New Developments in OSPF

BRKIPM-3006

Peter Psenak

Cisco Networkers 2007

HOUSEKEEPING

- We value your feedback, don't forget to complete your online session evaluations after each session and complete the Overall Conference Evaluation which will be available online from Friday.
- Visit the World of Solutions on Level -01!
- Please remember this is a 'No Smoking' venue!
- Please switch off your mobile phones!
- Please remember to wear your badge at all times including the Party!
- Do you have a question? Feel free to ask them during the Q&A section or write your question on the Question form given to you and hand it to the Room Monitor when you see them holding up the Q&A sign.

Agenda - Shipped Features

- Bidirectional Forwarding Detection for OSPF
- OSPF Link State Database Overload Protection
- MPLS LDP-IGP Synchronization and Autoconfiguration
- Interface-based OSPFv2 enable
- RFC 3623 OSPF Graceful Restart
- Scaling Enhancements
- Troubleshooting enhancements

Bidirectional Forwarding Detection for OSPF

BFD: Goal

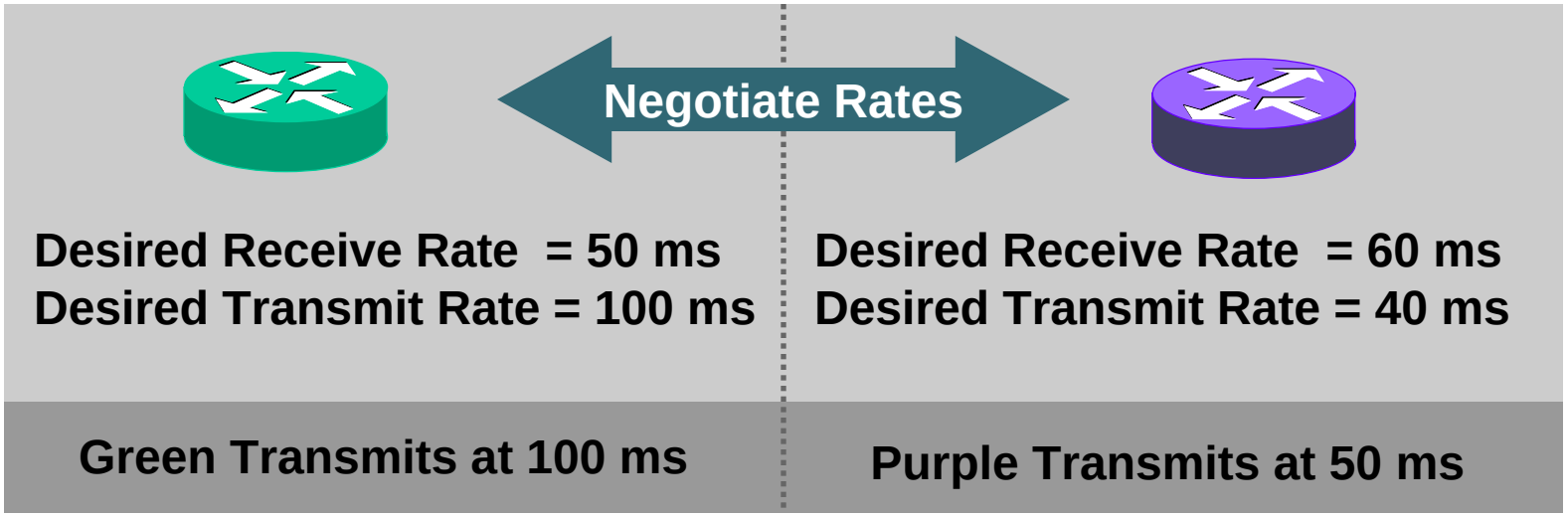
- Goal is sub-second neighbor loss detection
- Some underlying media types support sub-second detection (e.g., POS)
- Cisco IOS® OSPF supports fast Hellos
  - one second dead interval
  - prone to spikes in processing, occasionally generating false positives
- Need solution that works on all interface types and isn't prone to false positives due to processing spikes
  - BFD uses process pseudo preemption in IOS

BFD Protocol Overview

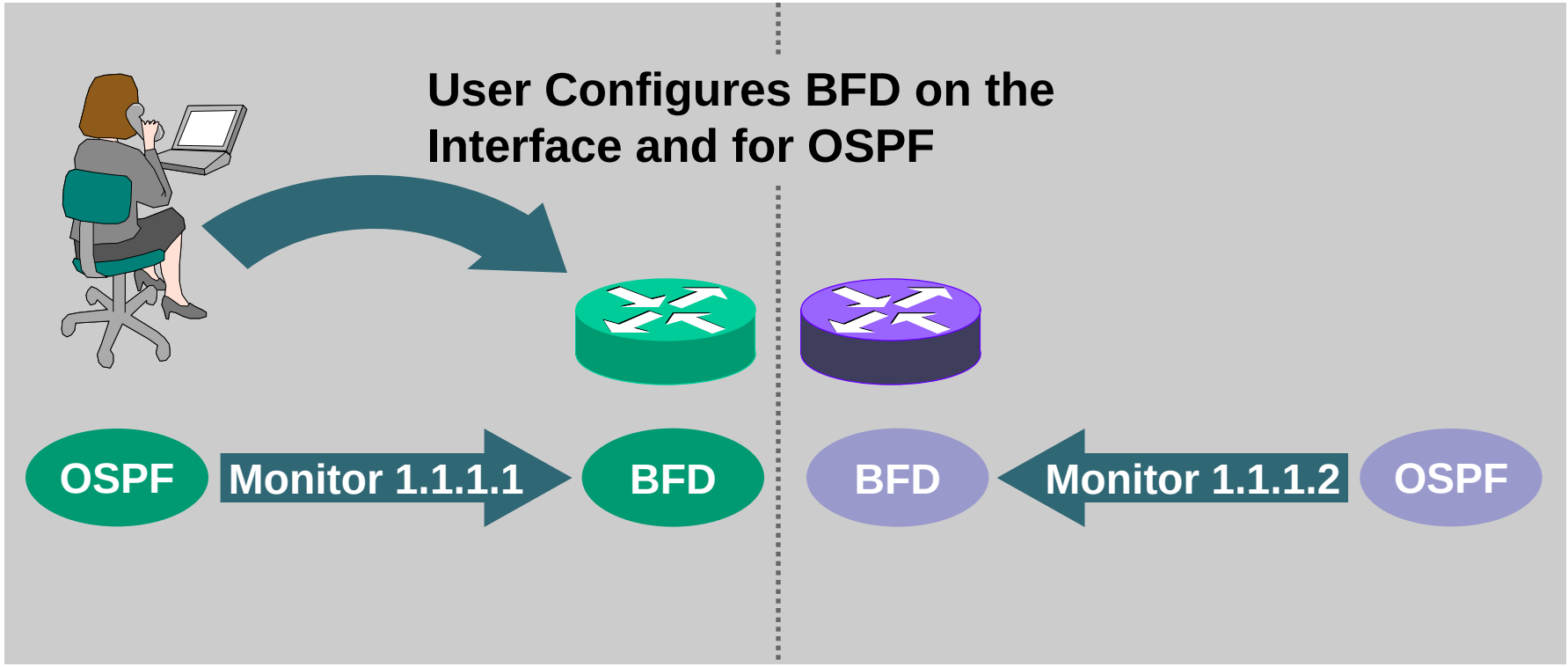
- Media independent: communications is always point-to-point between two systems
- Application independent (OSPF, ISIS, EIGRP, BGP…)
- There is no discovery mechanism in BFD
- Still need application keepalive mechanisms
  - Hellos needed for control plane verification
  - Hellos provide discovery mechanism
  - Hellos carry other protocol information
- Fast failure detection: lightweight, easy-to-parse, implemented on line cards (e.g., GSR)

Timer Negotiation

- Neighbors continuously negotiate their desired transmit and receive rates in terms of microseconds
- The system reporting the slower rate determines the transmission rate

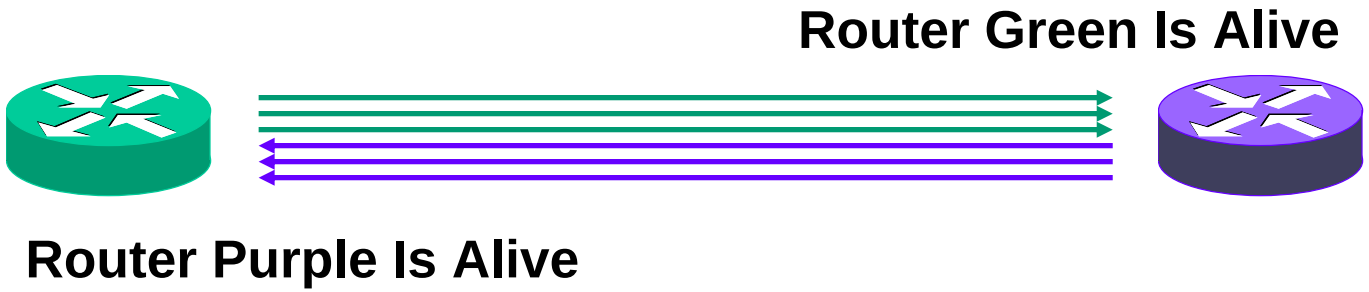


Modus Operandi



Asynchronous Mode

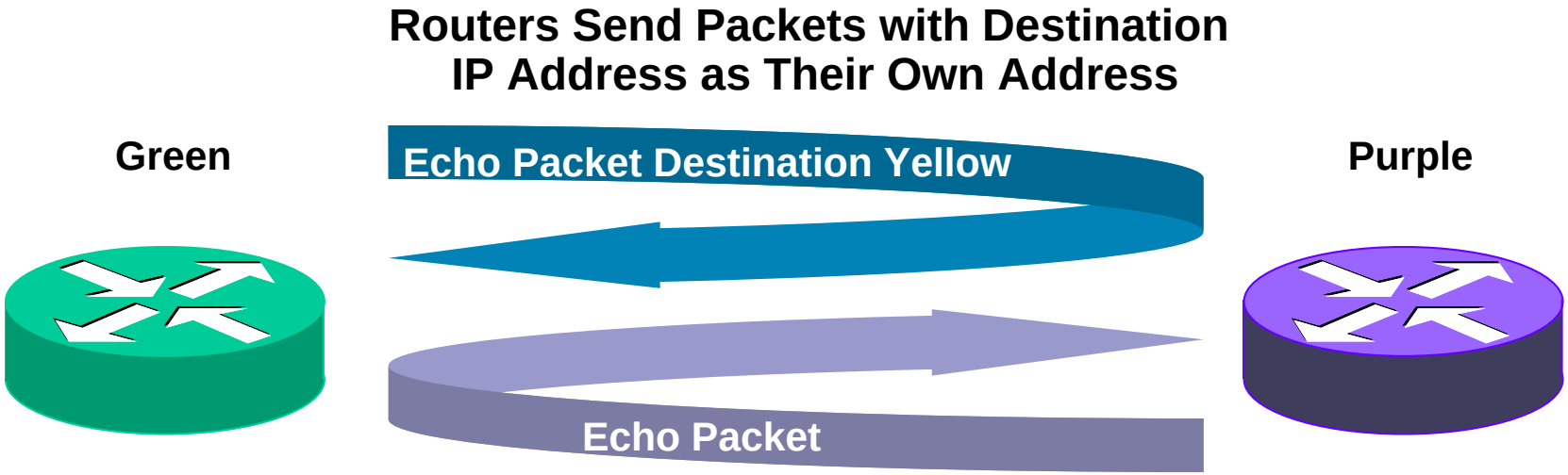
- Asynchronous mode–control packets flow in each direction



- Systems periodically send BFD control packets to each other
- If a number of those packets in a row are not received by the other system, the session is declared to be down
- Similar to OSPF Hello/Dead Interval operation

