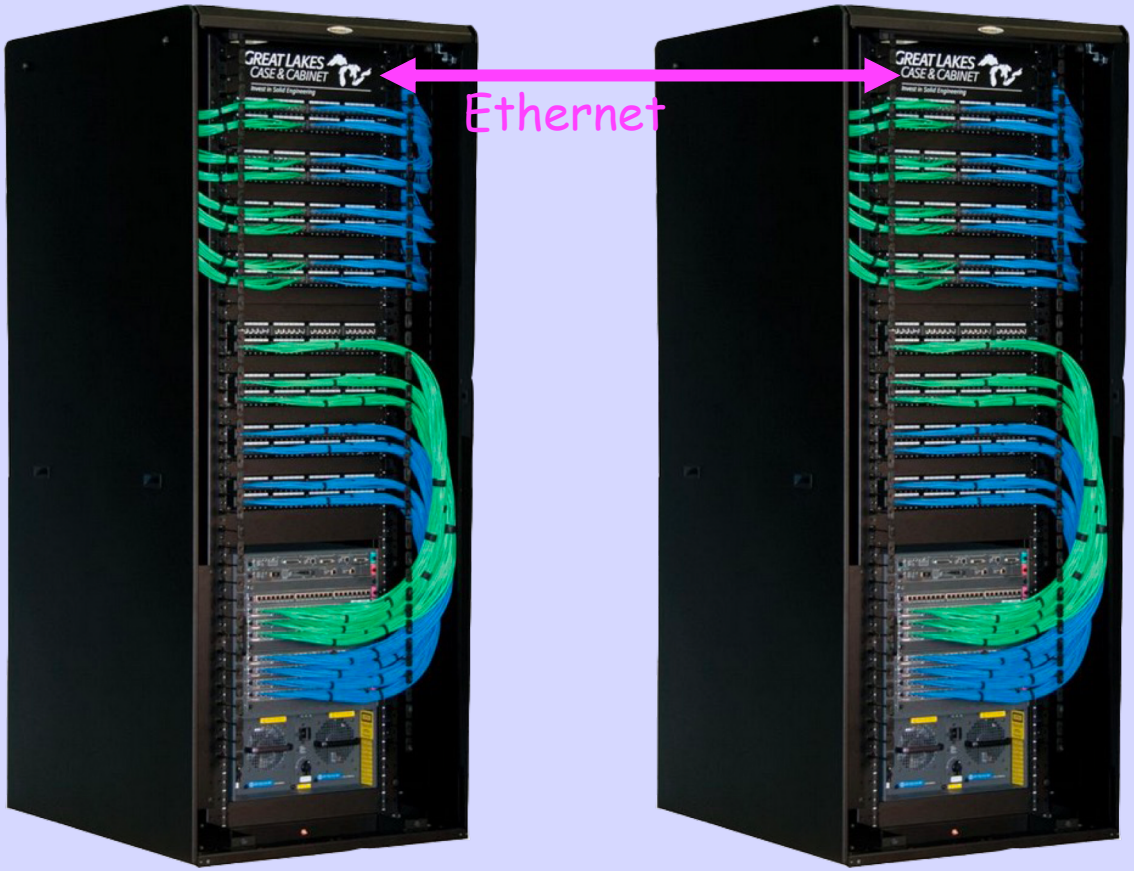
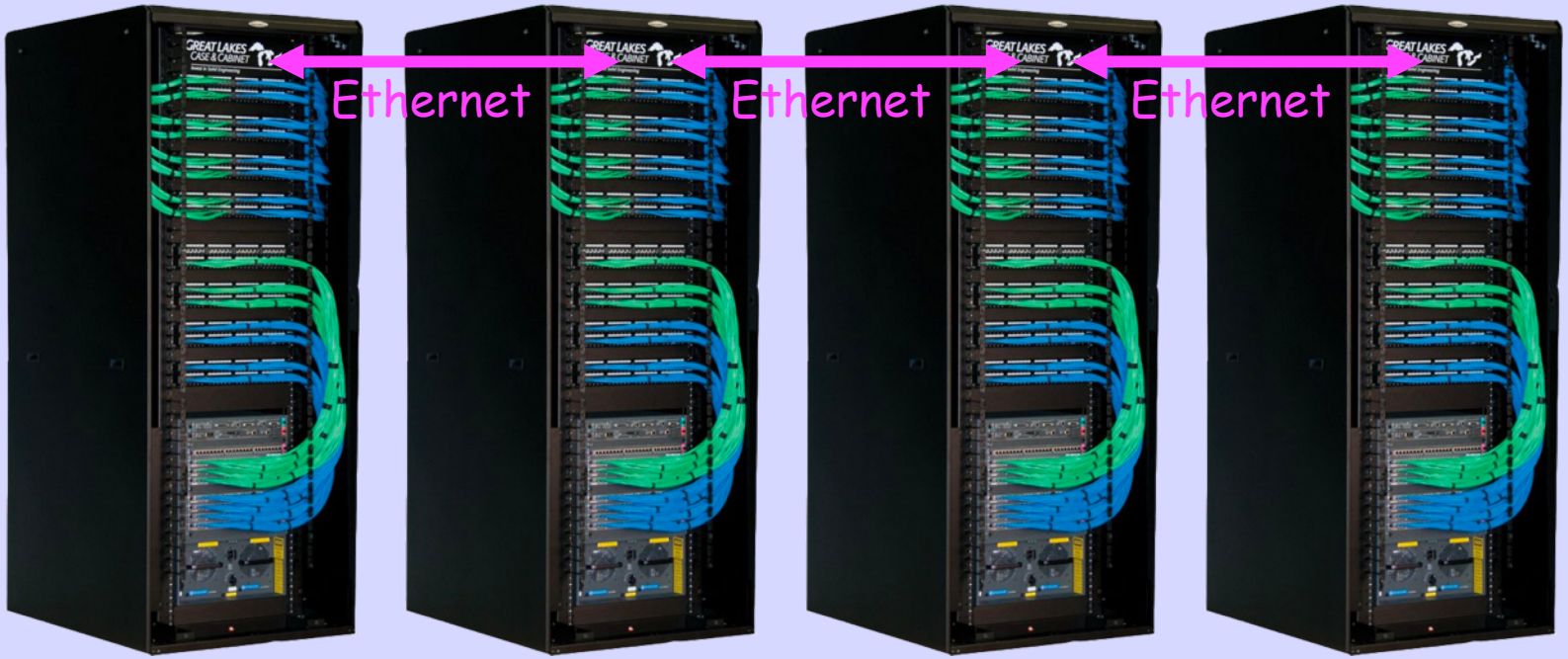
An Approach to Routing in a Clos

Randy Bush <randy@~~psg~~bogus.com>
IIJ Lab & Arrcus
with Keyur Patel, Arrcus
and a cast of dozens

This Works

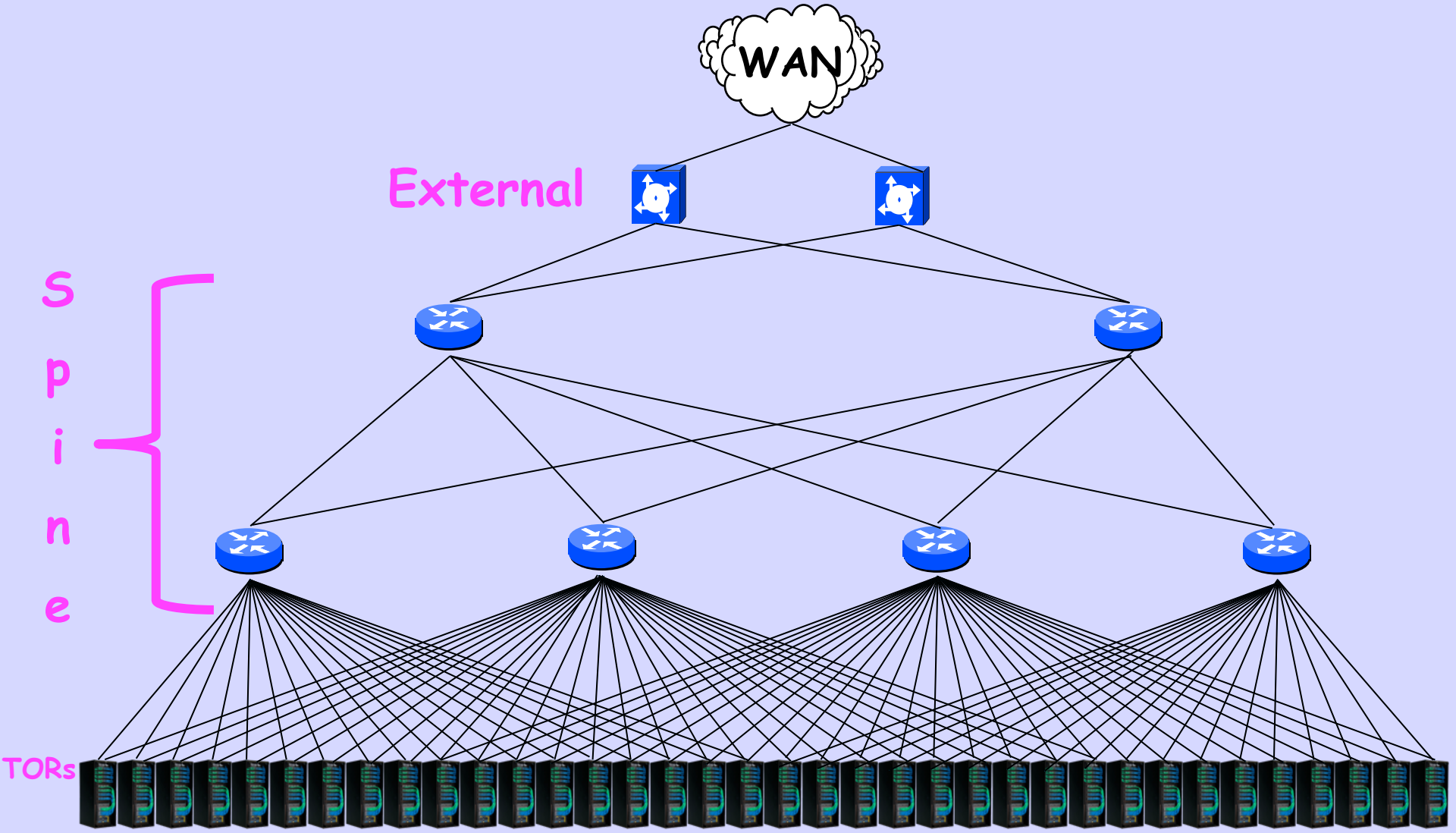


This Might Work



This Won't Work

This Works (Clos Network)



Clos is Not an Acronym

Clos, Charles (Mar 1953)
"A study of non-blocking switching networks"
Bell System Technical Journal. 32 (2): 406–424

For Example:

IIJ is Building a Second Medium Scale Data Center (MSDC) in Shiroi/Chiba Capacity of 6k Racks

How Do You Route In Something of This Scale?

OSPF OK to 500 Nodes

IS-IS good to 1,000

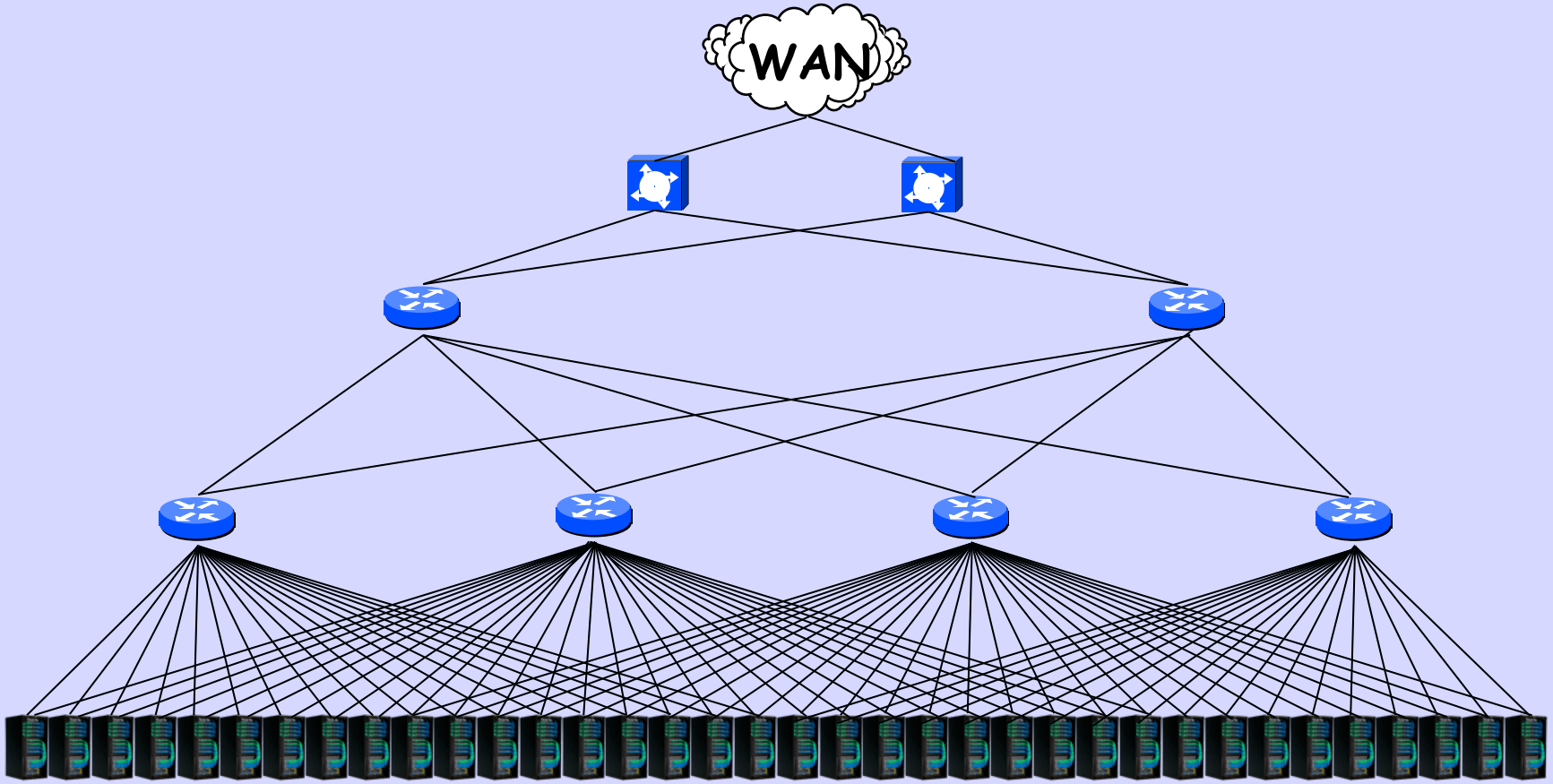
Limited Because They Repeatedly Flood Everything

