Locator/ID Separation Protocol (LISP) Tutorial

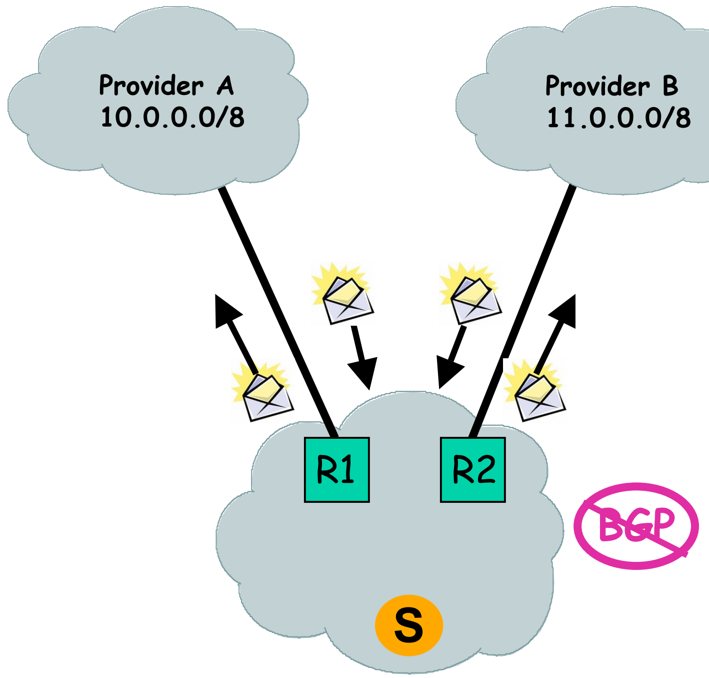
IETF Vancouver - December 2007

Dave Meyer, Vince Fuller, Darrel Lewis, Eliot Lear, Scott Brim, Dave Oran, Noel Chiappa, John Curran & Dino Farinacci

Agenda

- What problem is LISP solving?
- Why Locator/ID Separation?
- Data Plane Operation
- Mapping Mechanisms
  - CONS, NERD, ALT, EMACS
- Incremental Deployability
- Prototype and Testing

What Problem is LISP Solving?



(1) Improve site multi-homing

a) Can control egress with IGP routing

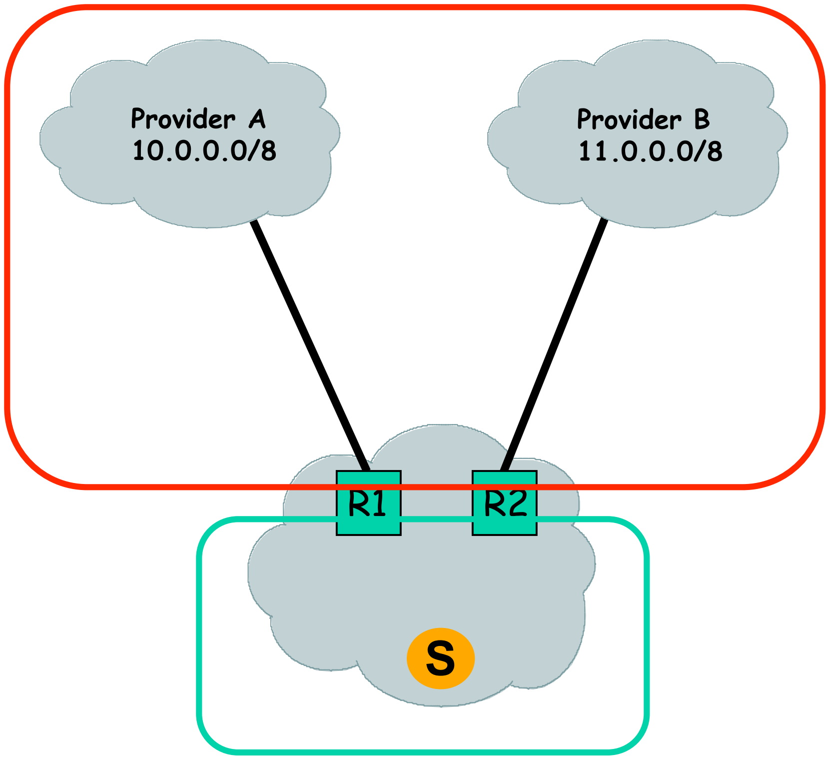
b) Hard to control ingress without more specific route injection

c) Desire to be low OpEx multi-homed (avoid complex protocols, no outsourcing)

(2) Improve ISP multi-homing

a) Same problem for providers, can control egress but not ingress, more specific routing only tool to circumvent BGP path selection

What Problem is LISP Solving?



(3) Decouple site addressing from provider

a) Avoid renumbering when site changes providers

b) Site host and router addressing decoupled from core topology

(4) Add new addressing domains

a) From possibly separate allocation entities

(5) Do 1) through 4) and reduce the size of the core routing tables

What Provoked This?

- Stimulated by problem statement effort at the Amsterdam IAB Routing Workshop on October 18/19 2006
  - RFC 4984
- More info on problem statement:
  - http://www.vaf.net/~vaf/apricot-plenary.pdf

Why the Separation?

