

# Techniques for Scaling LDP

**Kireeti Kompella**  
**Juniper Networks**

[www.mpls2007.com](http://www.mpls2007.com)

# Agenda

---

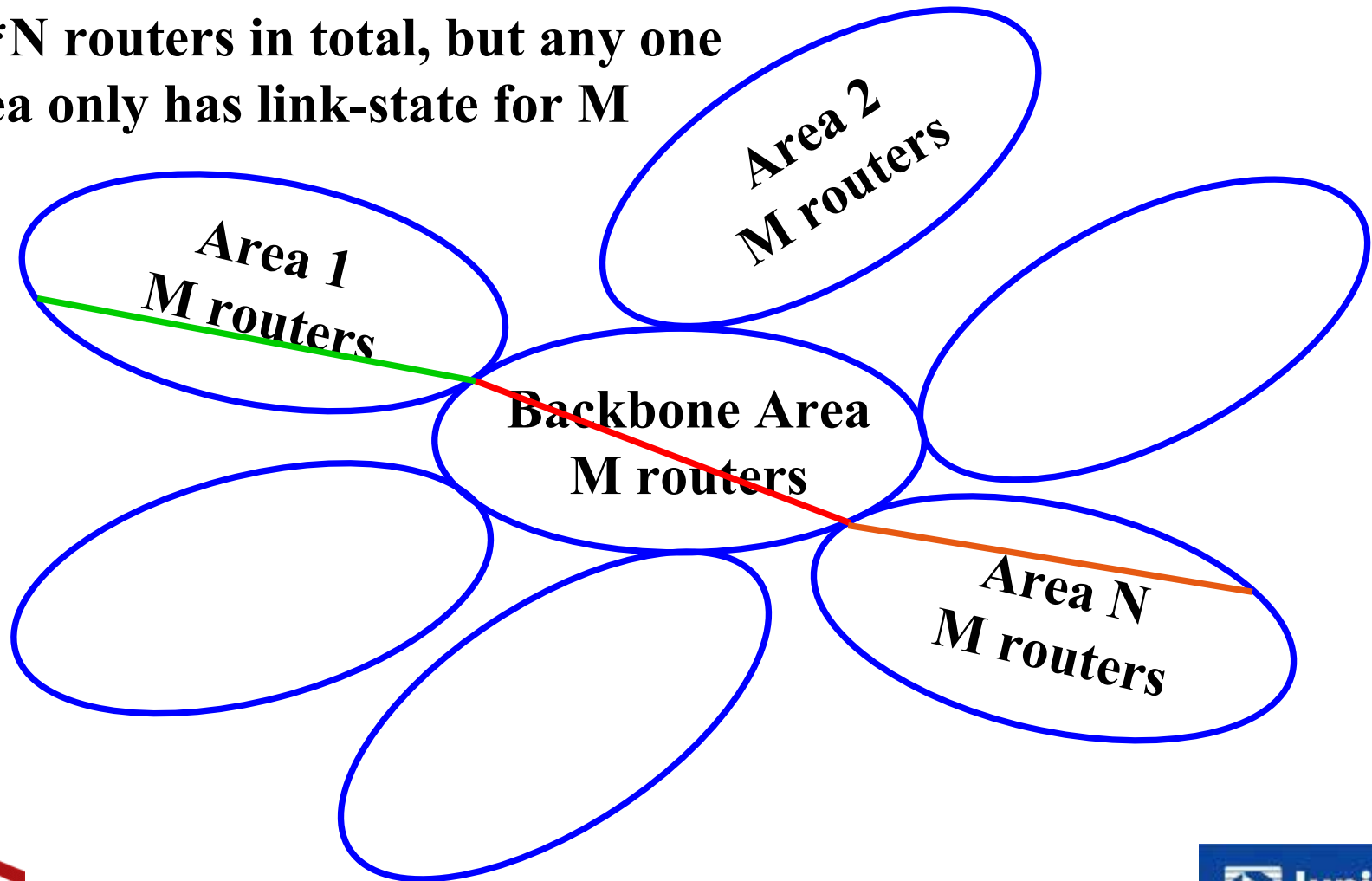
- **Problem statement**
  - **How IGPs solve the problem**
- ***Other* solutions**
- ***This* solution**
  - **Outline**
  - **Improvements**
  - **Considerations, ECMP**
  - **Other applications**
- **Summary**

