A taxonomy for New Routing & Addressing Architecture Designs

Lixia Zhang & Scott Brim

RRG Meeting @ IETF69

July 27, 2007

Goal

- Build a framework, to allow us to
  - position each proposed solution in the design space
  - facilitate the evaluation of various design tradeoffs
- How:
  - Identify the solution directions
  - Find the open issues to be addressed
  - Find the dimensions of the design space
  - Look thru proposed solutions, find missing points
  - *Iterate !*

Routing system scalability

(**GRA**: Globally Routable Address space)

- Problem: Too many entries in GRA, too many updates
- Solution space:
  - A. Reduce the table size, or
  - B. Find a way to handle large routing tables

The proposed solutions

(so far) fall into the first category, which can be further sorted into:

A1: Only using topologically aggregatable addresses

Multihomed sites → multiple prefixes

A1a: the site uses site-local prefix internally (GSE)

A1b: the site uses GRA prefixes internally (SHIM6, Six/One)

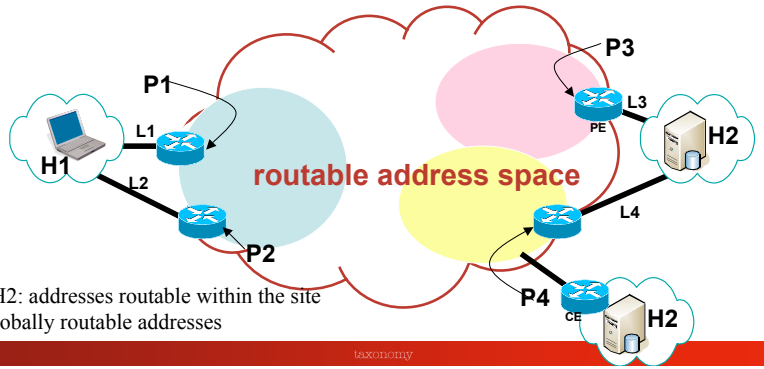
A2: Moving "edge" prefixes out of the global routing system

- Find "edge" attachment points in the routable space
- Deliver packets by tunneling to their attachment points

A rough picture for solution space discussion

grossly simplified; the boundary between GRA and the rest of the world (ROW) vary among different solutions

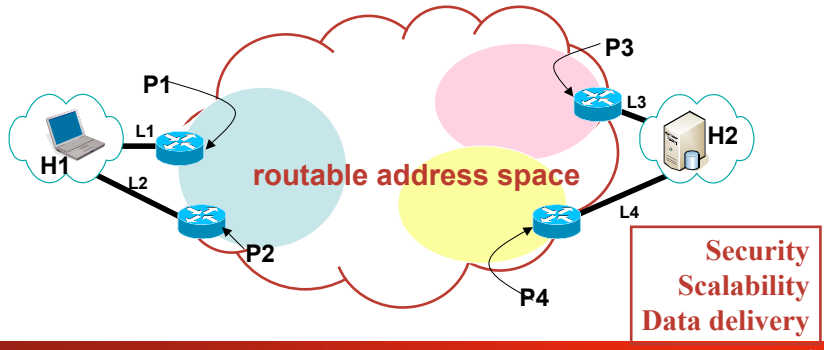
– Ranging from inside hosts (SHIM6) to stopping at site border (GSE, LISP)



H1, H2: addresses routable within the site
Pi: globally routable addresses

Common issues among the solutions

Q1: How to get mapping info
Q2: How to detect failure (e.g. P3 unreachable, or L3 failure)
Q3: How to handle failure

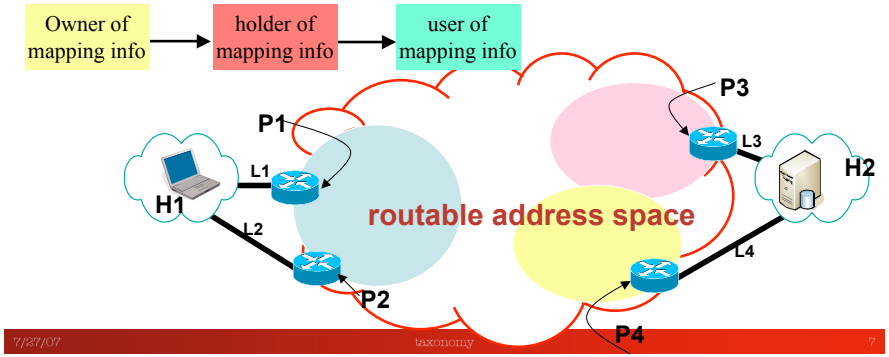


Q1. How to get the mapping info

Q1.1 How to inject the mapping info into the system

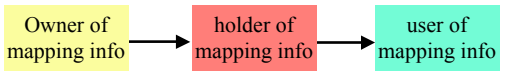
Q1.2 Where to distribute, who holds the mapping info