Echo Mode

- Echo mode



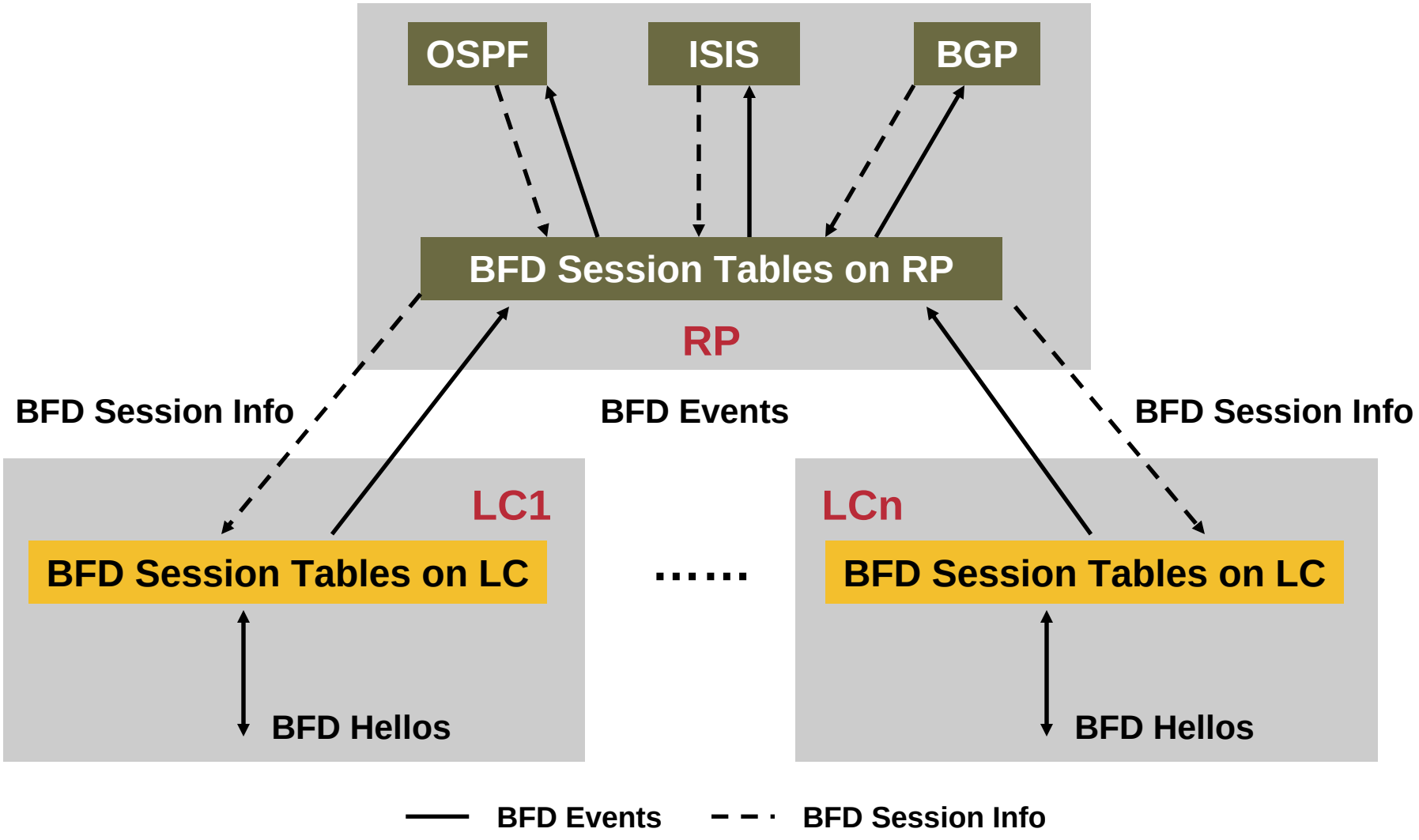
Echo Packets Loop Through the Remote System

- The session is established using control session and then echo mode can be negotiated between the systems

Demand Mode

- Both systems must agree on running demand mode
- Relies on alternate mechanisms to imply continuing connectivity
- Demand mode may be enabled or disabled at any time by setting or clearing the demand (D) bit in the BFD control packet, without affecting the BFD session state
- Periodically, at a negotiated rate, the system trying to verify connectivity sends a packet with ‘P’ bit set and the neighbor replies with ‘F’ bit set
- If no response is received to a poll, the poll is repeated until the detection time expires, at which point the session is declared to be down

BFD GSR Distributed Architecture



OSPF BFD Configuration

- Configure BFD on interfaces

```
R(config)# interface <interface-name>
R(config-if)# bfd interval <10-30000> min_rx <10-30000> multiplier <3-50>
```

- Configure BFD in OSPF

```
R(config)# interface <interface-name>
R(config-if)# ip ospf bfd
        or
R(config)# router ospf 1
R(router-config)# bfd all-interfaces
R(config)# interface <interface-name>
R(config-if)# ip ospf bfd disable
```

OSPF/BFD Interaction (Recap)

- OSPF neighbors discovered using Hello or manual configuration
- OSPF process registers neighbors with BFD when neighbor state machine goes to FULL
- BFD session is brought up with each willing neighbor
- Session timers negotiated
- BFD monitors liveliness of forwarding plane
- Swiftly notifies OSPF of BFD session failures
- Upon notification, OSPF brings down neighbor, regenerates a new Router-LSA, and runs SPF to reroute via any alternate paths

BFD Standards

- Bidirectional forwarding protocol: draft-ietf-bfd-base-04.txt
- BFD for IPv4 and IPv6 (single hop): draft-ietf-bfd-v4v6-1hop-04.txt
- Originally only Async mode was supported, later Echo mode was added
- Available in: 12.4(4)T 12.2(18)SXE 12.0(31)S 12.2(33)SRA

OSPF Link State Database Overload Protection

Database Overload Protection

- Protects router from the large number of received LSAs
  - possibly as a result of the misconfiguration on remote router
- Router keeps the number of received (non self-generated) LSAs
- Maximum and threshold values are configured
- When threshold value is reached, error message is logged
- When maximum value is exceeded, no more new LSAs are accepted.
- If the counter does not decrease below the max value within one minute we enter ‘ignore-state’

Database Overload Protection (cont.)

- In ‘ignore-state’ all adjacencies are taken down and are not formed for ‘ignored-interval’
- Once the ‘ignored-interval’ ends we return to normal operations
- We keep the count on how many times we entered ‘ignore-state’ – ‘ignore-count.’
- When ‘ignore-count’ exceeds it’s configured value, OSPF process stays in the ‘ignore state’ permanently
  - – Ignore-count is reset to 0, when we do not exceed maximum number of received LSAs for a ‘reset-time’
  - –The only way how to get from the permanent ignore-state is by manually clearing the OSPF process

Database Overload Protection (CLI)

- Router mode

```
max-lsa <max> [<threshold> [warning-only]
              [ignore-time <value>]
              [ignore-count <value>
              [reset-time <value>]]
```

- Available in: 12.3(7)T 12.2(25)S 12.0(27)S 12.2(18)SXE 12.2(27)SBC

MPLS LDP-IGP Synchronization and Autoconfiguration

LDP-IGP Sync: Problem Statement

- Lack of coordination between the IGP and LDP
- Results in packet loss if IGP adjacency is UP, but LDP session is DOWN on interface
  - On link up IGP adjacency comes up faster then the LDP label exchange completes
  - LDP session is lost but the IGP adjacency stays FULL

LDP-IGP Synchronization: Link Up

- If LDP session is up IGP brings adjacency up as normal
- If LDP session is not up
  - – If the LDP neighbor is reachable the IGP delays sending Hellos until:
    - • LDP session is up or
    - • Hold down timer fires (default: wait forever)
  - – If the LDP neighbor is not reachable the IGP starts sending Hellos (LDP needs IGP for its session)
    - • When the adjacency is up it initially announces the link using max-metric
    - • Use of the configured link metric begins when the LDP session is up

LDP-IGP Synchronization: LDP Session Down

- After IGP adjacency is up and IGP is advertising configured link metric if LDP session goes down
  - IGP starts advertising max-metric for the link
  - When LDP session is back up, IGP resumes advertising the configured link metric

LDP-IGP Synchronization (CLI)

- LDP-IGP synch enabled by new router mode command

  mpls ldp sync

- IGP holddown configured by new global mode command; default is infinite

  mpls ldp igp sync holddown <msecs>

- LDP-IDP synch disabled for an interface by new interface mode command

  no mpls ldp igp sync

LDP Auto-Configuration

- Service provider customers consider current LDP configuration cumbersome and error prone
- LDP auto-configuration provides a short cut to automatically enable LDP on links for which a specified IGP has been configured
- New router-mode configuration command

  mpls ldp autoconfig [area <area-id>]

  – Enables LDP on interfaces for which the associated IGP is enabled

  – if area is provided it is limited to interfaces attached to the specified area
- May be disabled on a per-interface basis

  no mpls ldp autoconfig igp

LDP Auto-Configuration (Cont.)

- Existing interface **mpls ip** command may still be used as just another way to enable LDP on interface
- LDP may be enabled on an interface either automatically by LDP autoconfig or explicitly by interface mpls ip
- Available in: 12.3(14)T 12.0(30)S

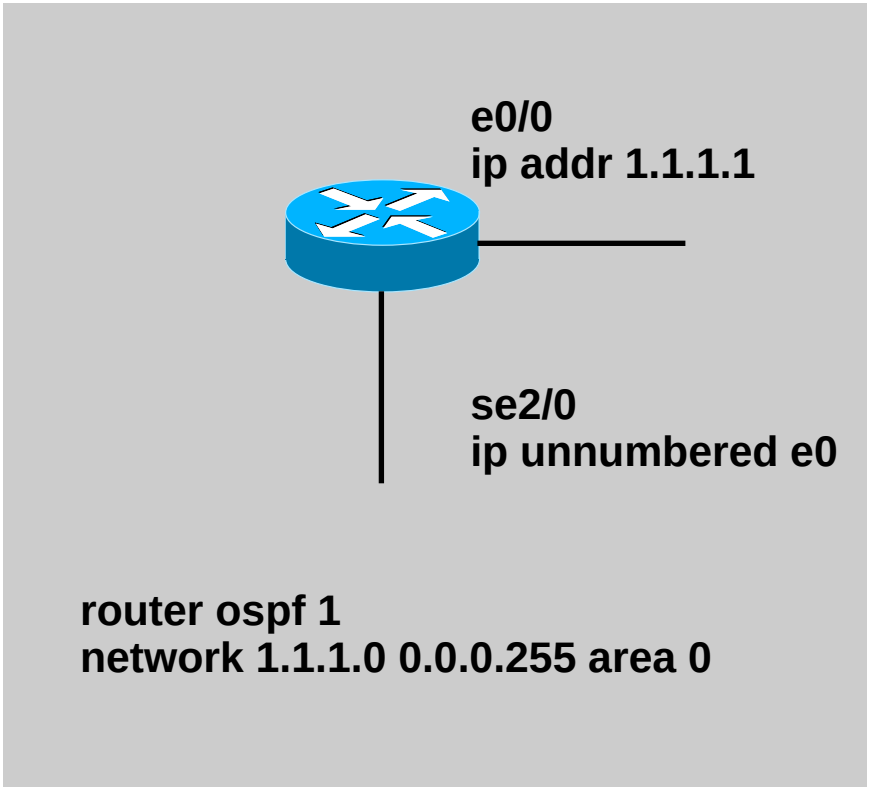
LDP Sync & Auto-Configuration

- New router mode mpls ldp commands require LDP to be enabled globally (the default)
- If LDP has been globally disabled new router mode mpls ldp commands will fail
- If a new router mode command has been configured, global no mpls ip will fail

Interface based OSPF Enable

Interface-Based OSPF Enable

- OSPF has traditionally supported enabling the protocol on interfaces by a network command
- A new tool which can be used instead of network command
- On unnumbered interfaces, OSPF automatically assumes the same area as the corresponding numbered interface
- What if the unnumbered interface is desired to be in a different area?

