LISP-NERD/CONS, eFIT-APT and Ivip – and some challenges common to them all

Robin Whittle - First Principles

Melbourne Australia

http://www.firstpr.com.au/ip/ivip/

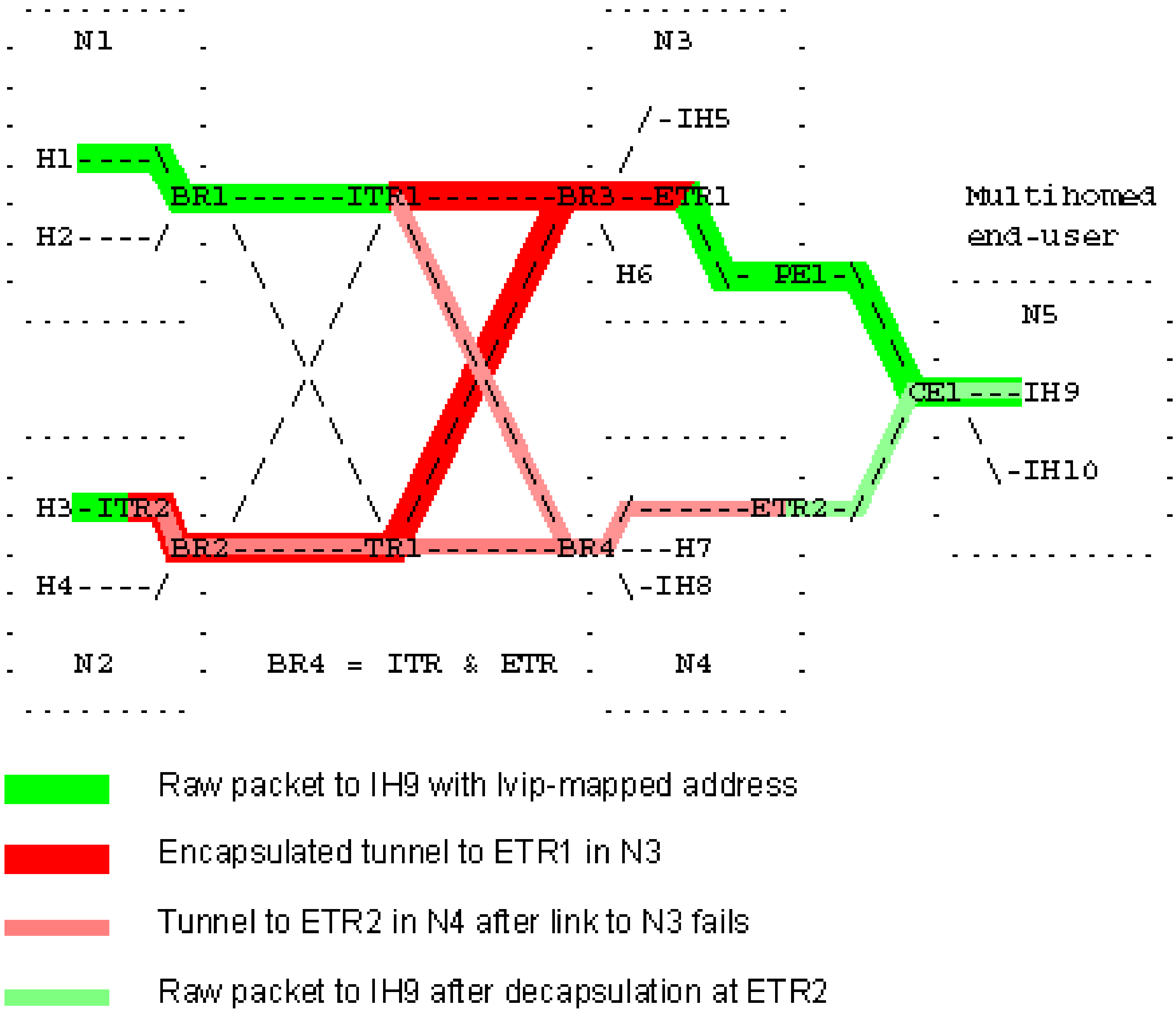
Ivip (Internet Vastly Improved Plumbing) is my proposal.

