

Ivip (Internet Vastly Improved Plumbing) Architecture
draft-whittle-ivip-arch-03.txt

Abstract

Ivip (Internet Vastly Improved Plumbing) is a core-edge separation solution to the routing scaling problem, for both IPv4 and IPv6. It provides portable address space which is suitable for multihoming and inbound traffic engineering (TE) to end-user networks of all types and sizes. The subset of the global unicast address space which is used in this fashion is called SPI (Scalable Provider Independent) space. End-user networks divide their SPI space into "micronets", each with a common mapping to a particular ETR (Egress Tunnel Router) address. ITRs (Ingress Tunnel Routers) receive packets which are addressed to SPI addresses and, after looking up the mapping at a local query server, tunnel traffic packets to the ETR which delivers them to the end-user network. Ivip pushes mapping changes in real-time to local full-database query servers - such as in each ISP. These answer ITRs' mapping queries and push subsequent changes to the mapping to any ITRs which would still be caching the mapping of an earlier reply. The mapping for a micronet consists of a single ETR address. ITRs make no decisions between multiple ETR addresses. End-user networks would typically contract a separate company to change the mapping of their micronet, in response to the reachability of their ETRs and TE and portability between ISPs. Ivip includes two extensions for ITR-to-ETR tunneling without encapsulation and the resulting Path MTU Discovery problems - one for IPv4 and the other for IPv6. Both involve modifying the IP header and require most DFZ routers to be upgraded. Ivip is a good basis for the TTR (Translating Tunnel Router) approach to mobility, in which mobile hosts retain an SPI micronet of one IPv4 address (or IPv6 /64 prefix) no matter what physical addresses they are using, including behind NAT. TTR mobility for both IPv4 and IPv6 involves generally optimal paths and works with unmodified correspondent hosts.

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that

Commentaire [Med1] : After reading the document, I have the following comments (some of them may be valid for other CES proposals)
-a figure to show all involved functional elements would ease the readability of this document.
-An example of call flow would be also more than welcome
-The document does not discuss if single homed networks needs to deploy ITR. Why they have to pay this cost?
-The document does not assess the impact of the presence of several LQSD on the validity of the stored information
-QSD may be seen as a single point of failure
-Due to traffic growth, QSD must be able to handle a big amount of request
-Several interconnection layers may be defined: the physical one with BGP interconnection, on top of it service providers who deploy the CES, interconnection between these SP is required. During the bootstrap, the SPI must be advertised in the core, then it does not solve the scalability issue. This situation will be valid unless a global deployment is adopted
-Deployability issues: what to do when several version of table structures, protocol exchange are to co-exist?
-This document encloses some business considerations, this is an added value compared to other proposal but the concern I have is that some statement are subjective
- How to assess the flexibility of the proposed system. Being part of the system should not lead to a frozen situation where no modification is possible: for instance adding/remove/modifying reachability information of ITR/ETR/QSR/RUAS/LQSR should be doable without impact

other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/lid-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on July 17, 2010.

Copyright Notice

Copyright (c) 2010 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the BSD License.

Table of Contents

- 1. Introduction 5
- 2. Goals 9
 - 2.1. IPv4 and IPv6 9
 - 2.2. Portability, multihoming and TE for billions of end-user networks 9
 - 2.3. Modular separation of the control of mapping from the core-edge separation architecture itself 11
 - 2.4. Simple ITRs and ETRs with little or no communication between them 11
 - 2.5. Maximise the flexibility with which ITRs and ETRs can be located 12
 - 2.6. Mobility 13
 - 2.7. Elimination of encapsulation and PMTUD problems 14
 - 2.8. No requirement for new host functionality 15
 - 2.9. Full benefits to all adopters 15
 - 2.10. Business incentives to deploy new infrastructure 16
 - 2.11. Maintenance of existing levels of security and robustness 17
- 3. Non-goals 19
 - 3.1. Complete core-edge separation is not required 19
 - 3.2. Mapping changes need not be free of financial cost 20
 - 3.3. No attempt to cope with partially reachable ETRs 21
 - 3.4. No attempt to avoid all the mapping data being stored at any one location 23
 - 3.5. It may not be possible to completely eliminate unfair burdens 24
 - 3.6. No attempt to mix IPv4 and IPv6 25
 - 3.7. Not Locator - Identifier Separation 25
- 4. Architectural Choices 27
 - 4.1. Core-edge separation rather than elimination 27
 - 4.1.1. Core-Edge Elimination (CEE) architectures 27
 - 4.1.2. Core-Edge Separation (CES) architectures 29
 - 4.2. Local query servers 31
 - 4.3. Real-time mapping distribution 32
 - 4.4. SPI address management 32
 - 4.5. IP in IP encapsulation 35
 - 4.6. MHF initially or in the long term to avoid encapsulation and PMTUD problems 35
 - 4.7. Outer header address is that of the sending host 35
 - 4.8. IPTM (ITR Probes Tunnel MTU) PMTUD management 36
- 5. Architectural Elements 39
 - 5.1. ITRs 39
 - 5.1.1. Types of ITR and their addresses 39
 - 5.1.2. DITRs - Default ITRs in the DFZ 40
 - 5.1.3. Modified Header Forwarding - MHF-only ITRs 41
 - 5.1.4. Encapsulation and PMTUD management 41

- 5.1.5. Mapping lookup and caching 43
- 5.1.6. ITFH - ITR Function in Host 45
- 5.1.7. ITRs auto-discovering local query servers 46
- 5.1.8. ITRs regulating their advertisements 46
- 5.2. ETRs 47
 - 5.2.1. In servers or dedicated routers 47
 - 5.2.2. ETRs in ISP networks 47
 - 5.2.3. ETRs at the end-user network site 47
 - 5.2.4. MHF ETR functionality - EAF and PLF 48
 - 5.2.5. ETR functionality for encapsulation 49
- 5.3. QSDs - full database query servers 50
 - 5.3.1. Placement in ISP and end-user networks 50
 - 5.3.2. QSD initialization and reception of mapping updates . 51
 - 5.3.3. Responding to queries 53
 - 5.3.4. Sending mapping updates to ITRs and QSCs 54
- 5.4. QSCs - caching query servers 55
- 5.5. FMS - Fast-Push Mapping Distribution System 56
- 5.6. MHF - Modified Header Forwarding 63
 - 5.6.1. EAF - ETR Address Forwarding for IPv4 63
 - 5.6.2. PLF - Prefix Label Forwarding, for IPv6 64
- 5.7. TTR Mobility 65
- 6. Security Considerations 67
- 7. IANA Considerations 68
- 8. Informative References 69
- Appendix A. Acknowledgements 72
- Author's Address 73

1. Introduction

Version 03 of this memo is a freshly written document which is shorter than the original from 2007. Some terminology has been changed and the presentation is optimised for people who are involved in the RRG. Please see [I-D.whittle-ivip-glossary] for definitions of some terms and acronyms. Please refer to the RRG mailing list and <http://www.firstpr.com.au/ip/ivip/> for the latest developments.

The Ivip (pr. "Eye-vip") project began in June 2007 and in early 2010 is one of the three Core-Edge Separation (CES) architectures being considered by the RRG (IRTF Routing Research Group) [I-D.irtf-rrg-recommendation] - the others being LISP [I-D.ietf-lisp] and TIDR [I-D.adan-idr-tidr].

The routing scaling problem can be summarised as there being practical limits and unfair cost-burdens to providing portability, multihoming and inbound traffic engineering (TE) for the potentially large number of end-user networks which want or need this. The same problem exists in principle for IPv4 and IPv6, but only IPv4 has a problem at present.

The burden placed on the inter-domain routing system (often referred to loosely as the Default-Free Zone - DFZ) by the advertised prefixes of ISPs (Internet Service Providers) is generally thought not to be a problem. So the challenge is to find a way of providing address space and new methods of routing so that the portability, multihoming and TE needs of potentially millions or billions of end-user networks can be served in a "scalable" manner: efficiently, robustly and without unfair burdens falling on anyone, such as those who operate the DFZ routers.

The unfair, unsustainable, burden is caused by the number of separately advertised PI (Provider Independent) prefixes of end-user networks today - and the rate at which these prefixes have their point of advertisement changed. (Also, if an end-user network changes the type of advertisement frequently, such as with more or less ASNs, this too is a burden.)

The most important part of the burden is on the DFZ's "BGP control plane". This is partly the inter-router BGP traffic and the overall behaviour of routers - particularly any difficulty which the excessive number of prefixes causes in the system converging to good enough best-paths in the event of an outage. It is also the burden of CPU effort and storage in the RIB of each router. This includes the effort of writing changes to the FIB when RIB information changes. Also, FIBs may have their ability to handle packets temporarily disabled while new information is written.

Commentaire [Med2] : GENERAL COMMENT: I think that a motivation section to question the current needs which justify the introduction of CES or CEE need to be elaborated: what is the trend of PI advertised. If BGP system is able to manage that maximum, then it is more about education effort to enforce best practices of prefix aggregation, etc.

Supprimé : Ivip-arch ID

The actual number of prefixes each DFZ router has to handle is a major part of the problem, though the total RIB burden also depends on how many neighbours each router has. The number of prefixes in the FIB is a serious burden too, but it is widely believed that this is not the most important problem. Any solution which only helps with the FIB is not really a solution to the problem.

In order to provide portability, for instance to millions or perhaps billions of end-user networks in a scalable manner, it follows that the DFZ routers must not have to consider the prefixes of each individual network at all in their RIB or FIB. Consequently, most approaches to solving the problem concentrate on providing a new type of address, or a subset of existing addresses, which is suitable and attractive for providing end-user networks with portability, multihoming and TE, but which places only very slight burden on the DFZ compared to the burden each PI prefix places today.

Supprimé : etc.

Support for mobility has not generally been considered part of the routing scaling problem. However, with the proliferation of cellphones, other IP applications it is reasonable to assume that in the future, most hosts will be mobile devices, generally running on limited battery power and relying on wireless links which are frequently slow, unreliable and/or expensive.

Commentaire [Med3] : No value to mention VoIP here.

It think that the current trend is inline with this statement. The mobile traffic is drastically increasing. Developing countries adopt mobile infrastructure rather than fixed one.

Mobility is arguably an extreme form of portability and/or multihoming. To embark on a major architectural enhancement for scalable routing, in a manner which did not support billions of mobile devices, would make little sense. So while mobility is not a formal part of the routing scaling problem, Ivip and some other proposed architectures (e.g., [REF]) seek to provide mobility on a massive scale.

In addition, new use cases such us sensor networking and M2M may advocate for more and more devices to be connected.

In these scenarios, each mobile device is generally considered to be a separate end-user network. An entire corporation's network, or that of a large university, is also an "end-user network". So in the following discussion, this term could mean a wide variety of things - far beyond the small subset of end-user networks which are currently able to gain and advertise PI space.

Supprimé : VoIP,

Commentaire [Med4] : In the future, this may not be valid since some ISPs offer already mobile broadband services.

Apart from Core-Edge Separation architectures, there are three other classes of architectural enhancement for solving the Internet's routing and addressing scaling problem.

Commentaire [Med5] : MIM O techniques or any other technique to ensure robustness of mobile traffic may be envisaged.

Firstly, improving BGP so the DFZ can cope with vastly greater numbers of end-user prefixes - which is impossible. Secondly, replacing BGP with some other arrangement so the inter-domain routing system can cope with this increase in the number of end-user network prefixes. It may be technically possible to devise such a scheme - but there appear to be insurmountable difficulties in introducing it to the operating IPv4 or IPv6 Internets. Thirdly, Core-Edge

Elimination (CEE) [REF] architectures promise scalable routing and a simplified (compared to CES architectures), arguably more elegant, network - with hosts gaining new routing and addressing responsibilities and with the DFZ working much as it does today, but only for ISP prefixes.

The reasons for choosing Core-Edge Separation over Elimination for Ivip are described in detail in Section 4. In summary, the CES approach is the only one which meets the constraints imposed by the need for widespread voluntary adoption. Furthermore, and contrary to some widely held beliefs, I argue that network-centric approach of CES architectures produces a better performing Internet than one in which hosts do all the routing and addressing work. This is particularly so considering that in the future, the majority of hosts ~~would be operating over typically slow, unreliable and potentially costly wireless last-mile links.~~

Supprimé : will

Commentaire [Med6] : As mentioned above, this may not be a valid argument.

Ivip is based on some unique architectural choices, including: ITRs (Ingress Tunnel Routers) receiving mapping changes in real-time; local full-database query servers; migration to Modified Header Forwarding (MHF) to avoid encapsulation and its PMTUD (Path MTU Discovery) difficulties; and (when encapsulation is used) the use of the sending host's address as the outer header's source address, so that ETRs can easily enforce ISP BR (Border Router) source address filtering on de-encapsulated packets.

The following description assumes that Ivip will be introduced with encapsulation, with long-term migration to MHF (Modified Header Forwarding).

However, it is

possible that by the time the introduction date is set that most DFZ routers will have ~~firmware-based FIBs~~, and so could be easily upgraded to support MHF. In that case, ITRs and ETRs could be much simpler, since they would not need to handle encapsulation or PMTUD management.

Supprimé : firmware

Below, Ivip is generally assumed to be introduced as a single system for the purposes of solving the routing scaling problem. However, multiple independent systems along the lines of Ivip (with encapsulation) could also be introduced without need for standardisation for the purpose of supporting commercial TTR Mobility services.

Commentaire [Med7] : What means system here?

I publicly disclose and discuss all Ivip developments as rapidly as possible in order to gain support and constructive critiques - and in the hope that any novel ideas will remain free from patent encumbrances.

This ~~memo~~ is intended for readers who are broadly familiar with the routing scaling problem and RRG discussions and who have, ideally,

Supprimé : ID

familiarised themselves with LISP.

This ~~document~~ provides not only a general description of Ivip, but the rationale for architectural choices which distinguish Ivip from other approaches. Some aspects of Ivip's architecture are discussed in greater detail in separate documents:

Supprimé : ID

The fast push mapping distribution system to the local query servers could be implemented in various ways, such as flooding mapping data through a meshed network of servers. The current plan involves a global network of servers in a structure resembling crosslinked multicast trees. Please refer to [I-D.whittle-ivip-db-fast-push] for further details.

The TTR approach to mobility is described in [TTR Mobility].

The IPv4 approach to Modified Header Forwarding (MHF) is described in detail in [I-D.whittle-ivip-etr-addr-forw]. The IPv6 approach is described in ~~[PLF-for-IPv6]~~ and the best summary of its operation can be found at the end of the ~10k word Ivip Conceptual Summary and Analysis: ~~[Ivip-Summary-and-Analysis]~~ .

Supprimé : PLF

Supprimé : for

Supprimé : Ivip

Supprimé : Summary

Supprimé : and

Ivip's approach to Path MTU Discovery, when ITRs tunnel using encapsulation, is discussed in [PMTUD-Frag].