- The level of indirection allows us to:
  - Keep either ID or Location fixed while changing the other
  - Create separate namespaces which can have different allocation properties
- By keeping IDs fixed
  - Assign fixed addresses that never change for hosts and routers at a site
- You can change Locators
  - Now sites can change providers
  - Now hosts can move

Locator/ID Separation Solutions

- First let's look at Locator/ID solutions
- Host Based
  - shim6, HIP, Six/One
- Router Based
  - LISP, GSE, EFT, IVIP, Six/One

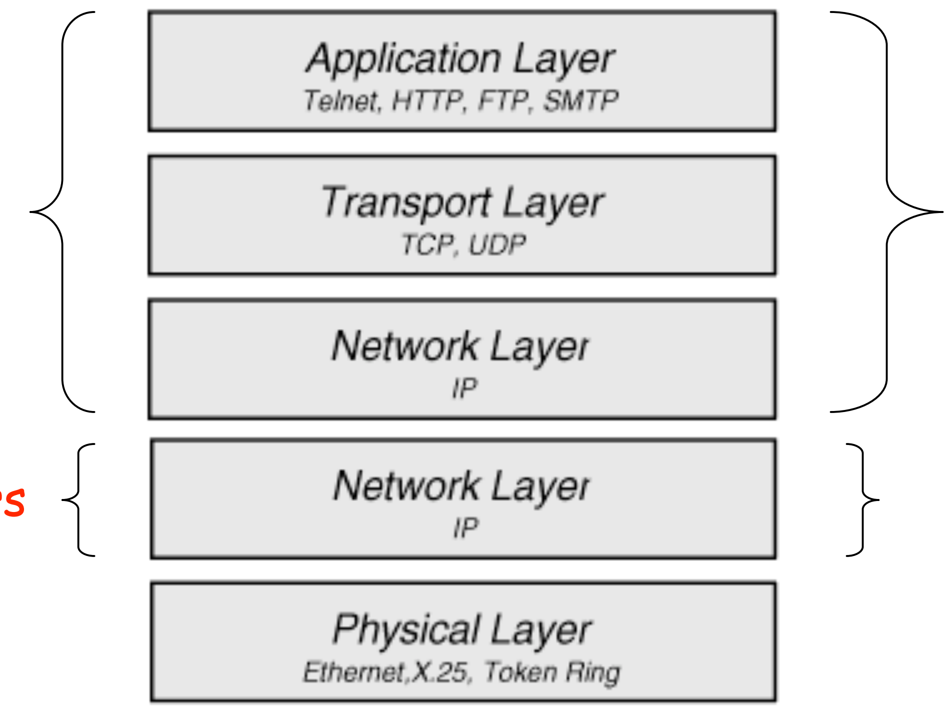
What is LISP?

- Locator/ID Separation Protocol
- Ground rules for LISP
  - Network-based solution
  - No changes to hosts whatsoever
  - No new addressing changes to site devices
  - Very few configuration file changes
  - Imperative to be incrementally deployable
  - Address family agnostic

What is LISP?

Uses IDs

Uses Locators



Host Stack

(end-to-end)

LISP

“Jack-Up” or “Map-n-Encap”