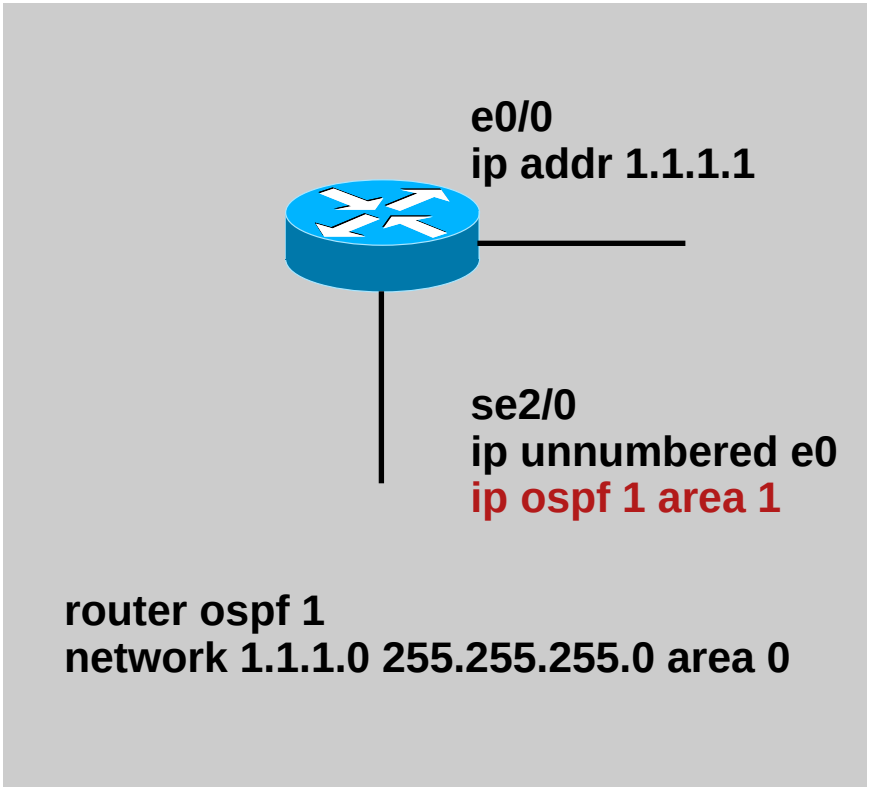


Interface-Based OSPF Enable

- Allow an interface specific command to enable OSPF in a different area
- Interface scoped command has higher precedence
- By default secondary addresses are announced;
  can be turned off with an option

  ip ospf 1 area 1 secondaries none
- Available in:

12.3(11)T 12.2(28)SB 12.0(29)S



RFC 3623 OSPF Graceful Restart

RFC 3623 OSPF Graceful Restart

- Prior to RFC 3623, we implemented Cisco-proprietary NSF (Cisco NSF)
- RFC 3623 uses Grace LSAs to notify neighbors about the restart event, compared to LLS in Cisco NSF
- Helper routers, similar to Cisco NSF
- CLI under the router-mode:

  nsf ietf

  nsf ietf helper disable

  nsf ietf helper strict-lsa-checking

- Available in 12.0(32)S 12.2(33)SRA
  Helper mode available in 12.4(6)T

Scaling Enhancements

Scaling Enhancements

- Several improvements have been made to help OSPF scale better
- Quantum Based CPU Releasing CSCdu44804
- OSPF Hello Process Optimization CSCuk56641
- Tunable OSPF Queue Sizes CSCeg85971 (OSPFv3 - CSCsc36160)

Quantum Based CPU Releasing

- OSPF was too friendly in releasing the CPU – we attempted to suspend too often
- In many cases we used to suspend based on the number of processed entities (packets, LSAs, timers, etc.)
  - dependent on the platform
  - processing of single entity may not be constant
  - processing of single entity may be affected by code changes
- If some higher priority process was scheduled we suspended without doing much work
  – we were put at the end of the NORMAL priority queue
- When IOS process gets CPU it has certain amount of time that it can use to do it's job before releasing the CPU (Quantum)

  -process-max-time <value> (default 200ms)

Quantum Based CPU Releasing (cont.)

- We may have suspended after few ms and had to wait several hundred ms, not having much time to do our tasks
- We introduced a ‘guaranteed’ percentage from quantum we use, before giving way to higher priority processes
- If there is no higher priority process running, we continue to run our full quantum
- New command under the router mode:

  process-min-time percent <value> (default 25)
- We use min-time when performing regular tasks and 2x (min-time) when performing more critical tasks

OSPF Hello Process Optimization

- OSPF Hello process has two main tasks:
  1. absorb incoming packets
     - Hello packets are processed
     - other OSPF packets are enqueued to Update Queue for later processing by OSPF Router process
  2. process various Neighbor and Interface related timers – Hello Timer, Dead Timer, etc.
- Our Quantum was not equally divided between these two tasks
- Problem was observed with large number of neighbors
  - we were able to send Hellos fast enough
  - we were not able to absorb packets fast enough
  - Result: adjacency loss

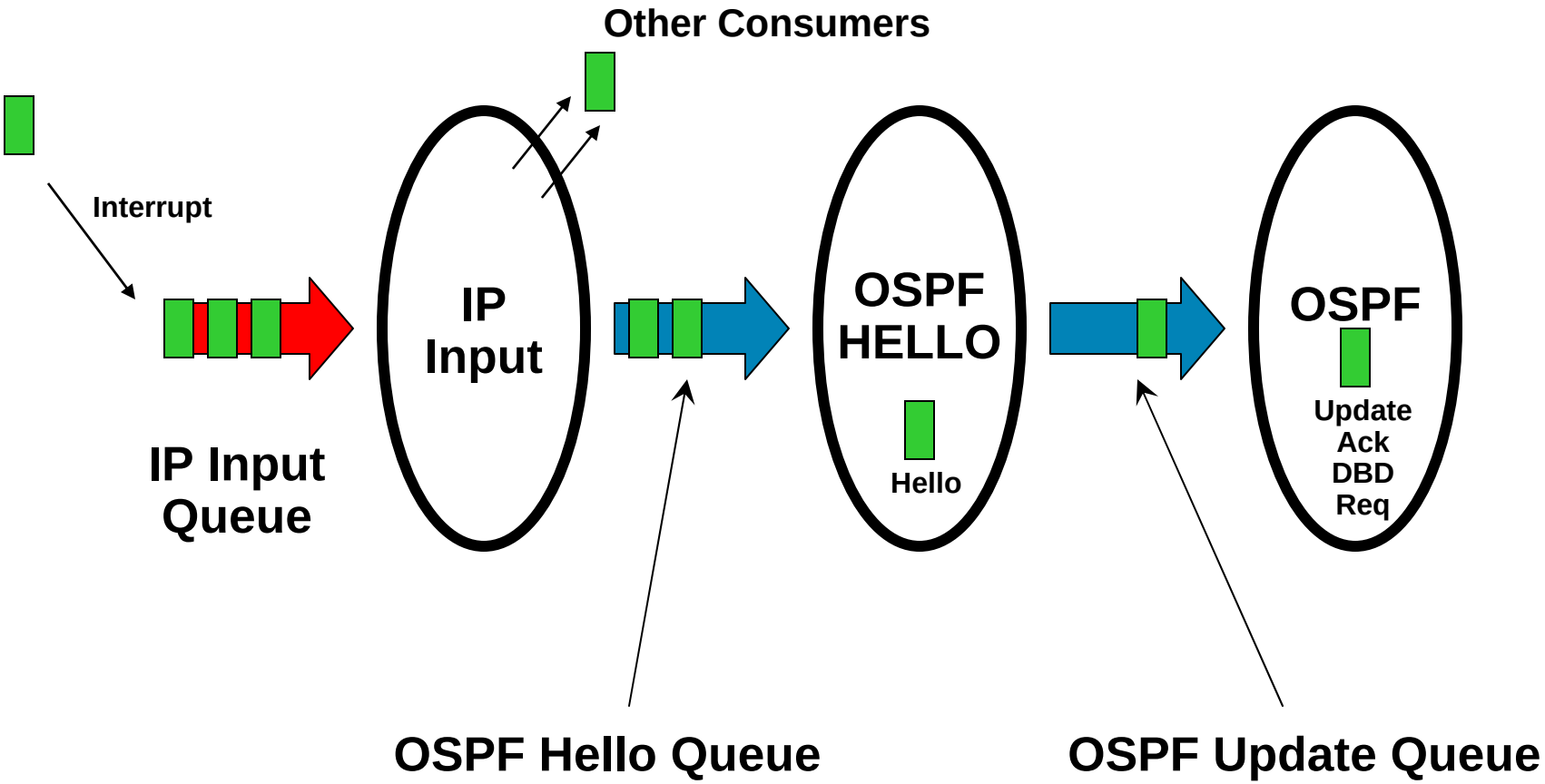
OSPF Hello Process Optimization

- Improvement has been made in a way we divide our quantum between the two tasks
- We measure the time we spend in each of the two tasks and check it after each packet/timer is processed
- When we have spent half of our quantum in one activity we switch to the other
- Balanced quantum usage
- Test has been performed with NPE-300:
  - 700+ OSPF peers
  - Dead Interval of 1 second
  - compared to 9 seconds prior to the change

Tunable OSPF Queue Sizes

- Multiple Queues are involved before OSPF packet gets processed
- IP Input Queue – packets whose destination is a local address on the system are enqueued to this queue at interrupt level
- OSPF Hello Queue – all OSPF packets are enqueued to this queue by ‘IP Input Process’
- OSPF Update Queue – all OSPF packets except OSPF Hellos are enqueued to this queue by OSPF Hello process

Tunable OSPF Queue size



Tunable OSPF Queue size

- IP input queue can be tuned by hold-queue in command
  - additional room for IGP packets can be created by enabling SPD
- OSPF Hello Queue and Update Queue sizes can now be changed:

  queue-depth {hello | update} {<queue size> | unlimited}
- Increasing Update Queue Size
  - helps in situation where we need to exchange large database with many neighbors simultaneously
- After Hello Process Optimization packets are enqueued to the Update Queue faster

Troubleshooting Enhancements

Interface Scoped Debugging

- Enhancement in limiting the OSPF debug output to just a selection of interfaces
- Example below will generate debug output for only two interfaces specified below

  debug condition interface Ethernet 0/0

  debug condition interface Ethernet 1/0

  debug ip ospf hello

  debug ip ospf adjacency

  Available in: 12.4(4)T 12.2(30)S 12.0(32)S

  debug ip ospf flood

  Further enhancement to allow an access-list filter, and a “detail” option (for verbose output)

  Available in: 12.4(4)T 12.2(30)S 12.0(32)S

OSPF Traffic Statistics

show/clear ip ospf [process-id] traffic [interface]

- Output consists of:
  - Global summary section
  - Per-process sections
    - OSPF queues
    - Interface details
    - Per-process summary

```
router2#show ip ospf traffic

OSPF statistics:
 Rcvd: 29 total, 0 checksum errors
    7 Hello, 8 database desc, 2 link state req
    8 link state updates, 4 link state acks

 Sent: 29 total
    8 Hello, 6 database desc, 2 link state req
    8 link state updates, 5 link state acks

      OSPF Router with ID (200.1.1.2) (Process ID 1)

OSPF queues statistic for process ID 1:

 OSPF Hello queue size 0, no limit, drops 0, max size 2
 OSPF Router queue size 0, limit 200, drops 0, max size 2
```

- Available in: 12.4(6)T 12.0(28)S 12.2(30)S
- OSPFv3 support in 12.2(31)SB

OSPF Traffic Statistics – Interface Details

Interface Serial2/0

OSPF packets received/sent

Type	Packets	Bytes
RX Invalid	0	0
RX Hello	8	384
RX DB des	8	496
RX LS req	2	72
RX LS upd	8	740
RX LS ack	4	236
RX Total	30	1928
TX Failed	0	0
TX Hello	10	792
TX DB des	6	624
TX LS req	2	112
TX LS upd	8	708
TX LS ack	5	460
TX Total	31	2696

- Per interface filter:
  show ip ospf traffic <if_name>

OSPF header errors
Length 0, Checksum 0, Version 0, Bad Source 0,
No Virtual Link 0, Area Mismatch 0, No Sham Link 0,
Self Originated 0, Duplicate ID 0, Hello 0,
MTU Mismatch 0, Nbr Ignored 0, LLS 0,
Authentication 0, TTL Check Fail 0,

OSPF LSA errors
Type 0, Length 0, Data 0, Checksum 0,

OSPF Traffic Statistics – Per Process Summary

```
Summary traffic statistics for process ID 1:

OSPF packets received/sent

 Type     Packets      Bytes
 RX Invalid   0          0
 RX Hello    8          384
 RX DB des   8          496
 RX LS req   2          72
 RX LS upd   8          740
 RX LS ack   4          236
 RX Total    30         1928

 TX Failed    0         0
 TX Hello    10         792
 TX DB des   6          624
 TX LS req   2          112
 TX LS upd   8          708
 TX LS ack   5          460
 TX Total    31         2696

OSPF header errors
 Length 0, Checksum 0, Version 0, Bad Source 0,
 No Virtual Link 0, Area Mismatch 0, No Sham Link 0,
 Self Originated 0, Duplicate ID 0, Hello 0,
 MTU Mismatch 0, Nbr Ignored 0, LLS 0,
 Authentication 0, TTL Check Fail 0,

OSPF LSA errors
 Type 0, Length 0, Data 0, Checksum 0,
```