# Problem

- A single network grows ...
- ... and with growth come scaling issues
- In plain IP networks, the focal point of scaling is the (link-state) IGP
  - Typically, this is addressed by hierarchy (OSPF areas or IS-IS levels), and sometimes by prefix aggregation
  - Sometimes this is addressed by hierarchy by introducing BGP
- In Traffic-Engineered MPLS networks, this is addressed by TE LSP hierarchy

# Scaling an IGP Using Hierarchy

$M \cdot N$  routers in total, but any one area only has link-state for  $M$



# How This Helps

1. Hierarchy: a router has topological and reachability information for its own area alone; it has only reachability information for all other areas
2. Aggregation: a router only carries full addresses (/32s) for routers within its own area; it has aggregated prefixes for all other areas  
(For this to work, all the loopbacks of routers in an area must come from a single, aggregated prefix)

# Problem Statement

---

- Can we make LDP scale similarly?
  - Can hierarchy help?
  - Does aggregation help?

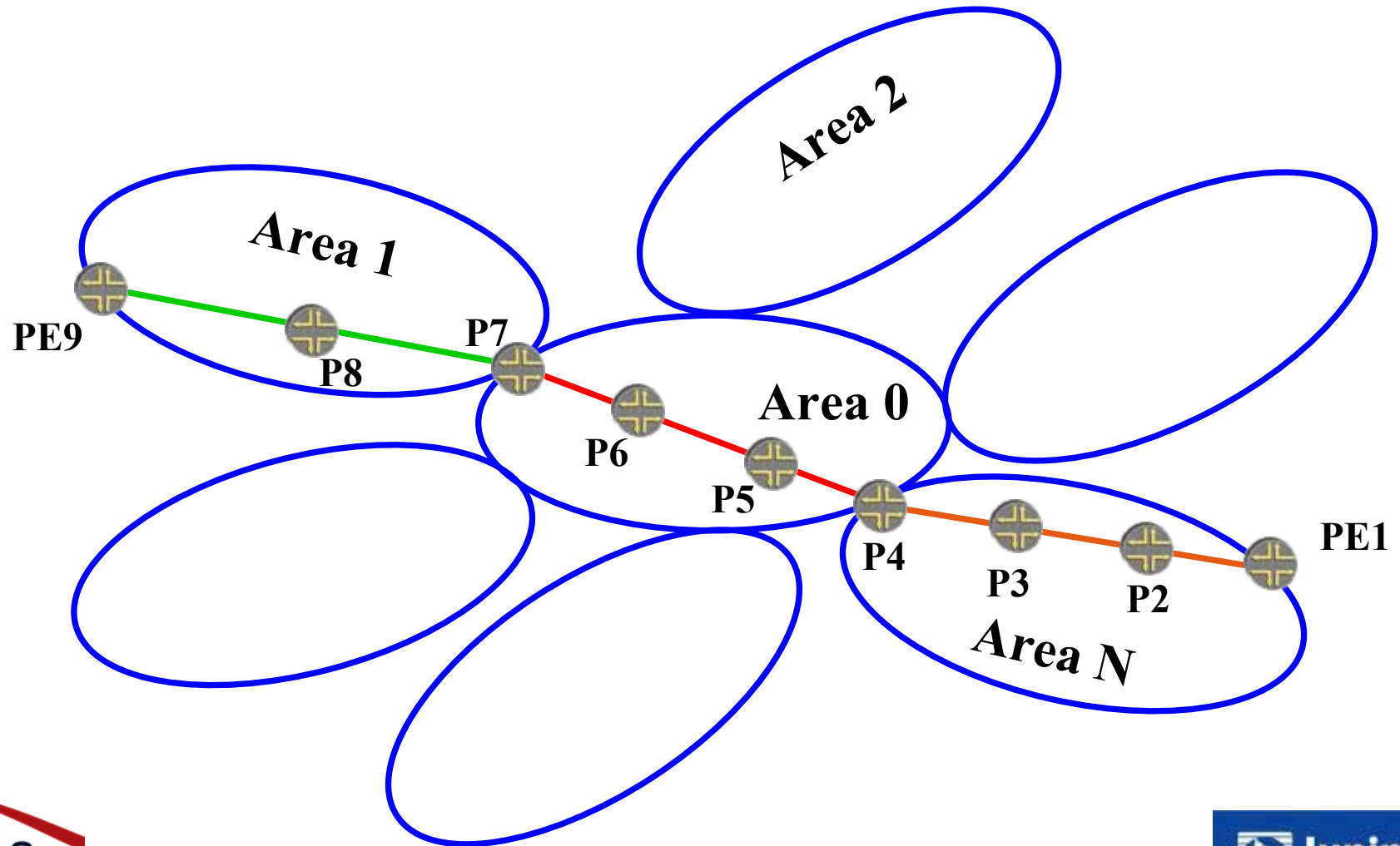
## *Other Solutions*

- Scale the IGP using hierarchy and aggregation: carry prefixes instead of /32s for areas other than yours
  - However, keep LDP “flat” by allowing LDP addresses to match against a prefix in the IGP instead of a /32
  - OR: use “label aggregation” to more-or-less correspond to loopback aggregation (2 label stack)
- Scale using BGP
  - Each ABR advertises the loopbacks of non-backbone areas into labeled BGP (RFC 3107) with itself as the nexthop (2 label stack)
  - Natural if IGP scaling is dealt with by using BGP