draft-farinacci-lisp-05.txt

```
              0                   1                   2                   3
              0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
             +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
           / |Version|  IHL  |Type of Service|          Total Length         |
          /  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
         /   |         Identification        |Flags|      Fragment Offset    |
        /    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      OH     |  Time to Live | Protocol = 17 |         Header Checksum       |
        \    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
         \   |                    Source Routing Locator                     |
          \  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
           \ |                 Destination Routing Locator                   |
             +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
           / |           Source Port         |         Dest Port (4341)      |
        UDP  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
           \ |           UDP length          |        UDP Checksum           |
             +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
           / |                    Locator Reach Bits                        |
       LISP  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
           \ |                            Nonce                              |
             +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
           / |Version|  IHL  |Type of Service|          Total Length         |
          /  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
         /   |         Identification        |Flags|      Fragment Offset    |
        /    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      IH     |  Time to Live |    Protocol   |         Header Checksum       |
        \    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
         \   |                           Source EID                          |
          \  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
           \ |                         Destination EID                       |
             +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

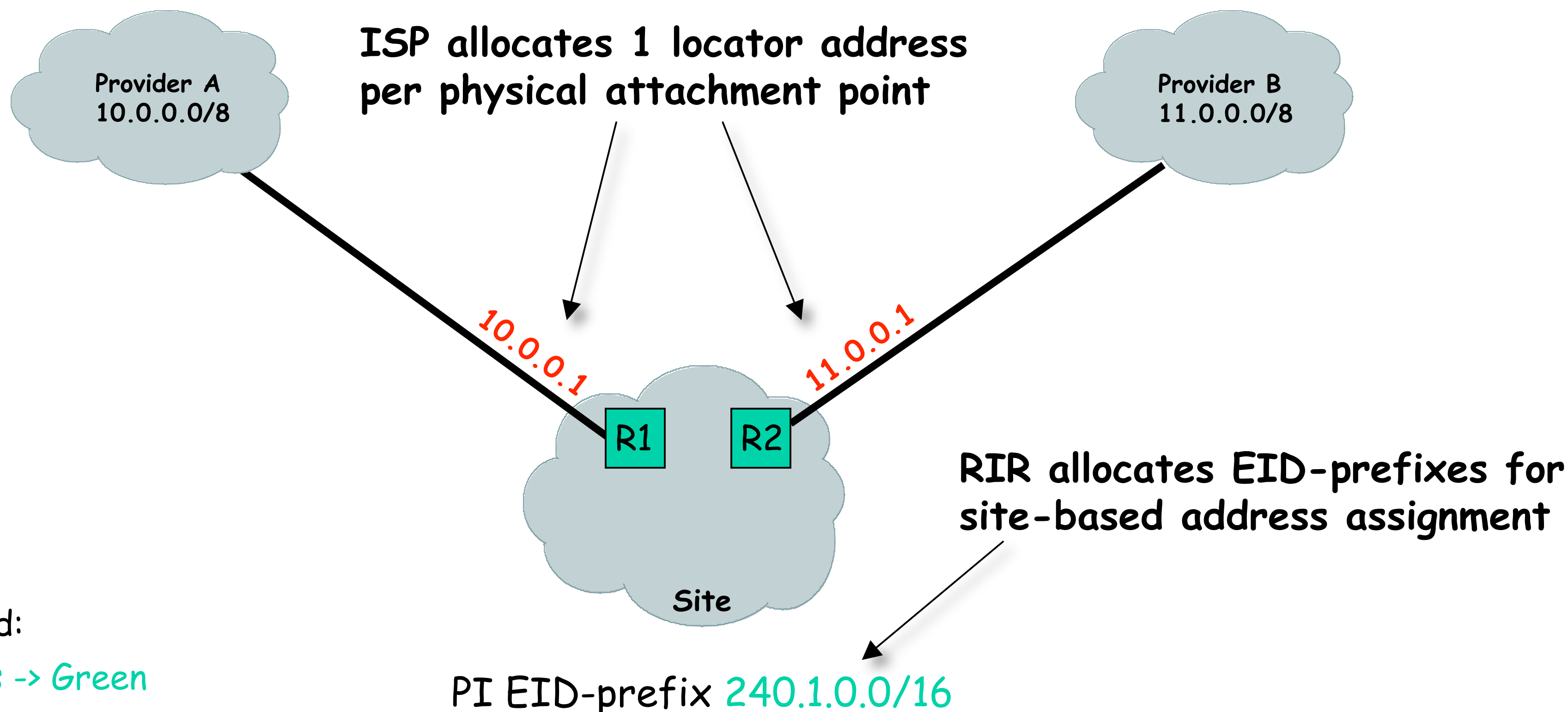
What is LISP?

- Data plane
  - Design for encapsulation and tunnel router placement
  - Design for locator reachability
  - Data-triggered mapping service
- Control plane
  - Design for a scalable mapping service
  - Examples are: CONS, NERD, ALT, EMACS

Some Brief Definitions

- IDs or EIDs
  - End-site addresses for hosts and routers at the site
  - They go in DNS records
  - Generally not globally routed on underlying infrastructure
  - New namespace
- RLOCs or Locators
  - Infrastructure addresses for LISP routers and ISP routers
  - Hosts do not know about them
  - They are globally routed and aggregated along the Internet connectivity topology
  - Existing namespace

Different Address Allocation Authorities



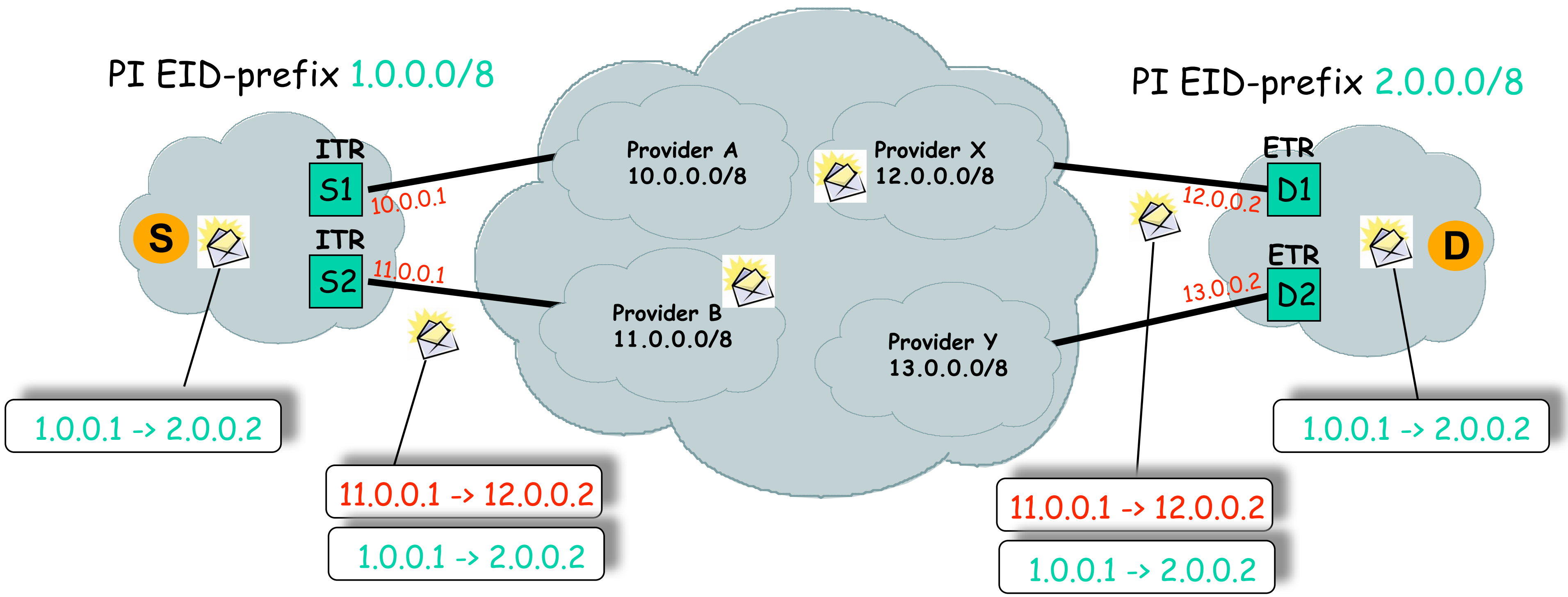
Legend:

- EIDs -> Green
- Locators -> Red

2 Network Elements

- Ingress Tunnel Router (ITR)
  - Finds EID to RLOC mapping
  - Encapsulates to Locators at source site
- Egress Tunnel Router (ETR)
  - Owns EID to RLOC mapping
  - Decapsulates at destination site

Packet Forwarding



Legend:

EIDs -> Green

Locators -> Red

DNS: D -> 2.0.0.2

Mapping Entry
- EID-prefix: 2.0.0.0/8
- Locator-set:
  - 12.0.0.2, priority: 1, weight: 50 (D1)
  - 13.0.0.2, priority: 1, weight: 50 (D2)

When the ITR has no Mapping

- Need a scalable EID to Locator mapping lookup mechanism
- Network based solutions
  - Have query/reply latency
  - Can have packet loss characteristics
  - Or, have a full table like BGP does
- How does one design a scalable Mapping Service?

Confusion on LISP Variants

- There is only <u>one</u> version of LISP
- The variants are ways the LISP data plane finds mappings