- Per process filter:
  sh ip ospf <process_id> traffic

OSPF SPF Detail Statistics

- Original ‘show ip ospf stat’ output:
  - included time values for various phases of SPF
  - area independent
  - process independent
- Command has been extended with:
  - per process
  - per area and per SPF statistics
  - SPF Type (Full or Incremental)
  - LSAs that triggered SPF
  - number of LSA processed
- CLI

  show ip ospf <process-id> statistics <detail>

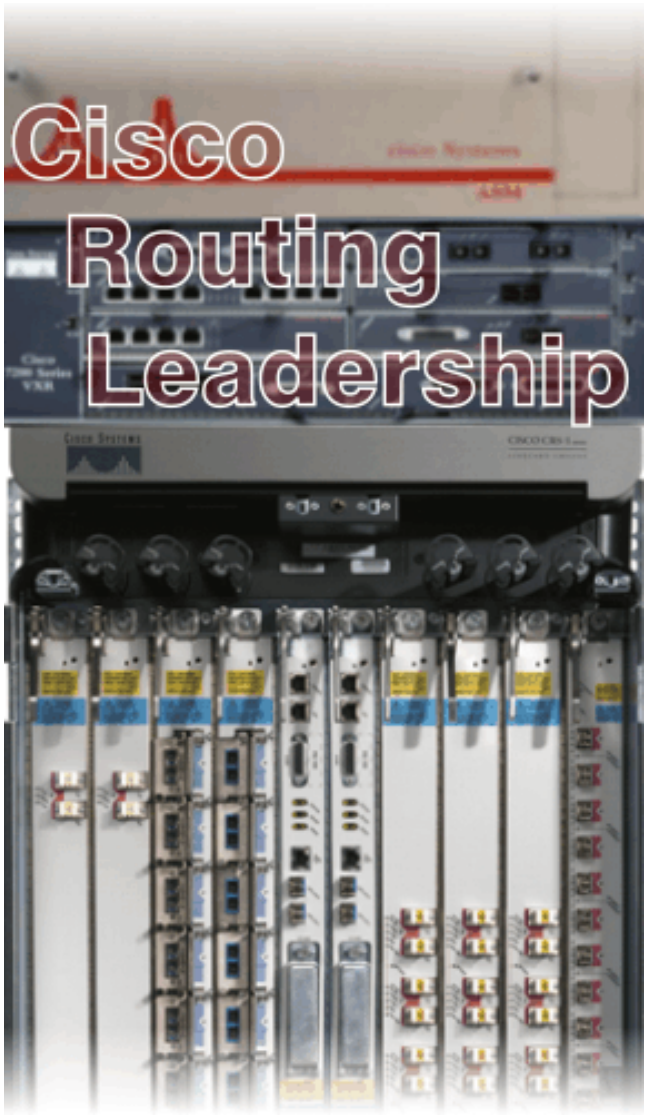
OSPF SPF Detail Statistics (cont.)

```
OSPF Router with ID (1.0.0.100) (Process ID 1)

  Area 1: SPF algorithm executed 5 times

SPF 1 executed 00:48:13 ago, SPF type Full
  SPF calculation time (in msec):
  SPT    Intra  D-Intr Summ   D-Summ Ext7   D-Ext7 Total
  8       16        4        0         0           0       0         28
LSIDs processed R:120 N:30 Stub:240 SN:0 SA:0 X7:0
  Change record R,
  LSIDs changed 1
  Last 10 LSIDs:  1.0.0.100(R)

....
....
```

Available in 12.0(24)S 12.2(18)S 12.3(2)T

Agenda - Currently Active Features

- Multi-Topology Routing (MTR)
- Local RIB
- Advertisement of Prefix/Link Attributes
- Generalized TTL security mechanism
- OSPF Prefix Suppression
- OSPF Graceful Shutdown
- OSPFv3 Fast Convergence
- OSPF Event Logging



OSPF and Multi Topology Routing

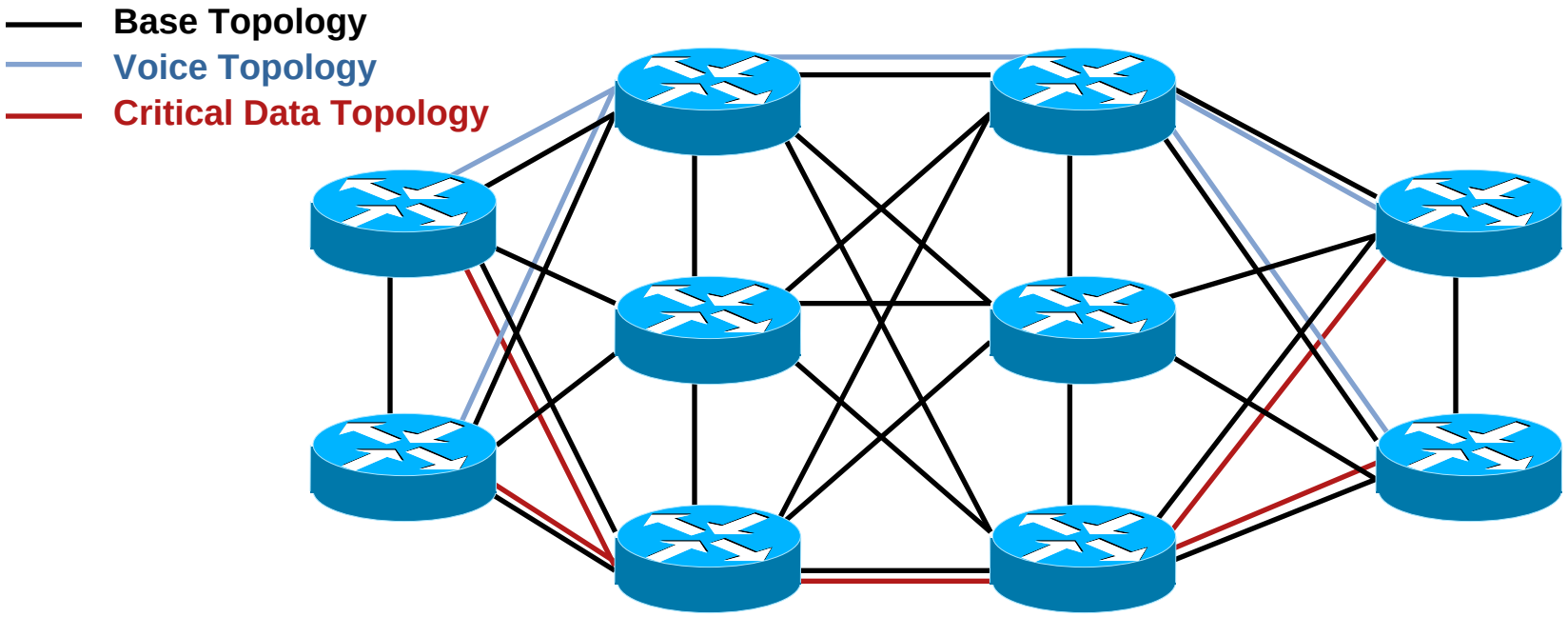
What is MTR?

- Multi-Topology Routing (MTR) allows:
  - – classification of the traffic to classes and class specific forwarding of the traffic
  - – efficient use of the network infrastructure by mapping classes of traffic to L3 topologies
- MTR adds another dimension to L3 destination based routing – traffic class
- Goal

  Influence the path that certain types of traffic would take to reach a given destination based on attributes such as DSCP, application type etc., in addition to the destination address.

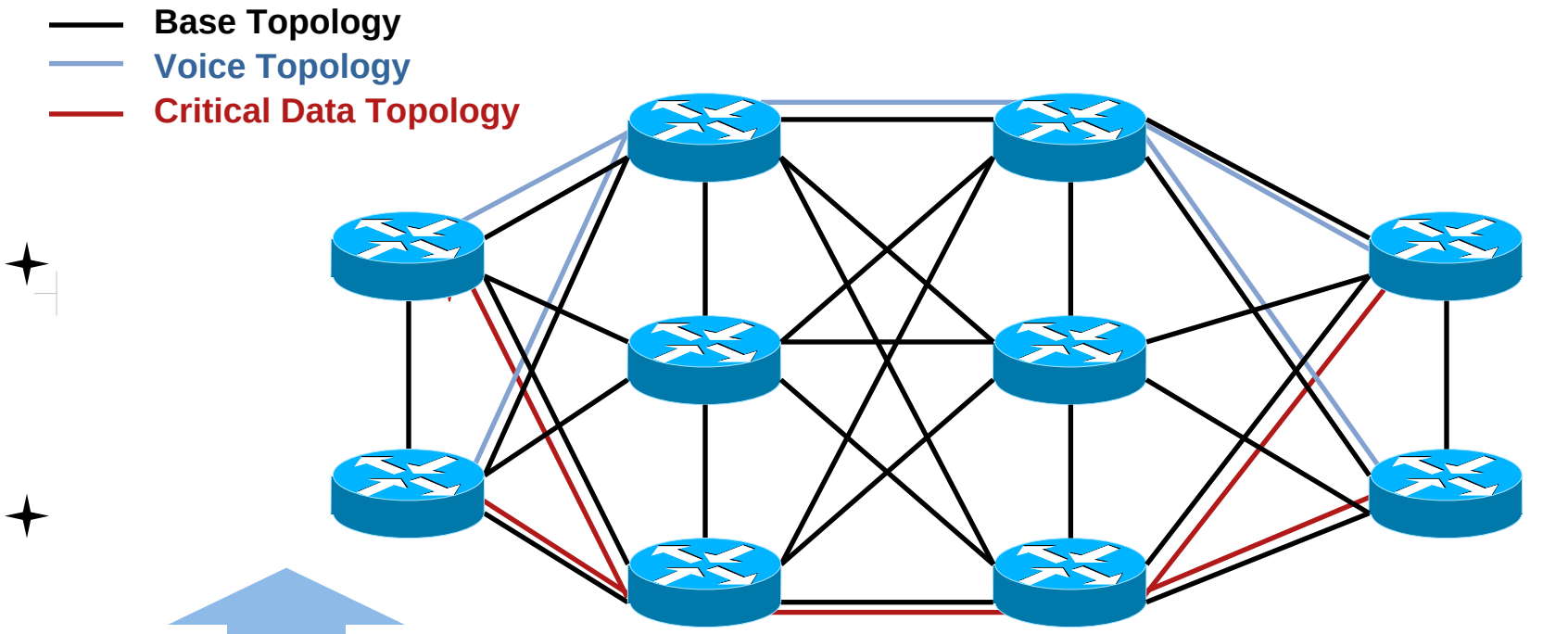
Multi-Topology Routing
Defining Topologies

Start with a Base Topology Includes All Routers and All Links

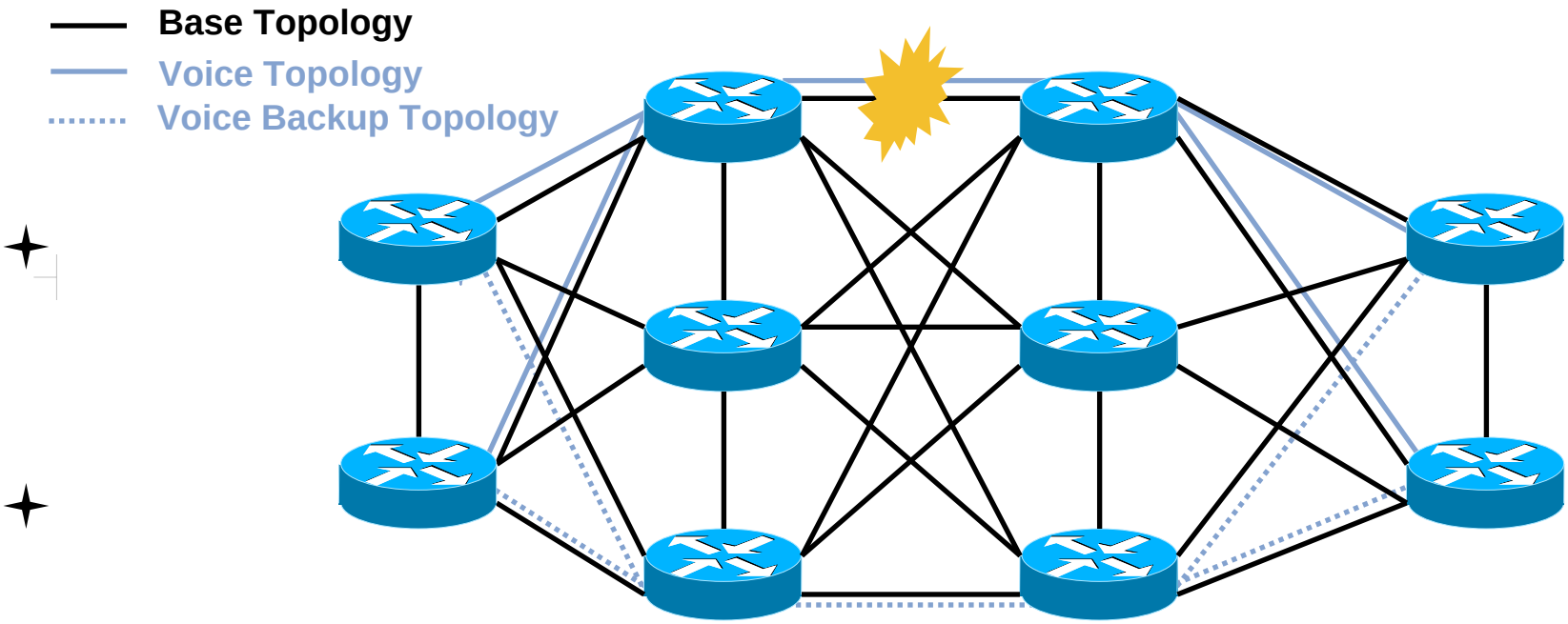


- Define the colored topology across a contiguous section of the network
- Individual links can belong to multiple topologies