Your Clos on IS-IS or OSPF

BGP Scales Because It Signals Only Changes

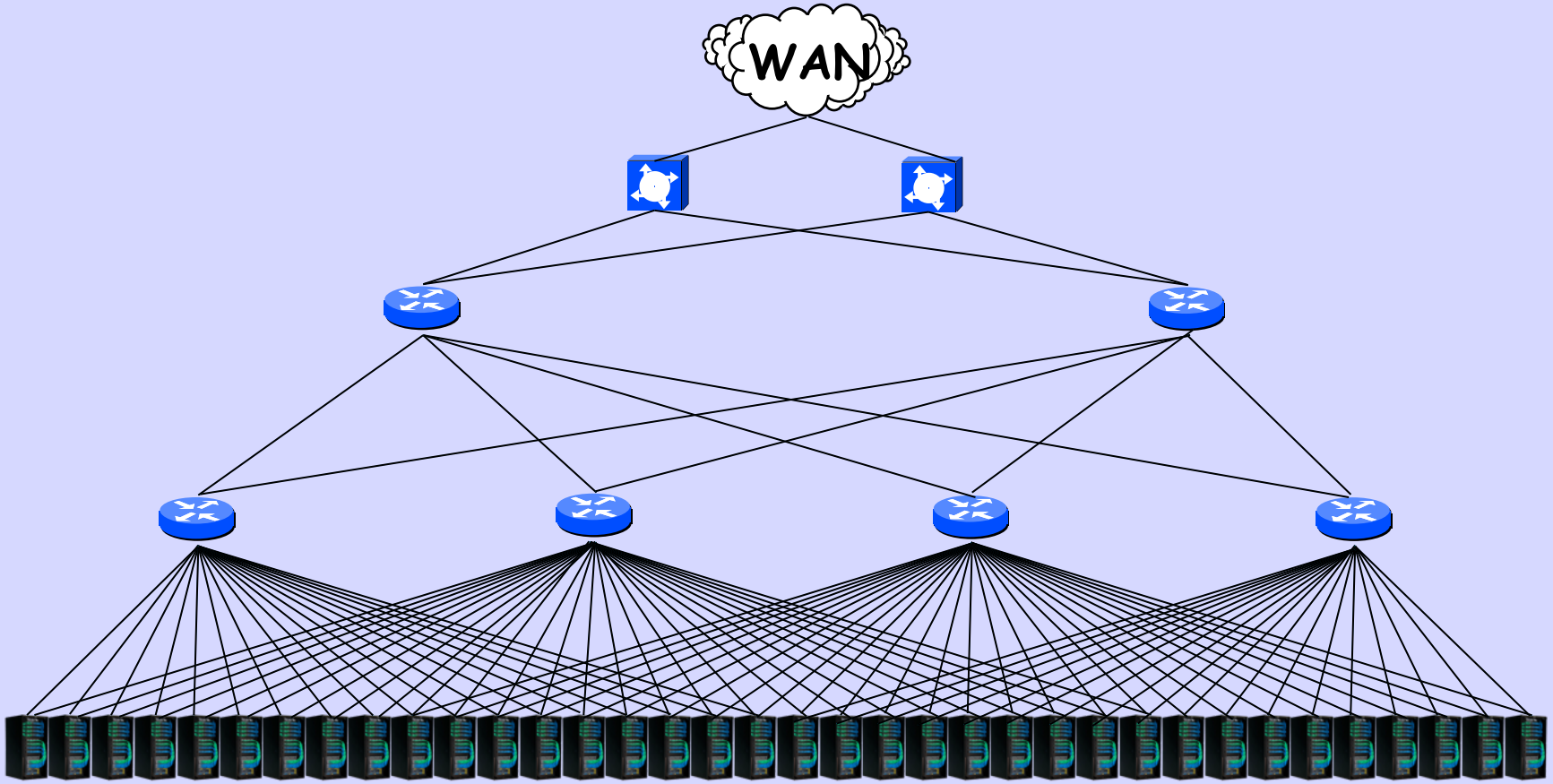
So BGP has become common in MSDCs

BGP Is Great as Updates are Infrequent

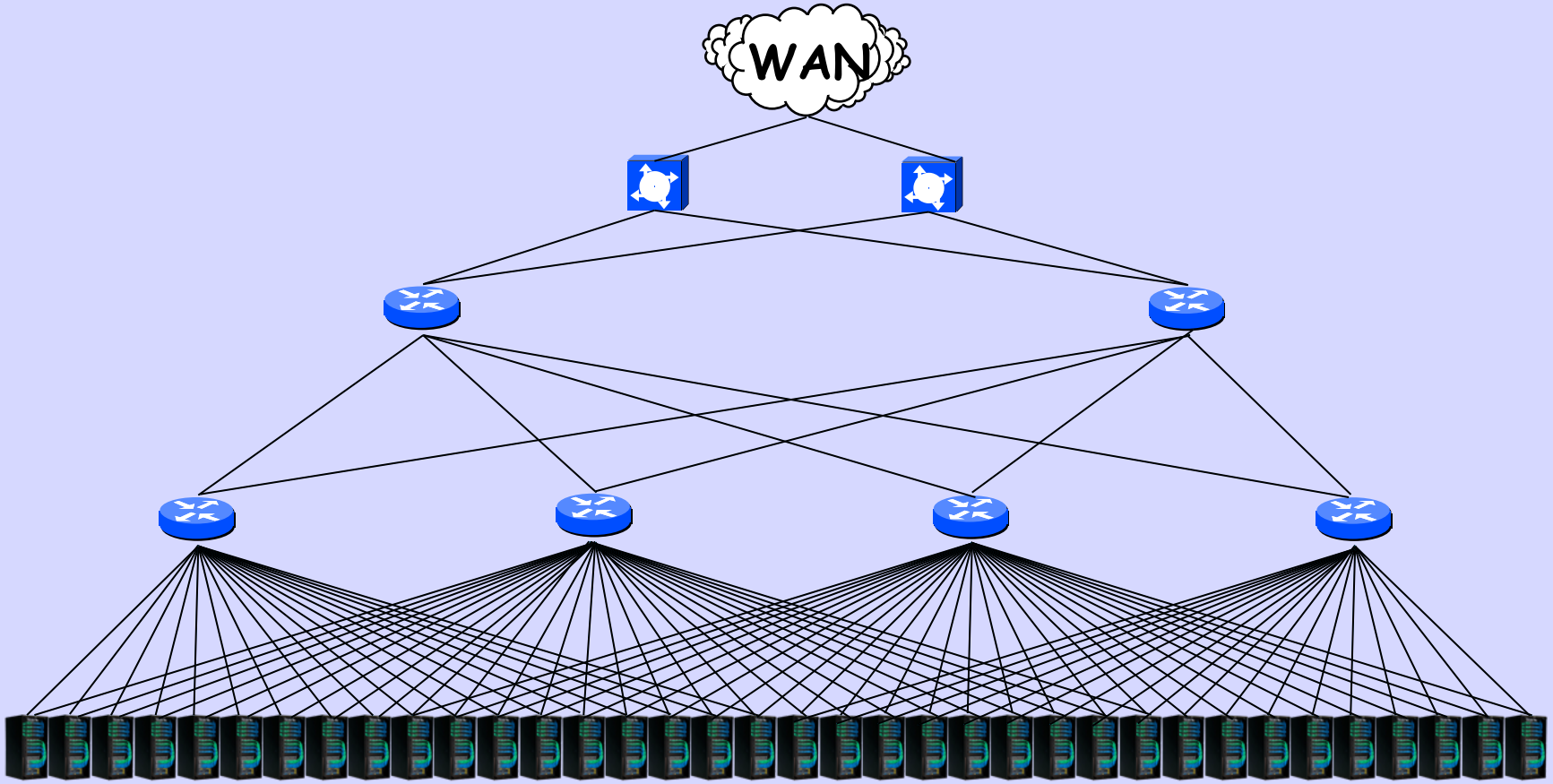


ECMP can be Very Wide

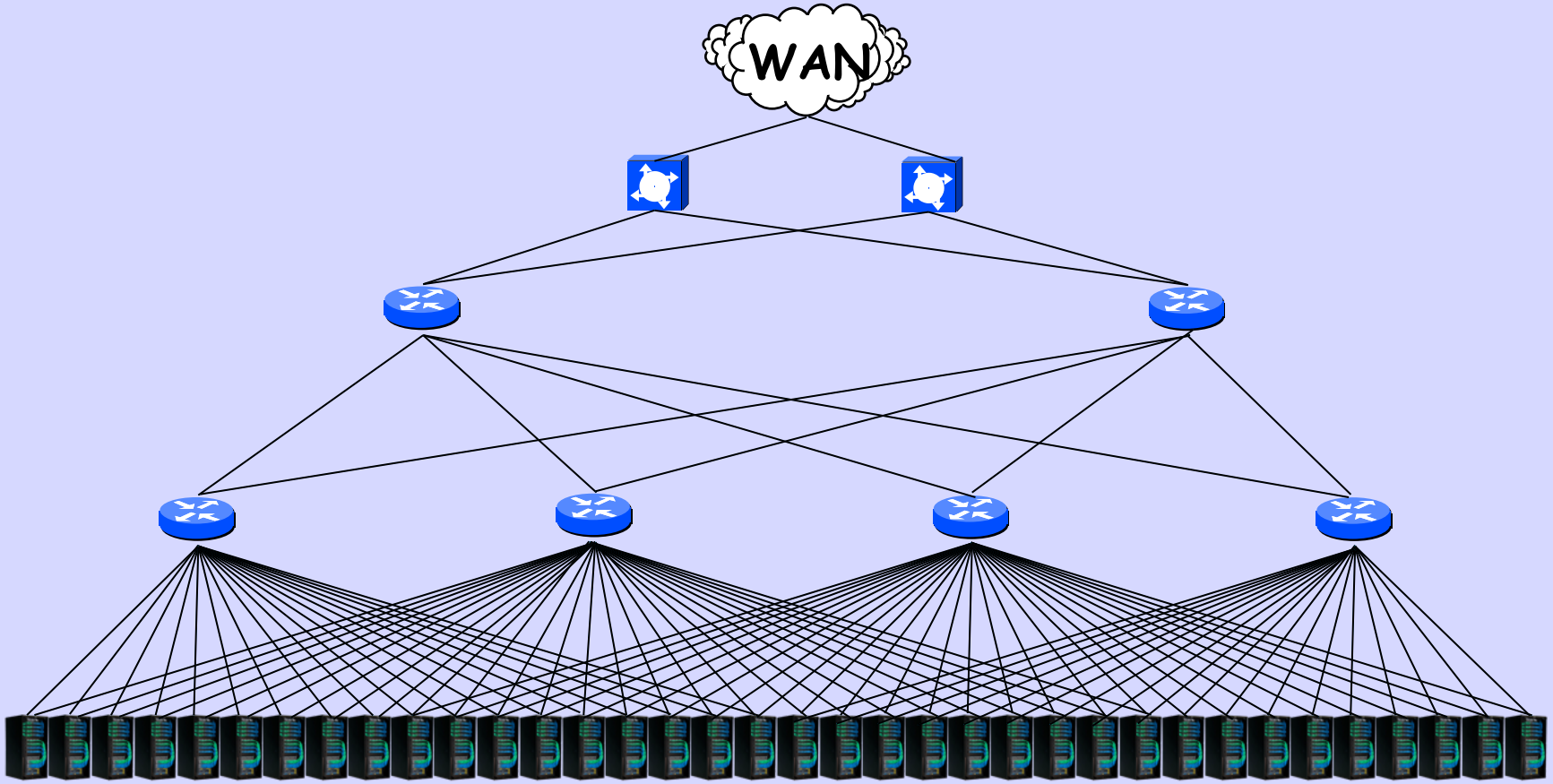
32, 64, even 128



But What is the Decision Process?



