

The Locator/ID split, its implications for the IP Architecture, and a few current approaches

David Meyer

Future of Routing Workshop

APRICOT 2007

dmm@l-4-5.net

<http://www.l-4-5.net/~dmm/talks/apricot2007/locid>

Agenda

- A Brief History of the (R&A) Universe
- Problems: Hot Boxes and Hot Wires
- Currently Proposed Approaches
 - Briefly: SHIM6, HIP, SCTP, GSE, TIDR, and LISP
- Longer Term Solutions
 - Briefly: Nimrod, DHTs, Compact Routing
- Characterizing Architectural Models
- Discussion

Brief History of the R&A Universe

- BGP remains largely unchanged (circa 1994)
- Concern about growth of the DFZ RIB
 - In particular, due to multihoming, and traffic engineering (TE)
- IAB R&A Workshop -- 10/2006
 - SPs voiced concern over the trajectory and properties of the DFZ RIB
- Lots of discussion since, little clarity on what is a problem and what isn't
 - And as a result, little work on solution spaces

Hot Boxes and Hot Wires¹

- Hot Boxes

- Do trends allow continued scaling of the (DFZ) RIB at the present rate?
- If not, does the RIB have to scale sub-linearly in the number of end sites? If so...
 - Would that break multihoming via PI?
 - And how do you do TE w/o more specifics?
 - Or server referrals (for the content folks)?

- Hot Wires

- Will BGP (UPDATE) dynamics kill us even if the RIB size is “controlled”?

¹Terminology due to Brian Carpenter

Is there really a *Hot Box* Problem?

- Issue: Growth of the RIB
 - *Is Hot Box a real problem?*
 - <http://www.nanog.org/mtg-0702/presentations/fib-scudder.pdf> suggests it is not
- As John (and others) have observed...
 - The problem is solvable, in some time frame, by either adding more memory (DRAM RIB/FIBs), and/or by changing our engineering practices (e.g., BGP free core), and/or by social engineering (eliminating the influence of bad actors, e.g., “CIDR policy”, or by cutting off the heavy tail of the UPDATE stream)
- Q: Can we have the DFZ RIB grow at current rates and still scale the control plane at *constant relative cost*?
 - tli’s point from the AMS and Cisco workshops
 - And or the rrg@ietf.org mailing list

Is there really a *Hot Wires* Problem?

- Issue: BGP UPDATE Dynamics
 - The core network can be exposed to “edge dynamics” by virtue of the fine grained information being carried by the routing system (according to gih, the “core” is relatively stable)
 - Of course, this is related to the size of the DFZ RIB
 - Issue: If there is some kind of Long Range Dependency (LRD)¹ present in the UPDATE stream, then these dynamics could be “non-linearly cumulative”
 - Route Flap Dampening was a proposed solution, but that had a negative impact on the Internet’s ability to respond to *real* topological changes

¹ i.e., Is the UPDATE time-series is a long memory process (events at a time index in the past effect current events, independent of the “lag”)?

On Observing The *Hot Wires* Problem

- Anecdotal Evidence
 - Example: Baltimore Tunnel Fire and the ensuing instability
- Historical Control Plane “Traces”
 - i.e., Routing data
 - To the best of my knowledge, no one has (yet) *rigorously* characterized the “burstiness” of the UPDATE stream¹
 - *Rigorously* likely involves estimating the Hurst parameter H ²
 - H is a statistical measure describing persistent correlations
 - There is also some research that suggests that the UPDATE stream is heavy tailed (see <http://www.potaroo.net>)

¹ A good start can be found, however, in “Modeling BGP Table Fluctuations”, Flavel, et. al.

² See, e.g., http://www.richardclegg.org/pubs/rgc_ukpewtalk05.pdf

Summary: The *Hot Wires* Problem

- While we don't really know if there is LRD in the UPDATE stream, if LRD is present, it is unlikely to be characterized by a “simple” (or even single) H value
 - i.e., discovering the presence of LRD in the UPDATE stream requires a non-trivial study
 - Do we even have the right/good data?
- Relationship between RIB size and UPDATE dynamics?
- Conclusion: More study needed

So what's out there now?