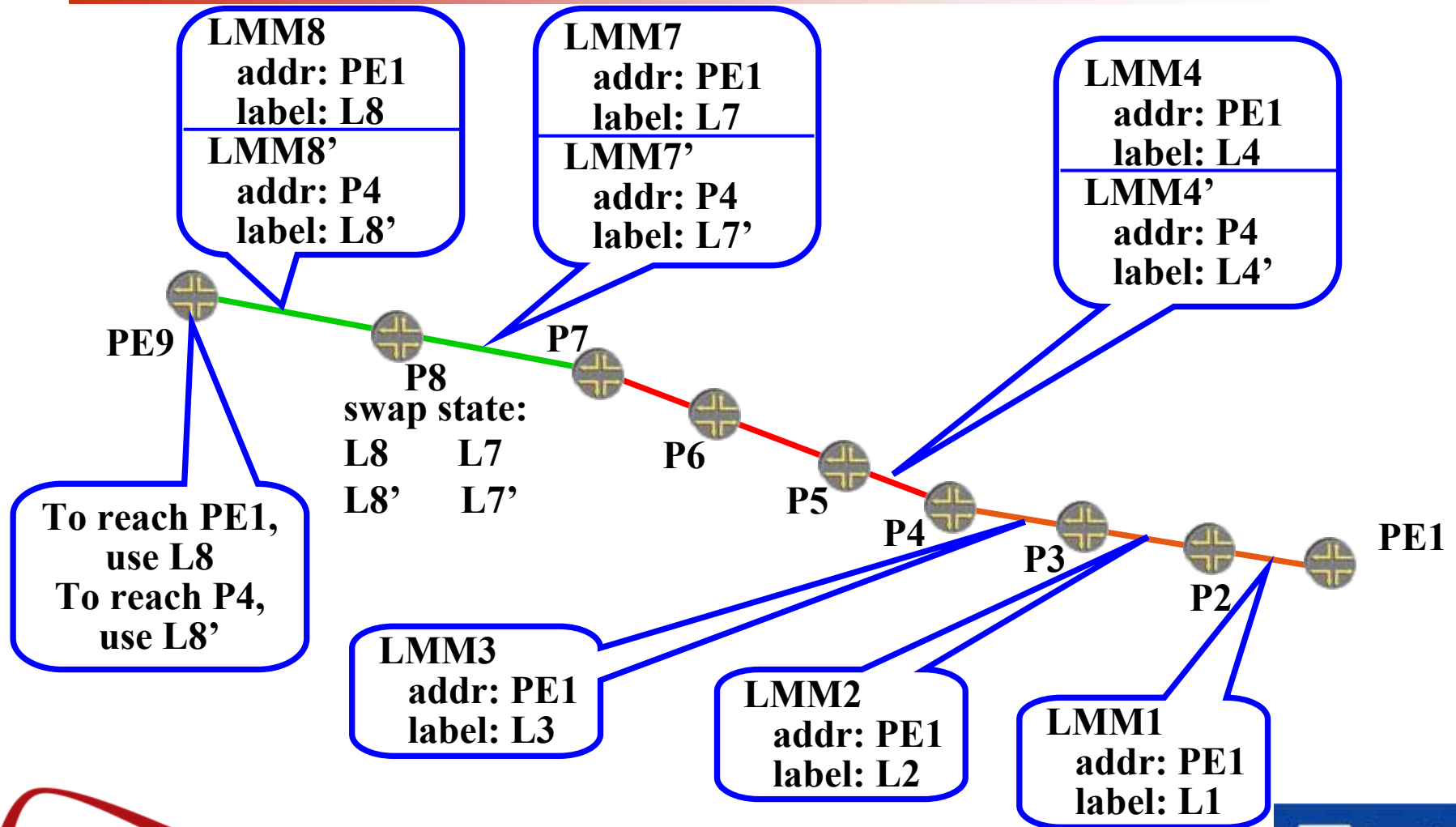
# ***This Solution***

- Scale the IGP using hierarchy ...
  - Scale LDP also using hierarchy
  - This yields Hierarchical LDP (H-LDP)