Do You Want to Write BGP Policy for Massive ECMP?



Consult the Professor

Edsger W Dijkstra
1930-2002

Shortest Path First

BGP-SPF

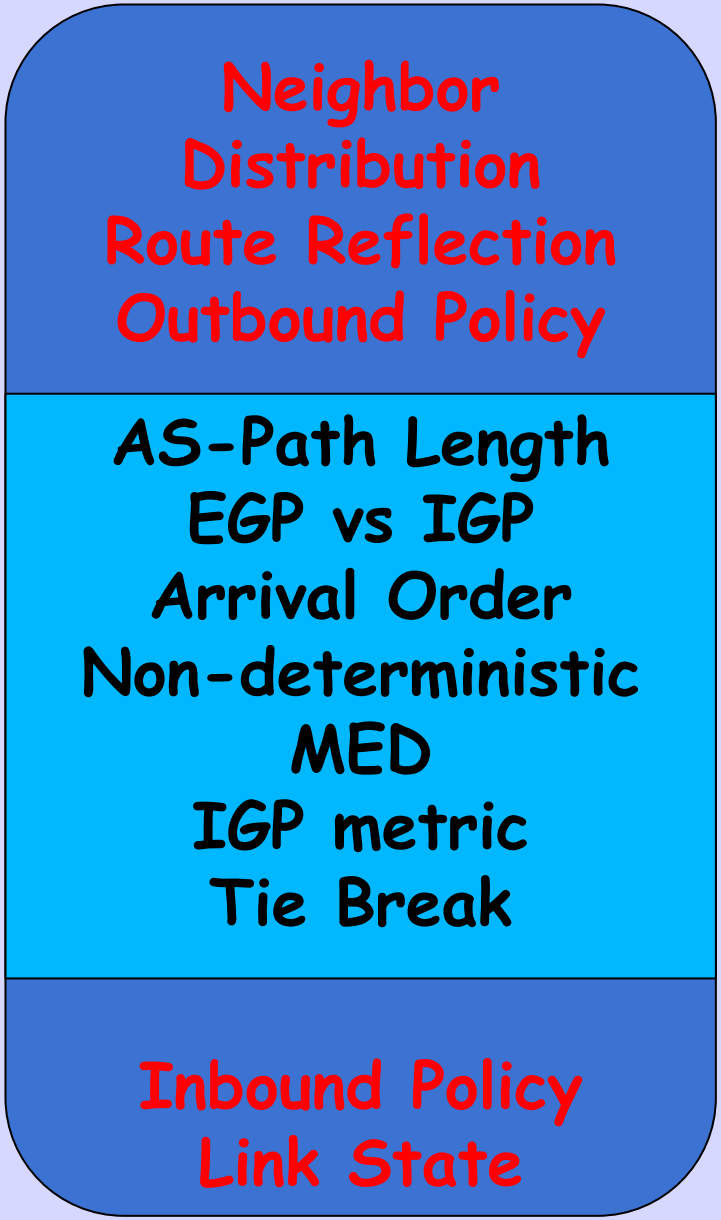
The Path Calculation of IS-IS With the Update Rate of BGP

SPF?

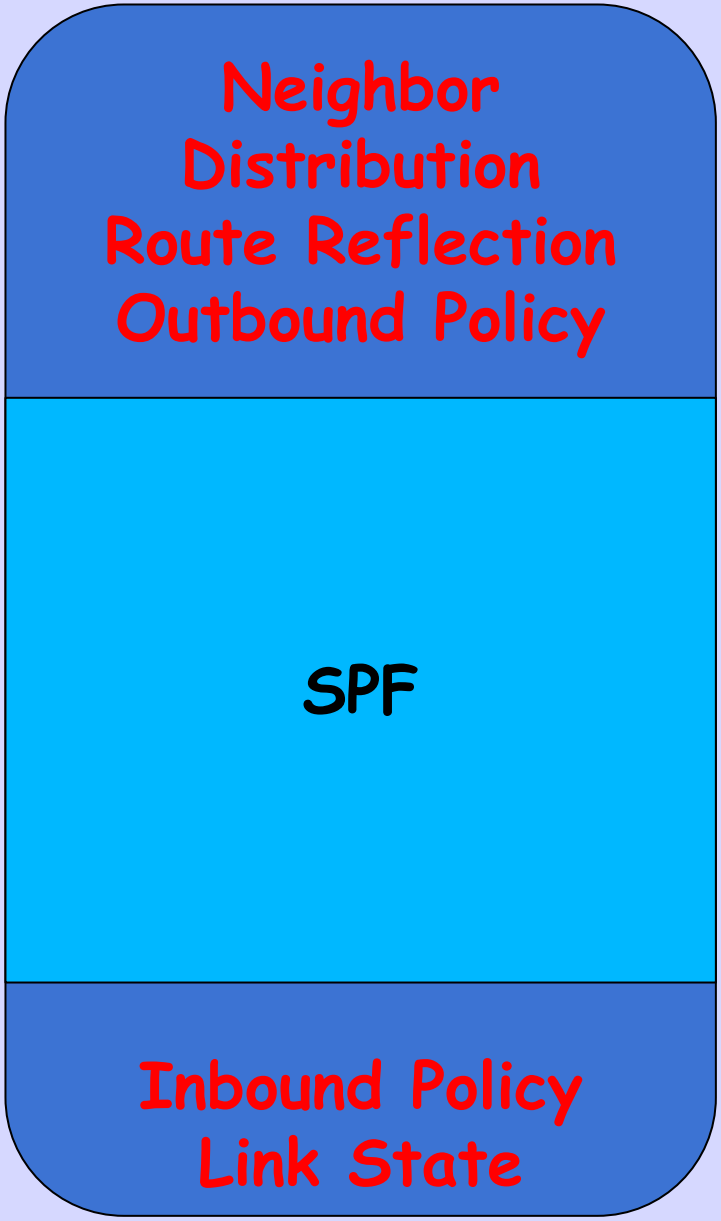
I thought BGP was path vector, not link state!

s/Best Path/SPF/

- New SAFI
- NLRI format exactly same as BGP LS (RFC 7752) Address Family to carry link state information
- BGP runs Dijkstra instead of Best Path Decision process
- BGP MP (new SAFI) and BGP-LS Node attribute for compatibility
- Peering Models: eBGP, iBGP, RR



BGP4 Classic



BGP-SPF

AS-Path Length
EGP vs IGP
Arrival Order
Non-deterministic
MED
IGP metric
Tie Break

Removed!

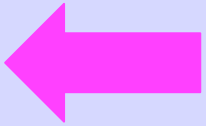
BGP-SPF

- Next-Hop and Path Attributes come for free with BGP Link-State Address Family
  - Needed for RFC 4271 error handling
- Decision Process Phases 1 and 2 (best path) replaced by SPF algorithm (AKA Dijkstra)
- Decision Process Phase 3 (tie break) may be skipped as NLRI is unique per BGP speaker
- Need to assure the most recent version of NLRI is always used and re-advertised
  - Augmented with sequence numbers

BGP-SPF

- Starting with greatly simplified SPF with P2P only links in single area (i.e., SPT)
- Should scale very well to many use cases
- Could support computation of LFAs, Segment Routing SIDs, and other IGP features
  - BGP-LS format includes necessary Link-State
- Link-State AF is dual-stack AF since both IPv4 and IPv6 addresses/prefixes advertised
  - BGP-LS format also supports VPNs but SPF behavior not defined
  - Work needed to define interaction with existing unicast AFs
    - Matter of local implementation policy

Peering Model

- BGP sessions, optionally with Route-Reflector or controller hierarchy
  - Link discovery/liveliness detection outside of BGP 
- RR hierarchy can be less than fully connected but must provide redundancy
  - Must not be dependent on SPF for connectivity
- Controller could learn the expected topology through some other means and inject it
  - SPF Computation is distributed though
  - Similar to "Jupiter Rising: A Decade of Clos Topologies and Centralized Control in Google's Datacenter Network"

BTW, Every Rack is (usually) an AS

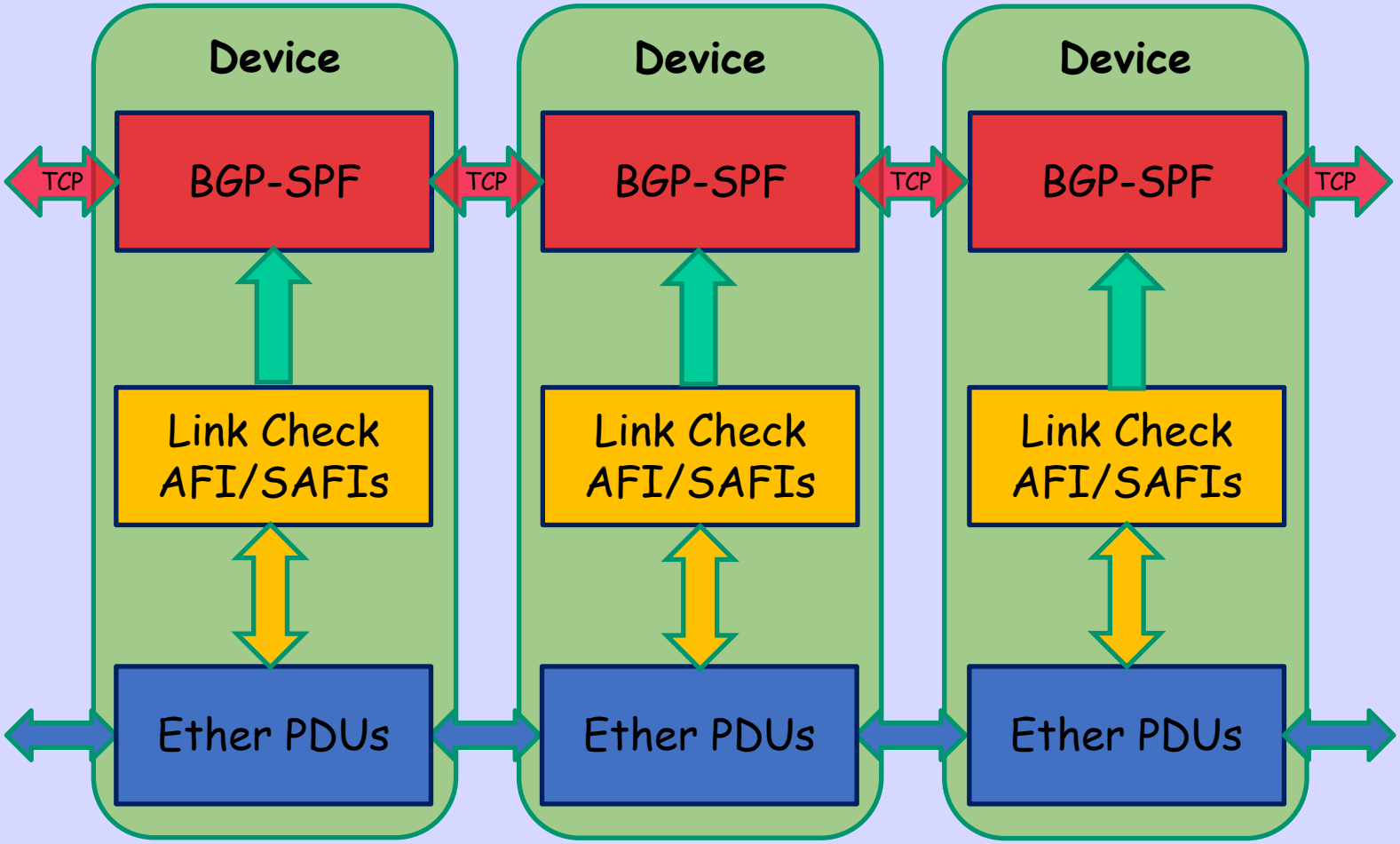
Get Over It

How Does BGP-SPF Learn Link State?