- Currently Proposed Approaches
 - SHIM6, HIP, SCTP, GSE, TIDR, and LISP
- Longer Term Solutions
 - Nimrod, DHTs, Compact Routing

Current Proposals -- SHIM6

- Pros

- Locators available before data sent
- Flexible locator selection
- Some thought for multicast

- Cons

- Requires host changes
- Only for IPv6
- No Traffic Engineering (TE) control in the network
- Some enterprise folks say too complicated

Current Proposals -- HIP

- Pros

- Locators available before data sent
- IDs are cryptographic
- IPsec - provides authentication

- Cons

- Requires host changes
- Only for IPv6
- No TE control in the network
- IPsec - adds NAT complexity
- No thought for multicast

Current Proposals -- SCTP

- Pros

- ID/Locator separation inherent in transport protocol
- Applicable for both IPv4 and IPv6
- Inherent authentication but cryptographic IDs could be used

- Cons

- Require host changes - but already under way
- All host apps need to be converted from TCP/UDP
- No TE control in the network
- No multicast support

Current Proposals -- GSE

- Pros

- Network-based, but host changes
- TE part of solution
- Flexible placement of tunnel routers

- Cons

- Requires host changes - not include RG in pseudo-header checksum
- IPv6 Only
- Rewrite versus encapsulation - original source RG lost
- No thought for multicast
- Really near term?

Current Proposals -- TIDR

- Pros

- Network-based, zero host changes
- Applicable for both IPv4 and IPv6
- TE part of Locator/ID split
- Flexible placement of tunnel routers
- Core routing tables reduces (also stops ASN depletion)

- Cons

- BGP changes
- No thought for multicast
- Edge routing tables still the same size

Current Proposals -- LISP

- Pros

- Network-based, zero host changes
- Applicable for both IPv4 and IPv6
- Supports TE
- Flexible placement of tunnel routers
- No routing protocol changes
- More thought for multicast

- Cons

- Need to get locators at data-plane time
- Security is hard
- Uses encapsulation versus header rewrite

Long v. Short Terms

- Should we be thinking more longer term?
 - Should we hack for now and put more focus on getting to a long-term?
 - Should we do a short-term and long-term concurrently?
 - Will short-term be long-term?
 - Will long-term never happen?
 - Is IPv6 long-term or short-term?

Long Term Solutions -- NIMROD

- Nimrod is a new architecture
 - However, fairly well specified
 - Reference implementations exist
- Should we just start engineering it?
 - Or are there just too many changes to consider?
 - How much experimentation can we afford?

Other Long Term Possibilities

- DHTs
- Compact Routing
- ROFL
- Many unanswered questions

Characterizing the Solution Space: Architectural Perspectives

- We will be focusing here on namespaces, and the various options for syntax and semantics of those namespaces (notably, locators and identifiers)
- This is not to say that there aren't solutions that don't necessarily involve *unwinding the overloading of IP addresses*
- For example, perhaps some form of *compact routing*?
 - Noting that there are name-independent forms of compact routing that *do* implement a form of loc/id split
 - See, e.g., “Compact Routing on Internet-Like Graphs”, Krioukov, et. al. <http://berkeley.intel-reserach.net/kfall/page/crig-infocom.pdf>

Characterizing the Solution Space: Architectural Perspectives

- There is little or no discussion of routing algorithms in the rest of this session (though we could change that...), in large part because
 - The fundamental problems (e.g., information complexity, amount of data, etc.) are relatively insensitive to the routing algorithm
 - Unless topological aggregation isn't key to scalability
 - You can think about these problems in either the distance vector, link state, or path vector contexts and arrive at similar results
- So lets dive in...

Characterizing the Solution Space: New Namespaces

- Which namespaces?
 - Locators
 - “where” (to a first approximation)
 - Identifiers
 - “who” (to a first approximation)
 - Both?
 - Something else?
 - Like “Service”?

Characterizing the Solution Space: Architectural Perspectives

- First, are we planning to use the existing routing-name (locator) name-space?
 - If so, implies natural limits on what is possible
 - Incremental improvements to BGP?
 - And perhaps that is the best we can do!¹
- And can we make sense out of using the existing routing-name namespace and re-architecting the allocations?
 - Doesn't seem likely (CIDR v2?)