Q1.3 Where/who makes selection decision from multiple (Pi → Hi)



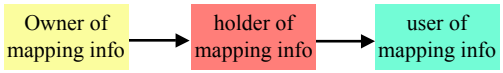
Q1.1 How to inject mapping info into the system

- Mapping info:
  - A1: DNS name → GRA address(es)
  - A2: ROW prefix → GRA address(es)
- Injection
  - – Manual configuration
  - – Automated protocol exchange
- Important consideration: Authentication of mapping info



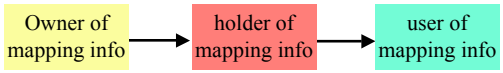
Q1.2 where to distribute mapping info

- Flood
- Push: to *specific node*(s)
- Poll/look-up:
  A1: by hosts
  A2: by site edge router, or any "responsible" router
- What is the system structure each of the above operates in?
  - Combining mapping info into DNS
  - Establishing a new/separate mapping info system
  - Combining mapping info with routing
- What is the trust model/relation between neighbor nodes in the distribution chain; how to insure info authenticity



Q1.3 Where/who makes selection decision from multiple (Pi → Hi)

- The holder is a database, a user (e.g. host, or ITR) receives complete (Hi → Pi's) mapping
- The holder makes decision on which mappings are given to which ITR

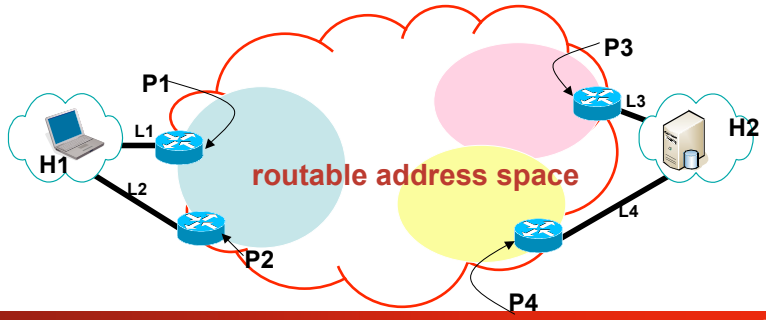


Q2. How to detect failures

A1: host detects failures

A2: Look at the picture again:

- failures within GRA space: can do business as usual (or can do better!)
- failures at Pi or Li: *need new solutions*



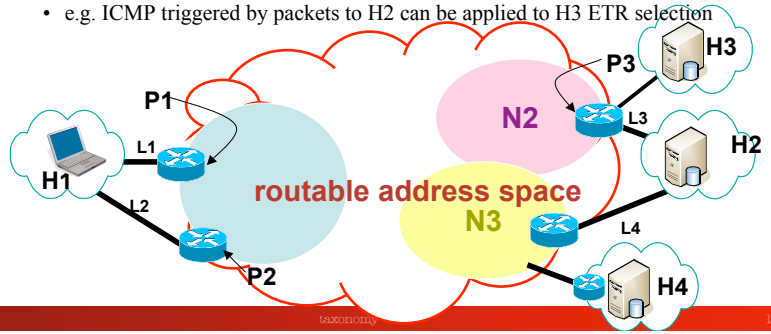
Q2: means for failure detection

Data-traffic triggered failure detection

A1: up to transport/upper level protocols

A2:

- data traffic triggered ICMP message (or equivalent)
- Piggyback TR status on data packets
- Indirect inference
  - e.g. ICMP triggered by packets to H2 can be applied to H3 ETR selection



Q3: How to handle failures

A1: host handles detection and recovery

- Potentially duplicate detection efforts by multiple hosts
  - e.g. multiple hosts suffer data losses caused by the same failure before they can react

A2:

Q3.1 Which nodes to inform

Q3.2 How to handle in-flight packets

Shared question: which party holds the temporary failure info, and how to promptly remove it when failure recovered?

Summary of questions

Q1: How to get mapping info

1.1 How to inject the mapping info into the system

1.2 Where to distribute, who holds the mapping info

1.3 Where/who makes selection decision from multiple (Pi→ Hi)

Q2: How to detect failure

Q3: How to handle failure

3.1: Which nodes to inform

3.2: How to handle in-flight packets

3.3: which party holds the temporary failure info, and how to promptly remove it when failure recovered?

Evaluation criteria

- Data delivery performance
  - Delay due to mapping look up
  - Delay due to suboptimal paths
  - Loss due to lack of mapping info
  - Loss during transient failure
  - Traffic concentration
- Scalability: with regard to the sizes of GRA system and "edge" population
  - Table size at mapping info holding nodes
  - Control data distribution overhead
- How to secure mapping info distribution

Stop here