2. Goals

2.1. IPv4 and IPv6

Ivip is intended to solve the routing scaling problem (as described in the introduction), for IPv4 and IPv6, for very large numbers of end-user networks.

It may be argued that, since IPv6 allows for larger prefixes, fewer IPv6 inter-domain routing entries would be required per AS (Autonomous System). Nevertheless, even if recommended practices for aggregation are followed by a given AS for its prefixes, the routing table size would grow due to multi-homing and the assignment of PI IPv6 prefixes to end sites.

Supprimé : ¶

Much of Ivip is identical in principle for both Internets. However the mapping information for IPv6 is lengthier and there are other differences, such as in Path MTU Discovery (PMTUD) when encapsulation is used, and in the IPv4 and IPv6 approaches to MHF which remove the need for encapsulation.

2.2. Portability, Multihoming and TE for Billions of End-User Networks

Ivip is intended to provide scalable address space for billions of end-user networks - for both IPv4 and IPv6. The new kind of address space - SPI (Scalable Provider Independent) space - is suitable for end-user networks to use in a portable fashion, meaning they can keep this space when choosing another ISP for Internet connectivity service.

- Supprimé : multihoming
- Supprimé : billions
- Supprimé : end
- Supprimé : user
- Supprimé : networks

Portability of the end-user network address space which is used to identify hosts, routers and networks is an absolute requirement of scalable routing. Even if a network could reliably and inexpensively renumber all its hosts and routers, and change all its configuration files which contained such addresses, it would never be able to reliably and securely alter all the other places where these identifying addresses reside in other networks. These includes the use of these addresses in referrals, existing communication sessions, configuration files of VPNs and hard-coded (however questionably) into firmware and software.

Assuming the end-user network has two or more ISPs, SPI space will also support multihoming and inbound traffic engineering. In the following, TE refers to Inbound Traffic Engineering - the ability to steer incoming traffic streams between two or more ingres points (e.g., two ISP connection links). Ivip's

approach to TE differs from that of other core-edge separation schemes. It is potentially finer-grained, more flexible and more able to respond to rapid changes in traffic patterns.

Ivip is intended to provide SPI space for billions of end-user networks.

- Supprimé : "
- Supprimé : "
- Supprimé : "i
- Supprimé : traffic
- Supprimé : engineering
- Supprimé : "
- Supprimé : ISPs

With about 3.7 billion unicast IP addresses, it seems unlikely that billions of end-user networks could exist within IPv4, but it would be technically possible if most of the space was reserved for such networks, if most of them used a single IP address for their

- Commentaire [Med8] : Already mentioned above. Can be deleted.
- Mis en forme : Surlignage

micronets, and if most of the unicast address space was used for SPI space.

An upper estimate of the number of non-mobile networks which would want or need SPI space is in the 10^7 range. This is on the basis of a population of 10^{10} , with there being only one organisation per 10^3 people which needs portability, multihoming and/or TE enough to establish a physical link to a second ISP and pay two ISPs rather than one for Internet access.

This is an order-of-magnitude estimate. Perhaps with inexpensive wireless links to another ISP, increasing numbers of end-user networks would want to multihome instead of relying on their single fibre, DSL or HFC cable connection. Already, many homes and offices can connect via two or more of these types of cabled connections. Nonetheless, even if it was convenient and inexpensive, it is hard to imagine more than 10^7 non-mobile end-user networks being sufficiently concerned about multihoming or portability to use SPI space, rather than the small amount of PA space they currently receive from their single ISP.

The goal of scalable routing is to scalably provide portability, multihoming and TE to all networks which want or need it. However, it is reasonable to assume that most home and SOHO networks, and some smaller factory and office networks, will remain happy with the reliability of their single-provider service, and will not be concerned about portability when choosing another ISP.

On this basis, scalable routing architectures which are intended to scale to more than about 10^7 end-user networks will be doing so primarily for mobile end-user networks, typically consisting of a single hand-held device which operates from a single IPv4 or IPv6 address or from a single IPv6 /64 prefix.

A small number of end-user networks will have multiple sites or some other reason to split their SPI space into multiple micronets, but in any realistic scenario involving billions of such networks, the great majority of such networks will be a single site or device, with little or no need for TE or greater address space than a single IPv4 address or an IPv6 /64. Therefore, it is reasonable to expect that most of these billions of networks will require only a single micronet of SPI addresses. So, for these scenarios of billions of end-user networks, the total number of separately mapped micronets of SPI address space will be only marginally greater than the number of end-user networks.

Commentaire [Med9] : I think all this figures are some kind of speculative data . Can be removed.

2.3. Modular ~~Separation of the Control of Mapping from the Core-Edge Separation Architecture~~

Ivip's real-time mapping system means that the tunneling behaviour of all ITRs can be controlled directly. The mapping consists of a single ETR address, so unlike with other ~~Core-Edge Separation~~ architectures, Ivip ITRs do not need to make any choices between multiple ETRs for the purposes of multihoming service restoration or TE.

Control of the tunneling behaviour of Ivip ITRs ~~remains~~ entirely outside the Ivip system. It is the responsibility of end-user networks to control this mapping at all times - and they are likely to delegate this responsibility to a ~~third party~~. Exactly how end-user networks make their decisions about mapping - and how, for instance, a Multihoming Monitoring (MM) ~~system~~ might detect ETR failure, and alter mapping accordingly - is entirely separate from Ivip's mapping system, ITRs and ETRs.

It would be a private, ~~flexible~~, arrangement between an end-user network and ~~a MM company it hires to continually probe the network's reachability via its two or more ETRs~~. This means the frequency and type of probing, and the decision-making algorithms, can be completely open-ended and subject to development and customisation - without any constraints or need for changes in the ~~specification document~~ which define Ivip.

This modular separation of the detection and decision-making functions from the ~~Core-Edge Separation~~ is ~~claimed to be a good engineering practice~~ and ensures that the Ivip subsystem can be used flexibly, including for purposes not yet anticipated.

Other CES techniques monolithically integrate the following functions into the ~~Core-Edge Separation~~ architecture itself - primarily by specifying exactly how all ITRs must behave: (1) multihoming failure detection, (2) decisions about how to choose between ETRs to restore service and (3) how to implement TE. This would add greatly to the complexity of the system itself, make it harder to introduce new methods of testing reachability, etc. and restrict all end-user networks to relying on the necessarily restricted set of functions which can reasonably be built into all ITRs.

2.4. Simple ITRs and ETRs with ~~Little or no Communication Between Them~~

With encapsulation, the only time ITRs engage in two-way communication is when probing the Path MTU to ~~a given ETR~~, by using a special pair of packets which carry a larger traffic packet than has previously been successfully received by the ETR from this ITR.

- Supprimé : separation
- Supprimé : control
- Supprimé : mapping
- Supprimé : core
- Supprimé : edge
- Supprimé : separation
- Supprimé : architecture
- Supprimé : itself
- Supprimé : core
- Supprimé : edge
- Supprimé : separation
- Supprimé : rests
- Supprimé : company they hire for this purpose
- Supprimé : company
- Commentaire [Med10] : Why flexible ? compared to what ?
- Supprimé : the
- Supprimé : RFCs
- Supprimé : core
- Supprimé : edge
- Supprimé : separation
- Commentaire [Med11] : Can be dropped
- Commentaire [Med12] : This is somehow subjective. At this stage, the reader can agree or disagree with what is claimed.

Personally, when it reads "flexibility" in several places, this worries me and I'm looking where the complexity is hidden.

I'd suggest that this discussion to be removed or at least moved to the conclusion section for instance.
- Supprimé : core
- Supprimé : edge
- Supprimé : separation
- Supprimé : 1
- Supprimé : communication
- Supprimé : between
- Supprimé : them
- Supprimé : the

Apart from this, ITRs do not communicate with any remote entity but their local query servers. ETRs do not communicate with any other entity than the ITR for this PMTU management function.

Supprimé : anything
Supprimé : part of the I
 Ivip system

If MHF is used rather than encapsulation, there is no need for ITRs to communicate with ETRs - so ITRs only communicate with query servers and ETRs do not communicate at all.

Consequently, ETRs and ITRs can be simple functions in existing routers or in standalone devices. The ITR function can also be implemented in the sending host (ITFH), though this is not advisable if the sending host is on a slow and unreliable link. ETRs must be addressed using conventional global IP (v4/v6) unicast addresses - not SPI addresses. ITRs can be both kinds of address. Ivip may include an option for an ITR or ITFH to set up a two-way persistent tunnel to its one or more local query servers, which would allow an ITR function to be behind one or more layers of NAT. This tunnel could be as simple as TCP from the ITR, or ITFH, to each query server, with keepalive packets.

Supprimé : servers
Supprimé : ,
Supprimé : ,
Supprimé : such as a wireless link
Supprimé : on
Supprimé : "
Supprimé : "
Mis en forme : Surlignage

It is important to make ITRs as simple as possible, in order that they may be inexpensive and therefore, if desired, more numerous - so as to reduce the load on each one. ETRs are simpler than ITRs, since they simply de-encapsulate received packets with a comparison between outer and inner source addresses and do not look up or cache mapping information.

Ivip with encapsulation uses simple IP-in-IP encapsulation. There is no special header and no other data piggybacked onto traffic packets. This minimises encapsulation overhead and reduces the complexity of both ITRs and ETRs. Other Core-Edge Separation architectures use their own headers to carry extra information with each traffic packet, with that header behind a UDP header. These other architectures also require ITRs to determine reachability to multiple ETRs.

Supprimé : core
Supprimé : edge
Supprimé : separation

2.5. Maximise the Flexibility with which ITRs and ETRs Can be Located

Commentaire [Med13] : Wh at means flexibility here ?
Supprimé : flexibility
Supprimé : can
Supprimé : located
Supprimé : N

Ivip ITRs can be located in the sending host, in the sending-host's end-user network (which may be an ISP network or an end-user network using either SPI or conventional PI space) or in the ISP network which the host's end-user network connects to the Internet through. If there is no such ITR, the packets will enter the DFZ and be forwarded to the nearest (in BGP terms) DITR (Default ITR in the DFZ, previously known as OITRD for Open ITR in the DFZ).

ETRs can be located in ISP networks with a link to the end-user network. ETRs can also be located at the end-user network end of a

link from an ISP, and so be physically located at the end-use site. In both cases, their address must be a conventional global unicast address (usually from one of the ISP's prefixes) - not an SPI address or behind NAT devices.

2.6. Mobility

One of Ivip's goals is to support massive mobility, since this is surely a major facet of the future of Internet communications. It would make no sense to introduce one set of architectural changes to solve the routing scaling problem as it appears today, and then have to devise and introduce a second set to provide for billions of mobile devices.

It is frequently assumed that in order for a Core-Edge Separation scheme to support mobility, the Mobile Node (MN) must be its own ETR. LISP-MN makes this assumption. So does [I-D.jen-mapping] - a critique of which is [Critique of draft-jen-mapping-00].

TTR mobility does not involve mapping changes every time the MN (Mobile Node) gains a new physical address, since it continues to use the same one or more TTRs as its one or more ETRs. Mapping changes are needed only to use a new TTR, which is desirable after the MN moves a large distance, such as 1000km or more.

Although the TTR approach to mobility could be used with other core-edge separation schemes, Ivip is a better basis for TTR mobility than other core-edge separation schemes such as LISP, APT [I-D.jen-apt], TRRP [TRRP] or TIDR. None of these proposals provide a method of ITRs gaining updated mapping within a few seconds, as Ivip does. With Ivip's real-time mapping system, MN can begin using a new, nearby, TTR within seconds and, most importantly, within a few seconds no ITR will be tunneling packets to the previous, and now more distant, TTR. Therefore, the MN can promptly end the tunnel to the previous TTR and use the new TTR exclusively. Without this real-time mapping, the MN would need to retain tunnels to one or more previous TTRs for as long as the mapping system takes to ensure no ITRs are tunneling packets to them. This could take 10 to 30 minutes or more for the non-Ivip core-edge separation schemes.

TTR Mobility is not required to solve today's routing scaling problem. It may be regarded as separate to Ivip, because it could be used with other core-edge separation schemes. However, it is best to consider TTR Mobility as a natural extension of the basic Ivip

Supprimé : Ivip is a good basis for the TTR approach to mobility, and would be attractive to deploy for this reason alone.

Supprimé : core

Supprimé : edge

Supprimé : separation

Supprimé : draft-jen-mapping-00

Commentaire [Med14] : What do you mean by « physical address » ?

Mis en forme : Surlignage

Supprimé : other

Commentaire [Med15] : Add a ref.

Supprimé : the Mobile Node (

Supprimé :)

Commentaire [Med16] : Add a ref for measurement

architecture, which does not place any constraints on the basic architecture other than that its mapping system will need to scale to billions of (mostly mobile, handheld device) end-user networks.

2.7. Elimination of ~~Encapsulation~~ and ~~PMTUD Problems~~

Supprimé : encapsulation

Supprimé : problems

When ITRs use encapsulation to tunnel traffic packets to ETRs, there are serious problems with Path MTU Discovery (PMTUD) for the sending host. If the packet with its encapsulation header is too long for the next hop link of some router between the ITR and ETR, then there needs to be a mechanism by which the sending host receives a valid ICMP Packet Too Big message, with an MTU value which will result in an encapsulated packet of the correct length.

It is challenging to solve this problem securely and without unreasonable amounts of state in the ITR. Ivip's solution - ITR Probes Path MTU [PMTUD-Frag] - involves extra complexity and state in ITRs and to a lesser extent in ETRs. This, and the transmission overhead of the encapsulation header (particularly heavy with IPv6 VoIP packets) provides strong motivation to either avoid encapsulation entirely, or to introduce Ivip with encapsulation, but in the long-term transition to an alternative system which lacks these problems.

Ivip has two techniques, known collectively as Modified Header Forwarding (MHF) which replace encapsulation as the ITR to ETR tunneling technique. They are:

Supprimé : ¶

1. ETR Address Forwarding (EAF) - for IPv4. [I-D.whittle-ivip-etr-addr-forw]
2. Prefix Label Forwarding (PLF) - for IPv6. [~~PLF-for-IPv6~~].

Supprimé : PLF

Supprimé : for

If Ivip is introduced with encapsulation, all ITRs and ETRs will be capable of supporting MHF. At some date in the future, the DFZ routers will be upgraded to support this, probably without any significant cost.

Ideally, it would be possible to establish Ivip from the outset without encapsulation. This would save having to develop the more complex ITR and ETR functions required by encapsulation - especially the PMTUD functionality. It would also eliminate the need to design a transition arrangement.

It may be many years before Ivip or something like it is scheduled for deployment. I have not been able to reliably determine what

proportion of current DFZ routers have firmware-based FIBs. Any such router could be upgraded with a firmware update in order to support MHF. As the years pass, the chance that most or essentially all DFZ routers could be upgraded in this way, for very little cost, increases. Initial deployment with MHF is a goal, with the alternative goal being eventual transition to MHF.