Multi-Topology Routing
Traffic Paths

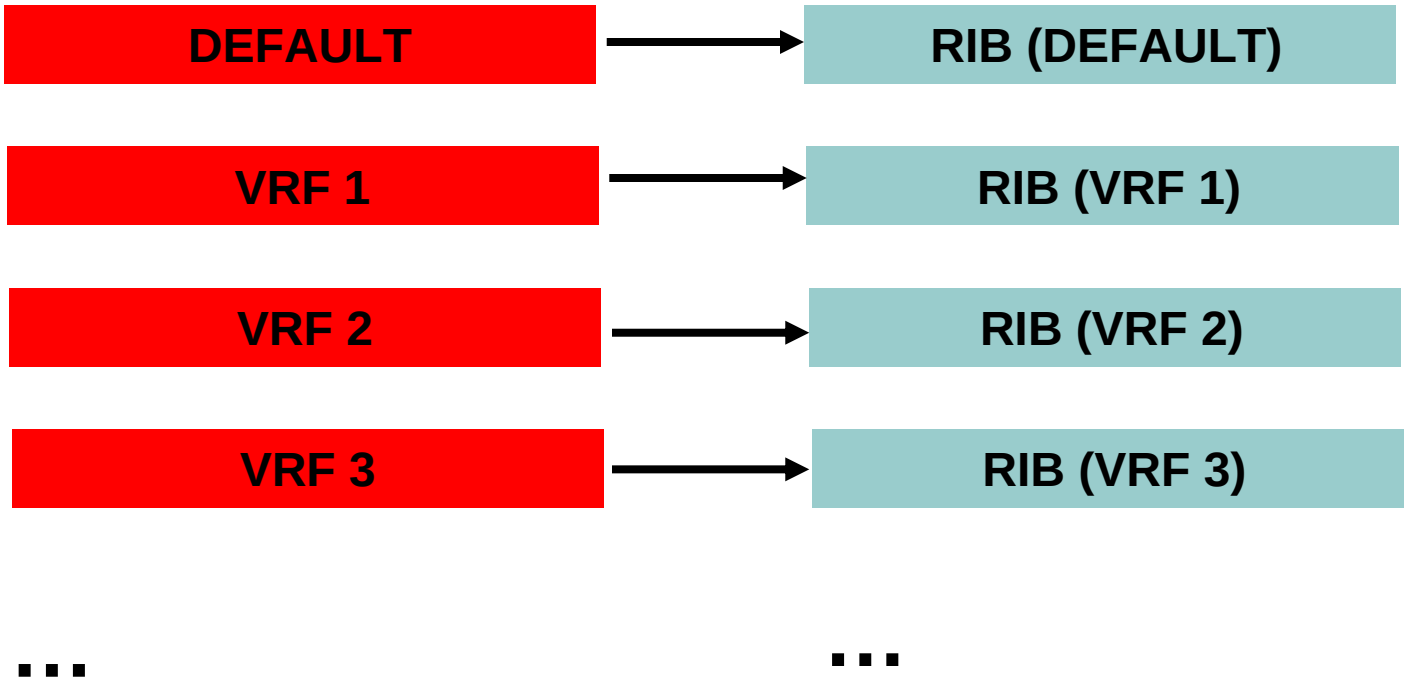


Multi-Topology Routing
Backup Topologies

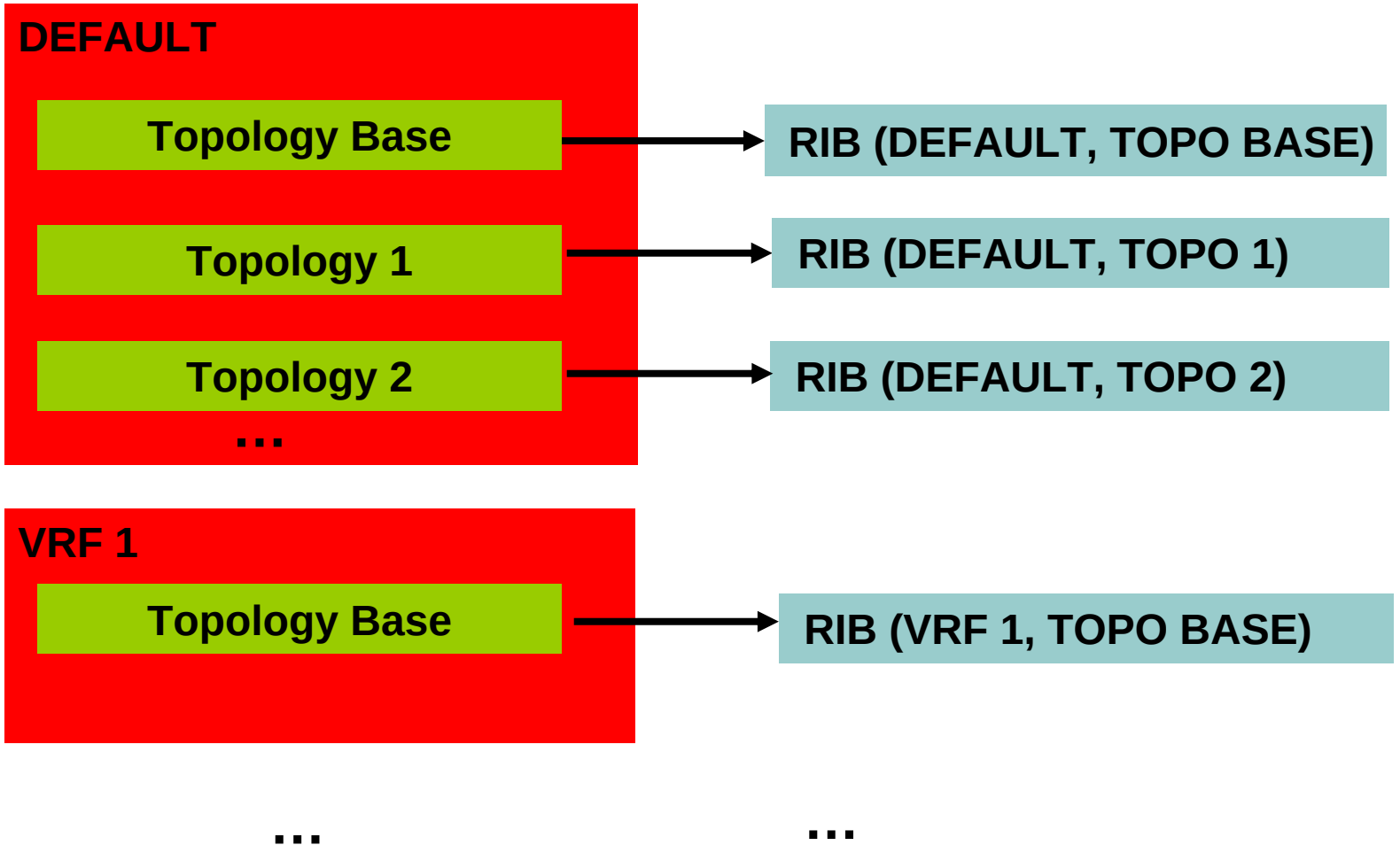


- Topologies can have configured backup paths
- Selection of primary/backup path based on cost—no different than how it is done today

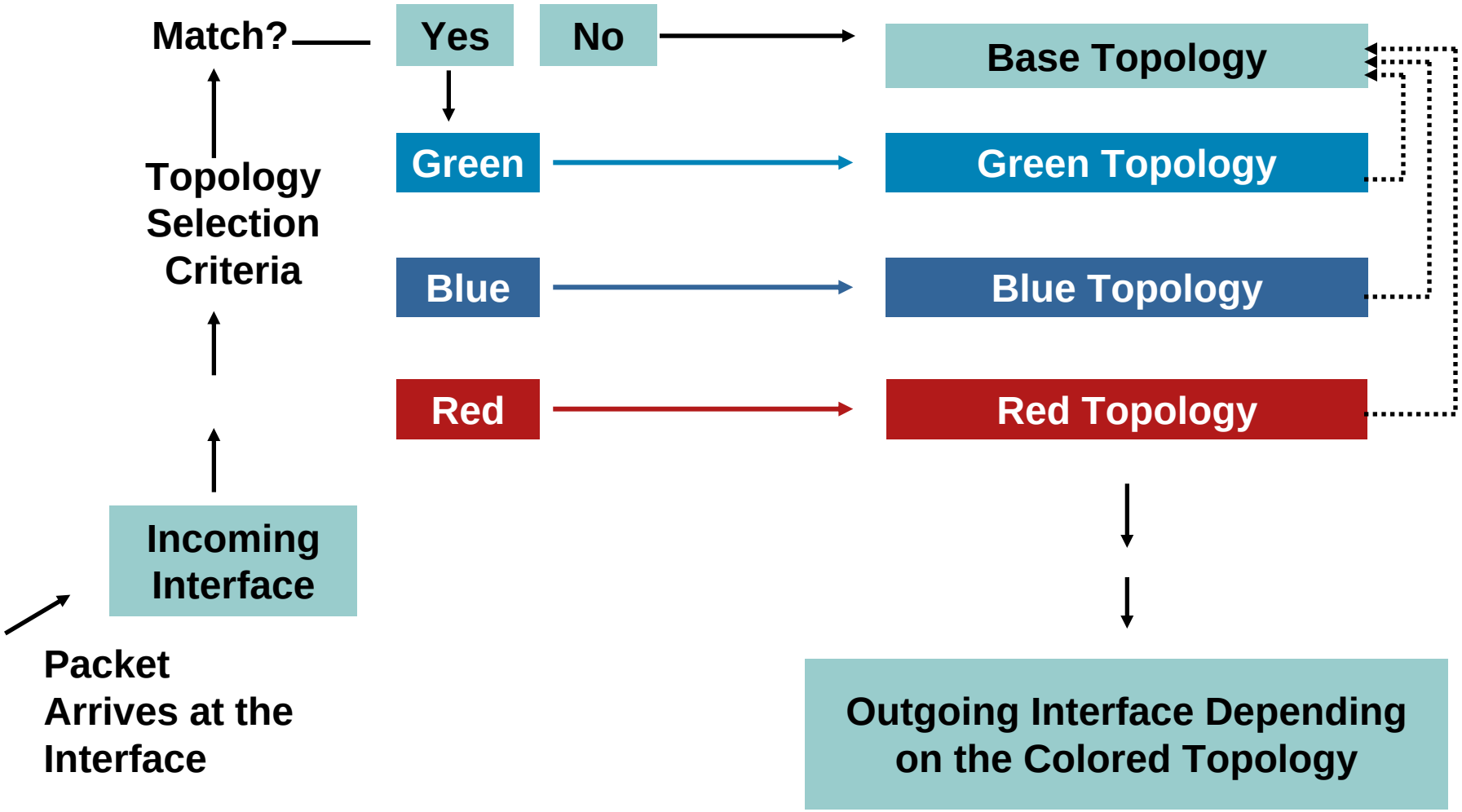
MTR Routing Infrastructure – pre MTR



MTR Routing Infrastructure – MTR



MTR Operation at the System Level



Supported Forwarding Behaviors

- Strict Forwarding
  - – if the packet can not be forwarded based on the topology specific forwarding table, packet is dropped
- Fallback to the Base (Incremental Mode)
  - – if the packet can not be forwarded based on the topology specific forwarding table, packet is forwarded based on the forwarding table associated with the Base topology