  - LISP 1 and LISP 1.5 use Data Probes
  - LISP 2 uses DNS - abandoned
  - LISP 3 are new Database Mapping Algorithms
- Current LISP mapping mechanisms
  - ALT and EMACS are LISP 1.5 variants
  - CONS, NERD and ALT are LISP 3.0 variants

Why so many Mapping System Designs?

- Tough questions need answering:
  - Where to put the mappings?
  - How to find the mappings?
  - Is it a push model?
  - Is it a pull model?
  - Do you use secondary storage?
  - Do you use a cache?
  - What about securing the mapping entries?
  - How to secure control messages?
  - What about protecting infrastructure from DOS-attacks?
  - What about controlling packet loss and latency?

Mapping Service

- Build a large distributed mapping database service
- Scalability paramount to solution
- How to scale:

    `(state * rate)`

- If both factors large, we have a problem
  - `state` will be $O(10^{10})$ hosts
    - Aggregate EIDs into EID-prefixes to reduce state
  - `rate` must be small
    - Damp locator reachability status and locator-set changes
    - Each mapping system design does it differently

Mapping Service

- NERD
  - Push design
- ALT and EMACS
  - Push and data-triggered pull design
- CONS
  - Push EID-prefixes at top levels of hierarchy
  - Pull mappings from lower levels of hierarchy

LISP-NERD

- A signed compact database of EID to RLOC mappings
- A CDN is used to distribute signed databases and updates
- Successive incremental updates are used to keep databases up to date without having to retrieve entire copies
- ITRs contain entire mapping database
- Never a failed lookup
  - No packet drops
  - No lookup latencies

LISP-NERD

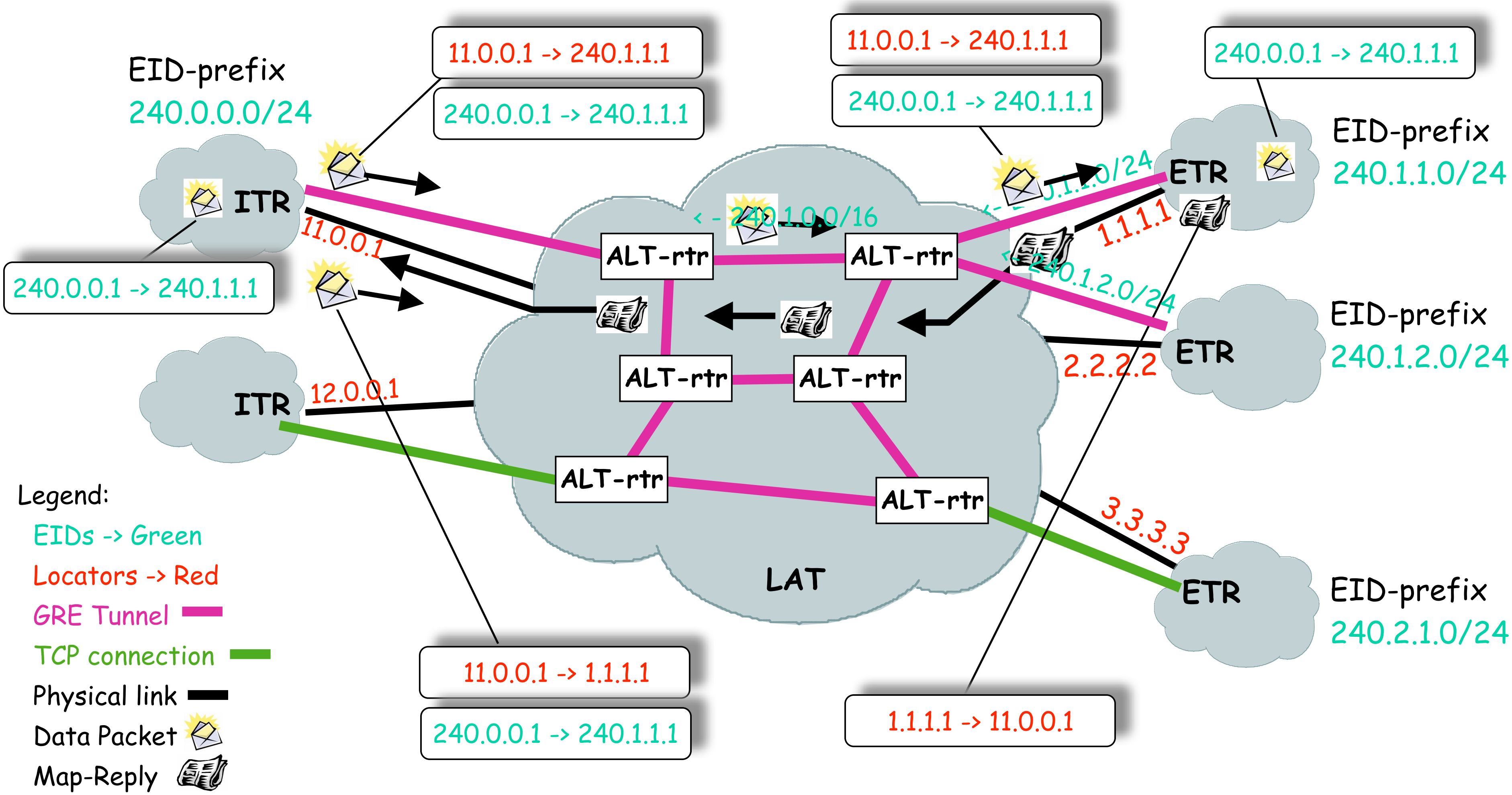
	# EIDs	2 RLOCs	4 RLOCs	8 RLOCs
Today	10^5	3.6 MB	6 MB	10.8 MB
	10^6	36 MB	60 MB	108 MB
	10^7	360 MB	600 MB	1.08 GB
	10^8	3.6 GB	6 GB	10 GB
	10^9	36 GB	60 GB	600 GB

Assume top 64 bits of all IPv6 addresses

LISP-ALT

- Use a logical topology of LISP-ALT routers