2.8. No ~~Requirement~~ for ~~New Modifications on the Host~~.

It is a primary goal not to require any new host functionality - in stacks or applications. However, as an option, the ITR function can be integrated into sending hosts when this is desired.

Mobile hosts using the TTR Mobility approach will have a little extra functionality, which could be implemented in the stack or perhaps outside it, as a separate piece of software. The IP stack itself and all applications remain unchanged and communicate with all other hosts, mobile or not, using by current protocols and addressing.

One reason for avoiding the need for new host functionality is to enable the system to be widely enough adopted to solve the routing scaling problem, given the constraints imposed by the need for voluntary adoption. [Constraints-Voluntary]

Another more fundamental reason is to ensure there is no extra burden on hosts, which would be particularly a problem for hosts which are on slow, expensive and unreliable links. This includes hosts on 3G wireless links - and in the future it is reasonable to expect this to be true of the majority of hosts.

While many people are attracted to the idea of hosts doing more, and leaving the network to be simple, there are objections to this. I intend to write these up as an ID, but for now they are on a web-page and in RRG discussions. [Host-Responsibilities] In summary, it is highly undesirable for a new architecture to require all hosts to do more routing and addressing management than they currently do: just DNS lookups. The delays which are inherent any such arrangement are highly undesirable and the way these delays are worsened by one or both hosts being on high latency, unreliable, wireless links is particularly objectionable. Also, it is desirable not to enforce extra complexity or communication requirements on all hosts, since many of them will be constrained by battery power limitations.

- Supprimé : requirement
- Supprimé : new
- Supprimé : host
- Supprimé : functionality

2.9. Full ~~Benefits~~ to ~~All Adopters~~

Ivip provides the full benefits of portability, multihoming and TE to all end-user networks which adopt its SPI space.

- Supprimé : benefits
- Supprimé : all
- Supprimé : adopters

In order to do this, packets from hosts in networks which lack ITRs must be forwarded to an ITR and tunneled to the correct ETR.

This is achieved by placing a number of ITRs in the DFZ. These are known as DITRs. When Ivip was first announced [Ivip-2007-06-15] these were named (erroneously): "Anycast ITRs in the DFZ". By placing DITRs widely around the Net, path lengths from any sending host to the ETR are minimised.

LISP Proxy Tunnel Routers (PTRs) perform the same function. [I-D.lewis-lisp-interworking]

For a scalable routing solution to be widely enough adopted, it must provide compelling benefits to all adaptors, including the earliest. Without DITRs, PTRs or their equivalent, only a small fraction of packets being sent to an end-user network would use the new system. Yet the goal is for all adaptors to use the new form of addressing entirely, and so not to have to use the existing unscalable BGP and PI space approach to portability, multihoming and TE.

Commentaire [Med17] : Already mentioned

Supprimé : (Default ITRs in the DFZ) and were previously known as OITRs (Open ITRs in the DFZ)

Supprimé :)

2.10. Business Incentives to Deploy New Infrastructure

Some scalable routing proposals involve no additions to the network - just the adoption of new functionality in the end-user networks which use it. These are generally "Core-Edge Elimination" architectures. [C-E-Sep-Elim]

Supprimé : incentives

Supprimé : deploy

Supprimé : new

Supprimé : infrastructure

No such proposal meets the constraints imposed by the need for widespread voluntary adoption. Firstly, most or all of them involve changes to host stacks and applications, which is impractical in the absence of compelling motivations for the authors of this software to make such major changes. Secondly, all such proposals only provide portability, multihoming and TE benefits for packets sent from other networks which have adopted the scheme. Therefore, only if all networks adopted it would any one network be able to abandon its current routing and addressing arrangements. The benefits of scalable routing in a global sense, and for each adopter, the abandonment of unscalable alternative routing and addressing arrangements, are only achieved after full (or almost full) adoption by all networks. Yet there is insufficient direct incentive for early adopters for even a fraction of networks to adopt it.

Supprimé :

A Core-Edge Separation architecture with DITRs, PTRs or some equivalent functionality provides full benefits to all adopters, and so is capable of being widely enough adopted to solve the routing scaling problem. Scalable routing benefits accrue in direct proportion to the number of adopting networks. The problem can be

substantially solved by widespread adoption. Complete adoption is desirable, but not at all required.

Core-Edge Separation schemes typically do not require any changes to hosts - to stacks or applications. They do however involve the creation of at least two items of global infrastructure, before any end-user network can use the system.

The first is some kind of mapping system, by which ITRs can decide which ITR to tunnel a packet to, when they receive a packet addressed to an SPI address ("EID" in LISP terminology).

The second is one or more sets of DITRs, PTRs or their equivalent.

Ivip's technical structure lends itself to business models in which those who construct and run these two types of infrastructure can do so on a potentially profitable basis, by charging end-user networks according to the use they make of the mapping system and of the DITRs.

2.11. Maintenance of Existing Levels of Security and Robustness

All scalable routing schemes complexify the Internet - so it is unlikely that the goals of not degrading security and robustness to any degree can be fully realized. Only once Ivip is fully designed and carefully analysed can there be a realistic estimation of the security and robustness problems it will entail.

- Supprimé : existing
- Supprimé : levels
- Supprimé : security
- Supprimé : robustness
- Mis en forme : Surlignage

However, it remains a goal to minimise and ideally to eliminate any such degradation.

Ivip's approach to handling the PMTUD problems inherent in encapsulation is intended to be secure against attacks - such as from spoofed ICMP Packet Too Big messages.

Ivip is the only core-edge separation scheme to provide an inexpensive method of ETRs enforcing the source address filtering ISPs may impose on packets arriving at their Border Routers (BRs). Such filtering is imposed to prevent outside attackers spoofing the address of any host inside the ISP's network - and includes dropping packets with private (RFC 1918) source addresses.

- Mis en forme : Surlignage

This is achieved by the simple arrangement of the ITR using the sending host's address as the outer header source address in all the encapsulated packets in the tunnel to the ETR. ETRs simply compare the inner source address with the outer, and drop any de-encapsulated packets where the two differ.

This is only for encapsulation. When MHF is used, there is no need for ETRs to perform any such task, since the original packet is sent across the DFZ, with the sending host's source address in the IP header - so BR filtering occurs normally and the ETR never receives a packet which violates these filtering rules. (With encapsulation, when the ITR occasionally probes the PMTU to an ETR, it sends an additional packet with the source address being that of the ITR, but this does not alter the ETR's ability to enforce BR source address filtering.)

Global query server CES architectures such as LISP-ALT involve multiple devices, often in distant networks and other countries, handling mapping queries which include the ITR's and the destination host's address. While the ITR doesn't necessarily imply that the query came from a given sending host, nonetheless this information, if it fell into the hands of an attacker, would represent a serious security breach. To prevent this, every ALT router would need to be secured at least as well as DFZ routers. This is not necessarily impossible, but Ivip's global mechanism, the fast-push system of Replicators, does not carry queries or anything else which could reveal sending host or receiving network activity. Authentication of mapping changes at full-database query servers means that an attacker who gains control of one or more Replicators will be unable to alter the mapping of ITRs.

3. Non-goals

3.1. Complete ~~Core-Edge Separation Is Not Required~~

At least one ~~Core-Edge Separation~~ scheme - APT (which is no longer being developed) - had the goal of completely separating the global unicast address space into two subsets: core and edge. In this scenario, only ISPs would have core addresses and all end-user networks (or perhaps all end-user networks which needed portability, multihoming and TE) had edge addresses. Then, in theory, it would be possible to prevent any host in an edge network from sending packets to the core - which was supposed to provide some security benefits.

Ivip does not rely for its benefits (improvements to routing scalability, or the benefits for end-user networks) on complete adoption of SPI (edge) space by all end-user networks, or by the subset of them which want or need portability, multihoming and TE.

Ideally, for scalability, the only prefixes advertised in the DFZ would be those of ISPs (including those used to serve many end-user networks with PA space) and the relatively small number of prefixes which encompass the SPI space. "Relatively small" is in comparison to the very large number of micronets these prefixes contain and to the likewise very large numbers of end-user networks which are using this SPI space.

The full benefits for end-user networks which adopt SPI space - portability, multihoming and TE - do not depend at all on how many other end-user networks adopt SPI space.

The benefit of routing scalability depends on how many end-user networks which need or want portability, multihoming and TE actually do adopt SPI, rather than the two undesirable alternatives of either not getting these benefits, or getting them by the unscaleable method of advertising conventional PI prefixes in the DFZ.

In order to maximise routing scalability, the more end-user networks which adopt SPI space, the better. But there is no need or intention to have them all adopt it.

A satisfactory outcome for scalable routing would be for some or many of the end-user networks which currently advertise PI prefixes in the DFZ to continue doing so - and for the great majority of all other end-user networks which want or need portability, multihoming and TE to use SPI space instead.

- Supprimé : core
- Supprimé : edge
- Supprimé : separation
- Supprimé : is
- Supprimé : not
- Supprimé : required
- Supprimé : core
- Supprimé : edge
- Supprimé : separation

3.2. Mapping ~~Changes Need Not Be Free of Financial Cost~~

It appears that the designers of other ~~Core-Edge Separation~~ architectures have a goal of mapping changes being free of financial cost. This is not a goal of Ivip.

Ivip is the only core-edge architecture to contemplate or require mapping changes to be paid for - by the end-user network whose micronet of SPI space the mapping applies to. All other proposals avoid financial costs such as this.

In the case of the global query server systems - LISP-CONS, LISP-ALT and TRRP there is no need for payment, since changing the mapping has no direct impact beyond the authoritative query server(s) in which the mapping is changed. (Unless there are provisions for sending mapping changes to particular ITRs which might need it, which may be a part of LISP.)

APT, like Ivip, involves pushing all mapping changes to a large number of local full-database query servers, via a flooding technique. In APT, these local full database query servers reside in each ISP and are called Default Mappers. There is no mention of charging for mapping changes, but this raises questions about what disincentives there might be to discourage an end-user network from making large numbers of mapping changes, and so burdening the mapping distribution system and all APT-using ISPs with these changes.

Ivip's arrangement for charging end-users for each mapping change, and for each change to the way their SPI space is divided into micronets, is intended to solve a large part of the problem of funding the fast-push mapping distribution system. It is not a complete solution, since it could be argued that there is no incentive for a given ISP's full database query server to keep on receiving and storing mappings for some end-user networks which, for some reason, that ISP's ITRs never handle packets to those end-user networks.

Still, this partial solution enables market forces to set the cost of mapping changes - so the end-user networks effectively pay for the cost of delivering their mapping changes most or all of the way to the full database query servers.

The cost of changes should be low enough to be a trivial issue in the rare events of multihoming service restoration and portability to another ISP. The cost should also be low enough to make reasonably frequent changes for TE attractive, when it allows significantly better utilization of multiple links to ISPs. It should also be low enough to present no problems for TTR Mobility, whenever mapping

Supprimé : changes

Supprimé : need

Supprimé : not

Supprimé : b

Supprimé : free

Supprimé : financial

Supprimé : cost

Supprimé : core

Supprimé : edge

Supprimé : separation

Commentaire [Med18] : Can be dropped. Any new architecture induces extra cost.

Add a ref if any where the no cost is claimed.

Commentaire [Med19] : This will be efficient only if all SPs accept to play this game. In order to gain new customers, new service packages can be offered to end users. This may dilute your argument.

I agree with you that business consideration may be critical if we want to change the current state.

Commentaire [Med20] : This also may have an impact on interconnection agreement.

An additional interconnection layer may be defined above the BGP/Physical one: between SP which offers exchange of mappings.

Mis en forme : Surlignage

changes due to the MN moving more than about 1000km.

Although mapping changes needs to be transmitted reliably to all full database query servers, within a few seconds, via a specially constructed set of servers, each change is reasonably compact, and they are sent en-masse, multiple changes per packet. So the fast-push network will be much more efficient in terms of bandwidth and CPU resources than the nearest current equivalent - changes to DFZ advertisements percolating slowly, neighbour to neighbour, through expensive DFZ routers.

3.3. No ~~Attempt~~ to ~~Cope~~ with ~~Partially Reachable~~ ETRs

Ivip's use of a single ETR address in the mapping is different from the use of multiple ETR addresses in the mapping information of all other core-edge separation schemes. This gives rise to a potential benefit of those other schemes which is not a goal of Ivip.

Ivip ITRs all over the Net tunnel packets which are addressed to any particular micronet to a single ETR at any one time. (This is ignoring perhaps a second or less when the mapping is changed, and some ITRs receive the cache update from their query server earlier than others.) It is up to the multihoming end-user network to ensure that the mapping changes in a manner which maximises the connectivity of its network during a multihoming service restoration event.

For instance, an end-user network has two ISPs ISP-A and ISP-B, and can map its one or more micronets to either ETR-A or ETR-B. Whether the ETRs are in the ISP or at the end-user site is not important. ETR-A's connection to the rest of the Net is via ISP-A and likewise ETR-B's is via ISP-B. In this example, only one micronet is considered, but the same principles apply with multiple micronets.

When both ISPs and ETRs are working well - that is to say both ETRs are reachable from any router in the DFZ and both ETRs are connected to the end-user network - the end-user network may have the mapping set to ETR-A. If an external monitoring company (contracted by the end-user network) detects that the end-user network is no longer reachable via ETR-A, then it will issue a mapping change so that the micronet is mapped to ETR-B instead. As long as ETR-B is connected to the end-user network and is reachable from any router in the DFZ, then this is a perfectly good outcome: full connectivity is restored within a few seconds of the mapping change being issued.

However, if ETR-B is unreachable from some subset of the DFZ routers (and therefore from a subset of sending hosts in end-user and ISP networks) AND this subset of DFZ routers can reach ETR-A (and assuming ETR-A is still connected to the end-user network), then Ivip

- Supprimé : attempt
- Supprimé : cope
- Supprimé : partially
- Supprimé : reachable

Commentaire [Med21] : Depends on the reactivity of the monitoring system ?

Furthermore, if a ETR is not reachable from a location A, it does not mean that it will be unreachable from a location B also. Did I missed something?

cannot ensure complete connectivity, since neither ETR is reachable from all other networks, and since Ivip can only steer packets to one ETR or another. (Actually, practical connectivity only concerns the fraction of DFZ routers and other networks which are currently sending packets to this end-user network - but the ideal is that the end-user network is always reachable from all other networks.)

Other ~~Core-Edge Separation~~ techniques have a potential advantage in this scenario, since it is possible that all the ITRs which are currently sending packets may be able to discern the reachability of the two ETRs and adapt their tunneling by choosing an ETR which is reachable. In this circumstance, the non-Ivip ~~Core-Edge Separation~~ schemes would be able to restore full connectivity when Ivip could not.

However, this set of circumstances - both ETRs being partially ~~reachable~~ and the patterns of reachability being complementary - is likely to be a highly transient state, since the DFZ routers will rapidly adapt their best-paths to restore full connectivity to both ISPs and their ETRs. Also, it cannot be assured or assumed that the non-Ivip ITRs would choose the reachable ETR fast enough to take advantage of this difficult situation.