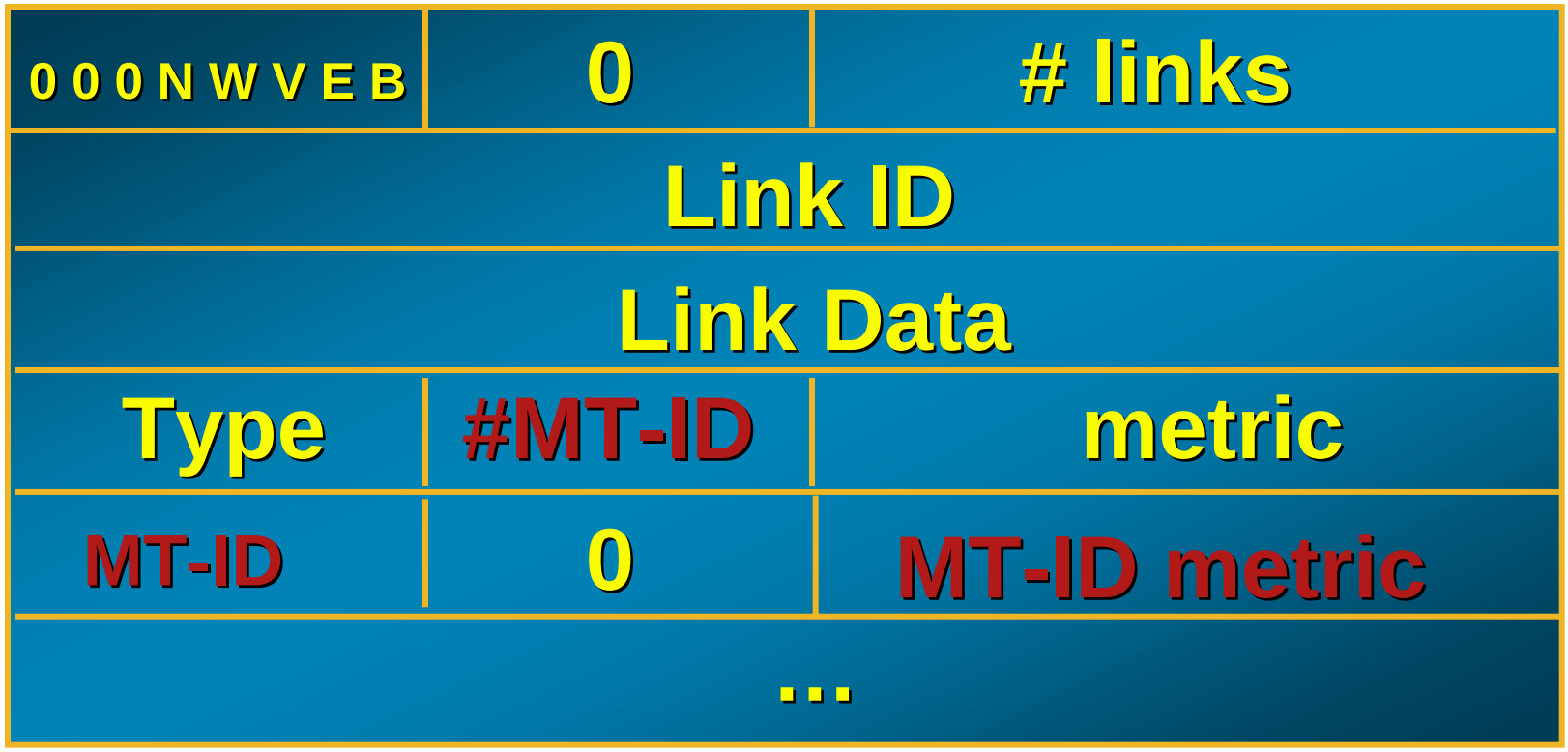
MTR OSPF Protocols Extension

- Requirement for an MTR aware OSPF
  - attach interfaces to multiple topologies
  - advertise topology specific link and/or prefix information
  - compute topology specific reachability for IP prefixes
  - populate topology specific RIB
  - receive and process topology specific redistribution event notifications
  - backward compatible with pre MTR version

OSPF MTR - Topology Specific Metrics

- Ability to advertise multiple metrics in LSAs
  - Per link/prefix per topology metric
- RFC1583 defined way to achieve this (TOS specific metrics) but RFC2328 removed it
  - TOS metric fields remained in LSAs for backward compatibility
- We redefined TOS to MT-ID and TOS metrics to MT metrics
- MTID range 0–127
  - 0 reserved for base unicast topology
  - 1 reserved for base multicast topology
- Used in Router LSA, Summary LSAs, External LSAs to describe the metric for multiple topologies

OSPF MTR – Router LSA



OSPF MTR - Topology Specific Metrics (cont.)

- Explicit topology metric advertisement
  - valid for ‘Base’ (TOS-0) metric too
- If the metric for certain topology is not advertised, link/prefix does not participate in the topology
- Difference to the RFC1583
  - If T-bit was set and certain TOS metric was not advertised, it was considered equal to TOS-0 metric (implicit TOS metric advertisement)

OSPF MTR – Adjacencies

- Single adjacency formed – even if multiple topologies defined on the interface
  - reduces overhead
  - reduces amount of adjacencies
- Adjacency is formed even if there is no topology in common over the link
- OSPF control packets are sent using Base (TOS0) topology

OSPF MTR – SPF Computations

- SPF scheduling is topology specific and independent of SPF in other topologies
  - SPF throttling timers are topology specific
- If SPF for multiple topologies is ready to run, priority mechanism is used to order the SPF runs
  - priority can be configured for each topology
- SPF calculation for each topology is independent from SPF in other topologies

Excluding the Link/Prefix from Base Topology

- How to Exclude Link/Prefix from Base Topology
- In Summary LSA and External LSA

  Set the TOS0 metric to 0xFFFFFF (compatible with RFC2328)
- In Router LSA there is no ‘unreachable’ metric for router links
  - area must have DefaultExclusionCapability enabled:
    - TOS0 metric in Router LSA is ignored
    - MT#0 metric is used instead and is optional

Interoperability with Non-MTR Aware Routers

- If there is no need to exclude link from the base topology, there is no problem in interoperability
  - MTR unaware routers will only participate in base topology
- When there is a need to exclude the link from the base topology, DefaultExclusionCapability must be enabled on all routers in the area
  - MT-bit (old T-bit) set in Hello packets
  - If received Hello does not have MT-bit set, it's ignored
  - TOS0 metric is not used, MT#0 metric used instead

OSPF MTR MIB Enhancement

- OSPF MIB doesn't specify a mechanism to access objects in more than one instance or topology
- SNMP has a mechanism to pass a "context" in it's queries
- We will allow context-string configuration at the instance and topology level
- When queried with these contexts, right OSPF objects can be returned

```
router ospf 1 vrf CUST_A
snmp-context-string OSPF-1
!

router ospf 2 vrf CUST_B
snmp-context-string OSPF-2
!
```

```
router ospf 2
address-family ipv4
 topology VOICE tid 33
  snmp-context-string OSPF-2_VOICE
```

MTR CLI – Configuration Example Global Level

```
ip multicast-routing
ip multicast rpf multitopology
!
class-map match-any TRAFFIC_VOICE
  match  dscp af11
class-map match-any TRAFFIC_DATA_CRITICAL
  match  dscp af12
!
!
policy-map type class-routing ipv4 unicast MTR_POLICY
  class TRAFFIC_VOICE
   select-topology VOICE
  class TRAFFIC_DATA_CRITICAL
   select-topology DATA_CRITICAL
!
!
global-address-family ipv4
 topology DATA_CRITICAL
 !
 topology VOICE
 !
 service-policy type class-routing MTR_POLICY
!
```

MTR CLI – Configuration Example Interface Level

```
!
interface Loopback0
 ip address 10.0.0.1 255.255.255.255
!
interface Serial2/0
 ip address 10.1.0.1 255.255.255.252
 ip pim dense-mode
 serial restart-delay 0
 !
 topology ipv4 base
 ip ospf topology disable
 !
 !
 topology ipv4 VOICE
 ip ospf cost 10
 !
 !
 topology ipv4 DATA_CRITICAL
 ip ospf cost 10
 !
 !
 topology ipv4 multicast base
 ip ospf cost 20
 !
!
```

```
interface Serial3/0
 ip address 10.1.0.5 255.255.255.252
 ip pim dense-mode
 ip ospf cost 10
 serial restart-delay 0
 !
 topology ipv4 DATA_CRITICAL
 ip ospf cost 20
 !
 !
 topology ipv4 multicast base
 ip ospf cost 10
 !
!
```

MTR CLI – Configuration Example Router Level

```
router ospf 1
 log-adjacency-changes
 network 10.0.0.0 0.255.255.255 area 0
 !
 address-family ipv4 unicast
 !
 topology DATA_CRITICAL tid 20
 timers throttle spf 50 100 4000
 priority 100
 !
 !
 topology VOICE tid 10
 snmp context OSPF1_VOICE_TOPO
 timers throttle spf 10 20 2000
 priority 120
 !
 exit-address-family
 !
 address-family ipv4 multicast
 !
 topology base
 priority 10
 !
 exit-address-family
!
```

Configuring MTR – OSPF topology aware commands

- area <area-id> default-cost
- area <area-id> filter-list
- area <area-id> nssa default-information-originate
- area <area-id> nssa no-redistribution
- area <area-id> nssa no-summary
- area <area-id> range
- area <area-id> stub no-summary
- area <area-id> virtual-link
- discard-route
- max-metric
- neighbor <ip address> cost <value>
- summary-address
- timers throttle spf
- priority

OSPF Local RIB and Prefix Prioritization

OSPF Local RIB

- OSPF Local RIB is a local cache in OSPF which keeps all active and backup OSPF routes
- Manipulating or walking OSPF routes is fast
- OSPF Local RIB is the only place where routes are updated during SPF
- Once certain part of the SPF is finished OSPF Local RIB is synchronized with Global RIB
- Prevents churns in the Global RIB and FIB during SPF in some cases

OSPF Local RIB (cont.)

- Local RIB is an infrastructure for more advanced features to come
- For a large number of prefixes, the time between the installation of the first and the last prefix can be significant
- A VoIP gateway address may need to be processed faster than other types of IP packets; Same for BGP peering (loopback address)

Priority Driven IP Prefix RIB Installation

Local RIB Defines Three Queues to Prioritize IP Prefix Installation in the RIB

- High priority prefixes

  Prefixes configured with a high priority will be inserted into the RIB first

- Medium priority prefixes

  Today /32 prefixes— e.g. loopbacks for BGP next-hops

- Low priority prefixes

  All other prefixes

- More queues could be defined to achieve higher degree of granularity

OSPF Link and Prefix Attributes

OSPF Link and Prefix Attributes

- The goal is to advertise link or route attributes for:
  - Further routing decision making
  - Opaque information for applications
- The proposal should be extensible enough to allow for a number of different types of attributes
- One of the applications is to advertise Tags for:
  - links in the topology
  - intra-area prefixes
  - inter-area prefixes
- Tags can be used for various purposes – e.g. priority treatment during SPF for faster convergence

OSPF Link and Prefix Attributes (cont.)