- They connect to each other via GRE tunnels
- They run eBGP over the GRE tunnels
- EID-prefixes are advertised and aggregated along this topology
- ITR sends Data Probes and Map-Requests over this topology to find destination ETR
- Destination ETR replies with Map-Reply

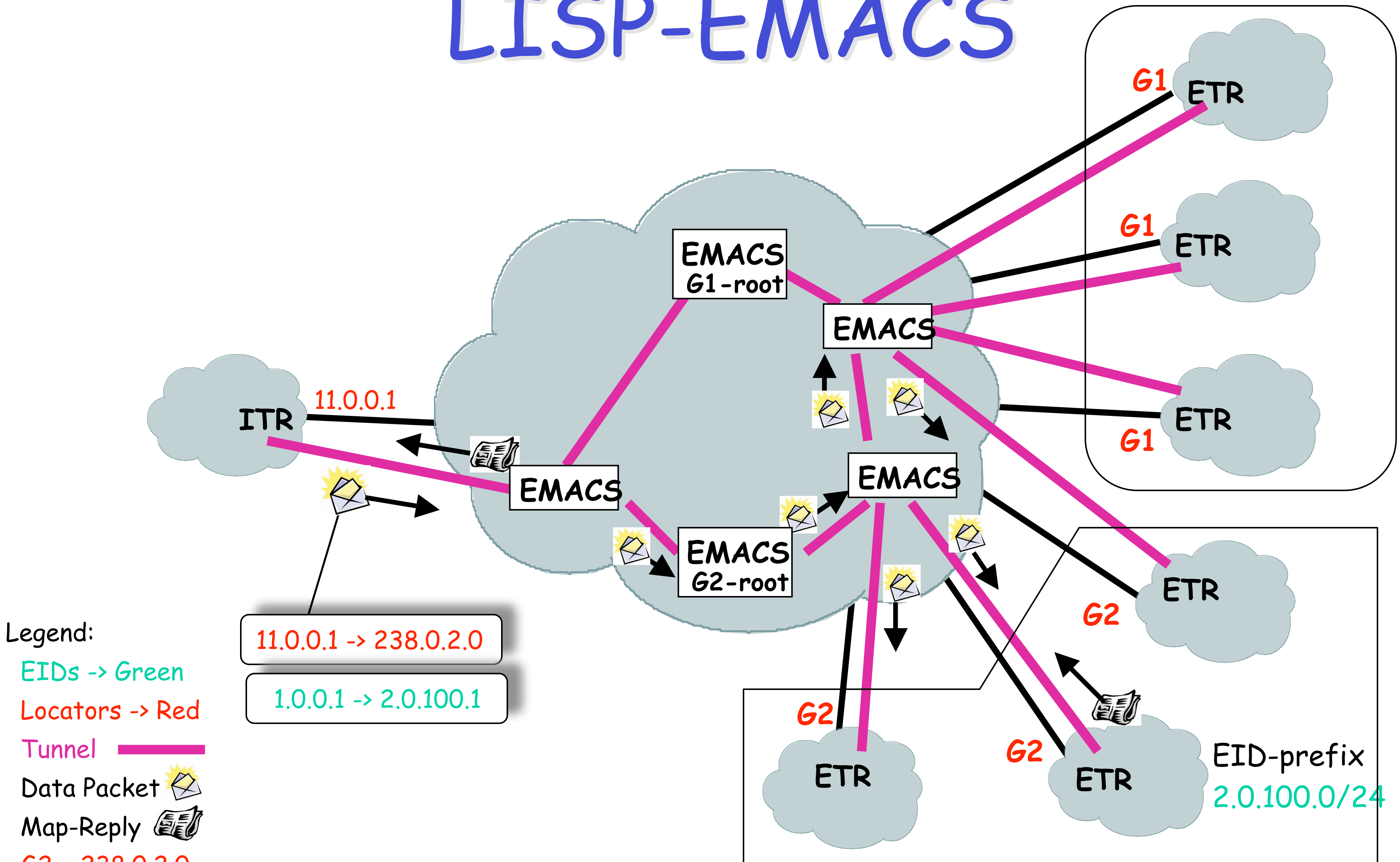
The LISP Alternate Topology



LISP-EMACS

- Uses alternate topology like LISP-ALT
  - BGP over GRE
- Find ETR by multicast Data Probe
- PIM Bidirectional shared tree used
  - Over GRE topology only
- ETRs hash their EID-prefixes to join a multicast group
- Wrong ETRs ignore
- Right ETR responds with Map-Reply over alternate or direct topology

LISP-EMACS

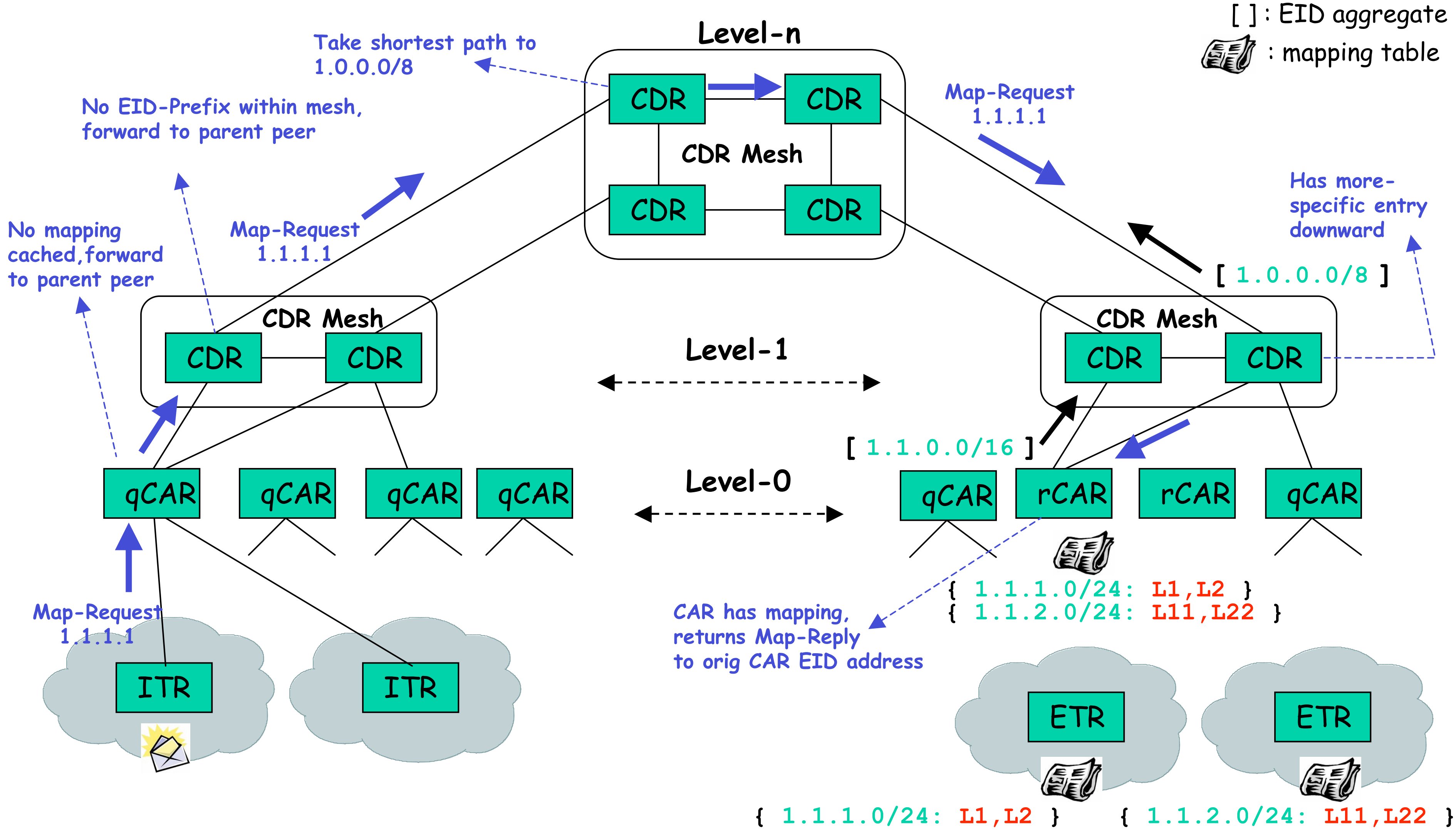


ITR
11.0.0.1
EMACS
EMACS G1-root
EMACS
EMACS
EMACS G2-root
G1
ETR
G1
ETR
G1
ETR
G2
ETR
G2
ETR
G2
ETR
EID-prefix
2.0.100.0/24
11.0.0.1 -> 238.0.2.0
1.0.0.1 -> 2.0.100.1
Legend:
EIDs -> Green
Locators -> Red
Tunnel
Data Packet
Map-Reply
G2 = 238.0.2.0

LISP-CONS

- LISP-CONS is a hybrid push/pull approach
- Push EID-prefixes (but not mappings) at upper levels of hierarchy
- Pull from lower levels of hierarchy
- Mappings stay at lower-levels
  - Map-Requests get to where the mappings are
  - Map-Replies are returned
- Getting to the lower-levels via pushing of EID-prefixes

LISP-CONS



Legend:
{ } : mapping entry
[ ] : EID aggregate
: mapping table
Level-n
Take shortest path to
1.0.0.0/8
CDR
CDR
CDR Mesh
CDR
CDR
Map-Request
1.1.1.1
No EID-Prefix within mesh,
forward to parent peer
Has more-
specific entry
downward
No mapping
cached,forward
to parent peer
Map-Request
1.1.1.1
[1.0.0.0/8]
CDR Mesh
CDR
CDR
Level-1
CDR Mesh
CDR
CDR
[1.1.0.0/16]
Level-0
qCAR
qCAR
qCAR
qCAR
qCAR
rCAR
rCAR
qCAR
{ 1.1.1.0/24: L1,L2 }
{ 1.1.2.0/24: L11,L22 }
Map-Request
1.1.1.1
CAR has mapping,
returns Map-Reply
to orig CAR EID address
ITR
ITR
ETR
ETR
{ 1.1.1.0/24: L1,L2 }
{ 1.1.2.0/24: L11,L22 }

Hybrid Approaches

- Run ALT or CONS at lower levels
  - Have NERD push mappings at higher levels