It is possible that a non-Ivip ITR may be able to detect non-reachability of a particular ETR when the Ivip approach would not. In that case, the non-Ivip approach would be superior due to this particular ITR changing its mapping and so retaining connectivity.

With Ivip, end-user networks will be able to choose between many Multihoming Monitoring (MM) companies and each company would have a range of options for how frequent the reachability probing occurs, how many servers in the DFZ are used to probe the path via each ETR and how decisions should be made if there appears to be a reachability problem. A MM company with probing servers scattered widely around the Net should be able to detect most reachability problems experienced by in any part of the DFZ, but it can't necessarily detect every one. How the MM company decides which outages to respond to, with a mapping change, is a matter for the company and the end-user network to decide.

Ivip's external, user-supplied, detection of reachability problems and creation of mapping changes can be the subject of ongoing innovation and choice, with the intention that it be more effective at restoring full connectivity than the individual, isolated, efforts of non-Ivip ITRs - which have a difficult task reliably and inexpensively testing reachability to various ETRs. This is particularly the case if tens or hundreds of thousands of ITRs are tunneling to one ETR. Such ITRs may not actually probe reachability

- Supprimé : core
- Supprimé : edge
- Supprimé : separation
- Supprimé : core
- Supprimé : edge
- Supprimé : separation
- Supprimé : reachability

of ETRs with ping or the like, but rely on ICMP messages due to traffic packets not reaching the ETR. A difficulty with this is that ICMP messages may be lost, and are not necessarily always generated if there is an outage. Furthermore, it is costly for ITRs to be able to securely distinguish genuine ICMP messages from spoofed packets.

3.4. No ~~Attempt to Avoid All the Mapping Data Being Stored at Any One Location~~

The global query system ~~Core-Edge Separation~~ schemes (LISP-CONS, LISP-ALT and TRRP) seem to be predicated on the belief that it is either impossible, or at least extremely undesirable, to require the full set of mapping data to be sent to or stored at any one location. The attraction of this position is that these architectures are not prevented from handling very large numbers (billions) of EID prefixes (SPI micronets in Ivip terminology) due to any constraints of storage, CPU power or transmission bandwidth.

Since it is clearly feasible to transmit and store numbers of mappings such as 10^6 to 10^8 to single servers or routers, these architectures are presumably intended to scale to billions of EIDs. Of these three, only LISP-ALT is being actively developed. After two years development, questions about its ability to scale to these numbers have not yet been resolved. ~~[LISP-ALT-Critique]~~. In early 2010, version 07 of draft-lear-lisp-nerd claims that NERD (each ITR downloads the full mapping database) will scale to at least 10^8 EIDs. There is no work yet which shows how LISP-ALT could scale to such numbers.

Ivip and APT both incur significant costs and technical challenges moving the mapping changes to large numbers of full-database query servers and storing a copy of the complete mapping database (as it is constantly updated) in each such server. Ivip does not have a goal of avoiding this transmission and storage, of as many as 10^10 mappings, for two reasons:

- Firstly, by the time such adoption occurs, it will not be too difficult to store this data in RAM. (Consumer hard drives today already provide sufficient storage.)
- Secondly, the costs of transmission and storage are judged to be a worthwhile price for the benefits of local full-database query servers: ITRs getting mapping reliably and without significant delays - and so not leading to any lost or significantly delayed packets, which are a problem with the global query server approaches.

So this central goal of some other core-edge separation schemes is not a goal of Ivip.

- Supprimé : attempt
- Supprimé : avoid
- Supprimé : all
- Supprimé : mapping
- Supprimé : data
- Supprimé : being
- Supprimé : stored
- Supprimé : any
- Supprimé : one
- Supprimé : location
- Supprimé : core
- Supprimé : edge
- Supprimé : separation
- Supprimé : .

3.5. It ~~May Not Be Possible To Completely Eliminate "Unfair" Burdens~~

Another motivation for the global query server ~~CES~~ schemes seems to be avoiding all "unfair burdens". Unfair burdens are central to the current routing scaling problem - that any end-user network with PI space can advertise it, as one or more prefixes, in the DFZ and so burden all DFZ routers with the task of maintaining a best path to each such prefix. This is a massive unfair burden problem, since it falls on all ISPs and some large end-user networks. These organisations who suffer the cost, including the need to buy more expensive routers, have no reasonable way of preventing these actions by end-user networks, or of collecting payment from them to offset the very real burdens which are imposed.

Local query server systems (APT and Ivip) do have an "unfair burden" problem in that every network which operates full database query servers needs to pay for bandwidth and storage for every mapping change in the entire system. Yet only a subset of those mapping changes will ever be used in that network, since hosts in that that network are only sending packets to a subset of SPI micronets (EID prefixes in APT terminology).

It is possible to imagine an ISP in Nova Scotia analysing the mapping changes and determining that 10% of them come from some range of SPI addresses, or some country, which the ISP never sends packets to. More generally, for almost any ISP in the world, the great majority of mapping changes are not needed - even if there is no discernable pattern.

Ivip does not have a goal of completely avoiding this particular kind of "unfair burden", for several reasons ~~as shown hereafter:~~

Firstly, the bandwidth used by the incoming mapping changes and the RAM or hard drive space required to store them is expected not to present significant technical or cost difficulties, in the context of hardware development and general traffic volumes. A fully deployed system with up to 10^10 IPv6 micronets would involve considerable storage. Each mapping would consist of 64 bits for the starting /64 prefix of the micronet, up to 64 bits for the length of the micronet, also in units of /64s - and then 128 bits for the ETR address. This is 32 bytes per mapping - a total of 320 gigabytes. By the time such levels of adoption arise, there should be no difficulty fitting this in RAM on a COTS (Commercial Off The Shelf) server.

Secondly, to the extent that this unfair burden is a problem for ISPs, it is a far better alternative to solve the scalable routing problem in this manner than to attempt to solve it with a global query server architecture. Such architectures inevitably delay

- Supprimé : may
- Supprimé : not
- Supprimé : be
- Supprimé : possible
- Supprimé : to
- Supprimé : completely
- Supprimé : eliminate
- Supprimé : unfair
- Supprimé : burdens
- Supprimé : core-edge separation

Supprimé : .

and/or drop a significant proportion of initial packets and so would be difficult or impossible to have adopted widely - and would degrade application responsiveness if they were widely used.

Thirdly, there is scope for market-based solutions to this unfair burden - such as the originators of mapping changes paying all ISPs to accept and use their changes.

Global query server architectures attempt to avoid this unfair burden problem, but create other problems of the same kind - such as difficulties financing the expensive and crucial global query server system in an environment where queries cannot be charged for, and in which mapping changes likewise incur no fee.

Ivip's goal is to minimise unfair burdens and to eliminate them where possible. It appears impossible to devise a scalable routing system which completely avoids unfair burdens, without creating serious technical difficulties or barriers to voluntary adoption, such as the need to rewrite host stacks and applications.

3.6. No ~~Attempt To Mix~~ IPv4 and IPv6

Ivip for IPv4 is intended to be a free-standing system completely independent of Ivip for IPv6. An IPv4 ITR could be implemented in the same server or router as an IPv6 ITR - just as ITR, ETR and query server functions could be performed in the one device.

Likewise, the mapping distribution systems for IPv4 and IPv6 are intended to be separate and independent - but there's nothing to prevent one server being a Replicator for both systems.

3.7. Not Locator - Identifier Separation

There is considerable terminological inexactitude regarding the use of the term "Loc/ID Separation". True Locator - Identifier separation involves hosts handling packets using two addresses of different types, usually called Locator and Identifier, which therefore are in different namespaces. If both types of address are numeric and a Locator and an Identifier were numerically identical they would refer to different things because this numeric value has different meanings in each namespace. (Further discussion of the meaning of "namespace" is at: [Namespace]).

HIP and ILNP [I-D.rja-ilnp-intro] are examples of Locator / Identifier Separation. LISP (Locator/Identifier Separation Protocol), Ivip, APT, TRRP and TIDR are not.

An architecture which uses FQDNs as Identifiers and IP addresses

Supprimé : attempt
Supprimé : t
Supprimé : mix

Commentaire [Med22] : I fully agree.

(always PI, to ensure scalability) as Locators is also an example of true Loc/ID separation - for instance Name-Based Sockets [Vogt-2009].

| LISP, Ivip and other Core-Edge ~~Separation~~ architectures do not present hosts with separate Locator and Identifier addresses. The host sees only IP addresses, which perform both functions simultaneously - just as they do without ~~CES~~. ITRs are the only devices which treat packets differently if their destination address is in the "edge" subset of the global unicast address range.

Supprimé : separation

| The full arguments about why ~~CES~~ cannot correctly be construed as "Locator / Identifier Separation" are at:
[loc-id-sep-vs-ces]

Supprimé : core-edge separation

Supprimé : Core-Edge Separation

4. Architectural Choices

4.1. Core-Edge Separation Rather Than Elimination

4.1.1. Core-Edge Elimination (CEE) Architectures

Core-Edge Elimination (CEE) involves hosts dealing with two kinds of address: Identifiers and Locators. Other terminology may be used, such as "host identifier" or "logical address" instead of "Identifier" and "physical address" instead of "Locator". The simplest adaptation of existing protocols is to retain IP addresses as Locator addresses and develop a separate namespace for the Identifier addresses.

Each host retains its one or more Identifiers, no matter which one or more Locator addresses it is using.

The Locator addresses are global unicast addresses which are supplied by ISPs as PI space. The simplest form of multihomed end-user network would gain a PI prefix from each of its ISPs and each of its hosts would use one address from each prefix as a Locator address. Each such prefix is part of a larger (in terms of number of addresses - shorter in terms of prefix length) prefix the ISP advertises in the DFZ. The ISP can split one such advertised prefix into many smaller (longer) prefixes for multiple end-user networks. This solves the routing scaling problem because the total number of large (short) prefixes advertised by all ISPs is scalable, whereas - if not for the core-edge elimination scheme - the number of prefixes advertised in the DFZ by multihomed end-user networks would be an unacceptable burden on all DFZ routers and on the entire DFZ BGP control plane.

Applications connect to other hosts solely in terms of their Identifier addresses. It is the task of each host's stack to adapt to changes in other hosts' Locators, and to inform other hosts which need to know about this host's changed Locators. The Identifier addresses may be numeric or have some other form, and there is typically a DNS mapping from FQDNs to one or more Identifier addresses, just as there are to IP addresses today

Some key points about Core-Edge Elimination architectures include:

- 1. Identifiers are from a completely different namespace than Locators. If both are numeric, and a Locator is numerically equal to an Identifier, there can be no confusion about the separate entities each refers to, since the Identifier is interpreted in a different namespace from that used for Locators. Therefore, if IP addresses are used as Locators, IP

- Supprimé : edge
- Supprimé : separation
- Supprimé : rather
- Supprimé : than
- Supprimé : elimination
- Supprimé : architectures
- Supprimé : edge
- Supprimé : elimination

- Commentaire [Med23] : Whata does that mean?
- Mis en forme : Surlignage

- Supprimé : core
- Supprimé : edge
- Supprimé : elimination
- Supprimé : ¶

addresses cannot be used as Identifiers.

- 2. Host stacks are responsible for choosing which of a correspondent host's Locators to send a packet to. This work is not done by network elements, such as routers. (However some CEE architectures may have routers alter part or all of the outgoing destination address, or perhaps source address, to exert-network centric control over traffic flows.)
- 3. While there is typically a global, decentralised mapping system by which hosts can use another host's Identifier (perhaps in combination with one of its Locators) to look up that host's complete set of one or more Locators, the network itself remains simple and hosts take on more responsibilities than they have with existing IP protocols. This is regarded as a virtue by many people, and represents an extension of TCP/IPs "dumb network, smart end-points" approach, especially when compared to the telephone network.
- 4. Since applications need to work with a different kind of address element than an IP address for establishing and maintaining communications with other hosts, the host stack, its API and applications themselves need to be substantially rewritten in order to be able to work with a CEE architecture.
- 5. While it may be possible to slowly introduce such an architecture, the benefits of portability, multihoming and TE only apply to packets sent between hosts using the new system.
- 6. CEE architectures are subject to the critique that the extra management packets which hosts must send and receive as part of the new system is likely to create extra costs, delays and/or unreliability compared to current IP techniques.
- 7. This critique can be extended to argue that mobile hosts, due to their typically slow, not-necessarily reliable and potentially costly wireless links are especially impacted by these new responsibilities.
- 8. CEE schemes typically do not apply to IPv4 and so are based on IPv6 or on entirely new arrangements.

Points 4 and 5 constitute insurmountable barriers to the adoption of CEE architectures, since adoption must be very widespread, within a period of years, rather than decades, and since adoption must occur on a voluntary basis. [Constraints-Voluntary]

Point 6 is an argument that while CEE architectures are theoretically

Commentaire [Med24] : This won't be true anymore in the future.

Supprimé : Core-edge elimination

Commentaire [Med25] : Which is coherent in the sense that IPv4 addresses will be exhausted soon. The proposed solutions if adopted are likely to be deployed after IPv6 is in the network I guess because IP address depletion is one of our hot topic. Routing scalability is not in the same level.

elegant and simple, the facts of delay and loss of packets across global query server systems such as DNS, or whatever mapping system is used to securely determine a host's full set of Locators will contribute to delays in sending application packets. Also, if the two hosts have to exchange management packets with each other, for authentication purposes, before any application packets can be sent, then this will slow down the establishment of communications - especially if the hosts are far apart, on high latency links or if packets are lost.

Point 7 implies that in order to create a network which performs best, given the vagaries of slow and unreliable last-mile links, all hosts should not have to perform these additional Routing and Addressing management functions - that such functions be handled by better-connected devices, such as routers in ISPs' data-centers. [Host-Responsibilities]

The only existing routing scaling problem is in the IPv4 Internet. In early 2010 the IPv4 DFZ has about 300k prefixes with a doubling time of about 4.5 years. The IPv6 DFZ has about 855 prefixes - 1/350th the IPv4 number. Even if IPv6 prefix numbers had a doubling time of 1.0 years, it would be mid 2018 before the number reached current IPv4 levels - which are not yet unworkable. IPv6 adoption rates have consistently disappointed IETF expectations. Despite the run-out of unallocated IPv4 space, there is no sign yet that large numbers of existing users can have their Internet needs adequately served via IPv6 addresses alone.

For the reasons described in points 4 to 8, Ivip instead adopts a ~~CES~~ approach.

4.1.2. Core-Edge Separation (CES) ~~Architectures~~

Ivip uses a ~~CES~~ ~~architecture~~. CES does not involve the creation of new namespaces and does not require any changes to host stacks or applications.