- ... and if you scale the IGP using aggregation as well
  - Well, then scale LDP using aggregation
  - Or not, as you prefer (see later)

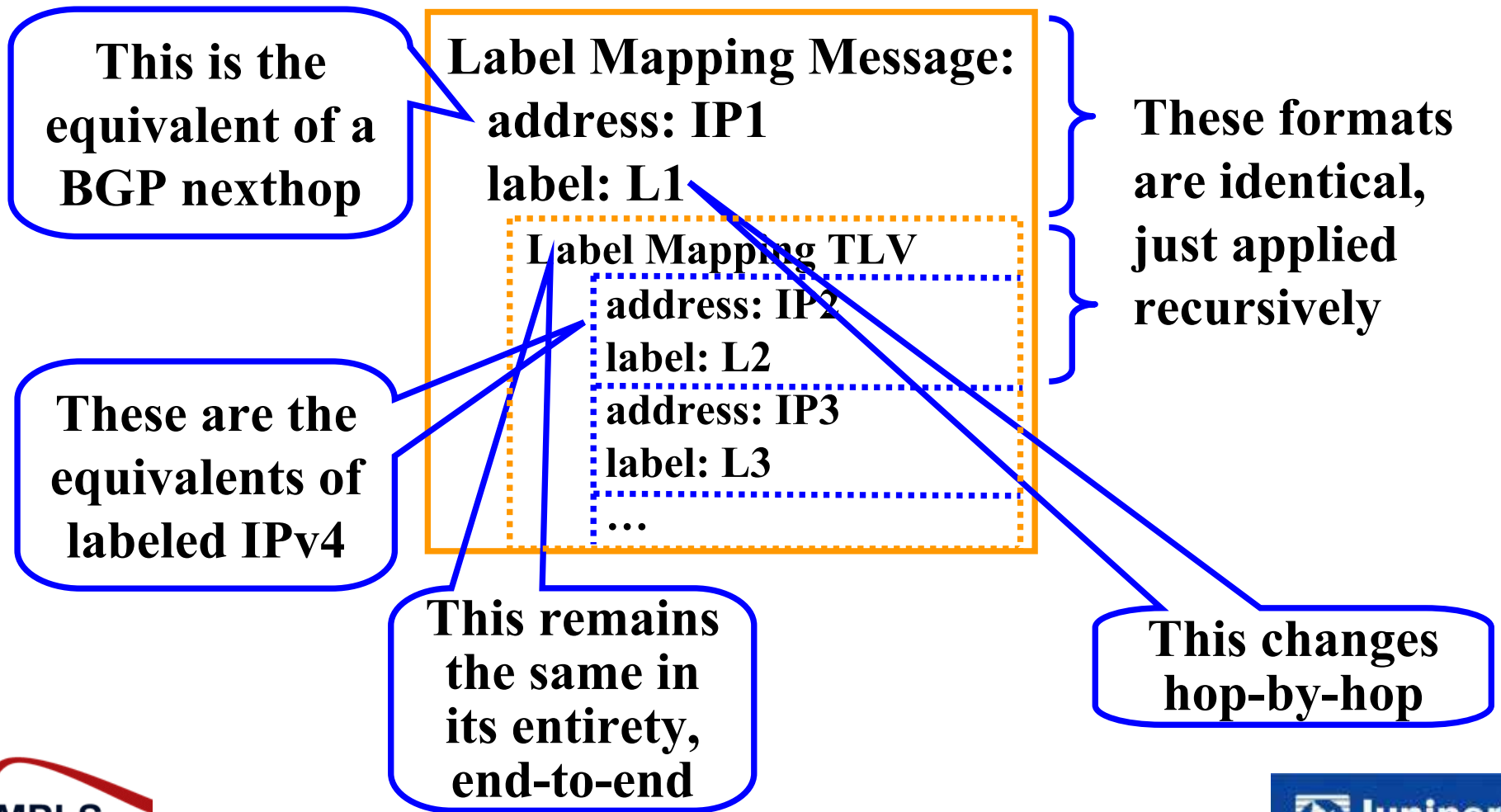
# Non-hierarchical LDP over Hierarchical IGP