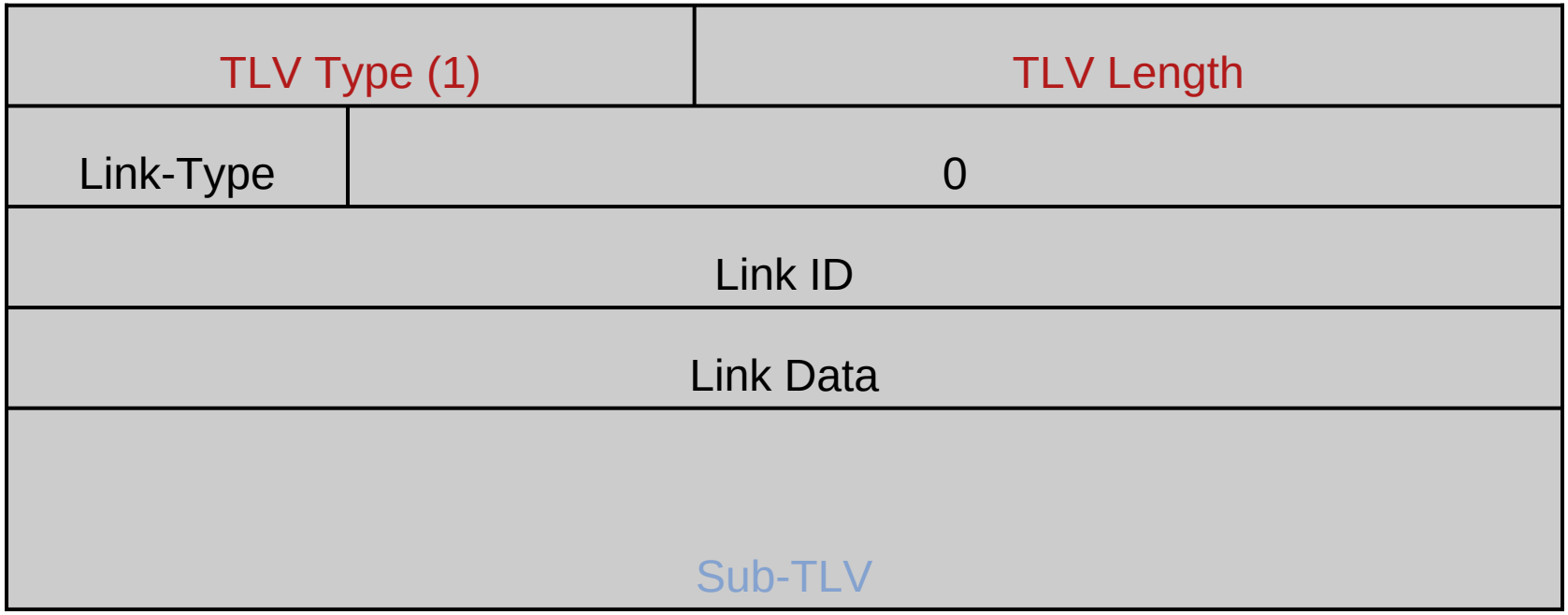
- Cisco's IETF Draft: draft-mirtorabi-ospf-tag-03.txt
- Current LSAs are not extensible
- A new LSA, called Route Attribute LSA is defined to advertise attributes for different LSA types
- RA LSA is an opaque LSA type 9, 10, or 11 depending on the desired flooding scope
- RA LSA payload is TLV-based (extensible)
- loose correspondence between prefix/link attributes and base OSPF LSAs
  - attributes for links advertised in a single Router LSA are not mixed with other links

RA LSA TLV Types

- Link attribute TLV (type 1)
  - Attributes for links and intra-area prefixes
- Inter-area route TLV (type 2)
  - Attributes for inter-area prefixes
- External route TLV (type 3)
  - Attributes for external prefixes
- NSSA external TLV (type 4)
  - Attributes for NSSA external prefixes

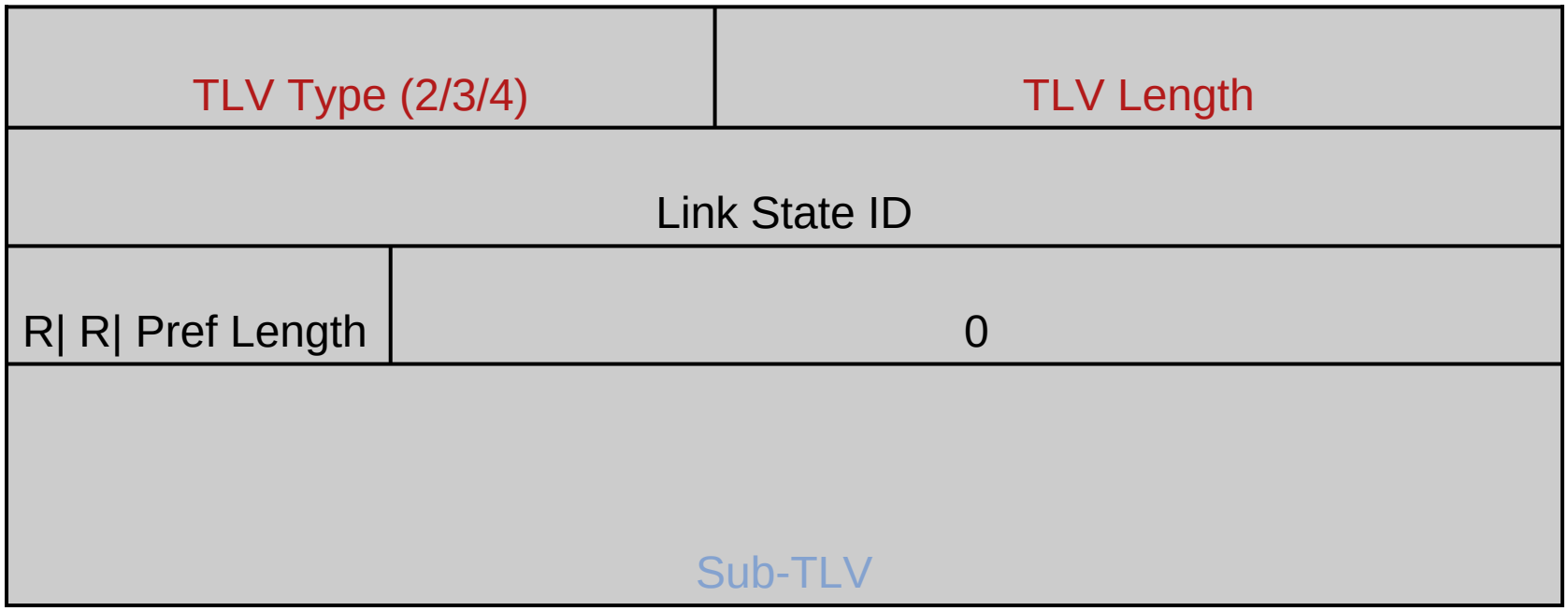
Link Attribute TLV

Contains Information to Uniquely Identify a Link



Inter-Area/External Route TLV

Contains Information to Uniquely Identify a Link



Generation of RA LSA

- A router configured to advertise link or route attribute, generates a RA LSA corresponding to one of its self-originated LSA
- RA LSA is regenerated only when there is a change in attribute value
- Across area boundary, an ABR will generate

  Area scoped RA LSA to associate attribute to type 3/type 4 LSA

  AS scoped LSA to associate attribute to translated NSSA routes

  Local policy could overwrite the default behavior

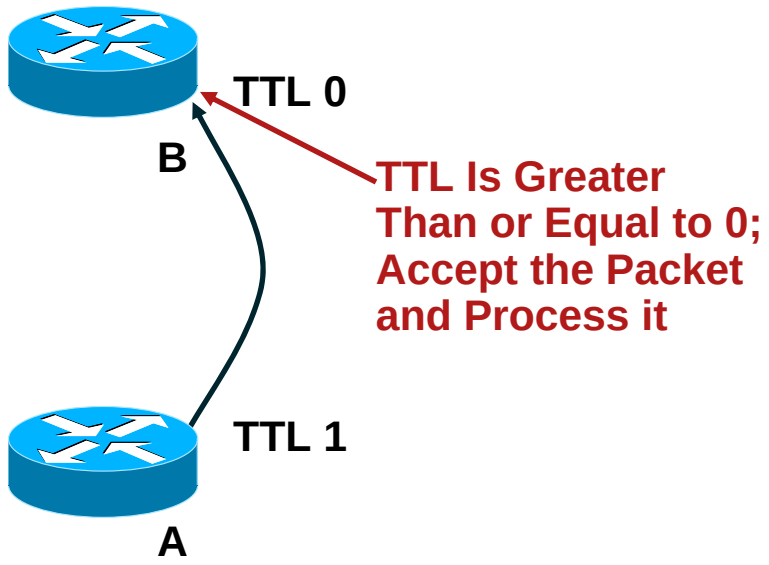
Backward Compatibility

- There are no backward compatibility issues as new LSA is ignored by non-capable router
- In order to advertise attribute across areas, ABRs should be capable or the flooding scope should be AS scope

Generalized TTL Security Mechanism

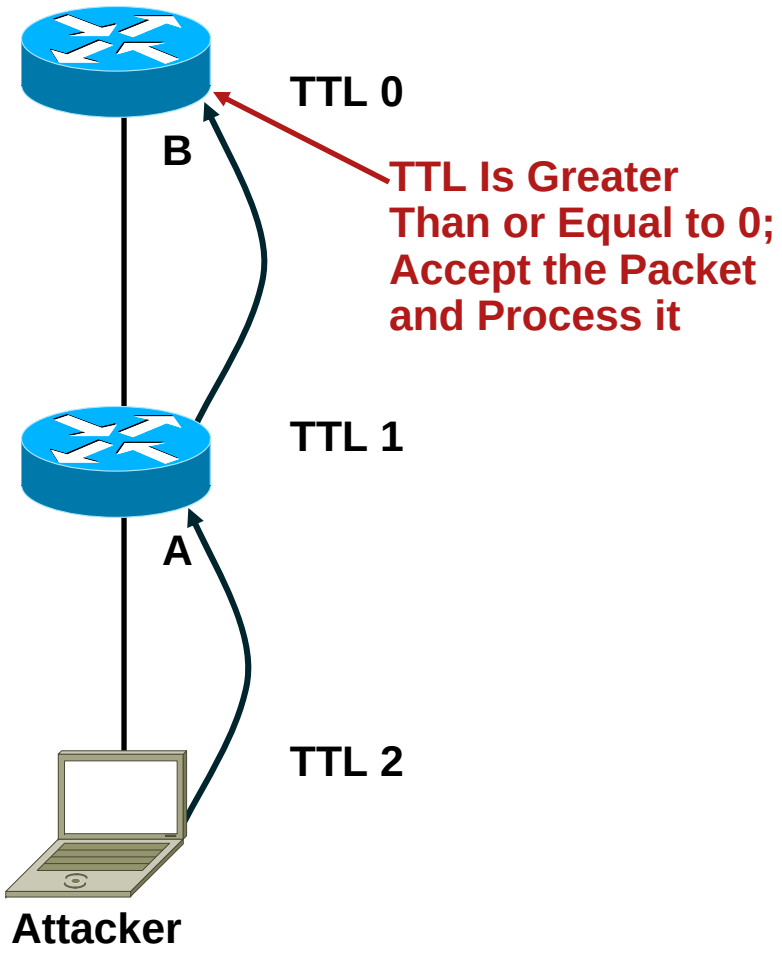
Original Protocol Design

- Routers send protocol packets with a TTL value of 1
- Receiving router accepts the packet, since it's destined to itself and TTL > =0



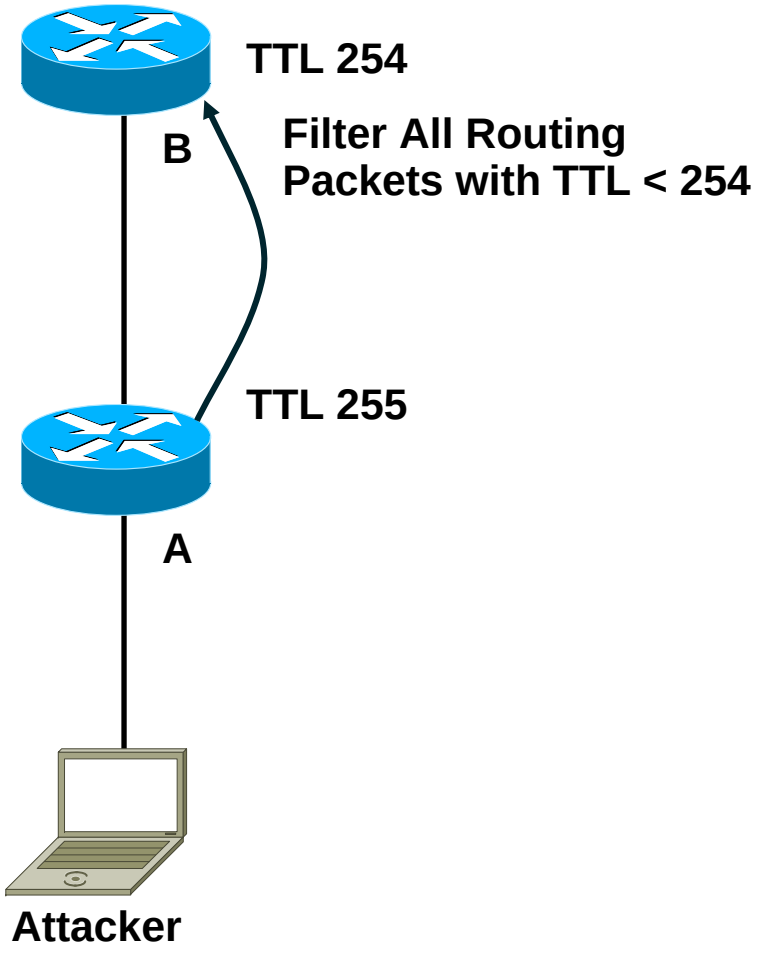
Attack Scenario

- An Attacker can send a packet to A with B's destination address
- A decrements the TTL by one, and forwards the packet to B
- B examines the TTL, finds it's still >=0, so it accepts the packet, and processes it
- Routing protocol Stack on B does not care about the TTL value
- This allows attackers to attack routers from multiple hops away
- DOS Attack possibility