Motivation

- BGP-SPF needs link neighbor discovery, liveness, and addressability
- LLDP is an IEEE protocol, complex, and 'hard' (IPR) to extend past 1500 bytes
- We wanted something simple and saw no real need for the complexities of CLNP, …
- So we propose a new EtherType with TLVs
- We discuss Ether payloads, not framing

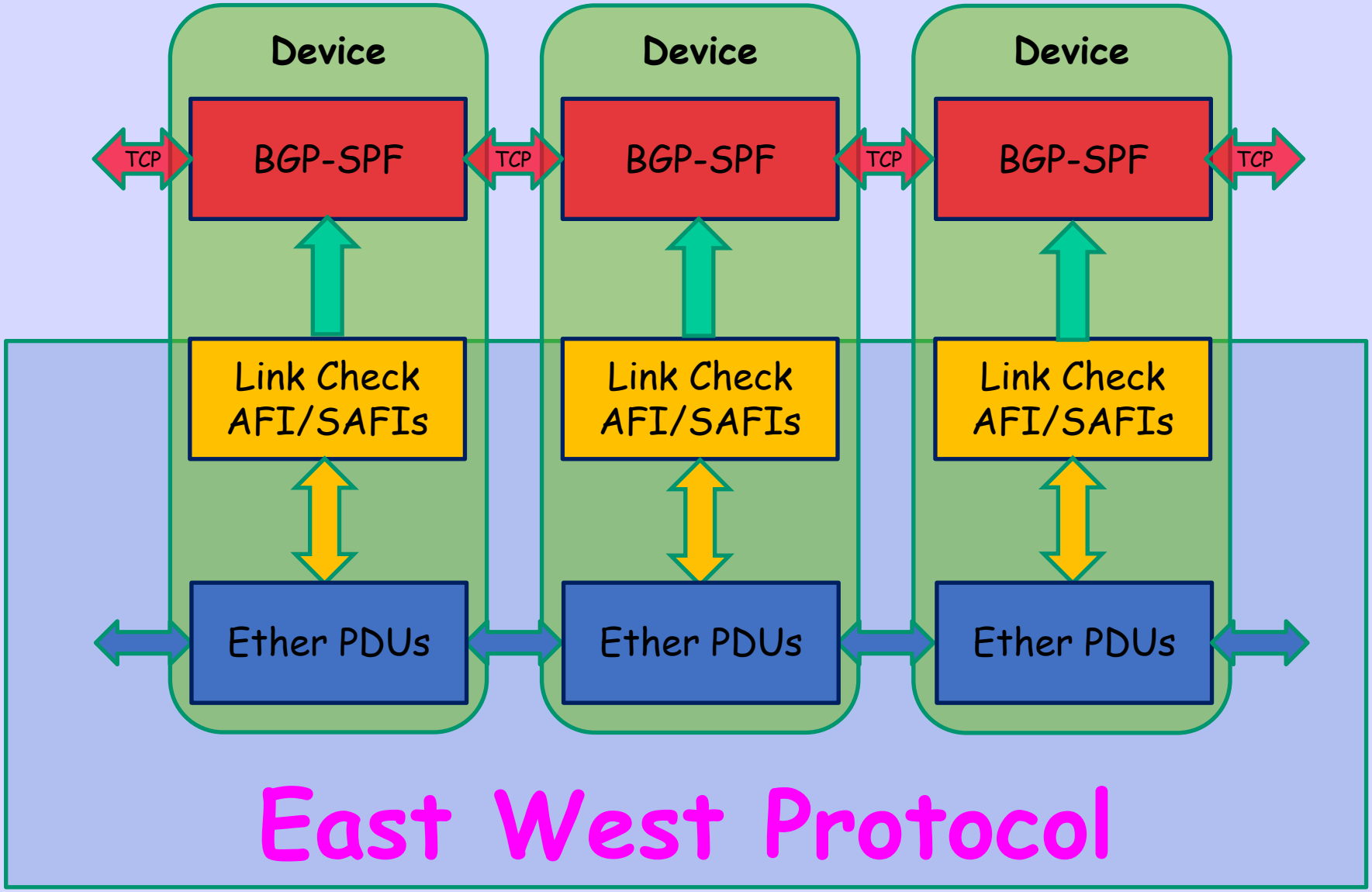
Topology / Routing Stack



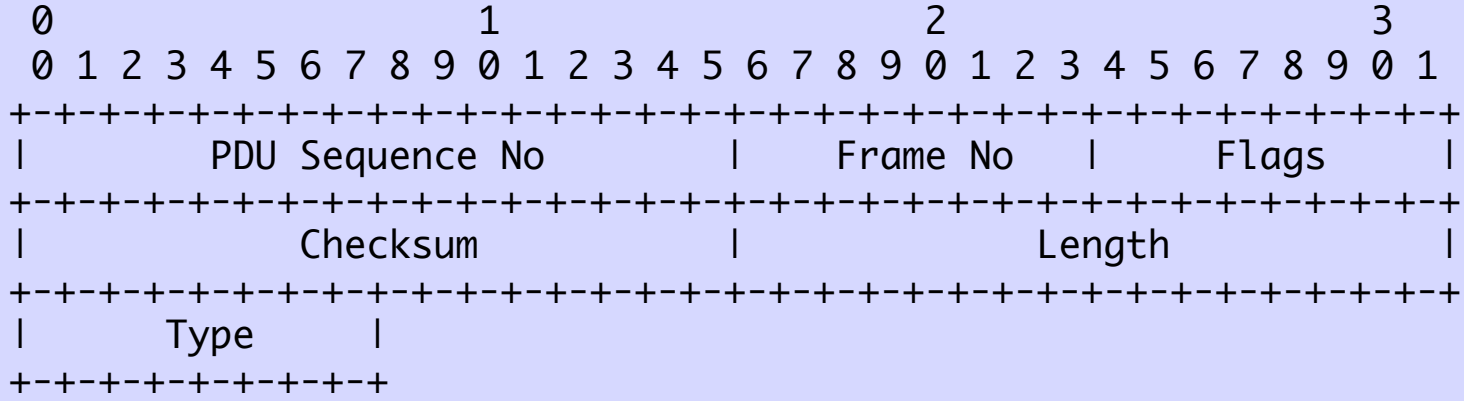
MAC Link State exchanged over raw Ethernet and pushed up stack

Add the AFI/SAFI data IP-Level Liveness Check

BGP-SPF uses link data to discover and build the topology database

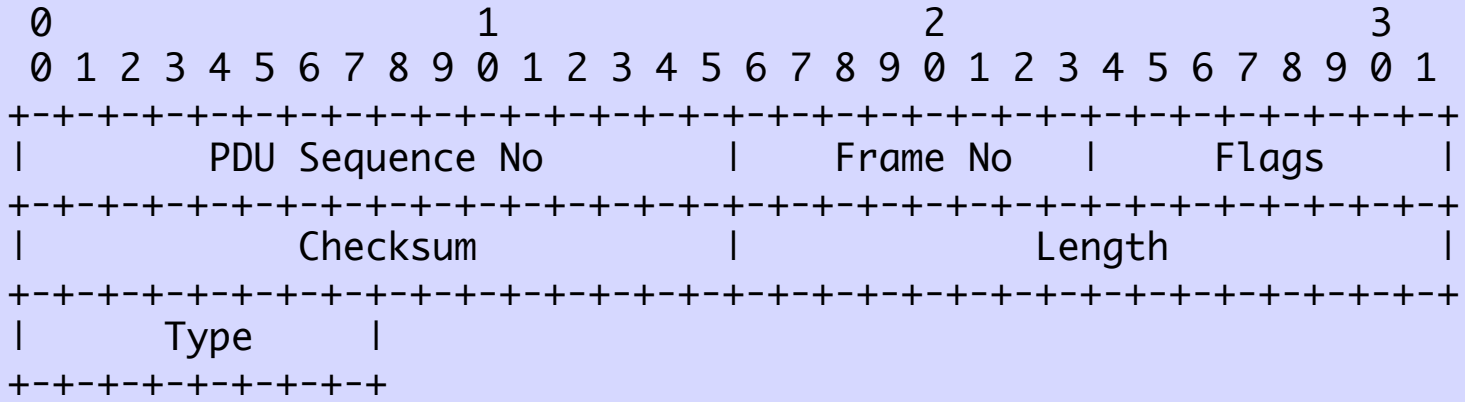


PDUs and Frames



- This is all about inter-device Link State
- A PDU is one or more Ethernet Frames
- A Frame has *PDU Sequence No* and a *Frame No* to allow assembly of out order frames
- Because BGP-SPF/IGP and Data Plane payloads are assumed to be IP over the same Ethernet, one worries about congestion

Every Frame a TLV



PDU Sequence No - Semi-unique identifier of a TLV PDU (e.g. UNIX time)

Frame No - 0..255 Frame Sequence Number Within a multi-frame PDU

Flags (bits)
- 0 – Sender has been restarted
- 1 – one of a multi-Frame sequence
- 2 – last of a multi-Frame sequence

Checksum - one’s complement over Frame, detect bit flips

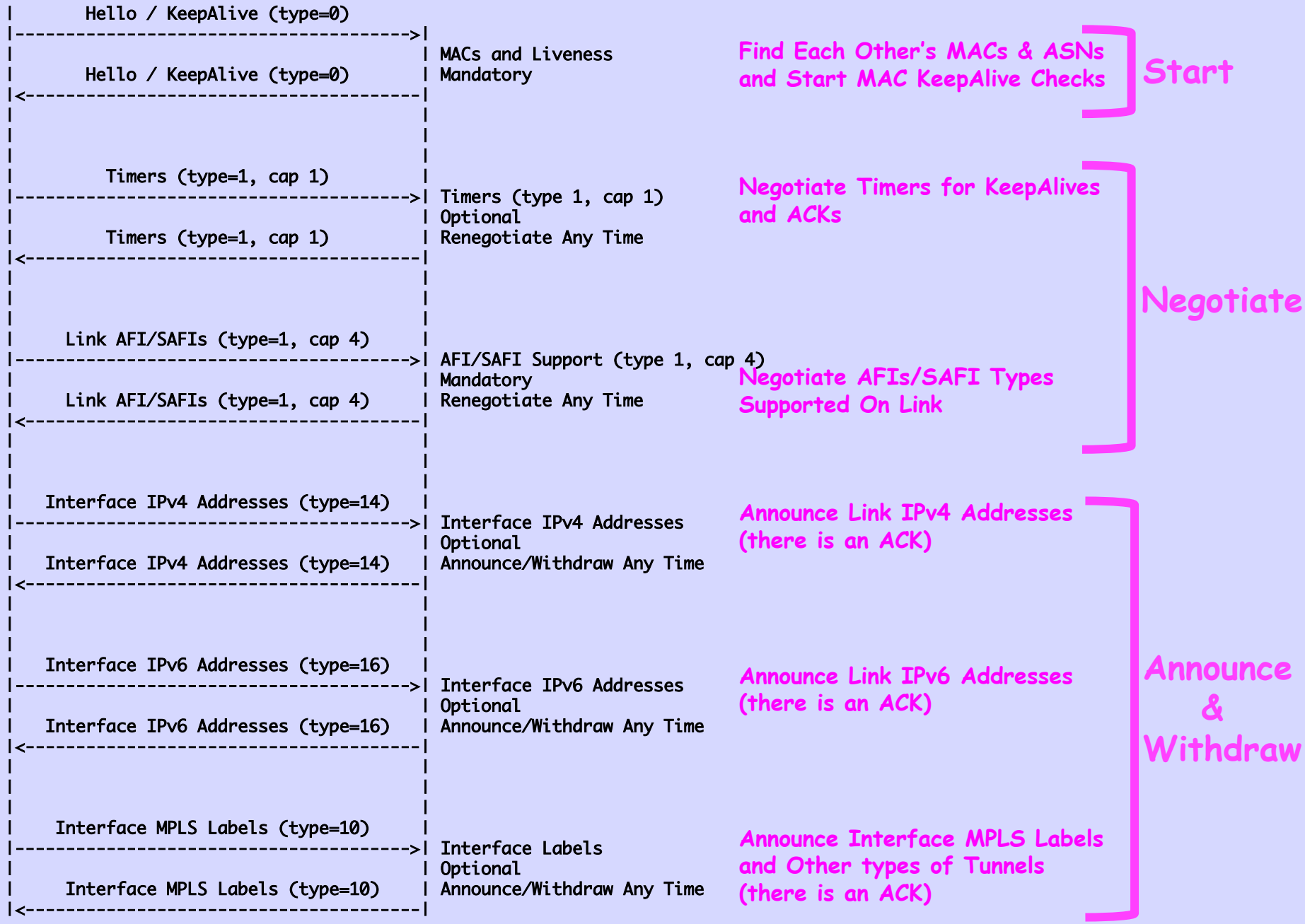
Length - Total Bytes in PDU including all frames and fields