A subset of the global unicast address space is converted to a new type of address which, in Ivip, is known as Scalable PI (SPI) space. This subset will consist of a growing number of prefixes, each of which is advertised in the DFZ. Within each such prefix, the SPI space can be divided up amongst many (thousands to potentially millions) of separate end-user networks. If a network gains more than one basic unit of address space ~~(e.g., an IPv4 address or an IPv6 /64 prefix)~~ ~~it can divide this space into multiple separately mapped "micronets".~~

As more and more space is converted for use as SPI space, this "edge"

Supprimé : core-edge separation
Supprimé : architectures
Supprimé : Core-Edge Separation (
Supprimé :)
Supprimé : Architecture

Supprimé : -
Supprimé : -

space grows to become a significant fraction of the total global unicast space. There must always be some conventional, non-SPI, space, since ETRs must be located on such addresses. There are many uses of space within ISPs which do not need to be on SPI space - including the large numbers of IPv4 addresses, or in the future IPv6 /64s, which are used for individual home and SOHO customers. Each such customer gets what is effectively a small (long) prefix of PI space, which is suitable for their purposes because they do not want or need portability, multihoming or TE.

Commentaire [Med26] : Home networks are assigned bigger prefixes /56.

As noted in the non-goals section, Ivip does not require or aim for complete conversion of all end-user networks to SPI space. Many will be happy with existing PI arrangements, and some larger existing end-user networks with their own (unscalable) PA prefixes will probably retain their current arrangements. Nonetheless, SPI space is intended to be attractive to all end-user networks, including the largest corporations, universities and government departments.

CES involves the progressive repurposing of existing address space. It does not involve the creation of any separate namespaces. If "separation" in "Locator/Identifier separation" means separate namespaces, then this term only properly applies to CEE schemes - which do exactly this.

Supprimé : core-edge elimination

CES can be introduced gradually, and with DITRs (or their LISP equivalent - PTRs) the benefits of portability, multihoming and TE can be supported for all packets sent to the adopting end-user network. Therefore 100% of traffic receives these benefits, in contrast to CEE architectures where only the subset of traffic originating from other upgraded networks has these benefits.

Assuming a CES architecture does not significantly reduce performance, robustness or security - and if it provides significant and immediate benefits to all adopters - then it meets the constraints due to the need for widespread voluntary adoption. [Constraints-Voluntary]

All CES architectures I am aware of do not require hosts to perform additional work to manage routing and addressing. So no CES architecture is subject to the critique which applied to CEE architectures, particularly with reference to mobile hosts. [Host-Responsibilities]

The historical roots of Core-Edge Separation architectures can be found in the mid-1990s - Steve Deering's "Map & Encap" for IPv4 [Deering-1996], Robert Hinden's "New Scheme for Internet Routing and Addressing (ENCAPS) for IPNG" (RFC 1955) and the 1992 crocker-ip-encaps-01.txt.

4.2. Local Query Servers

Probably the greatest challenge for a CES scheme is how to ensure ITRs can securely, reliably and rapidly obtain the mapping they need in order to be able to decide which ETR to tunnel a packet to. There are three basic approaches to this problem:

- 1. The complete global set of mapping changes is sent to each ITR, which maintains an up-to-date copy of the full mapping database.
- 2. Local full-database query servers are located in ISP networks and potentially in end-user networks in which ITRs are based. The complete global set of mapping changes is sent to each such query server, which maintains an up-to-date copy of the full mapping database. ITRs query one or more of these and so obtain mapping quickly and reliably.
- 3. No site or device stores a complete copy of the global mapping database. Instead, there is a global network by which ITRs can send query to the authoritative query server for the particular micronet of addresses which match the destination address of the packet the ITR needs to tunnel.

Supprimé : query

Supprimé : servers

Supprimé : core-edge separation

Supprimé : ¶

Commentaire [Med27] : This option may have synchronisation issues

Commentaire [Med28] : This approach may be compared to creating an overlay to maintain the mapping. A single overlay to exchange those mapping must be created otherwise the internet will be fragmented.

The only architecture to propose Option 1 is LISP-NERD. This is widely regarded as scaling poorly with large numbers of end-user networks. LISP-NERD was to be retired, but a new version 07 ID appeared in early January 2010. [I-D.lear-lisp-nerd].

Supprimé : option

Supprimé : was

Supprimé : .

Ivip and APT use Option 2. In Ivip, the local full database query servers are called QSDs. In APT, they are called Default Mappers, and also handle the encapsulation of some packets.

Supprimé : option

All other CES schemes to date use Option 3. The most prominent examples are LISP-CONS [I-D.meyer-lisp-cons], LISP-ALT and TRRP.

Supprimé : core-edge separation

Supprimé : option

The global query server network approach has obvious advantages in terms of there being no hardware-imposed limit to the number of query servers or end-user networks which can be supported. Furthermore, changes to mapping impose no direct burden on any other devices - whereas for Option 1 or 2, information must be sent to potentially hundreds of thousands of devices all around the world.

Supprimé : option

However, global query server systems pose apparently insoluble problems of delay and potential unreliability - due the delays and

risk of packet losses which are inherent in their global nature. Furthermore it seems to be impossible to make these systems scale to the very large numbers of EIDs required for ubiquitous mobile adoption. ~~[LISP-ALT-Critique]~~

Supprimé : .

Supprimé :

Local full-database query servers are the clear choice for Ivip. Even the largest imaginable mapping databases and the highest rates of change due to TTR mobility should not present insurmountable problems in terms of hardware, software and communications bandwidth, by the time such large numbers (10^9 to 10^10) of end-user networks are using the system.

4.3. Real-Time Mapping Distribution

Supprimé : t

Supprimé : mapping

Supprimé : distribution

If the mapping is distributed slowly to the QSDs (Query Server Databases) - or for any other reason, can't be sent to all ITRs which need it within a few seconds - then ITRs need to make their own decisions about which ETR to use. Therefore, the mapping needs to include multiple ETR addresses, and ITRs need to be much more complex to probe reachability (and/or somehow detect non-reachability with encapsulated traffic packets) and make the best choice between multiple ETRs.

By getting mapping changes to ITRs in real-time - within a few seconds at most - Ivip achieves several major benefits. Firstly, the mapping information can be more compact, since only a single ETR address is needed. Secondly, ITRs can be much less complex, and do not need to do any reachability testing. Thirdly, the real time control of all ITRs which is given to end-user networks modularly separates the reachability, multihoming service restoration and TE decision making systems from the ~~CES~~ scheme itself.

Supprimé : core-edge separation

Mapping needs to be delivered in a secure and robust manner, which presents some challenges in an essentially real-time global system. There may be multiple ways of achieving this, but the currently planned system for pushing all mapping changes to QSDs looks perfectly feasible.

The second stage of delivering mapping to ITRs is that QSDs are able to securely send mapping updates to all their client ITRs which need it. This would occur a small fraction of a second after the QSD received a mapping update from the global fast-push system.

4.4. SPI Address Management

Supprimé : address

Supprimé : management

Traditional IP techniques divide address space into binary boundary prefixes. Ivip uses traditional prefixes for the largest unit of SPI space - the "Mapped Address Block" (MAB). The smaller divisions of this do not use prefixes or binary boundaries. The units of dividing

SPI space are IPv4 addresses and IPv6 /64 prefixes.

A MAB is a prefix of address space which is devoted to use as SPI space. The single MAB is advertised in the DFZ, by all the DITRs which are located around the Net, attracting packets addressed to any address in the MAB. (It would also be possible to load share the MAB between multiple DITRs, each advertising a segment of it, but in general complete MABs will be advertised.) For instance, an IPv4 MAB may be 11.22.0.0/16.

A MAB might have previously been conventional PI space of an end-user network, and may now be used exclusively by this end-user network. In this case, it will presumably be used to serve the needs of many sites within this network, so achieving routing scaling by removing the need to advertise each such smaller (longer) prefix in the DFZ.

Most MABs will be operated by specialised companies - perhaps ISPs but not necessarily a company which actually provides Internet connectivity. The MAB operating company typically acquires rights to multiple prefixes of global unicast space, advertises each of them in a global system of DITRs and then rents out smaller portions of the MABs, on a yearly basis, to a large number of end-user networks.

Each end-user network rents a section of the MAB - a User Address Block (UAB). One end-user network might rent multiple non-contiguous UABs in the one MAB, and multiple UABs in multiple MABs. For simplicity, the following discussion assume they rent a single UAB, such as: 11.22.33.84 to 11.22.33.95 inclusive. This is an 18 IP address UAB. UABs could be as small as a single IPv4 address or IPv6 /64 or could be very large, including as large as the MAB itself.

The end-user network which rents this UAB is responsible for generating mapping changes to suit its needs - and for multihoming would typically hire a Multihoming Monitoring company and give them the credentials required to control the mapping.

Supprimé : (MM)

The end-user network can split their UAB up as they wish into typically smaller sections, known as "micronets". (Bill Herrin first used this term in TRRP.) A micronet is a contiguous set of any positive number of IPv4 addresses or IPv6 /64s which fit within the one UAB. This 18 IP address UAB could be used as a single 18 IP address micronet, or it could be split in any way - such as into as many as 18 single IP address micronets.

Each micronet is covered by a single Ivip mapping - it is mapped to a single ETR address.

MABs and micronets are important to ITRs and most of the mapping

system. UABs are not needed for these, but are an administrative construct of SPI space which an end-user network is authorised to change the mapping for.

The company which runs the MAB would provide a method by which the end-user network, or some other company it authorises, can change the mapping and the division of the UAB into micronets quickly and securely. This would involve the end-user network having complete control, but being able to give a username and password to another party such as the MM (Multihoming Monitoring) company, by the MM company could control the mapping of some or all of the end-user's UAB space. The technical and administrative arrangements for this are described in [I-D.whittle-ivip-db-fast-push].

For each mapping change and each change to the division of the UAB into micronets, the end-user network would incur a fee from the MAB company.

The MAB company would charge fees for renting the UAB space, and for the load placed on the DITRs which cover this MAB. The MAB company may run its own DITRs, or may contract this out to another company which specialises in this service. As long as there are significant numbers of hosts in networks without ITRs, it will be an important part of the MAB company's service to locate DITRs in all corners of the Net, to ensure good load sharing between them and to minimise the total path length from the sending host to whichever ETR the end-user network chooses to map their Micronet to.

This flexible integer-based approach to dividing SPI space is intended to maximise the efficiency with which it is can be used. Since a single physical site, such as a branch office, may be able to operate perfectly well on one or a few IPv4 addresses, or on a single IPv6 /64, a seemingly small UAB of 18 IPv4 addresses could be used to serve the needs of as many branch offices. Each such site could be multihomed with two or more local ISPs.

As fresh expanses of IPv4 space disappear, there will be continuing pressure to slice and dice the address space more finely so it can be used by more and more ISPs and end-user networks. However, the convention in the DFZ is not to propagate prefixes longer than /24. This 256 IP address granularity inherent in the current arrangement leads to considerable underutilization of space. With SPI address able to be sliced and diced freely in the smallest possible increments, a much greater utilization can be expected, in a scalable fashion, than is possible with current techniques.

4.5. ~~IP-in-IP Encapsulation~~

When encapsulation is used, there is a simple IP-in-IP header. There is no need for ITRs to communicate with ETRs, except for the purpose of PMTUD management. So, when the ITR tunnels traffic packets ordinarily (in all cases except for the special Path MTU measurement protocol, which is only used rarely) there is no need for a UDP header to enclose a special header with extra information. Architectures with slow mapping distribution and which therefore require ITRs to choose between multiple ETRs typically require the ITRs and ETRs to communicate - but this is not needed for Ivip due to its design choice elaborated above.

- Supprimé :** IP
- Supprimé :** in
- Supprimé :** encapsulation

4.6. MHF ~~Initially or in the Long Term to Avoid Encapsulation and PMTUD Problems~~

Both the IPv4 and IPv6 headers have un-used bits which can be employed to direct the packet from ITR to ETR. This path is primarily across the DFZ but typically includes routers inside ISP and end-user networks. These routers need to be upgraded - and in the long-term this can be done without significant cost, simply by building the new capabilities into new routers and implementing it in firmware updates.

- Supprimé :** initially
- Supprimé :** long
- Supprimé :** term
- Supprimé :** avoid
- Supprimé :** encapsulation
- Supprimé :** problems

4.7. Outer ~~Header Address is That of the Sending Host~~

When encapsulation is used, it seems natural to use the ITR's address as the outer header's source address. This is consistent with traditional tunneling, and ensures the ITR gets any ICMP messages, including especially Packet Too Big (PTB) messages.

There are two problems with this conventional approach, which is used by LISP and other CES architectures. Firstly, it is very expensive for the ITR to securely respond to PTB messages. Secondly, this approach means that any ISP BR filtering (dropping) of incoming packets according to their source address will not affect the packets at the BRs and must be replicated in the ETR. For more than a few such blocked prefixes, this is extremely expensive too - and we want ETRs to be as simple as possible.

The answer is to have the ITR use the sending host's source address in the outer header of the encapsulated packet. All ITRs will therefore generate packets with identical inner and outer source addresses. ISP BR filtering will drop the packets with source addresses matching any prefix inside the ISP's network and the ETR will never need to handle such packets.

The ETR needs to enforce this in the case where an attacker sends a packet to the ETR, with an inner packet having a banned source

- Supprimé :** header
- Supprimé :** address
- Supprimé :** that
- Supprimé :** sending
- Supprimé :** host

address and the outer header having a source address which is allowable. This enforcement is achieved by the ETR performing simple logic on each de-encapsulated packet: If its source address does not match the outer header's source address, the packet is dropped.

This arrangement of the outer source address being that of the sending host requires a novel approach to Path MTU Discovery management.

4.8. IPTM (ITR Probes Tunnel MTU) PMTUD Management

Supprimé : management

As long as encapsulation is used, there needs to be a method of informing sending hosts, via traditional RFC 1191 techniques of what length packet to send, so that once encapsulated, these packets may reach, but not exceed the MTU of the path between the ITR and ETR. This is true of any core-edge separation architecture which uses encapsulation. It is a complex topic and there is a solution, but it requires considerable thought and significant complexity in all ITR and ETR.

PMTUD management occurs naturally via RFC 1191 mechanisms for DF=1 traffic packets if the router with the too-small MTU is between the sending host and the ITR, or between the ETR and the destination host. Without encapsulation - with MHF - packet lengths are not increased in the ITR to ETR "tunnel", and the modified routers in this path will convert a too-long packet back to its original IP header format, before passing it to the ICMP PTB algorithm.

The difficult task is to make PMTUD work for the path between the ITR and ETR, where the original packet is encapsulated. I intend to write up IPTM in an ID. For now, the fullest description is on a web page. [PMTUD-Frag] Here is an overview of the process, which is much the same for IPv4 and IPv6.

This system involves restrictions on the length of IPv4 DF=0 (fragmentable) packets which are accepted by this system. It is reasonable to expect applications not to generate such packets, which place a serious burden on the network of they are too long. Google servers have been observed sending 1470 byte DF=0 packets. [DFZ-unfrag-1470] Such companies could presumably be persuaded to refrain from sending DF=0 packets altogether by the time a scalable routing solution is deployed. In the long-term, with EAF in place of encapsulation for IPv4, fragmentable packets addressed to SPI addresses will be dropped by all ITRs.