- Use ALT or CONS devices as Default Mappers
  - They would re-encapsulate packets
- Use ALT and EMACS together on BGP topology
  - When groups get too large inject EID-prefix

Interworking Deployability

- These combinations must be supported
  - Non-LISP site to non-LISP site
    - Today's Internet
  - LISP site to LISP site
    - Encapsulation over IPv4 makes this work
    - IPv4-over-IPv4 or IPv6-over-IPv4
  - LISP-R site to non-LISP site
    - When LISP site has PI or PA routable addresses
  - LISP-NR site to non-LISP site
    - When LISP site has PI or PA non-routable addresses

Interworking Deployability

- LISP-R site to non-LISP site
  - ITR at LISP site detects non-LISP site when no mapping exists
    - Does not encapsulate packets
  - Return packets to LISP site come back natively since EIDs are routable
  - Same behavior as the non-LISP to non-LISP case
    - LISP site acts as a non-LISP site

Interworking Deployability

- LISP-NR site to a non-LISP site
  - ITR at LISP site detects non-LISP site when no mapping exists
    - Does not encapsulate packets
  - For return packets to LISP site
    - ITR translates to a source routable address so packets symmetrically sent natively
    - PTR advertises NR prefixes close to non-LISP sites so return packets are encapsulated to ETR at LISP site

Prototype Implementation

- cisco has a LISP prototype implementation
  - Started the week of IETF Prague (March 2007)
- OS platform is DC-OS
  - Linux underlying OS
- Hardware platform is Titanium
  - 1 RU dual-core off-the-shelf PC with 7 GEs
- Based on `draft-farinacci-lisp-05.txt`
- Software switching only
- Supports both IPv4 and IPv6

Prototype Implementation

- Supports ITR encap and ETR decap
  - Load-balancing among locators
  - Respects priority & weight per mapping
- Multiple EID-prefixes per site
- Support for locator reachability
- Multi-VRF support for BGP-over-GRE

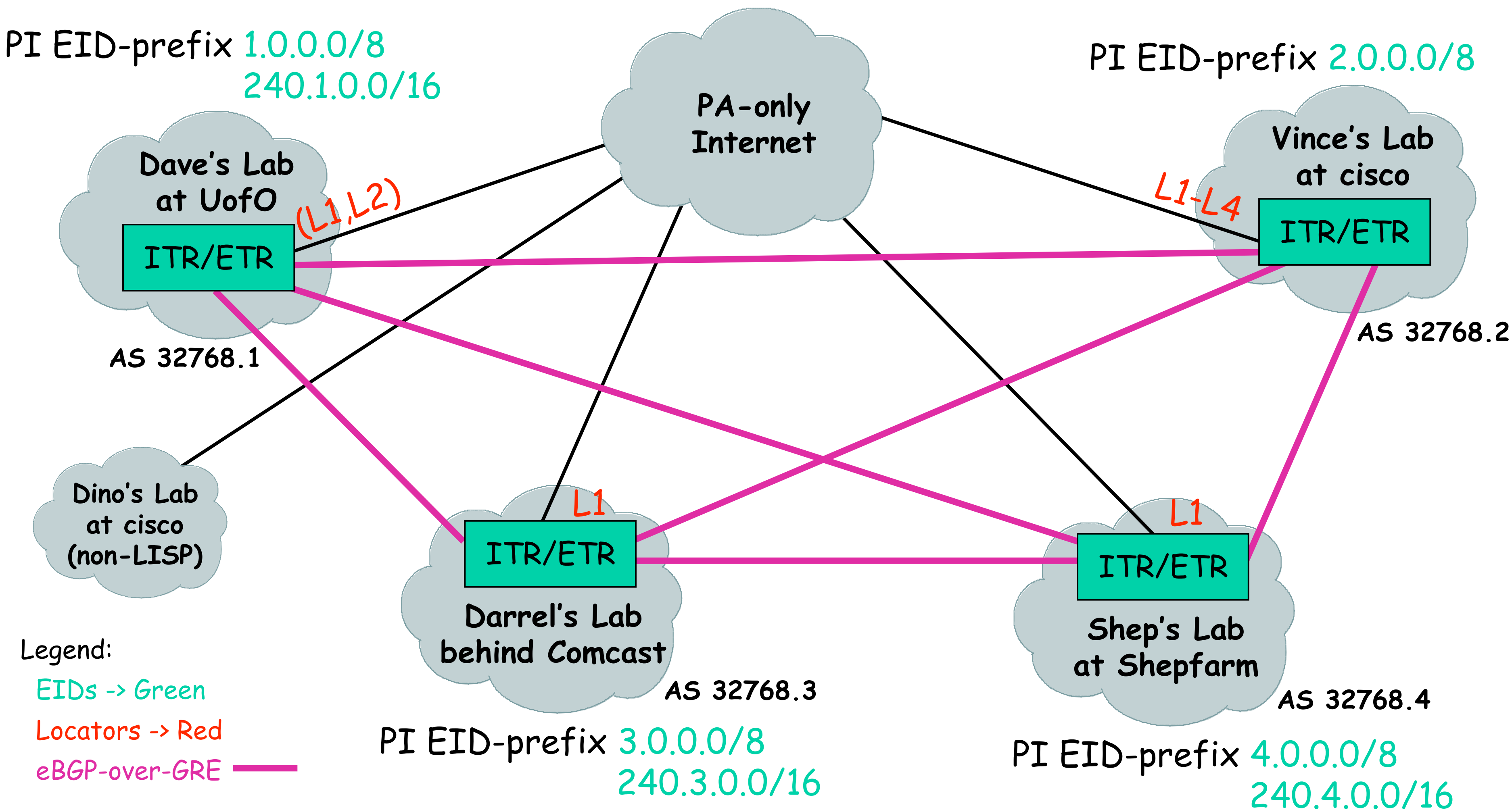
Prototype Implementation

- 240/4 support