My understanding of LISP and eFIT-APT may not be ideal.

2007-07-28

General features

	SHIM6	MIPv6	LISP-NERD	LISP-CONS	eFIT-APT	Ivip
Address portability			Y	Y	Y	Y
Multihoming	Y		Y	Y	Y	Y
Support for Mobility		Y				Y
IPv4 too			Y	Y	Y	Y


N1
N3
IH5
H1
BR1
ITR1
BR3
ETR1
Multihomed
end-user
H2
H6
PE1
N5
CE1
IH9
IH10
H3
ITR2
ETR2
BR2
TR1
BR4
H7
H4
IH8
N2
BR4 = ITR & ETR
N4
Raw packet to IH9 with Ivip-mapped address
Encapsulated tunnel to ETR1 in N3
Tunnel to ETR2 in N4 after link to N3 fails
Raw packet to IH9 after decapsulation at ETR2

Functional elements

	LISP-NERD	LISP-CONS	eFIT-APT	Ivip
Mapping data authority	Multiple servers	CARs	Default Mapper	Tree-structure of Update Authorisation Servers
Mapping data distribution	Poll & HTTP download	CAR-CDR-CAR network	Existing BGP routers	Ambitious Replicator system (servers)
ITR functions with full db	ITR		Default Mapper	ITRD
ITR functions with cache		ITR	ITR	ITRC, ITFH (in sending host, not behind NAT)
Query servers with full db				QSD
Query servers with cache		CAR	Default Mapper	QSC

Multihoming service restoration speed - 1

	LISP-NERD	LISP-CONS	eFIT-APT	Ivip
Method of distributing mapping info	Pull. * Poll & HTTP download database and updates	Pull. Ambitious CAR-CDR-CDR network	Push. New BGP messages	Push. * Ambitious Replicator system.
Changed mapping to ITR speed	Slow	Cache time + few secs?	Very slow	Fast – a few secs?
Trade-off cache-time vs. query-response traffic load?	Propagation of updates to ITRs can only be speeded up by reducing cache (poll) time and so increasing the global load of query (poll) and response packets.		Between ITRs and DMs, yes. DMs get db updates very slowly.	No need: caching ITRs get ‘Notify’ within few secs of update.

*Ivip involves large flows of mapping data to ITRDs and QSDs all over the Net, irrespective of the traffic or queries they handle. LISP-NERD also requires lots of downloads for all the ITRs (more numerous than Ivip’s ITRD and QSDs) to keep up-to-date.

Multihoming service restoration speed - 2

	LISP-NERD	LISP-CONS	eFIT-APT	Ivip
Multihoming service restoration time	Depends on how each ITR (& Default Mapper for eFIT-APT) performs its complex functions, including detecting loss of reachability. All MH service restoration (and TE) functionality must be built into the the protocols and implemented by the ITRs. The mapping database must be previously set to give the ITRs proper instructions within these limited parameters.			Depends on what external MH monitoring system the end-user employs to watch their system, and to automatically change the mapping database - plus (ideally) a few seconds for the mapping changes this system makes to propagate to all ITRDs, ITRCs and ITFHs.
Architectural approach	Database distribution, TE, MH reachability and restoration functions, etc. all defined in monolithic system which is hard to extend without major upgrades to ITRs etc.			Component approach – Ivip to be used with other user-chosen components for portability, MH, TE, optimal path Mobile IPv4/6 etc.