A simple approach to PMTUD management would be to choose some packet length, marginally below 1500 bytes and require all ITRs to accept only packets which are the encapsulation overhead number of bytes

shorter than this. Longer packets would cause the ITR to generate a PTB and the sending host would send a suitably shortened packet instead. This would be simple and perform reasonably well in today's DFZ, where the Path MTU can reasonably be assumed to be 1460 bytes or more.

However, such a scheme would fail to take advantage of jumboframe sized MTUs whenever they appear in the DFZ. ITR to ETR MTUs of around 9k bytes are likely to become more and more prevalent as more routers adopt Gigabit Ethernet interfaces, which handle these large packets.

The encapsulated packet has the sending host's source address. If such a packet reached a router with a next hop MTU which was longer than the packet, the router would transmit a PTB to the sending host. However, the sending host should ignore it, since the destination address in the enclosed packet headers will be that of the ETR, not of the destination host - and the rest of the enclosed headers will not match the packet it sent. Also, the MTU figure in the PTB is higher than the figure the sending host needs to adhere to.

So the challenge is for the ITR to generate RFC 1191 PTBs when necessary, in an inexpensive and secure manner, whilst adapting to potentially higher or lower MTUs to the ETR due to routing path changes - while making full use of jumboframe paths if and when they exist. Security in this case means being immune to spoofed PTBs - a single one of which could greatly reduce the MTU for all traffic from the ITR to a given ETR for at least ten minutes.

A careful decision will be made to assign a value such as 1200 bytes to a globally agreed constant MPMTU (Minimum Path MTU). Once set, this value must remain agreed to indefinitely. A BCP would require all DFZ routers, and all routers between the DFZ and any ITR or ETR (and of course the links between these) to handle packets of this length.

Any packets, which once encapsulated and so ENCAPS (Encapsulation overhead - 20 bytes for IPv4 and 40 for IPv6) bytes longer, have lengths less than or equal to MPTU are encapsulated without any extra processing. No PMTUD problems exist for these packets.

For any packet longer than this, assuming the ITR has not yet probed the PMTU to its ETR, the ITR performs some special processing. The packet itself is split into two sections and two packets are sent to the ETR as part of the ITR's attempt to probe the MTU to this ETR. One packet uses UDP encapsulation to convey a nonce, some flags and most of the traffic packet - with the ITR's address in the outer header's source address. This long packet is exactly the same length

as the original packet would be once encapsulated.

If this exceeds the PMTU to the ETR, then the ITR will be sent a PTB. Assuming this is received, the ITR will determine a new MTU to send in the PTB to the sending host. This process will repeat until the sending host's packets, once encapsulated, no longer exceed the MTU of the path to the ETR.

IPTM does not rely on these PTBs. The ETR is instructed, in a shorter packet to report to the ITR whether the long packet arrives or not - and the ETR repeats this report for a while until it is acknowledged. The long packet is accompanied by one or more copies of this shorter packet, which contains a matching nonce, flags and the remainder of the traffic packet. The shorter packet has the sending host's address in the outer header, so ISP BR source address filtering is still enforced.

The effect is that as the sending host (or multiple sending hosts whose packets must be tunneled to the one ETR) tries longer and longer packets, the ITR narrows its "zone of uncertainty" (cue Hammond organ, with reverb and ghostly sounds . . .) about the true MTU to this ETR. If the traffic packets necessitate it, the ITR will exactly determine the MTU, and so be able to stop probing it for a while and send PTBs to sending hosts which generate packets which, once encapsulated, would be longer than this reliably determined MTU. Further elaborations are required for the ITR to adapt to changing conditions and discover longer or shorter MTUs.

Without some kind of PMTUD system, core-edge separation architectures cannot use encapsulation. These techniques will require further design work and extensive testing, but are more secure and less expensive than the only other obvious alternative - using the ITR's address in outer headers and having the ITR maintain a large cache of details about recently sent "long" packets, in order that it can securely accept PTBs if they are too long.

5. Architectural Elements

5.1. ITRs

5.1.1. Types of ITR and their addresses

The ITR function can be implemented in a traditional hardware-based router, in a COTS (Commercial Off The Shelf) server, or as a piece of software in a sending host/node. The functions are much the same, but an ITR in a sending host does not advertise anything in a routing system - it simply handles outgoing packets which are addressed to any MAB.

Commentaire [Med29] : Avoid host since it is claimed to not be impacted by the proposed solution.

If an ITR is built with software and a COTS server, it doesn't need to be a "router" in most ordinary respects. For instance it doesn't need multiple interfaces. It may have a single Gigabit Ethernet link and advertise MABs in the local routing system, forwarding its encapsulated packets to a router to be forwarded like any other packet.

An ITR could be built into a DSL, HFC cable, fibre or WiMax / 3G router. However, it is probably best to do this only when the ITR function is on a reliable, fast, inexpensive link. Most wireless links are not like this and it would be better to let SPI packets flow out of the link, and be handled by ITRs in the ISP network, which have fast reliable paths to local query servers.

An ordinary ITR (not in a sending host, and not a DITR in the DFZ) is a device within an ISP or end-user network which attracts packets addressed to SPI addresses. It may do this by advertising every MAB - so the only packets forwarded to it, other than those addressed to the DITR itself, are those addressed to SPI addresses.

Alternatively, if the ITR is a router it may advertise the entire address space and so be forwarded all packets not addressed to prefixes advertised by local routers. Then, it would encapsulate packets which are addressed to SPI addresses and forward all other packets according to its ordinary router functions.

Supprimé : true

Supprimé : (hardware or software)

The ITR's address - the address it uses for tunneling packets from, and which is used for communication with the ETR for PMTUD management - may be on conventional global unicast space or, if in an end-user network, on SPI space. This address is also used for communication with local query servers and for receiving PTB messages.

Here is a description of what happens when a sending host in an ISP network, such as a QSD, on the ISP's conventional address space, sends a packet to a host in an end-user network on an SPI address - in this case an ITR or ITFH. The packet will go to an ITR in the ISP network (if the QSD doesn't have an ITFH installed already) and then

will be tunneled to the ETR for this end-user network. This ETR sends the packet to the SPI-addressed host, in this case an ITR or ITFH.

When MHF is used, there is no PMTUD management, no interaction with ETRs and no trace of the ITR's address in any outgoing packets. However, the ITR still needs an address for communicating with local query servers.

5.1.2. ~~Default ITRs in the DFZ (DITRs)~~

Supprimé : DITRs -

DITRs ~~stands for~~ Default ITRs in the DFZ. This first use of "Default" is different from the use of "Default" in "Default Free Zone". This term looks nonsensical when expanded fully.

Supprimé : are

Supprimé : "

Supprimé : "

The initial "Default" means that this ITR acts as one of (typically) many other such ITRs, all of them outside ISP and end-user networks. These DITRs advertise MABs from many places in the DFZ and so form multiple destinations which are the "default" - what happens to the packet if nothing else happens (it does not go into any other ITR before reaching the DFZ).

In principle, a DITR could advertise every MAB, or be an otherwise normal DFZ router and encapsulate every packet which is forwarded to it which is addressed to an SPI address. However, there is a burden of work looking up mapping, encapsulating packets and on occasions handling the PMTUD management functions to ETRs, which involves sending PTBs to sending hosts. It is unlikely that anyone running a DFZ router would want their device to do more work, unless they are paid for it by the beneficiaries. The beneficiaries of DITRs are the end-user networks which the packets are addressed to.

The most likely arrangement for DITRs is that the MAB companies who charge end-user networks rent for their SPI space will also run DITRs themselves or contract specialised companies to run DITRs all over the Net for them. In this scenario, a DITR would advertise only those MABs of the MAB companies who are paying the operator for this service. MAB companies would charge their SPI-renting end-user network companies for the traffic handled for their networks by DITRs, so DITRs in general would need to sample traffic reliably and generate reports in a form which would enable the MAB companies to bill their customers fairly. Only DITRs need this traffic sampling capability. Other ITRs would have monitoring and management functions, but would not need to collect usage statistics for billing.

Theoretically, DITRs could advertise all MABs and so handle packets addressed to every MAB. In practice, I expect DITRs will usually

only handle packets addressed to specific MABs. Other ITRs, including those in sending hosts, will handle packets addressed to any MAB. Consequently, these non DITR ITRs all need a reliable method of downloading the latest set of MABs. They will do this as part of discovering and communicating with their one or more local query servers.

DITRs may be implemented in hardware based routers, or in COTS servers. They are always located on conventional global unicast addresses - never on SPI addresses. DITRs are likely to be busy, so it makes sense to locate them in major datacenters, close to one or more full database query servers.

5.1.3. Modified Header Forwarding - MHF-only ITRs

Ivip for IPv4 and for IPv6 separately may or may not begin with encapsulation. If it does, then all ITRs and ETRs will also be capable of transitioning in the future to using MHF.

The MHF techniques are discussed in a later section, but involve much less processing than encapsulation. With MHF, there is no need for PMTUD management.

5.1.4. Encapsulation and PMTUD Management

Supprimé : management

When the ITR function is implemented in software - either inside a sending host, or in a COTS server, it will be relatively straightforward to write C code or the like to implement the functions of analysing the packet's destination address, deciding whether to encapsulate it or not, deciding which ETR address to encapsulate it to, and encapsulating it. Once encapsulated, the new packet is presented to the internal packet handling functions and forwarded normally.

This packet-handling code also needs to consider the length of the packet, with reference to a small set of variables it maintains for the ETR the packet will tunneled to. So the packet's destination address would firstly be used to find an ETR address. Generally, this would be found by reference to the ITR's cached mappings, but for initial packets in a new communication flow, the packet must be held for a few milliseconds or tens of milliseconds while the ITR retrieves the mapping information from its one or more local query servers.

Once the destination ETR address is known, the length of the packet is considered. If it is less than some constant, it can be encapsulated and sent without any further processing. If it is longer than this constant, then the ITR needs to perform PMTUD

management functions. In this case, the ITR establishes, or has already established, some variables for this ETR. These include an upper and a lower estimate of the MTU to this ETR. If these are different, then there is a "zone of uncertainty" about the MTU. If they are equal, then the ITR has already reliably established the MTU. If the packet length, plus the encapsulation overhead, exceeds the range of possible MTU values the ITR has previously determined for the path to this ETR, then the ITR will send part of the packet back to the sending host in an ICMP PTB message. If the encapsulated length would be less than the lower limit in the "zone of uncertainty" then the packet can be encapsulated without further processing.

If the encapsulated length falls within the "zone of uncertainty", then the ITR emits two packets - a long one and a short one - and communicates with the ETR in a way which will usually raise the lower limit of this zone, or lower the upper limit. In the former case, the ITR is able to determine that the encapsulated length did not exceed the MTU and that the ETR received it correctly. The traffic packet's contents are mainly contained in the long packet, which has the same length as the traffic packet would have had if encapsulated. The remainder of the traffic packet is conveyed in a short packet, of which perhaps a few will be sent. This is non-trivial process, which involves the ETR in some work - but it only occurs for packets whose encapsulated length falls within the "zone of uncertainty".

Except for rare error conditions, each such operation reduces the size of the "zone of uncertainty" - and typically the zone will be reduced to zero. Once this occurs, at least for the next 10 minutes or so, the ITR need not perform any such probing of the MTU. Every encapsulated packet which is to be sent to this ETR will be either shorter than the MTU, in which case it is encapsulated without any further work - or is longer, in which case a PTB is sent back to the sending host, with an MTU value such that the host will generate packets to this destination host of a length which, when encapsulated, will equal this reliably determined MTU.

This encapsulation and some kind of PMTUD management is required for any CES architecture which uses encapsulation. All other CES architectures use encapsulation exclusively. There is at least one other approach to PMTUD management which is probably more expensive to perform as securely as this one. The fact that this and other processes are explained in some detail in this Ivip ID and not in the IDs of other proposals does not mean that the other proposals, once developed to the point of proper operation, would be simpler than Ivip.

The encapsulation itself is straightforward. The sending host's

address is used for the outer source address and the ETR's is used for the outer destination address. For IPv4 packets, the Diffserv, TTL and other flags are copied to the outer header. For IPv6, Traffic Class and Hop Limit bits are also copied.

5.1.5. Mapping ~~Lookup~~ and ~~Caching~~

Apart from PMTUD management, looking up the mapping for an incoming packet is the most complex task that ITRs need to perform. This task is the same for encapsulation in both IPv4 ~~and IPv6~~ and for the IPv4 approach to MHF: ETR Address Formatting (EAF). For the IPv6 MHF technique - Prefix Label Forwarding (PLF) - the mapping lookup is similar, but only part of the ETR's address is actually needed for writing 19 or 20 bits into the header.

Supprimé : lookup
Supprimé : caching

Supprimé : and
Supprimé : IPv4

When encapsulation is used, for IPv4 or IPv6, the result of the ~~mapping lookup~~ is an IP address of the ETR, which will become the destination address of the outer header. The result of EAF is similar, and ETR address where the two least significant bits are zero. This will be written into the modified IPv4 header.

Commentaire [Med30] : An Address Family is indicated in the query?

The result of the PLF mapping will be a 19 or 20 bit value is written into the modified IPv6 header and which identifies one of 2^19 or one of 2^20 contiguous DFZ advertised prefixes, each of which is advertised by a different ISP site. These 20 bits do not uniquely identify an ETR if there are more than one at each ISP site, but they are sufficient for the packet to be forwarded across the DFZ to the nearest BR of that site, where a second mapping lookup may be performed on the destination address to determine which of multiple ETRs at that site the packet should be forwarded to. Please refer to the description and example of PLF operation at the end of ~~[Ivip-Summary-and-~~

Analysis].

Supprimé : Ivip-¶ summary.pdf.
Supprimé : Ivip
Supprimé : Summary
Supprimé : and

This following may appear somewhat complex, but it is a description of different approaches to handling ITR to ETR tunneling for both the IPv4 and IPv6 Internets. Ideally, encapsulation won't be necessary to at all. At worst, it will be necessary until DFZ and other routers are upgraded to handle EAF or PLF modified header packets.

The mapping lookup is driven entirely by the packet's destination address. Ivip does not attempt to send packets of differing types, service class or differing source address to different ETRs. (Nor do the other core-edge separation schemes.)

After a packet arrives, and has been classified as being addressed to an SPI address (meaning it matches one of the MAB prefixes) the next step is to find out whether the ITR has any mapping cached for the packet's destination address. For IPv4 the full destination address

is used. For IPv6, only the most significant 64 bits are used, since SPI space is divided on /64 boundaries.

Busy ITRs may have tens or perhaps hundreds of thousands of mappings already cached. An ITR function in a sending host may have only a handful or a few hundred. A carefully designed algorithm will be needed to find any existing mapping, or to determine that the destination address does not match any cached mapping.