Type (int)
- 0 – Hello / KeepAlive
- 1 – Capability

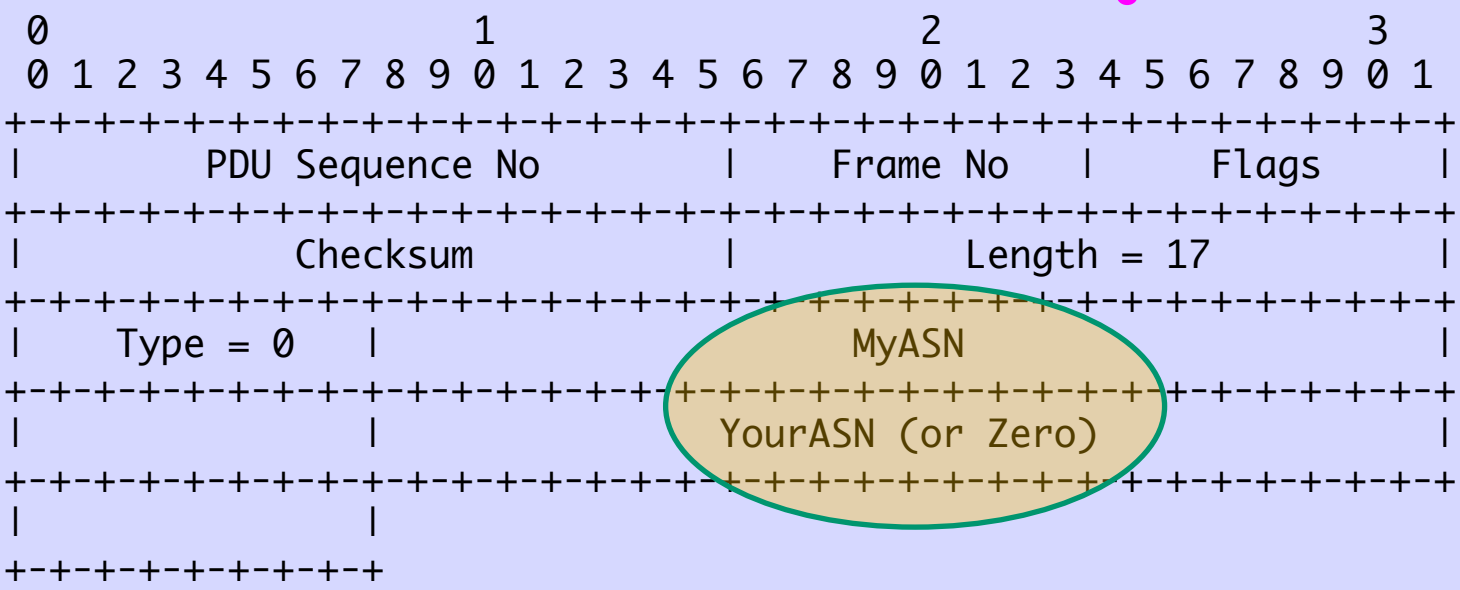
Checksum

- There is a reason conservative folk use a checksum in UDP
- And when the op stretches to jumbo frames …
- One's complement is a bit silly, though trivial to implement
- Sum up either 16-bit shorts in a 32-bit int, or 32-bit ints in a 64-bit long, then take the high-order section, shift it right, rotate, add it in, repeat until zero. -- smb off the top of his head

Inter-Link Ether Protocol



Link Hello / KeepAlive



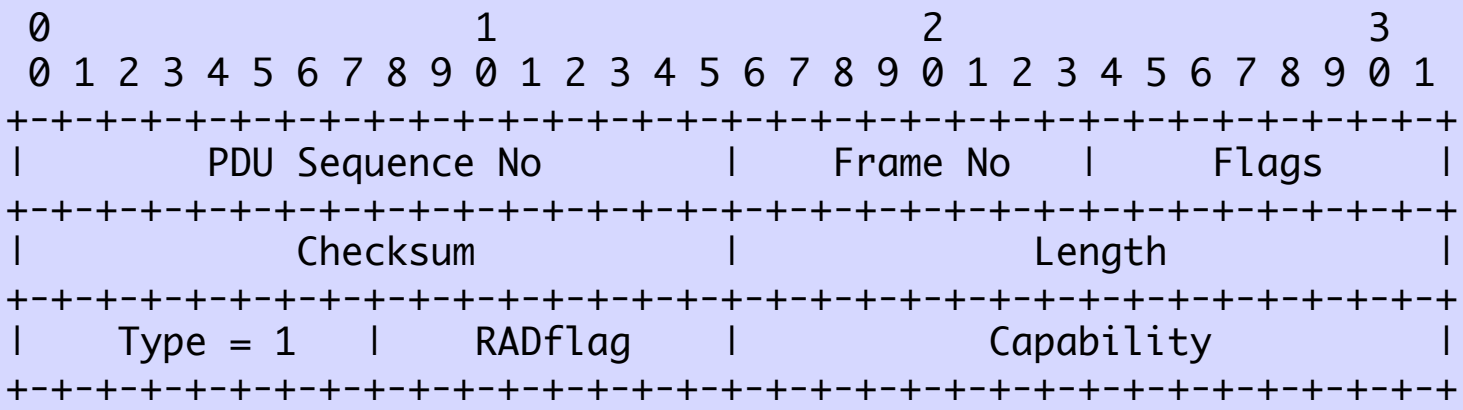
Type (int) - 0 – Hello / KeepAlive

- A multi-point topology is a set of point-to-point links
- Each device learns the other's MAC from its HELLO whining. All devices on a wire/interface know each others MACs and learn each other's ASNs

Once We Know Each Other's MACs

Ether Level KeepAlives are Started

Capability Negotiation (lowest MAC initiates)

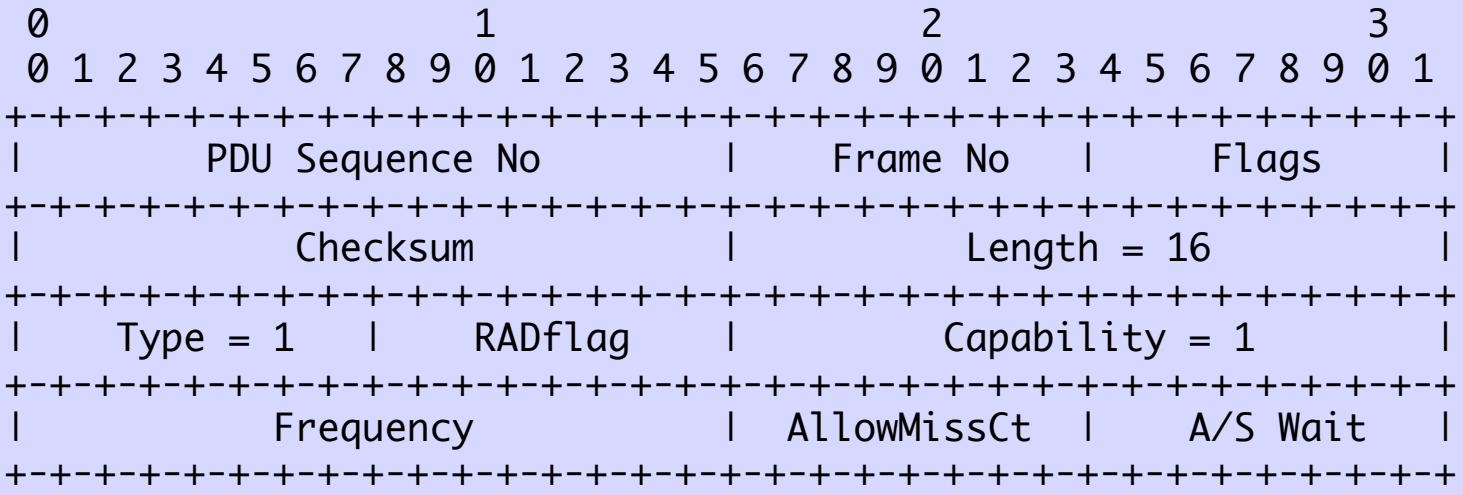


Type (int) - 1 Capability

RADflag (int) - 1 – Request
- 2 – Agree
- 3 - Deny

Capability - See Dictionary

Timer Negotiation



RADflag (int) - 1 – Request
- 2 – Agree
- 3 - Deny

Capability - 1

Frequency - Seconds/10 between KeepAlives (Default is 600)

AllowMissCt - Number of missed KeepAlives before declared down

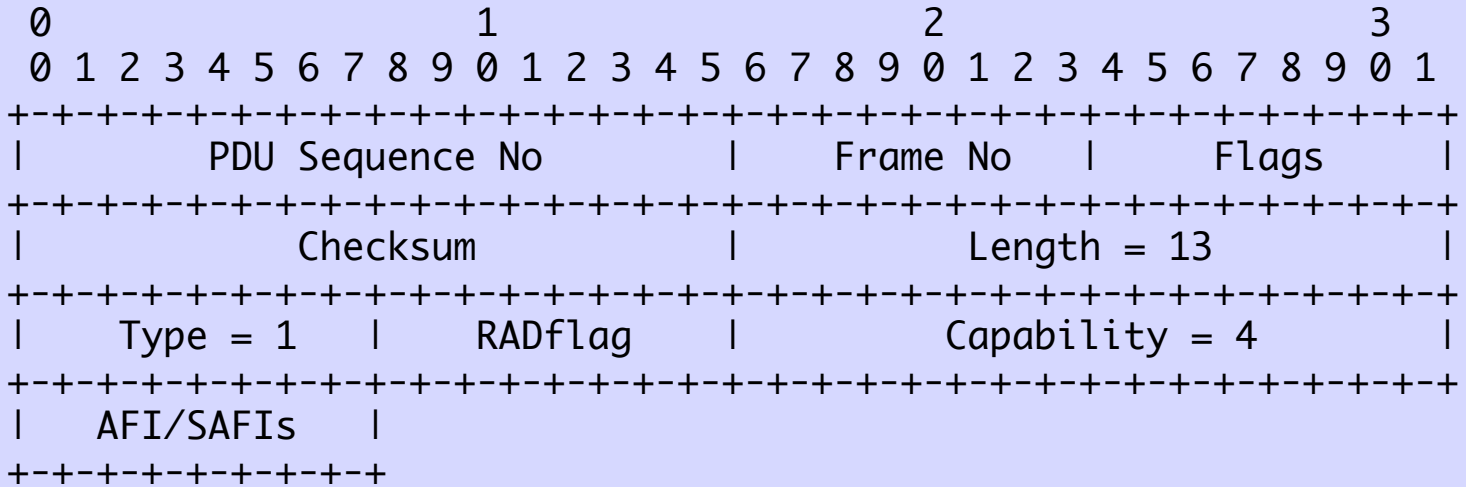
A/S Wait - AFI/SAFI ACK Timeout in Sec/10 (default 10)

We Know MAC/Ether Link State of This Device & Neighbor

And ASNs

Now Negotiate AFI/SAFIs of Link Interfaces

AFI/SAFI Capabilities



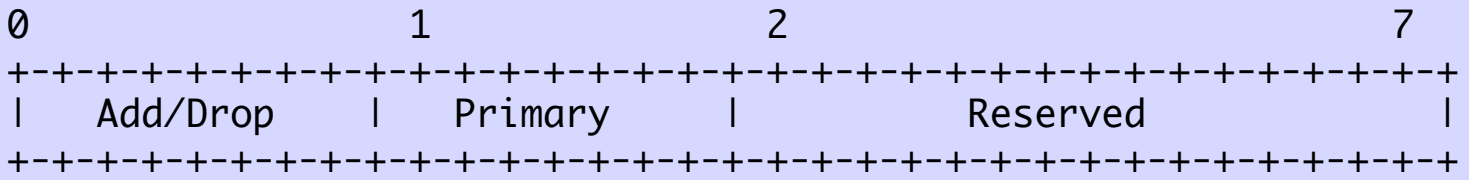
RADflag (int) - 1 – Request
- 2 – Agree
- 3 - Deny

Capability - 4

AFI/SAFI (int) – 10 – IPv4
- 11 – IPv6
- 12 – MPLS IPv4
- 13 – MPLS IPv6
- … other tunnels (e.g. GRE)

Now Both Sides Exchange Their Interface AFI/SAFI Configuration for the Negotiated AFI/SAFIs

Announce/Withdraw/Primary Flag



- An Interface may have multiple AFI/SAFIs
- For each AFI/SAFI there might be multiple Addresses
- One Address per AFI/SAFI SHOULD be marked as Primary
- It is a bit in the Announce/Withdraw Flag