Caching ITR and packets for which it has no mapping - 1

	LISP-NERD	LISP-CONS	eFIT-APT	Ivip
Caching ITR?	None – all are full db	ITR	ITR	ITRC & ITFH
How long does caching ITR take to get up-to-date mapping data?	*	CAR caching time plus a few seconds. Query and response traverse global CAR-CDR-CAR network.	Near instant*, since local Default Mapper has full db.	< 0.2 seconds access to local QSD's full database, which is (ideally) a few seconds behind user's updates.
What to do with packet for which there is no mapping?	(Does not occur.)	Hold it till mapping arrives. Bad!	Pass to DM, which tunnels it instantly.	Hold for a moment or let flow through to a full database ITRD.

* LISP-NERD's mapping timeliness is limited by its poll and download system.
eFIT-APT's DM mapping timeliness is very slow due to reliance on BGP.

Caching ITR and packets for which it has no mapping - 2

	LISP-NERD	LISP-CONS	eFIT-APT	Ivip
Can ITR decide it doesn't want to get mapping for this packet?	(Not applic-able, all ITRs are full db.)	No.	Yes – tunnel it to Default Mapper which will tunnel it to ETR – and send back mapping info, which this ITR may cache or ignore.	Yes – ITRC or ITFH can let it pass to an upstream ITRD, perhaps through other ITRCs, one of which will tunnel it. (Alternatively, tunnel it to an ITRD.) This does not constitute a query.
Packet must be handled by:	First ITR	First ITR	First ITR, which may tunnel it to one of potentially several Default Mappers.	ITRDs tunnel every packet they receive. ITFHs and ITRCs can choose not to tunnel packets, for instance to avoid delay, query-response traffic or load on their cache memory.

Packets from sending hosts in non-upgraded networks

	LISP-NERD	LISP-CONS	eFIT-APT	Ivip
Packets from non-upgraded networks?	?	One border router ITR might advertise and tunnel, so most paths will be sub-optimal. See RAM list msg01730.	?*	Anycast ITRDs in the core handle these packets, with optimal or generally close to optimal path lengths.
Prefixes advertised in BGP?	Not in the long term	Not in the long term	Not in the long term	Forever, but Ivip divides prefix much more finely and freely (in address space and time) than BGP allows – so supporting many more MH end-users than the average prefix of this length does now.

* Future eFIT-APT draft will have more on this question, which is vital to incremental deployment. (RAM list msg01745.)

Multihoming & Traffic Engineering functionality

	LISP-NERD	LISP-CONS	eFIT-APT	Ivip
ITRs do complex communications, accept ICMP?	Yes	Yes	Yes, Default Mappers too	No
ETRs communicate with ITRs?	Yes	Yes	Yes, with Default Mappers too	No
Real time decisions for MH service restoration and TE	Functionality fixed in system, controlled by mapping data and implemented in real time by ITRs etc.			No built-in MH or TE functions. Open-ended - relies on external systems, and fast replication of database updates.

Complex communications, responding to ICMP etc.
= security problems and heavy load on router's CPU.

Encapsulation, outer and inner Source Address

	LISP-NERD	LISP-CONS	eFIT-APT	Ivip
Encapsulation	UDP	UDP	UDP?	IP-in-IP*
Nonces & other stuff	Yes?	Yes?	?	No
Outer SA =	ITR	ITR	ITR?	Sending host
ITR handles Path MTU discovery ICMP packets?	Yes	Yes	Yes?	No
ETR’s task to prevent spoofed local SAs. (Assumes provider BR drops if outer SA = local.)	Assuming ITRs in provider network tunnel packets to the ETR, drop if (inner SA = local) & (outer SA != local). Really costly, since ‘local’ could involve thousands+ of prefixes.			Drop if inner SA != outer SA.

See Ivip Errata: It is impractical to make LISP/eFIT ITRs properly handle ICMP message so the sending host gets an ICMP message it recognises. Ivip’s “Outer SA = Sending host address” is not capable of getting recognisable ICMP messages to the sending host if they are created by routers in the tunneled section of the path. This clobbers traceroute and Path MTU Discovery in the tunneled section.