In the former case, the mapping consists of a starting address and ending address for the micronet which the destination address falls within - and a single ETR address. This ETR address (or set of PLF bits) is then applied to the packet - by writing it to the outer header when encapsulating, or by writing into the modified header for EAF or PLF. (PLF only uses 19 bits of the ETR address - just enough to distinguish between the 2^{19} contiguous prefixes which are reserved for this system. The resulting packet is then ready to be forwarded like any other - according to its outer header, or according to the bits just written into its modified header.

If no cached mapping is found, the ITR buffers the packet and sends a map query to a local query server. This includes a nonce which is used to secure the reply, and any later map update messages the query server sends if the mapping changes during the time the ITR caches it.

The local query server sets the caching time on the mapping. This time is not transmitted as part of the fast-push mapping system - it is locally configured and could be set differently for different replies by various algorithms in the query server to optimise its interactions with the ITRs, and to limit the number of mappings the ITR caches. (Further work: It may be desirable for each ITR to be able to communicate to its query server(s) the state of its cache and how close to any limits it is running, so the map replies can have their caching times adjusted downwards.)

The ITR flushes from its cache any mappings whose cache times have expired. The cache includes the starting and ending address of the micronet, the ETR address and the nonce which was sent in the query which returned this mapping.

At any time when a mapping is cached, the ITR may receive a Map Update message from its local Query Server. This will be secured by the nonce of the original query. The most common update will be that this micronet is now mapped to a different ETR address. Another type of update is that this micronet include it being mapped to no ETR (an ETR address of zero) - in which case the ITR will drop subsequent matching packets. If the end-user network splits a micronet into two

Mis en forme : Surlignage

or more smaller micronets, the original micronet is removed from the cache and the new micronets are written.

Alternatively, a micronet might be joined with adjacent micronets to make a new, larger micronet, with the same or a different ETR address. In this case, there will be an update for every already cached micronet affected by the change - and the previously cached micronet will remain, with the new ETR address.

None of these updates reset the caching time. So the mappings, however modified, will time out as usual. If the mapping update message from the query server reset the caching timeout process, then continued updates would keep a mapping in the ITR's cache for excessive periods - including if the ITR was not handling any packets for this micronet.

In this way, ITRs receive all the updated mapping they need, within a fraction of a second of the changed mapping being received by the local query server. This may be directly from the QSD, or from the QSD via one or more QSCs.

5.1.6. ITFH - ITR Function in Host

An ITR function in a sending host performs either encapsulation and PMTUD management or MHF as described above. This function is only for packets generated in the host. ITFH should only be used on hosts which have fast, reliable, connections to two or more local query servers. If there are delays, or packet losses, then the extra management traffic between the ITR function and the local query servers may not function well enough to ensure there are no significant delays to traffic packets.

In many settings, the software and hardware required to implement an ITR in the sending host will have zero incremental cost. RAM and CPU capacity is now extremely inexpensive. Hosts, such as desktop PCs and servers used in hosting farms and cloud systems come bristling with multicore CPUs and gigabytes of RAM for the price of a good shirt.

The host could be on a conventional global unicast address (PI or PA) or on an SPI address. If it is thought desirable to enable ITFHs in hosts behind NAT, then at least two additional measures would need to be taken. Firstly, if encapsulation was used, the PMTUD exchange with ETRs would need to work through the NAT - which it probably would. Secondly, the ITFH would need to set up and maintain a two-way tunnel to two or more local query servers. TCP with a keepalive would be sufficient. Then, instead of UDP mapping queries, replies and updates, the same messages would be sent over TCP. It is not out

of the question to link all ITRs, QSCs and QSDs with TCP, rather than use UDP packets, since the TCP will ensure reliable delivery of messages, and so reduce the complexity of the code for sending and receiving messages.

5.1.7. ITRs Auto-Discovering Local Query Servers

There is further work to do to enable ITRs to automatically discover the addresses of one or more local query servers. This is not absolutely necessary, but would greatly ease the deployment of ITRs in ISP and end-user networks. The more ITRs there are, the less work each one has to do and so the greater the chance that they can be implemented with little cost in a COTS server, rather than an expensive hardware-based router. This principle applies especially to ITFHs.

Supprimé : auto

Supprimé : discovering

Supprimé : local

Supprimé : query

Supprimé : servers

Commentaire [Med31] : Security and integrity should be part of the requirement of such a means.

In a first stage, static configuration would be sufficient just like BGP.

Supprimé : regulating

Supprimé : their

Supprimé : advertisements

5.1.8. ITRs Regulating Their Advertisements

Any ITR - including DITRs and ITFHs - should only process SPI packets and advertise MAB prefixes if it is able to obtain mapping, directly or indirectly, from a full database query server which is fully up-to-date.

An ITR which loses touch with its query servers, or which is informed in some way TBD that the query servers can no longer provide up-to-date mapping, should stop processing SPI packets immediately, and simply forward these packets like any other router. The packets will be handled by some ITR upstream, in the local network, the ISP network or by a DITR. To continue to process packets based on cached and potentially out of date mapping risks sending packets to the wrong ETRs. It is better to let the packets be handled by other ITRs which are working properly.

There is a potential difficulty if a local full database query server is online, but has recognised that its mapping for one or more MABs has corrupt or stale data. One option would be for this server to pass on queries for these MABs to other full database query servers - yet the nearest one with up-to-date mapping may be some distance away in another ISP, and there would need to be commercial arrangements for this. Another option would be for ITRs to be able to recognise that they can't get up-to-date mapping for one or more MABs, and to stop advertising these MABs and processing packets addressed to them. Either approach is probably preferable to a full database query server going offline just because one part of its database is corrupt, due to some transient failure in the fast-push mapping distribution. If the problem can be corrected promptly, within a few tens of seconds, then it is probably OK to let the dependent ITRs function normally, with the outdated mapping information in the QSD,

provided they all get any updates which were delayed by the outage. However, if the mapping information is corrupt, it should not be used to reply to mapping queries. To do so would risk traffic packets being tunneled to ETRs or to any address at all - which is unacceptable from a security point of view.

DITRs need to be closely connected to robust QSDs and should have a backup connection to some other, potentially more distant, QSDs. It would be highly undesirable for the DITR to be turning its advertisements of MABs on and off due to some problem with its query server(s), since this would unreasonably burden other DFZ routers.

5.2. ETRs

5.2.1. In ~~Standalone Devices~~ or ~~Dedicated Routers~~

The ETR function can be performed in a dedicated router or in a ~~standalone device~~, with appropriate software.

Whether the ETR function is performed in a server with one or more Ethernet ports, or a router with multiple ports of various kinds, depends on how the traffic packets are to be forwarded to the one or more end-user networks being served by this ETR. The methods of forwarding do not need to be part of the Ivip RFCs - just how ETRs handle the incoming packets, and for encapsulation, how they communicate with the ITR for PMTUD management purposes.

In the TTR mobility system, the TTRs perform ETR functions. The link to each end-user network is a separate two-way tunnel, established by the Mobile Node (MN) to the TTR.

- Supprimé : servers
- Supprimé : dedicated
- Supprimé : routers
- Supprimé : server

5.2.2. ETRs in ISP ~~Networks~~

An ETR in an ISP network can, in principle, handle packets for many end-user networks - all from a single global unicast address. This has a scaling benefit for IPv4 by supporting a potentially large number of end-user networks, with potentially large numbers of SPI addresses, while requiring only a one of the ISP's IP addresses. (For IPv6, inefficiency of address use is not a concern.)

- Supprimé : networks

5.2.3. ETRs at ~~The End-User Network Site~~

A multihomed end-user network with two links to ISPs might have two ETRs - one for each link. Each ETR will have a stable conventional (non-SPI) global unicast address to receive encapsulated packets on. So each ISP needs to devote at least one of its addresses, or more likely four, for each such ETR. This saves the ISP from having to run an ETR for this customer - all the ISP provides is connectivity

- Supprimé : the
- Supprimé : end
- Supprimé : user
- Supprimé : network
- Supprimé : site

and this small amount of stable address space.

There could be one physical ETR, with two links to the two ISPs, receiving encapsulated packets as above on the two addresses provided by the two links. This device would be a router ~~or even~~ if implemented on a server, since it would also be deciding which link to send outgoing packets on.

Supprimé : of some kind,

5.2.4. MHF ETR ~~Functionality:~~ EAF and PLF

Supprimé : functionality

Supprimé : -

If Ivip is introduced with encapsulation, its ITR and ETR functions will contain Modified Header Forwarding (MHF) functionality ready for a future migration from encapsulation to MHF exclusively. The IPv4 MHF technique - ETR Address Forwarding (EAF) - is very similar to the encapsulation arrangement, so the same ETR could do both, from the same address. However, with EAF, the ETR address is specified with the most significant 30 bits, giving a granularity of 4 IP addresses. To avoid having to change ETR addresses when encapsulation is turned off, only one ETR should be located in each /30 prefix.

The IPv6 approach to MHF ~~(denoted as Prefix Label Forwarding (PLF))~~ is conceptually different from the encapsulation approach in which the packet is tunneled to an ETR at a single IPv6 address. The ITR uses the mapping to write 19 or 20 bits into the IPv6 header. Upgraded routers in the DFZ forward the packet to ISP BRs advertising one of 2^19 or 2^20 separate prefixes. While the mapping still specifies an exact 128 bit IP address for the ETR, before MHF can be turned on, all ETRs must be given addresses within the special set of DFZ-advertised prefixes which the MHF system can forward these packets to.

Supprimé : -

Supprimé : -

On arrival at the BR, the packet itself contains no information of further use - it does not contain the ETR address, just 19 or 20 bits of the address bits which differentiate this contiguous set of prefixes. If there is only one ETR for each such prefix, then the BRs (or perhaps single BR) needs only to forward the packet to the ETR. Alternatively, the ETR function could be performed within the one or more BRs.

However, if this prefix has multiple ETRs, then the BR needs to behave like an ITR and perform a second mapping lookup, using the destination address of the packet, to decide how to forward (or perhaps tunnel) the packet to the correct ETR. There are various techniques for doing this, including the ISP using the PLF bits again, interpreted according to its own arrangements by its internal routers, to forward the packet to some internal prefix (perhaps in ULA space) which leads to the correct ETR. I have not yet explored the various ways an ISP could use to get PLF-tunneled packets to the

correct ETR, or how techniques and ETR placement arrangements for encapsulation can be made compatible with the PLF arrangements. Please refer to the description and example of PLF operation at the end of ~~[[Ivip-Summary-and-Analysis]].~~

With both EAF for IPv4 and PLF for IPv6, the work an ETR performs on each tunnelled packet is trivially simple: restore the altered bits so the IP header has its standard form again, and forward the packet to the destination network. The ETR does not communicate with the ITR or with any other part of the Ivip system, since the ITRs and ETRs have no Ivip-specific PMTUD problems to solve.

If there resulting packet is too long for the next hop, the existing IP stack of the server or router in which the ETR function is performed will implement conventional RFC 1191 PMTUD and generate a PTB to the sending host.

- Supprimé :** Ivip-summary.pdf.
- Supprimé :** Ivip
- Supprimé :** Summary
- Supprimé :** and

5.2.5. ETR ~~Functionality For Encapsulation~~

With encapsulation, ETRs receive IP-in-IP packets on a stable global unicast address. The ETR recognises all such packets and de-encapsulate them. If the outer header source address matches that of the inner packet, then the ETR forwards the packet to the end-user network. If the ETR handles multiple end-user networks, then it will have appropriate configuration or router functionality to forward the packet to the correct end-user network.

For PMTUD management, some more complex functionality is required. When the ITR uses special techniques to send a traffic packet, in two parts, as a probe of PMTU to this ETR, it sends a long packet and one short one (or multiple copies of the short one) to the ETR's address. However, these are not IP-in-IP encapsulated. They are both UDP packets - the long one with the ITR's address as the source address, and the shorter one(s) with the sending host's address as the source address.

If only the short packet arrives, then the long one was lost - probably due to it being longer than the PMTU from the ITR to this ETR. The ETR informs the ITR of this non-reception, and receives an acknowledgement of this. If both the long and short packets arrive, the ETR reconstructs the full traffic packet, forwards it to the end-user network, and informs the ITR that it has been received correctly. This involve significant complexity in the ETR, but does not involve storing state for more than a few seconds.

Once the traffic packet has been de-encapsulated, if the forwarding step leads to the packet being deemed too long for the next-hop MTU, then the conventional IP stack will generate a PTB to the sending host and

- Supprimé :** functionality
- Supprimé :** for
- Supprimé :** encapsulation

RFC 1191 PMTUD will proceed just as it would if there had been no ITR to ETR encapsulation.

5.3. ~~Full Database Query Servers (QSDs)~~

5.3.1. Placement in ISP and ~~End-User Networks~~

ISP networks and end-user networks do not absolutely need ITRs - since packets sent by hosts in these networks could always be forwarded to the DFZ and handled there by DITRs. However, as a service delivered to customers, to ensure the packets their hosts send to SPI addresses take the shortest path to the correct ETR, and to avoid depending on DITRs, ISPs will generally want to install ITRs in their network. This means they need to have at least one full database query server (QSD), for these ITRs to obtain mapping from. To rely on QSDs outside their network is probably not reliable or robust enough, since if an ITR can't get mapping, it won't be able to handle new flows of user traffic to SPI addresses.

- Supprimé : QSDs - full
- Supprimé : database
- Supprimé : query
- Supprimé : servers
- Supprimé : end
- Supprimé : user
- Supprimé : networks
- Commentaire [Med32] : Don't parse it.

Full database query servers could be implemented in a conventional router, but it makes most sense to implement them in COTS servers. (Currently we think of conventional routers being a very expensive place to run complex software. However, with cheap multicore 64 bit CPUs and gigabytes of RAM, future router models might have plug-in PCBs to add blade servers or the like to run such complex software in the same chassis.)

Query servers do not handle traffic packets. They must be reachable on stable global unicast addresses. Generally speaking, if the QSD is in an ISP network or in an end-user network which does not use SPI space, but which has ITRs, it will be on conventional (not SPI) space. QSDs in end-user networks which use SPI space will be located on SPI addresses. This means that the Replicators which send the mapping feeds to these QSDs will produce the feed packets addressed to the QSD's SPI address. Therefore, the feed packets will either need to be tunneled by a nearby ITR, or the Replicator itself could have an ITR function built in (ITFH).

- Supprimé : located
- Commentaire [Med33] : or by a fixed FQDN ?

If an end-user network (using PA, PI or SPI space) runs ITRs (including ITFHs), then it could have these ITRs send their queries to the QSDs of its one or more ISPs. This would be the best arrangement for smaller networks, since running one, or ideally two QSDs in its own network involves having two or four feeds of mapping data from Replicators outside its network.