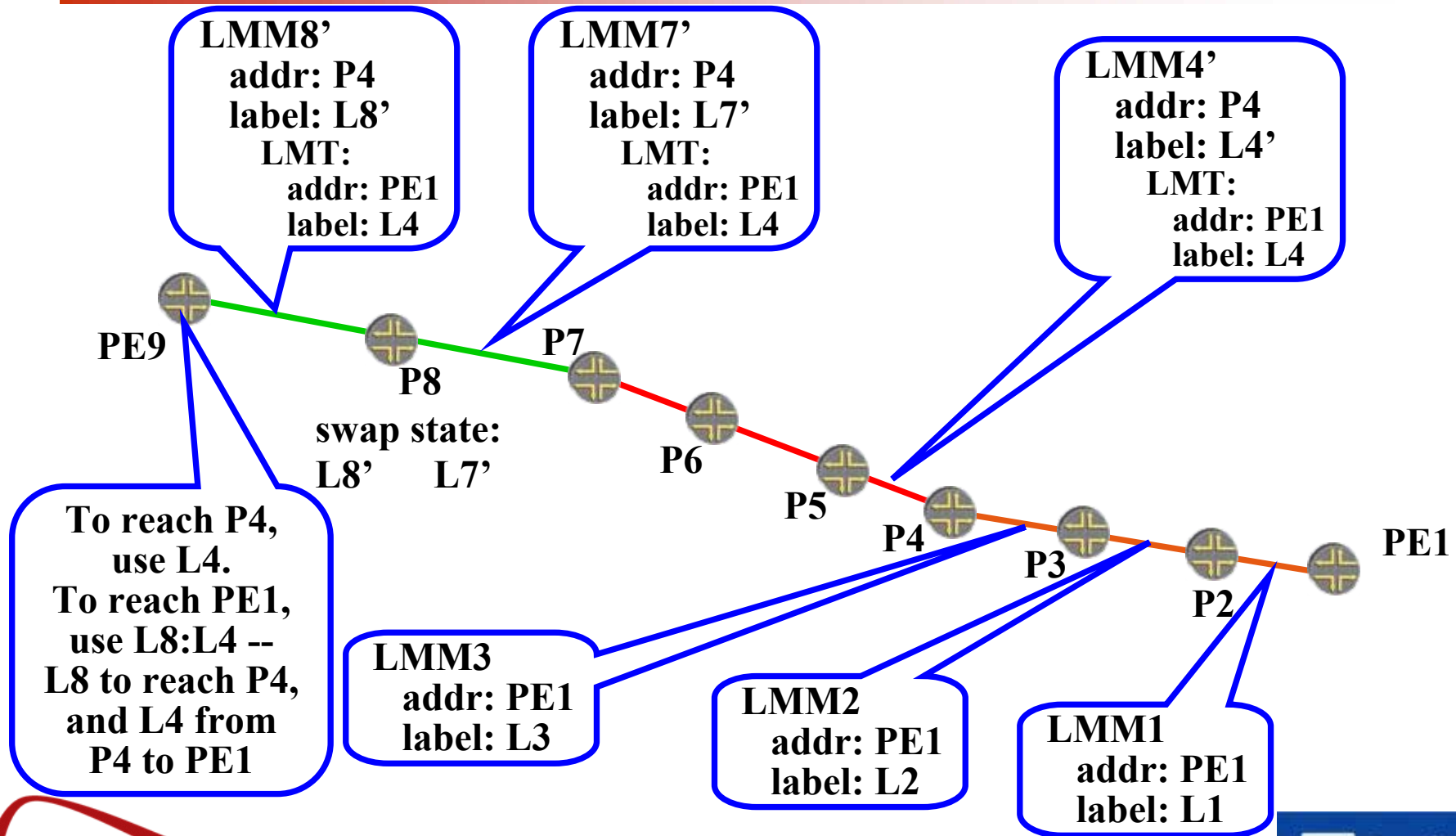
# Non-hierarchical LDP over Hierarchical IGP: Label Mapping Messages