* Ivip could use UDP (less efficient, but more flexible) if, as seems likely, every tunneled packet should have the ITR’s address within it.

Common Challenges 1: MTU and encapsulation

These proposals are potentially practical because they involve no new host functions and don't mess with BGP. The only way of achieving these goals is apparently to use encapsulation, which means they are all going to cause dropped or fragmented packets unless something is done . . .

New system shouldn't make Path MTU discovery harder, but how healthy is this at present anyway? (RFC 4459)

See Ivip Errata and notes on previous page – I now think the following is not true:

With 'outer SA = ITR' the ITR gets all the ICMP flak and needs to keep a (potentially prohibitive amount of) state about recent sending hosts, in order to somehow get an ICMP message back to the right host.

Ivip uses 'outer SA = inner SA = sending host', which absolves the ITR from all this state and ICMP packet handling trouble – and should preserve Path MTU discovery.

Common Challenges 2: ETR filtering of spoofed local source addresses

Assuming the provider border routers drop packets arriving from outside with SA matching one of the provider’s prefixes (spoofed local source address) then LISP and eFIT-APT require a major filtering task in the ETR to stop the ETR being used by attackers (implicitly outside the provider network) from launching packets through them with spoofed local source addresses.

Ivip uses the unconventional, and in some ways unfriendly ‘outer SA = inner SA = sending host’ approach, which makes it easy for the ETR: If inner SA != outer SA, then drop the packet.

I also think it is best for packets from hosts in the provider network to go via nearby ITRs and therefore to the right ETR, as controlled by the database, rather than relying on the local routing system to follow the intention of the mapping database. (See Ivip-arch I-D: 14.1.2.5 Note 2 - ETRs must handle packets from ITRs in the same network.)

Common Challenges 3: Incremental deployment

New architecture must maintain full reachability from hosts in non-upgraded networks.

New system must provide some benefits (portability and/or multihoming, with less cost, no AS or BGP stuff etc.) to end-users who choose to use the LISP/eFIT-APT/LISP-mapped addresses.

Some end-users will make very few updates to their mapping, others will make a lot. There probably needs to be a charging system per update, to partly finance at least some parts of the system which carry the load of those changes – otherwise, who would want to build and run those parts?

Common Challenges 4: Scrutiny and timeframe

Ideally, the new system would already be ready to deploy.

No matter what we wish, it would be 2010 at least before a new system is fully defined and passes what would be the most intense scrutiny ever. This will be the most ambitious change to the Internet for 2 decades or so – affecting all Internet communications.

The new system will probably marginally reduce the user packet sizes of all Internet communications, except when the sending host is smart enough to know the packet will not be handled by the mapping-tunneling system.

The new system needs to be carefully designed to minimise this impact, and to enable smart hosts to reliably know when they don't need to limit their packet length. This might be part of a more ambitious scheme for autodiscovery of potentially much larger MTUs for hosts who wish to try.

Common Challenges 5: Admin and address space

Unless the RIRs reserve some space – ideally some /8s – then by the time the new architecture starts running, it will have to work with a mess of address space already assigned to providers and end-users.

If we can develop the proposal fast, and show that it can be used to slice and dice IPv4 space much finer and use it much more efficiently than is possible with current BGP and address assignment practices, then maybe the RIRs will reserve some space for the 2012 timeframe when the new system is likely to be widely implemented.

Links

RRG’s wiki with links to proposals:
http://www3.tools.ietf.org/group/irtf/trac/wiki/RoutingResearchGroup

Ivip I-D includes a detailed section comparing Ivip with LISP.
http://www.firstpr.com.au/ip/ivip/

An updated comparison of LISP-NERD/CONS, eFIT-APT and Ivip, with links to latest versions of the proposals.
http://www.firstpr.com.au/ip/ivip/comp/