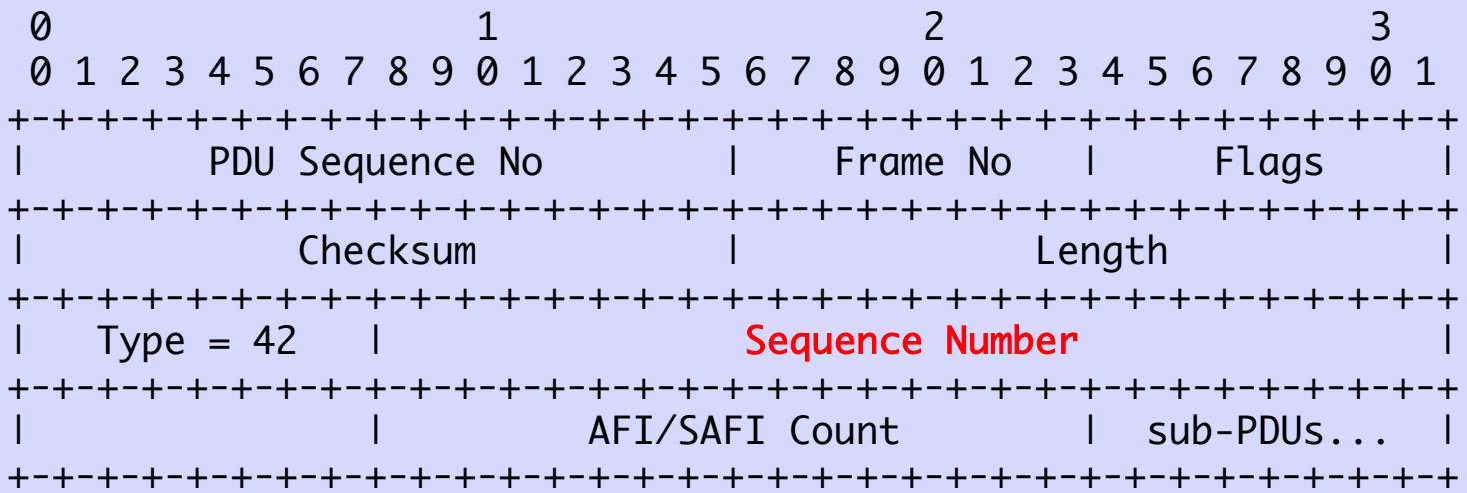
The AFI/SAFI Exchange

Is Over an Unreliable Transport

So There Are

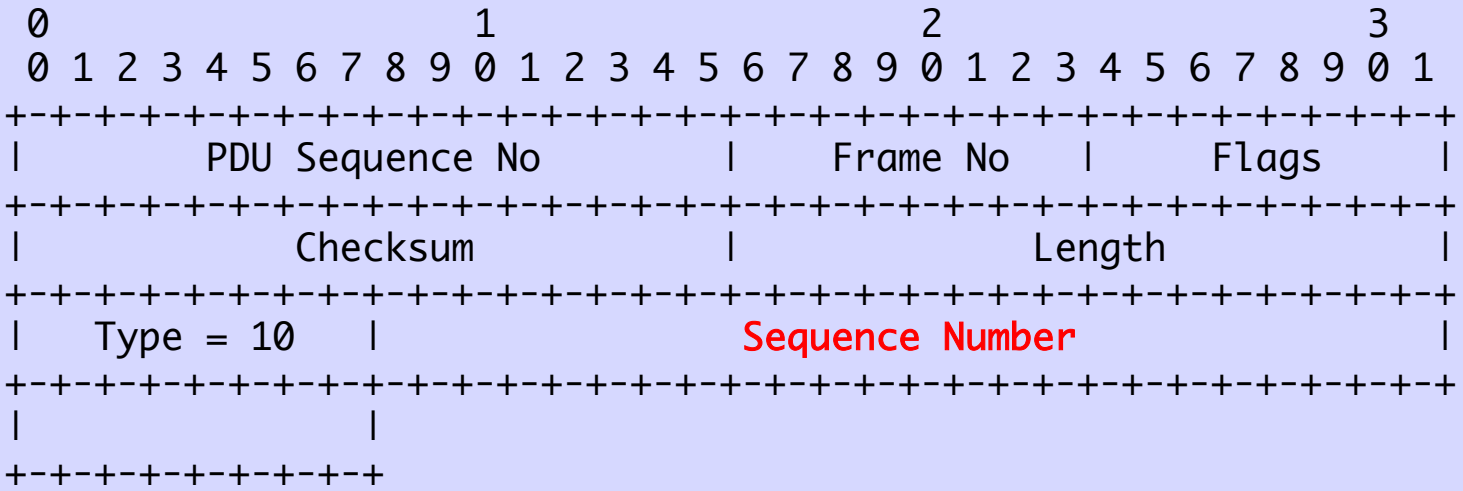
Sequence Numbers and ACKs

AFI/SAFI PDU Sequence Number



- The Sequence Number is a p2p link Announcement Counter
- The Receiver will ACK it with a Type=10
- If the Sender does not receive an ACK in one second, they retransmit. Other delay negotiated in Timing Capability

AFI/SAFI ACK

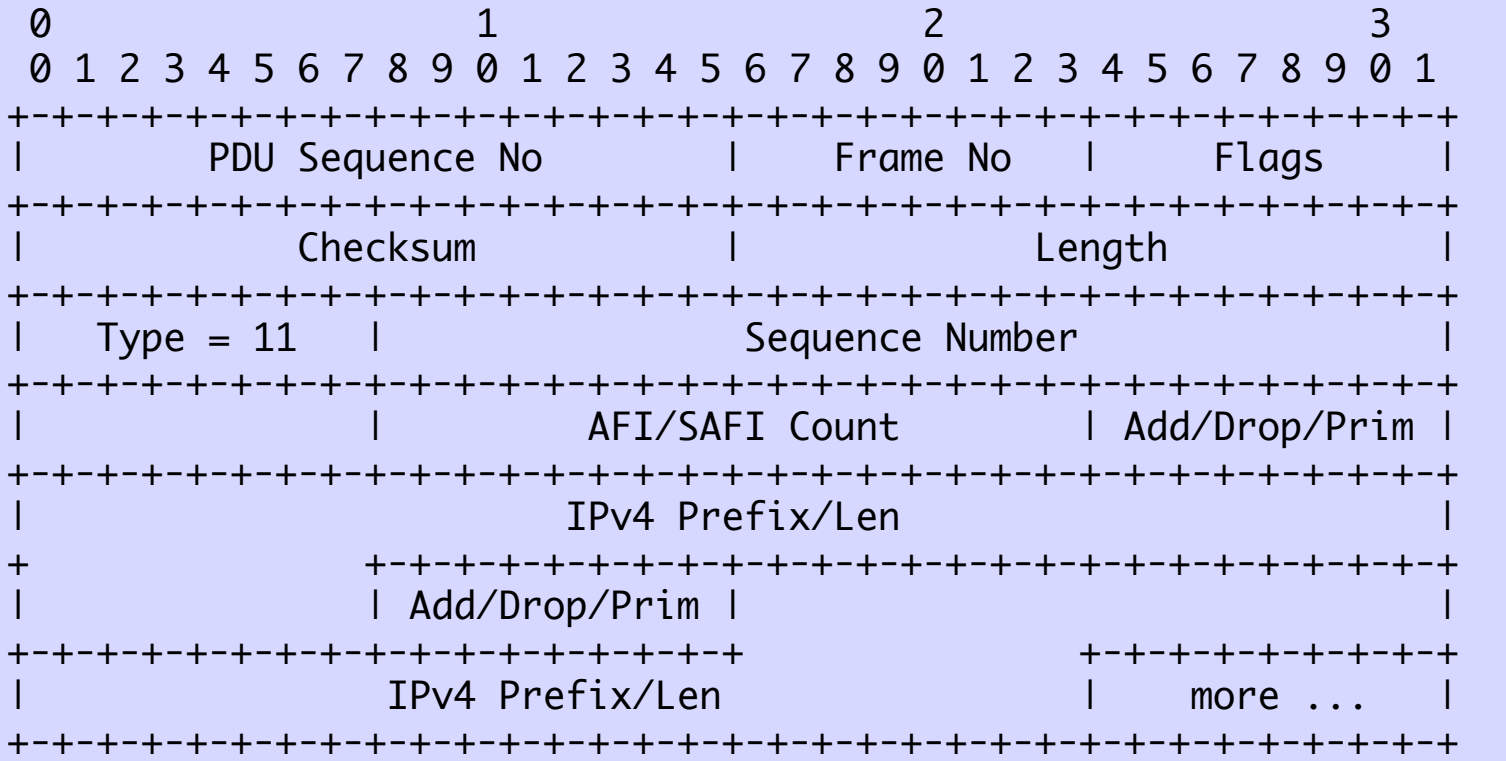


The sequence Number is the one being ACKed

Any PDU with a Sequence Number needs an ACK from the other end

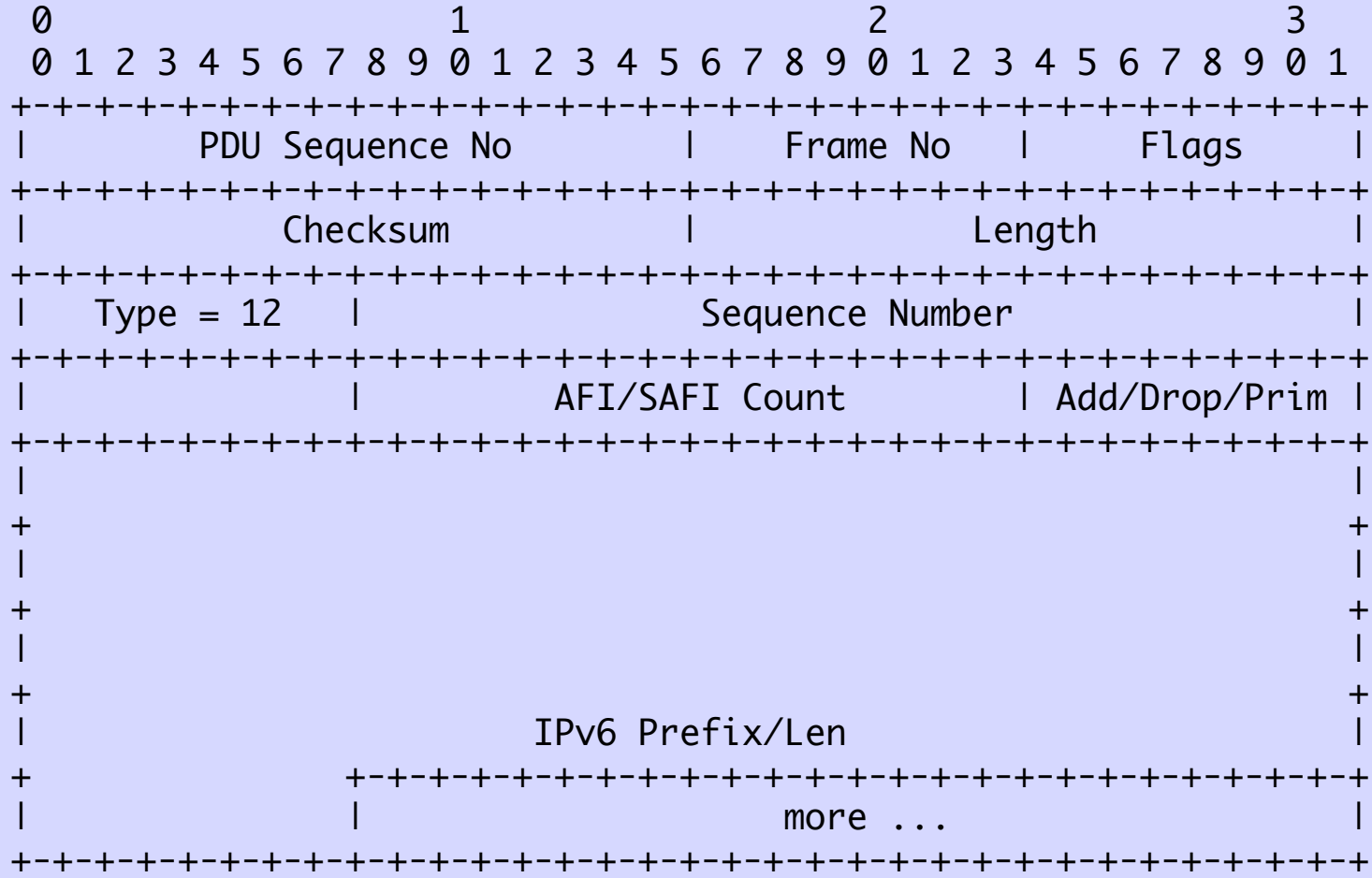
Any alignment freaks in the readership? It gets worse ☺

IPv4 Announce / Withdraw

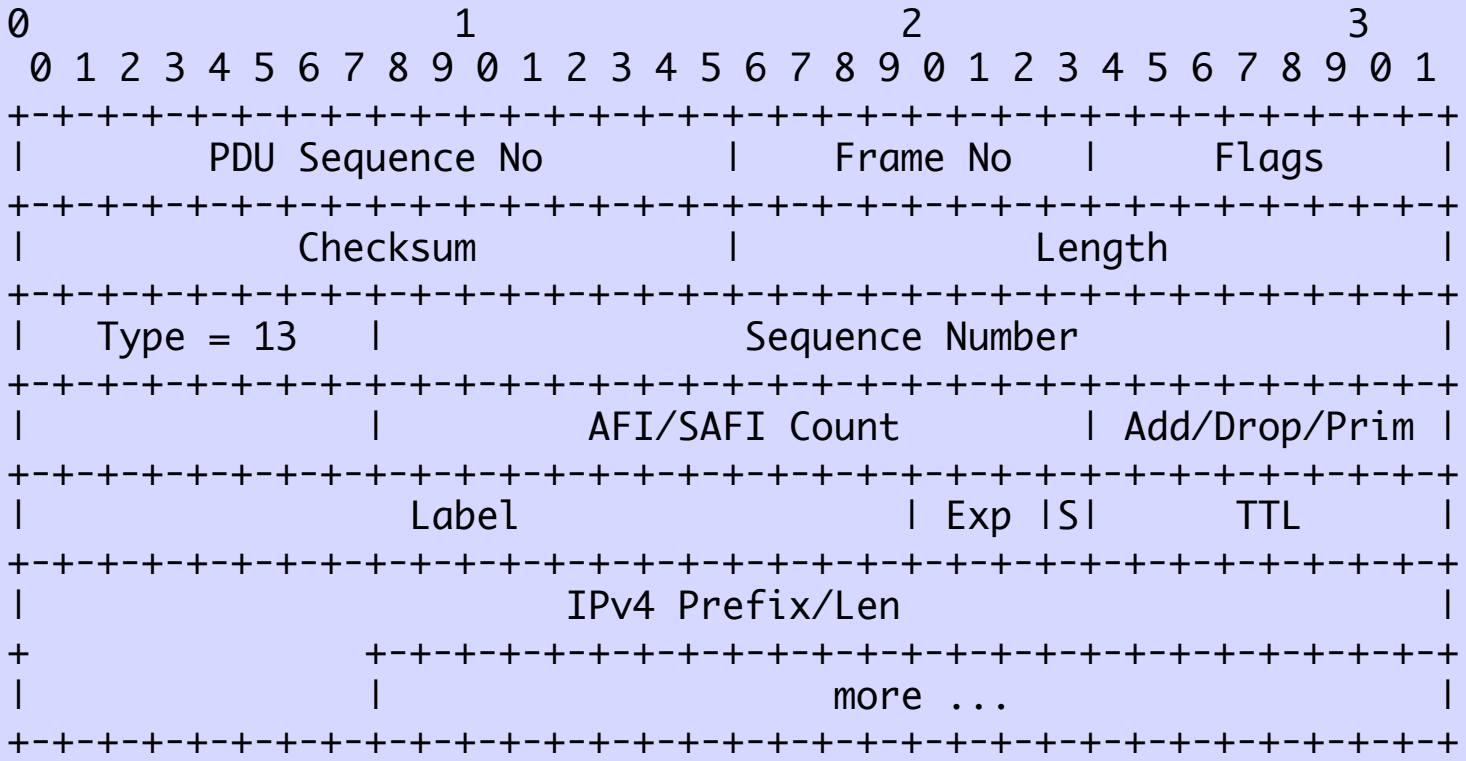


Add/Drop/Prim (bits) - 0 Announce / Withdraw
- 1 Primary of this AFI/SAFI on this Interface
- 2-7 Reserved