  - To use as EIDs
- 'glean-mapping' support
  - And route-returnability check for verifying when an EID has moved to a new ITR
- LISP-ALT support
  - BGP advertises EID-prefixes over GRE tunnels
  - Data Probes sent over GRE topology
  - Map-Replies returned over GRE topology

Prototype Testing

- Detailed Test Plan written and being executed against
- Multiple EID-prefix testing completed
- Multiple locator testing completed
- Started LISP-ALT testing

LISP-ALT Topology



What's Next for Prototype and Testing

- Deeper dive into LISP-ALT
  - Send Map-Requests over GRE topology
  - Experiment with re-encapsulating and recursive ITRs
- More testing on map entry changing
- Think more about security mechanisms
- Think more and experiment with hybrid models
  - LISP-ALT with NERD
  - LISP-ALT with CONS

What's Next for Prototype and Testing

- Think more and experiment with movement
- Think more about aggregation and anti-entropy models
- Implement Address-Family crossover support
  - IPv6 EIDs over IPv4 Locators
- Implement Interworking Draft

Status on Pilot Deployment

- Taking names for external pilot
  - Must be able to dedicate minimum of 1 day a week
- Shooting for Spring '08 start date
- Goals:
  - Test multiple implementations
  - Experience with operational practices
  - Learn about revenue making opportunities

Open Policy for LISP

- It's been 1 year since the IAB RAWS
  - Some of us committed to working in the IETF and IRTF in an open environment
- This is not a Cisco only effort
  - We have approached and recruited others
  - There are no patents (cisco has no IPR on this)
  - All documents are Internet Drafts
- We need designers
- We need implementers
- We need testers
- We need research analysis
- We want this to be an open effort!

Internet Drafts

draft-farinacci-lisp-05.txt

draft-meyer-lisp-cons-03.txt

draft-lear-lisp-nerd-02.txt

draft-fuller-lisp-alt-02.txt

draft-curran-lisp-emacs-00.txt

draft-lewis-lisp-interworking-00.txt


LISP
RULES