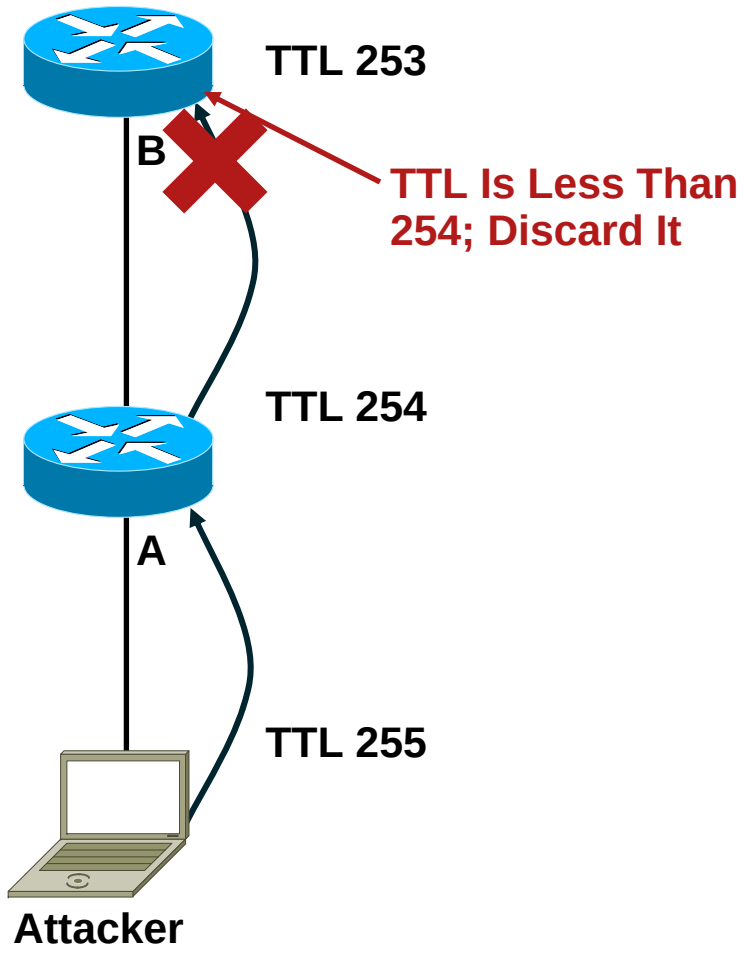
The Solution

- TTL is just a number
- A transmits all packets with a TTL of 255
- Routing protocol stack on B filters all routing protocol packets with a TTL less than 254
- Packets coming from more than one hop away will fail the TTL check and will get dropped



Generalized TTL Security Mechanism

- The Attacker sends a packet to A with B's destination address
- A decrements the TTL by one, and forwards the packet to B
- RP stack on B receives the packet with TTL < 254 and drops it
- The Attacker can no longer launch an attack against B
- TTL check will be initially implemented at the process level, eventually can be done in hardware later



OSPF
Prefix Suppression

OSPF Prefix Suppression

- When OSPF is enabled on the interface, it always advertises directly connected subnet
  - only way to prevent the connected subnet to be advertised is to make the link unnumbered (not possible on broadcast segments)
  - users may want to keep the links numbered for management and troubleshooting purposes
- Addresses associated with transit links are routable in the whole domain – security concern
- In large networks many prefixes are advertised for transit links - overhead
- We allow prefixes associated with the transit links not to be advertised
- Transit links can stay numbered

OSPF Prefix Suppression (CLI)

- Router Mode

  [no] prefix-suppression

  – suppress all prefixes except loopbacks and passive interfaces

- Interface Mode

  [no] ip ospf prefix-suppression [disable]

  – suppress all prefixes on interface

  – loopbacks and passive interfaces are included

  – takes precedence over router-mode command

  – disable keyword makes OSPF advertise the interface ip prefix, regardless of router mode configuration

OSPF
Graceful Shutdown

OSPF Graceful Shutdown

- Technique of removing the OSPF router from a network with minimal disruption to the network
- OSPF configuration is retained
- When router enters the shutdown mode it informs all other routers in the network about this event
  - floods it's own Maxed Router-LSA in each area
  - sends Hello packet with no neighbors listed on all it's interfaces
- OSPF process does not perform any activity when shutdown
  - OSPF packets received on any interface are dropped
  - no OSPF packets are sent
  - only configuration changes are processed

OSPF Graceful Shutdown (cont.)

- Router Mode command

  [no] shutdown

- Per interface shutdown is also available

  – Hello packet with no neighbors is sent on interface

  – Link(s) associated with the interface are removed from Router LSA

  – Network LSA is flushed for broadcast segment

  – OSPF operation is disabled on the interface

  – OSPF configuration is retained

- Interface Mode command

  [no] ip ospf shutdown

OSPFv3
Fast Convergence

OSPFv3 Fast Convergence

- SPF and LSA throttling has been implemented for OSPFv3
  - – Implementation of the throttling algorithm is now being shared by both OSPFv2 and OSPFv3
- Some enhancements have been made to the throttling logic and some bugs have been fixed ☺
  - – wait-interval is not being incremented if the event arrived after the current wait-interval expired
  - – MaxAged LSAs are kept until their current wait-interval expires to prevent the LSA to be deleted and new instance generated before the wait interval expires
  - – wait-interval is only incremented only if new version of the LSA is flooded

OSPFv3 Fast Convergence (CLI)

- OSPFv3 Router mode

  timers throttle lsa <start-interval> <hold-interval> <max-interval>

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

  timers lsa arrival <interval>

  – all values are in milliseconds

OSPF
Event Logging

OSPF Event Logging

- OSPF Event Logging facility has been enhanced for both OSPFv2 and OSPFv3
- OSPF event log kept per OSPF instance (used to be global)
- Configurable size
- Timestamp for each event
- Event logging can be paused while preventing the content of the log
- User can choose cyclical or one-shot operation
- Event log can be cleared
- User can choose the order in which events are displayed
- Filtering of events during display of the content

OSPF Event Logging (cont.)

- Content of the log was historically difficult to read
- Extended with new event types in a human readable form ☺
  - neighbor state changes
  - interface state changes
  - new or changed LSA arrival
  - LSA generation
  - SPF scheduling
  - SPF start, plus start of various phases of SPF
  - OSPF route update/deletion in/from the RIB
  - arrival of the redistribution events from the RIB

OSPF Event Logging (CLI)

- Configuration under Router Mode for both OSPFv2 and OSPFv3

  [no] event-log [size [number of events]] [one-shot] [pause]

- Exec mode

  clear ip ospf [process-id] events

  clear ipv6 ospf [process-id] events

  show ip ospf [process-id] event [spf] [lsa] [rib] [generic] [interface] [neighbor] [reverse]

  show ipv6 ospf [process-id] event [spf] [lsa] [rib] [generic] [interface] [neighbor] [reverse]

Meet the Experts
IP and MPLS Infrastructure Evolution

- Andy Kessler
  Technical Leader
- Beau Williamson
  Consulting Engineer
- Benoit Lourdelet
  IP services Product manager
- Bertrand Duvivier
  Consulting Systems Engineer
- Bruce Davie
  Cisco Fellow
- Bruce Pinsky
  Distinguished Support Engineer






Meet the Experts
IP and MPLS Infrastructure Evolution

- Gunter Van de Velde
  Technical Leader
- John Evans
  Distinguished Systems Engineer
- Oliver Boehmer
  Network Consulting Engineer
- Patrice Bellagamba
  Consulting Engineer
- Shannon McFarland
  Technical Leader





Meet the Experts
IP and MPLS Infrastructure Evolution

- Andres Gasson
  Consulting Systems Engineer

- Steve Simlo
  Consulting Engineer

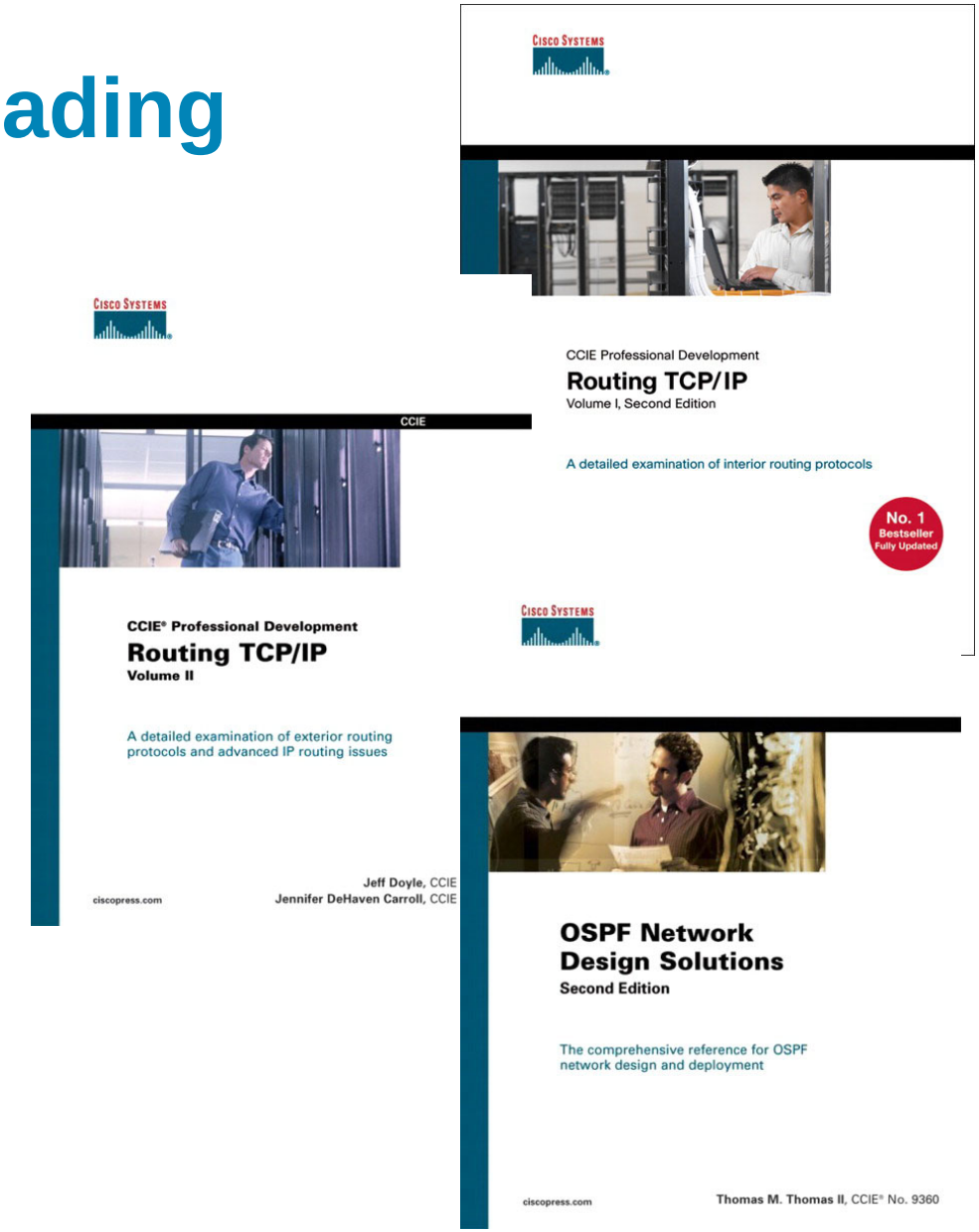
- Toerless Eckert
  Technical Leader

- Dino Farinacci
  Cisco Fellow & Senior Software Engineer

Recommended Reading

BRKIPM -3006

- Routing TCP/IP, Volume I
- Routing TCP/IP, Volume II
- OSPF Network Design Solutions



Available in the Cisco Company Store

Q and A