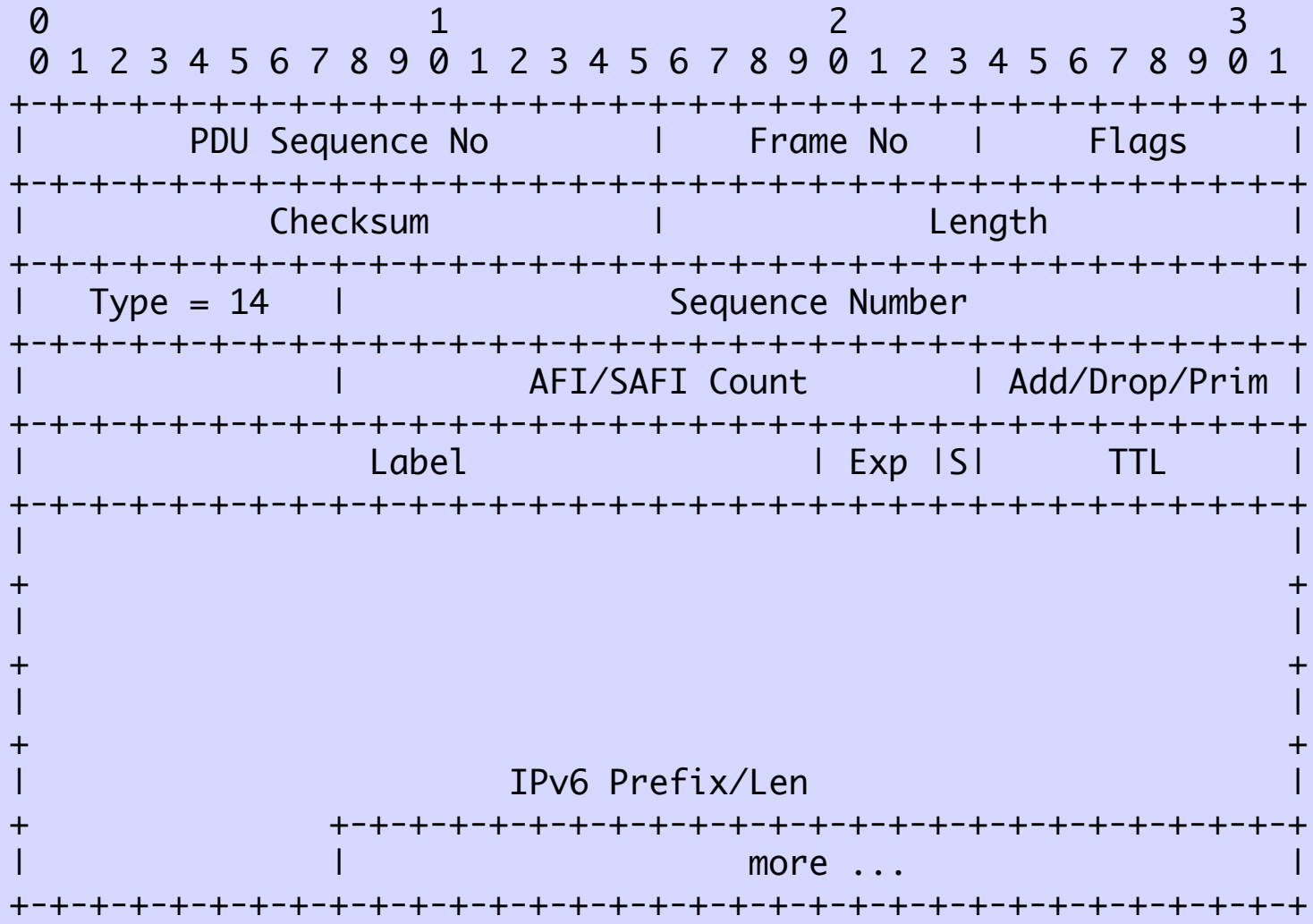
IPv6 Announce / Withdraw



MPLS IPv4 Announce / Withdraw



MPLS IPv6 Announce / Withdraw



Use Multiple MPLS Label PDUs to Allow One Label to be Associated with Multiple AFI/SAFIs and/or Multiple IP Addresses

Layer-3 IP/Label Liveness Should Also be Tested

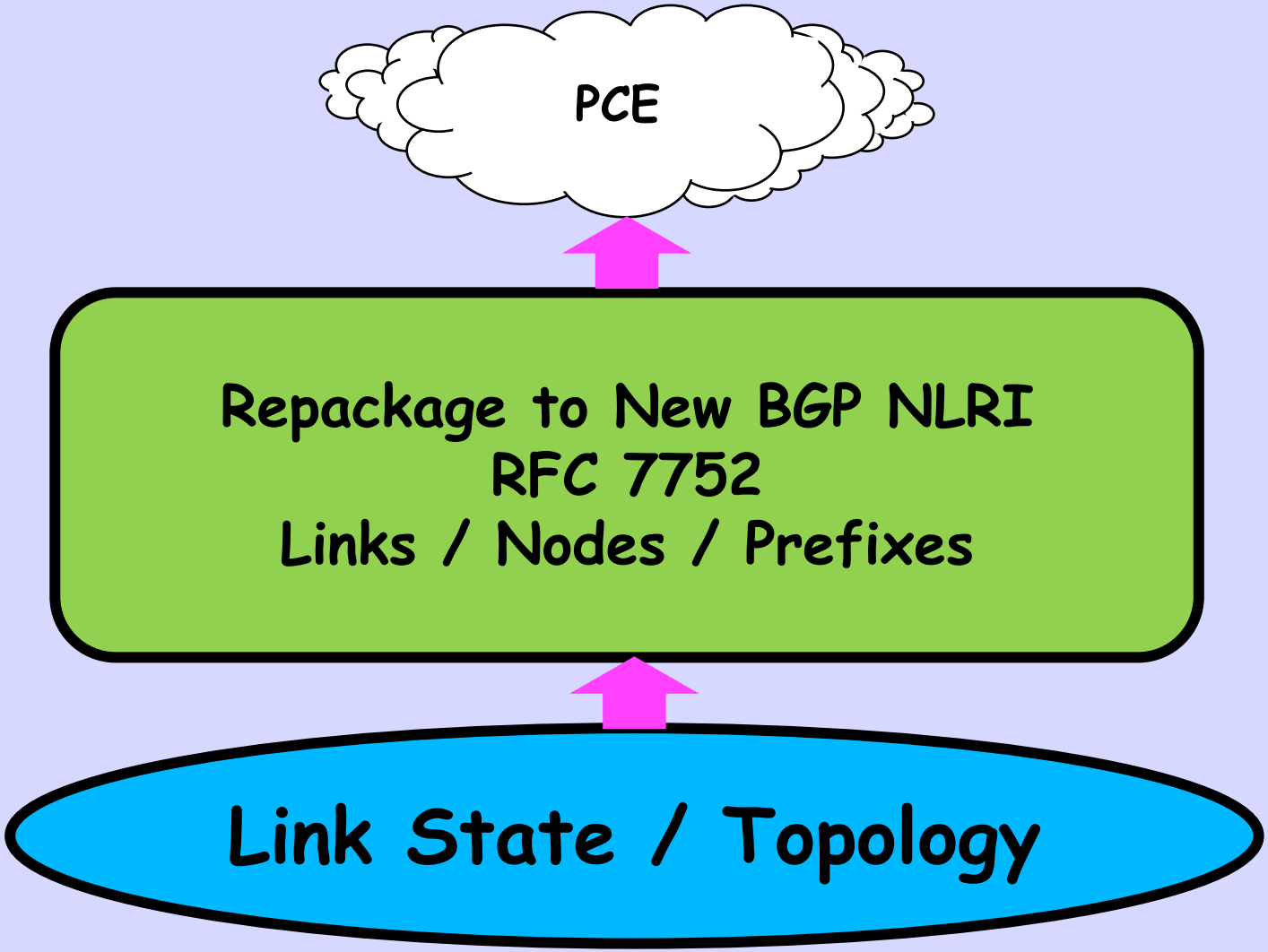
One or more Discovered AFI/SAFI Addresses Are Used to Ping, BFD, … to Assure Layer-3 Liveness

We Now Know all Links, ASNs, and AFI/SAFIs of This Device

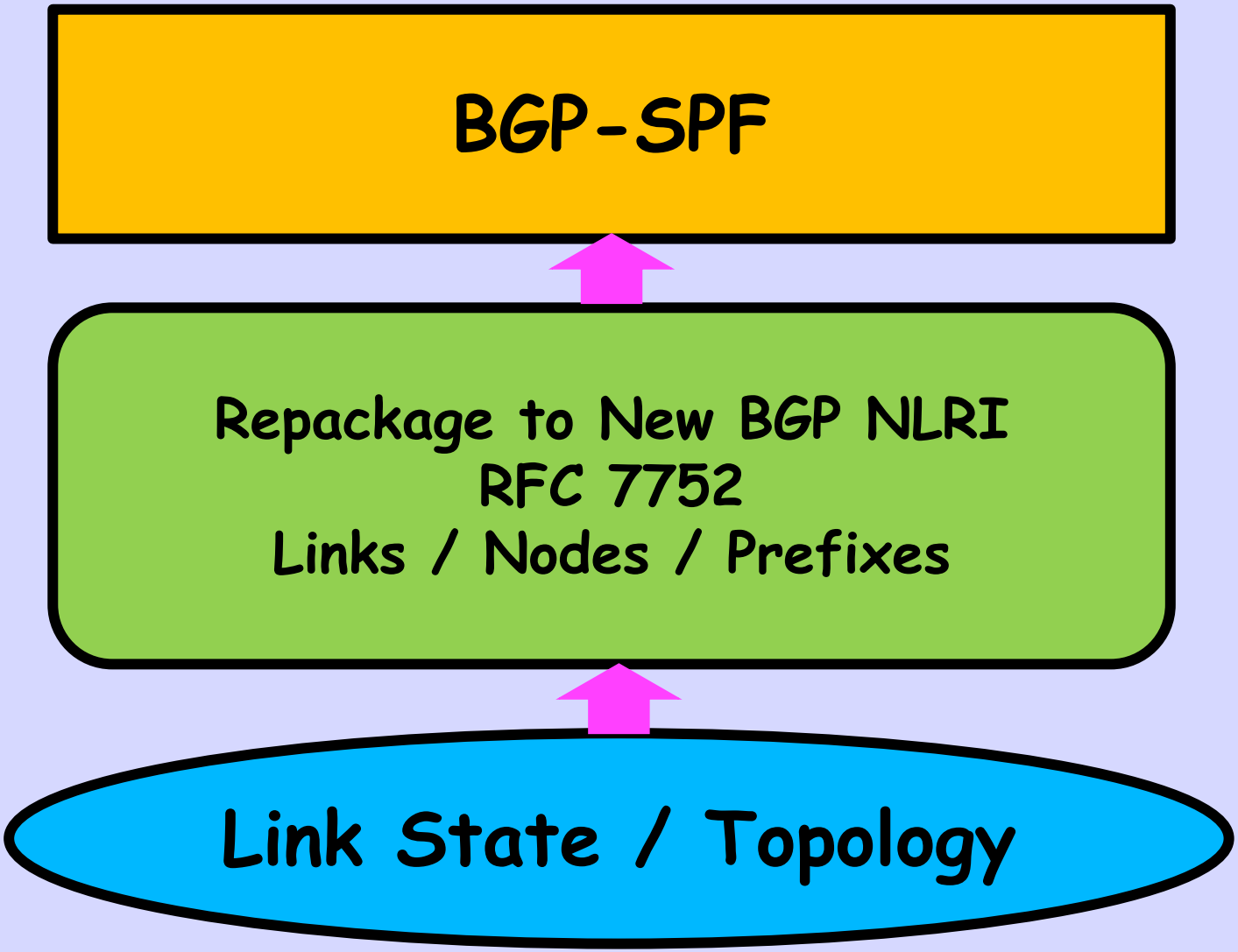
Now Push it Up to Topology and Dijkstra Layers