¹See Mark Handley's "The Internet Only Just Works"

Characterizing the Solution Space: New Namespaces

- There are really two cases here:
 - The new namespace *is not visible* to hosts
 - The new namespace *is visible* to hosts
- Lets take a closer look at each of these

The new namespace *is not visible* to hosts

- In this case mapping boxes are deployed at the customer edge (CE or PE) to encapsulate packets as they head to towards the core (and vice versa)
- Has the advantage that we can move forward without host modifications
- Has a two main sub-cases, depending on the syntax of the new namespace. Either
 - The syntax of the new namespace is the same as an existing namespace
 - The syntax of the new namespace is different than an existing namespace

Case I: The new namespace has the same syntax as an existing namespace

- Candidate namespaces: IPv4 and/or IPv6
- Has several advantages, including
 - Can use existing h/w and s/w in the core
 - Only the mapping boxes have new code
- Has various problems in the “half-deployed” state, since you have to carry the old *and* new names
 - See draft-nikander-ram-generix-proxying-00.txt
- David Conrad’s presentation from the AMS workshop outlined this approach
- LISP is an instance of this approach

Case 2: The new namespace has a new syntax

- Examples

- NIMROD

- In this case, both the core and the edges need to be changed

The new namespace *is visible* to hosts

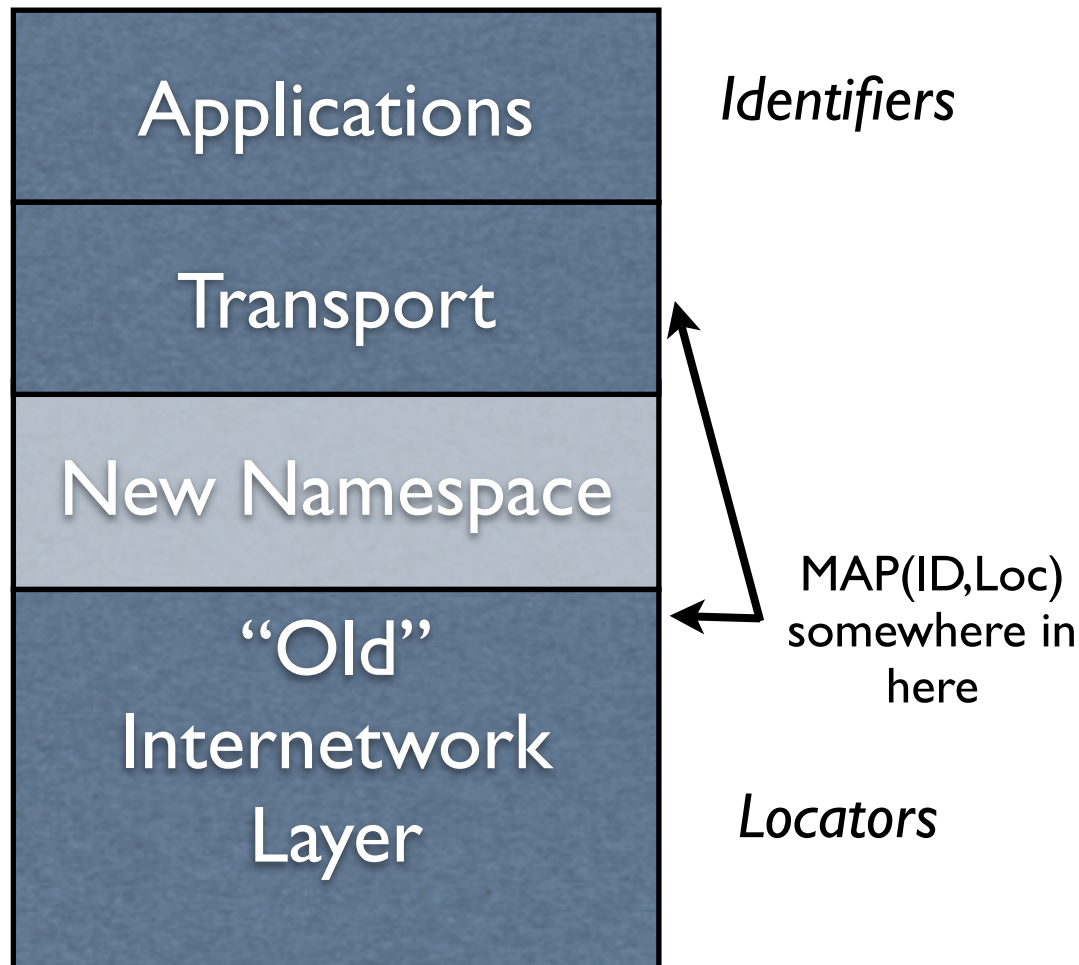
- In this case a new namespace is deployed, and the hosts see it, so there will need to be changes to hosts
 - Issue: What about the transition/coexistence space, and if we never get to IPv6, are IPv6-only solutions of interest (for example, SHIM6)?
- The new namespace can have the same sub-cases (same syntax or not), but this is less important here as you have to modify the hosts in any event
- Examples include SHIM6 and HIP, and GSE
 - GSE is a kind of a hybrid