# Hierarchical LDP Message Format



# Hierarchical LDP over Hierarchical IGP: Follow the Label Mapping Messages



## Improvements (backbone)

- A backbone LSR only sends out one Label Mapping Message per backbone LSR
  - Not one per LSR in the entire network
- A backbone LSR need only process Label Mapping Messages for other backbone LSRs; the Label Mapping TLVs need very little processing, and this can be done in bulk for all contained TLVs
- A backbone LSR only needs LDP state to reach other backbone LSRs

**In other words,  $O(M)$  state and processing instead of  $O(M*N)$ !**

## Improvements (non-backbone)

- A non-backbone LSR sends out (and processes) one Label Mapping Message per backbone LSR + one per LSR in its area
  - Only PEs need to fully process Label Mapping TLVs
- Non-backbone LSRs have LDP state for LSRs in their area and for backbone LSRs

**In other words,  $O(2M)$  state and processing instead of  $O(M*N)$ !**

# Improvements (ABRs)

- An ABR has to generate Label Mapping TLVs for each non-backbone Label Mapping Message it gets, and insert these into its own Label Mapping Message
  - As non-backbone Label Mapping Messages are sent/withdrawn, the ABR has to make corresponding changes in its Label Mapping Message

**In other words,  $O(n*M)$  state and processing instead of  $O(M*N)$  ( $n$  = number of non-backbone areas for a given ABR)**

# Improvements (PEs)

- End PEs have to process Label Mapping TLVs:
  - parse these, and install data plane state
- However, an end PE can (by policy) decide to install only the data plane state that it needs
  - In this case, end PEs can have only the state they need
  - Such a policy is hard to implement and deploy with “regular” (non-hierarchical) LDP
- With this policy, end PEs can have much less state with H-LDP than they would with “regular” LDP

# Considerations

- H-LDP achieves state/processing savings based only on IGP hierarchy, i.e., even if there is no aggregation of non-backbone loopbacks
- However, Label Mapping Messages can carry a lot of Label Mapping TLVs -- one per non-backbone LSR subtending on the generating ABR
  - If non-backbone loopbacks are aggregated, this can be used to aggregate Label Mapping TLVs
  - With aggregation, however, one doesn't know if a particular PE within the aggregate is up or not (both in IGP and LDP); thus one may prefer not to aggregate

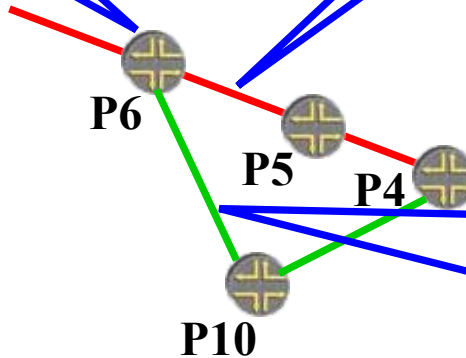
## Considerations (2)

- As mentioned before, the end to end tunnels with labeled BGP (RFC 3107) and with “label aggregation” have a two-label stack
  - H-LDP also creates tunnels with a two-label stack
- The only current solution with a single label stack is the “best match” approach

# H-LDP with ECMP

If the IGP cost from P6 to P4 is the same via P5 or P10, then P6 wants to do LDP ECMP to P4, i.e., use either L5' or L10'

This can be achieved with a little more work on P6's part, without full parsing of the LMT



**LMM5'**  
addr: P4  
label: L5'  
LMT:  
addr: PE1  
label: L4

**LMM10'**  
addr: P4  
label: L10'  
LMT:  
addr: PE1  
label: L4

## Other Applications of H-LDP

- In RFC 4364 VPNs, the “Inter-AS Option C” flavor requires PE-to-PE tunnels across the ASs
  - Achieving this in the context of LDP tunnels usually means leaking AS1’s loopbacks into AS2’s IGP, and vice versa, so that LDP can do label binding
- Instead, with H-LDP, an ASBR can add loopbacks learned from the other AS as Label Mapping TLVs in its own Label Mapping Message

# Summary

- Hierarchical LDP defines recursive Label Mapping TLVs within a Label Mapping Message
- This allows LDP to scale in a similar fashion to IGP scaling using hierarchy (areas/levels)
- H-LDP scales control plane processing and data plane state required by transit LSRs, both in the backbone and in non-backbone areas
  - Only ABRs initiating Label Mapping TLVs and end PEs requiring reachability need to process these TLVs
- H-LDP can further be applied recursively if needed

**Thanks!**

Questions?