BGP-LS (RFC 7752) an extension to BGP to distribute the network's link-state (LS) topology

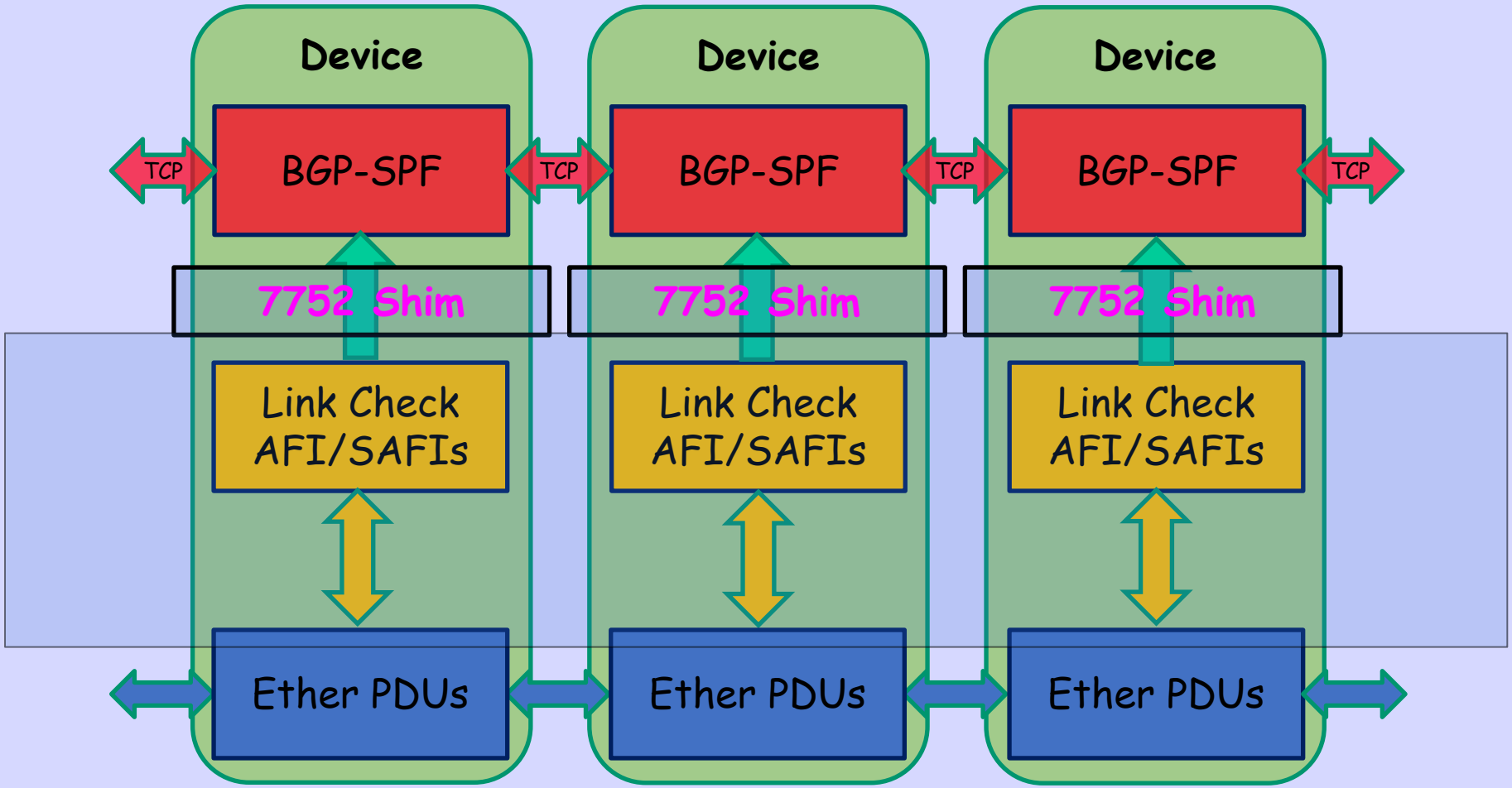
BGP-LS for SDN & TE



BGP-LS for BGP-SPF



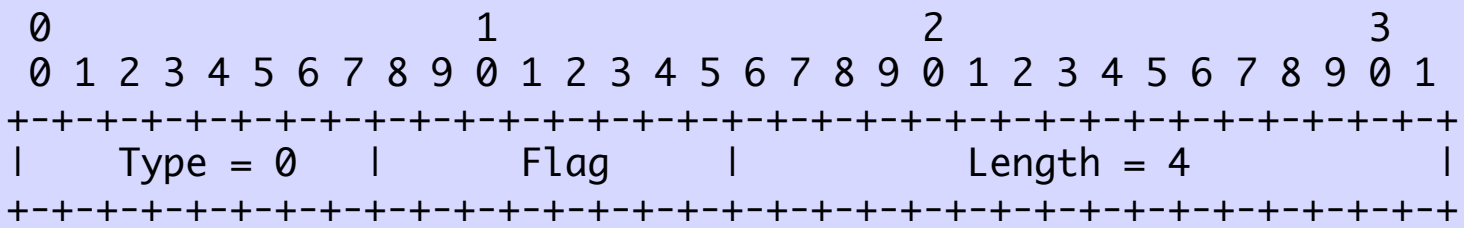
North/South Protocol



North/South Protocol

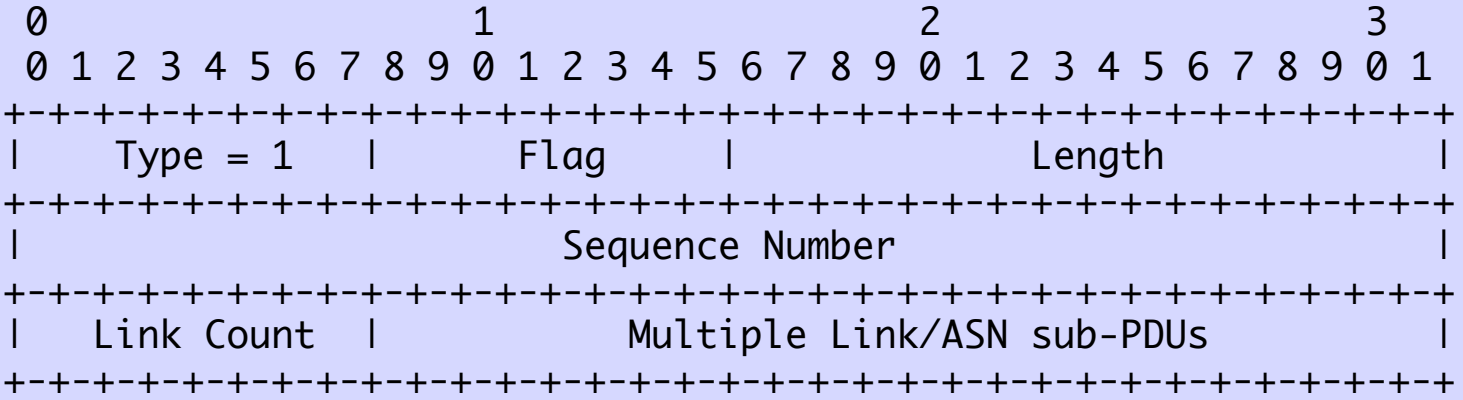
- We assume a reliable intra-device TCP transport, so no ACKs
- We assume a PDU capable of 64k
- The protocol is [re]started by a request from the 7752 Topology Shim Layer
- The Ether Layer then sends the full topology, its full link neighbor state, North
- The Ether Layer sends incremental updates as links and/or addressing change

Topology Request for Full State From 7752 Shim to Link Layer



Flag - 0 – Request Full Ethernet Topology (i.e. restart)

PDU from Link Layer to Shim

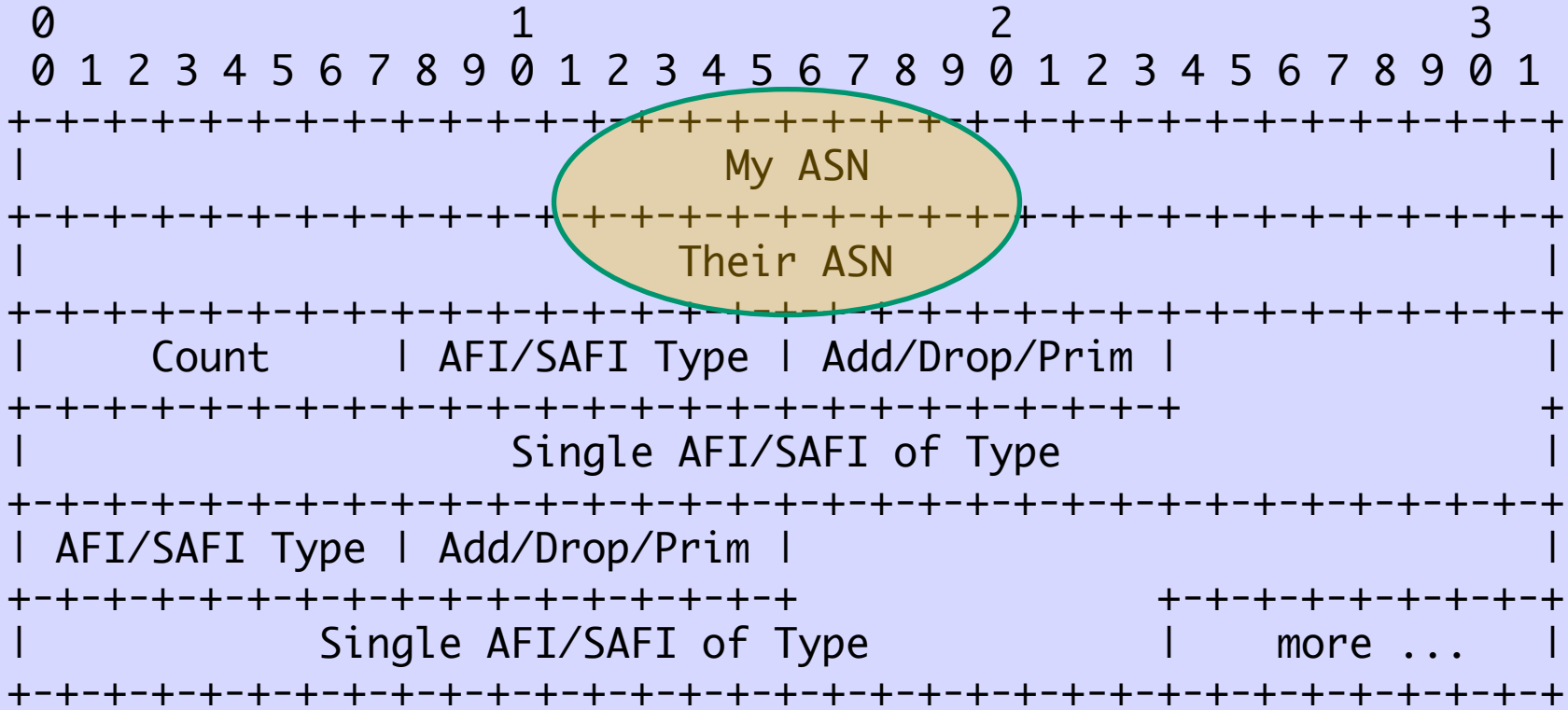


Flag
- 0 – This is the start of a Full State transfer
- 1 – Continuation PDU
- 2 – Last PDU of transfer
- 3 - This is the start of a Update for a state change

Link Count – Number of Link/ASN sub-PDUs to follow

Multiple Link/ASN LSAs, see following

Link/ASN sub-PDU



Count of AFI/SAFIs in this sub-PDU

AFI/SAFI Type - 11-IPv4, 12-IPv6, 13-MPLSv4, 14-MPLSv6, …

Add/Drop/Prim (bits) - 0 Announce / Withdraw
- 1 Primary
- 2-7 Reserved

Addressing and Routing Are Done in Upper Layers

- The 7752 Shim takes the above and translates to BGP-LS/SPF PDUs
- The Topology Layer, BGP-SPF, can now construct the full link state and IP/MPLS topology of the network
- Routing protocols such as BGP-SPF can then talk between devices to exchange reachability

And Bob's Your Uncle

7752 Reconsidered

- Despite the name, it is not just link state
- It encodes every feature and attribute of OSPF, IS-IS, LSP, RSVP, …
- It is only missing SMTP, HTTPS, … ☺
- So thinking of dumping 7752 and having LSOE go directly to BGP-SPF link state

BTW, There is No IPR