From the “Layers” Perspective

- Basically, you have an existing namespace which does two things: identification and location
- You can migrate one or both of these functions to a new namespace, with the new namespace inserted either above or below the “old” internetwork layer
 - If the new namespace is “above” the old internetwork layer, applications and/or transport must be modified to use the new namespace
 - If the new namespace is “below” the old internetwork layer, then you have to “jack up”¹ the existing internetwork layer

¹ The term “jack up” is due to Noel Chiappa

New namespace inserted above “old” internetwork layer



- Applications/transport are modified to use new namespace¹
- Essentially adds a new end-to-end naming layer
- Examples include SCTP, HIP and SHIM6
- SHIM6 keeps the old syntax, minimizing code changes; HIP has a new namespace, but provides a local mapping from the old namespace(s) to the new (again, minimizing code changes)

¹ As Pekka Nikander has observed, the separation of location and identification can be made at (at least) 3 different locations in the IP stack: Above the transport layer, in the transport layer, or in the network.

New namespace inserted below “old” internetwork layer



- Applications and/or transport use old namespace
- The old internetwork layer is “jacked up” and a new namespace and internetwork layer are inserted below it
- The old internetwork layer becomes a new end-to-end naming layer
- Rewriting “in the network” somewhere

Old Internetwork layer is “jacked up” to allow insertion of the new namespace

A Third Possibility

- Jim Bound has also suggested that one might try to migrate *both* location and identity out to new, specialized namespaces
- This has the effect of eventually either abandoning the existing IPv4 namespace entirely, or perhaps
 - keeping the IPv4 space for scope local forwarding
- Not a fleshed out idea

YAWTLAT¹

Approach	Encap	Protocol	Security
LISP	IP Tunnel	ICMP/New	undefined
SHIM6	Context Tag	SHIM6	HBA/CBA + RR
HIP	IPsec ESP	HIP BE	ORCHID + RR

¹Chart due to Pekka Nikander (see ram@iab.org); see also draft-nikander-ram-generix-proxying-00.txt

Aside: Interesting Alternative Model¹

- Use MPLS, augmented with a new routing-name namespace
 - BTW, MPLS could be considered to be in the “Not visible to hosts but with a new syntax” class of namespaces
- The idea would be to split the internetwork layer into a “host-to-first-hop-router” protocol and a “router-to-router” protocol
- This model is in “jack up” class of schemes, and posits a different architectural model for the internetwork layer to allow for easier TE and aggregation
- Bananas?

¹Suggested in conversations with Noel Chiappa

Other issues in the Loc/ID split space¹

- Do we need to allocate IDs in an aggregatable fashion?
 - Are IDs routable in some scope?
 - Perhaps with a different AFI/SAFI?
- Secure Locator/ID mapping service?
 - Can Unidirectional Mapping help?
 - ID -> Locator, but not the reverse?
 - Can DHT's be used for this purpose?
 - or the DNS?
 - perhaps in conjunction with Return Routing (RR)?

¹ NACL (Not A Complete List)

Other issues in the Loc/ID split space¹

- More generally, can we preserve secure end to end identity in the presence of loc/id split?
- And what about TE?
 - Presumably, part of the reason to do the loc/id split is to be able “more aggressively” topologically aggregate. If we don’t carry more specifics, then how do we do TE?
 - BTW, SPs seem to be wedged on the TE front
 - In particular, they need (given current practice) more specifics for TE, while at the same time they don’t want more specifics as they cause bloat in various places (e.g., RIBs)

¹ NACL (Not A Complete List)

Reviewing The Problem Space

- Scaling the Control Plane
 - Controversial
- IPv4 Address Depletion
 - But when?
 - Other deployment considerations
- Lots of Interdomain Issues
 - Provider Multihoming, Stateful Mobility, Address Migration, TE/Load sharing, Explicit Routing, Security Routing Paths (and for ER), Service Location, Network Partitioning, Single end-point in multiple service domains, Security behind proxies, ...

Reviewing The Problem Space

- Lots of Intradomain Issues Too
 - Site multihoming, Stateful Mobility, Clustering, Service Location, Connection Policy, Middle Box Services, ...
- And what role should hosts play?
 - Host-based (or host involved) solutions
- Not a complete list (NACL)

Discussion/Q&A

Thanks!