ISPs are not necessarily going to charge for this mapping feed traffic if the Replicators are within their network, since they are already accepting mapping feeds from outside their networks for their

- Commentaire [Med34] : This is subjective and should avoided as possible.

own QSDs, and it does not add to their upstream traffic load to replicate the mapping feeds so several can be sent to each substantial end-user network. However, once the Ivip system is handling large numbers of end-user networks, such as with a billion micronets, running QSDs is a non-trivial task. For instance, the initialisation operation described below could involve large quantities of bandwidth.

Every ISP which has ITRs, including ITFHs, needs at least one full database query server. For robustness reasons, each should have two or perhaps three or more. Each QSD should have at least two feeds of mapping information from two upstream Replicators, ideally via different upstream links. Ideally, each QSD in the ISP network should have its mapping feeds from different Replicators.

Supprimé : (QSD)

A Replicator could be implemented in the same server as a QSD - and this would be fine in the earlier years of deployment. Later, when the QSD has to handle more mapping updates and requests, it would probably make sense to put the Replicator function on another server.

5.3.2. QSD Initialization and Reception of Mapping Updates

Supprimé : initialization

Supprimé : reception

Supprimé : mapping

Supprimé : updates

Supprimé : servers

Booting up a QSD involves it downloading snapshots of the mapping database from RUAS (Root Update Authorisation Servers), whilst storing all mapping updates which were generated after the snapshot. Then, while still storing incoming mapping updates, the stored updates are applied in chronological order to bring the QSDs copy of the mapping database fully up to date and ready to receive updates as they arrive. This initialisation process results in the QSD having a complete list of all the MABs in the Ivip system. (Further work: plan how a running QSD finds out about additions and deletions from this list of MABs, including the commencement and cessation of mapping updates for these MABs.)

Mis en forme : Surlignage

Normally, each QSD will receive two (or perhaps more) feeds of mapping updates from two not-too-distant Replicators in topologically different locations. Each feed, in a given second, consists of a number of UDP packets, each with updates for one or more MABs. In the current design of the fast-push mapping distribution system, the feed consists of UDP packets secured with Datagram Transport Layer Security RFC 4347. If a packet is missing from the stream from one Replicator, a packet with the same updates will most likely be received from the other Replicator. (Each Replicator uses the same system of two redundant feeds to generate the mapping feed it sends out to other Replicators and to QSDs.)

If one or more packets are lost from both streams, then the QSD will retrieve the missing updates from a server operated by the RUAS which

generated these updates. In this way, even with temporary loss of mapping feed packets, the QSD is able to keep its local copy of the mapping database up-to date with all the mapping updates. The time delay from an end-user network making a mapping change (or the change being made by some company authorised to do so by the end-user network) to the update being integrated into QSDs all over the Net, will be in the order of 4 seconds.

The QSD also needs diagnostic and management functions, such as to be able to report problems with missing packets. It may also be configured to choose feeds from alternative Replicators if one or more of its current Replicators is not working properly.

There is considerable complexity in these functions, but the software, once written and tested, can run on COTS servers. Mapping data is most likely to be stored in a format which is also suitable for responding to mapping queries. Otherwise the mapping data itself needs to be updated, and then some other data structure suitable for query resolution needs to be replicated.

The QSD will store the full mapping database in RAM. For IPv4, the mapping data itself is very simple - just 12 bytes in its most compact form: 32 bits for the micronet's starting address, 32 bits for its length or end (though micronets would rarely exceed 2^{10} IPv4 addresses). If the IPv4 Internet ever supports 2 billion micronets, then this is 24 gigabytes, which can be done with DDR3 server RAM today. The storage format may not be this compact, but by the time such large numbers of end-user networks adopt SPI space, COTS servers will have more RAM. The difficulties of writing the code for, and running both QSDs and Replicators are justified by the avoidance of initial packet delays and the complexity of global query server systems such as LISP-ALT.

IPv6 mapping is more voluminous - 64 bits for the micronet's starting address, 64 bits for its length or end and 128 bits for the ETR address. This is 32 bytes. The worst case storage requirement, not counting any compaction or expansion due to how it is stored in an easily traversable form for answering queries, and making updates, is 10 billion micronets - 320 gigabytes. This fits on a consumer hard drive today, and will surely fit in COTS server RAM by the time such massive adoption of mobile IPv6 devices occurs.

In addition to applying the updates to the local copy of the database, any updates which match items in the "querier cache" are used to send mapping updates (using a protocol described below) to the matching querier.

5.3.3. Responding to Queries**Supprimé :** queries

The simplest model of ITRs and QSDs is that all ITRs send their queries to a single QSD. In practice each ITR may have two or more QSDs and may send a query to one if it does not get a response within a few tens of milliseconds from the first. ITRs send mapping queries when they receive a packet addressed to an SPI address which they do not have any cached mapping for.

Commentaire [Med35] : QSD must be robust and must handle a highly large number of incoming request per second.

Note that in this design: QSD is a single point of failure.

While waiting for the map reply, the ITR buffers the one or more packets which need this mapping. As soon as the mapping is received, the ITR tunnels these packets to the ETR specified in the mapping.

In practice, as is discussed below, it will often be desirable to place one or more levels of caching query server (QSC) between the ITRs and the QSD. So a QSD may receive mapping queries from both ITRs or QSCs. Queries from ITRs use exactly the same protocol as those from QSDs.

The mapping query is a UDP packet sent to a well-known port on the QSD. It contains a nonce which is specific to this query. (Any repeat of the same query will use a different nonce.) The packet also contains the destination address of the packet which the ITR needs mapping for. The ITR (or QSC) only knows that this address is within one of the MABs, and that it does not match any micronet in its cache.

The QSD looks up this address in its database and returns the ETR address. A zero address is an instruction to the ITR to drop the packet. The map reply packet contains the nonce from the query, and four items of information: the start address of the micronet, its end or length, the ETR address and a TTL caching time, in seconds.

If the querier is an ITR, it adds this mapping to its cache and is then able to handle any packets addressed to this micronet. The QSD sets the caching time, according to local policy and perhaps according to some functions which in some way help optimise the operation of this QSD, any QSCs and their dependent ITRs.

The QSD makes a record of the response in its querier cache. It does not need to store the address which was queried. It stores the address of the querier, the nonce in the query, the time the response was sent, the caching time (or the time at which the caching time will expire) as well as the micronet starting point and length and the ETR address. A background process deletes records whose caching time expires.

5.3.4. Sending Mapping Updates to ITRs and QSCs

Supprimé : mapping

Supprimé : updates

The following process is potentially expensive, due to the potentially large number of records in the querier cache. This may set some upper limits on the capacity of a single QSD to support ITRs directly - in terms of many ITRs, and or the traffic patterns of these ITRs, leading to a large and slow to index querier cache in the QSD. The use of QSCs between the ITRs and the QSD should significantly reduce the number of entries in the querier cache, and so enable a QSD to serve more ITRs.

The querier cache needs to be indexable on the addresses of its micronets. Micronets do not overlap, so the cache would have an index of the entries in ascending order by address - which will enable reasonably rapid searches.

Once a mapping update from the upstream Replicators has been applied to the QSDs mapping database, a check is performed to test whether it covers any address space in the querier cache. If so, then the QSD will generate a mapping update to that querier. The simplest example is when a micronet has its ETR address changed.

The resulting mapping update includes the nonce from the original query, and a nonce or sequence number to distinguish this mapping update packet from any others. The mapping update is sent as a UDP packet to the querier. (TCP sessions between ITRs, QSCs and QSDs might be a good idea rather than UDP packets.) The QSD expects an acknowledgement - also a UDP packet with the nonce and sequence number. It will resend the mapping update packet until it gets an acknowledgement - or too many retries generates an error event. This may signal that the querier is no longer alive. So within a few tens of milliseconds of the QSD receiving a mapping update from the fast-push mapping system, it relays a suitable one or more mapping updates to any queriers which are caching a mapping which is affected by the just-received update.

When a micronet changes its ETR address, the mapping update is simple - it contains the micronet's start and end (or length) and the new ETR address. If the querier is an ITR, it updates its cache and so tunnels any matching packets to the new ETR address.

Mapping changes could be more complex - since existing micronets could be divided into smaller ones or merged with adjacent ones. Some moderately complex logic will deal with the various possibilities subject to the constraints that no matter what mapping updates are received, the querier will not be given mapping for any addresses beyond the range of the micronet in the initial map reply message. This may involve multiple mapping updates which require the

querier to remove the existing micronet from its cache, and replace it with two or more smaller micronets.

Any altered cache items in the querier, including new ones when the initial micronet has been divided, will have exactly the same caching timeout as the original. So no matter what mapping changes arrive, the resulting cache items will expire in the querier at the same time as the original cached mapping would expire. If mapping updates extended the cache time, then there may be no limit to how long the querier needs to cache some micronets, even though it may not need this mapping any more.

From this it can easily be imagined that ITRs will be able to quickly and reliably receive the mapping they need from a local QSD and that the QSD will update the mapping cache of all ITRs which are caching mappings for which it receives updates.

5.4. QSCs - ~~Caching Query Servers~~

A caching query server (QSC) is a relatively simple function, typically implemented as software in a server. The software for ITRs, QSCs and QSDs would share some common components. A QSC receives and responds to queries from ITRs or other QSCs in the same manner as a QSD. The QSC sends queries to a QSC or QSD in just the same way as was described above for an ITR - and likewise receives map reply messages and map update messages from the upstream QSC and QSD as just described.

There could be zero, one, two or in principle any number of QSCs between an ITR and the one or more QSDs. All these devices are typically in the same ISP network - or in an end-user network with close connections to the QSD(s) in the ISP network. So communication between them is very fast, reliable and inexpensive. Typically, there will be little or no packet loss, but the protocols will need to cope with any losses in a robust manner. If a querier sends out a query and does not get a reply within some quite short time, such as 100ms, then it should try sending the query (with a different nonce) to an alternative upstream query server.

Further work: ITRs auto discovering query servers in general - and QSCs autodiscovering other QSCs and QSDs. Manual configuration of the tree-like structures of these devices should also be possible.

If the mapping needs of one ITR were completely uncorrelated with the mapping needs of other ITRs served by the same QSD, then there would be little or no benefit in deploying intermediate QSCs. However, there is likely to be sufficient commonality between the mapping needs of tens or hundreds of ITRs and ITFHs to make QSCs a good

Supprimé : caching

Supprimé : query

Supprimé : servers

investment in expanding the capacity of a single QSD to support more ITRs.

If 20 ITRs send their queries to QSC1 and another 20 to QSC2, then the queries, replies and map update exchanges which must be performed by the one QSD which both QSC1 and QSC2 query will be significantly reduced. This is because it will frequently be the case that QSC1 will already be caching the mapping which is needed to answer a query from one of its 20 ITRs. Without the QSCs, every ITR query would need to be handled by the QSD - and its querier cache would be correspondingly larger. Furthermore, if more than one of QSC1's ITRs is caching mapping for a micronet for which the QSD gets a mapping update, then the QSD only needs to send a single mapping update to QSC1, rather than sending one to each such ITR.

There is further work to do planning these protocols. The caching times do not affect Ivip's ability to get the mapping updates to all ITRs in real-time. Longer caching times will reduce the need for the querier, such as an ITR, to make another map request if it is still sending packets to the micronet. Longer caching times also increase the number of mapping updates which need to be sent - and perhaps the time is so long that the querier no longer needs the mapping. Shorter caching times reduce the number of cached items, but increase the load of mapping queries and responses.

While the exact details are TBD, it is clear that it will be possible to define relatively straightforward protocols by which ITRs, QSDs and optionally QSCs can be combined to efficiently support the mapping needs of many ITRs per QSD. Once the code for the ITR and QSD is written, writing the QSC code will be relatively easy. Since QSCs don't need a mapping feed from Replicators, they can be numerous and located close to ITRs - especially if there are a large number of sending hosts with ITFH ITR functions built-in. A QSC close to a bunch of ITRs is also attractive if this is not directly close to the QSD - for instance in an end-user network which uses one or more QSDs in the ISP network.

5.5. FMS - Fast-Push Mapping Distribution System

Ivip's most important architectural feature is that end-user networks are able to control all ITRs in real time - the address of the ETR to which packets which are addressed to their micronets will be tunneled. Previous sections describe how ITRs request and cache mapping from QSDs, and have their caches updated by the QSD if there is a change to this mapping within the caching time. While there is considerable complexity in the code required for ITRs, QSDs and QSCs, the protocols used between them are not particularly complex, and the communication takes place over physically and topologically short

distances, within ISP and end-user networks. So the delays and potential packet losses of the global Internet are not a concern for these elements of Ivip.

The most adventurous aspect of Ivip is the Fast-Push Mapping Distribution System (FMS), which enables mapping change commands originating from end-user networks or some other organisation acting on their behalf to be transmitted securely and robustly to all QSDs in the Net. There may be hundreds of thousands of these and the system should scale well to QSD numbers in the 10^7 range. The following material broadly describes the requirements of this system and one possible way of implementing it.

Perhaps there is a simple flooding protocol with a large mesh of servers which would work well. The following cross-linked multicast-like set of trees can be viewed as a flooding system, with each Replicator flooding the packet to 20 other Replicators as soon as it gets the first packet from one of its upstream Replicators.

More detailed material is in [I-D.whittle-ivip-db-fast-push]. Figure 1 depicts the structure of part of the FMS.

Figure 1: Multiple levels of Replicators drive hundreds of thousands of QSDs.

The FMS needs to be robust, secure and fast. Ideally it should take no more than four seconds for mapping changes to reach all QSDs. Once there, all ITRs which need the mapping change will be sent it within a small fraction of a second.

The FMS needs to be decentralised so there is no single point of technical, administrative or commercial failure. The main part of it will be run by a number of organizations working together. These organisations, of which there might be five to perhaps twenty, are called Root Update Authorisation Server (RUAS) organizations. The final parts of the FMS - the last one or more levels of Replicators fanning out the mapping update packets to QSDs - may be run by local ISPs rather than these organizations.

Each RUAS organisation operates its own RUAS system, which would be a multi-server system with no single point of failure within itself. Each RUAS is responsible for the mapping updates of particular multiple MABs. Within the FMS's Replicator system, which fans out the mapping updates to QSDs, each packet contains multiple mapping updates, and some other regularly inserted information regarding snapshots of each MAB's mapping information.

Each RUAS accepts mapping updates, after appropriate authentication, from end-user networks or whichever organisation they contract to control the mapping of their micronets. The RUAS may accept these updates directly or indirectly via other organisations.

Typically, mapping updates would be sent via some automated process, such as from a MM (Multihoming Monitoring) company - but manual control such as via a web-based form would also be possible. The most likely scenario for multihomed end-user networks is that their appointed MM company would send mapping changes to the MAB company from whom the end-user network is renting their SPI space. The MAB company's servers would convey the mapping change within a fraction of a second to the RUAS this company has chosen to handle the mapping updates for their MABs.

In the current design, RUASes are loosely interlinked and have some form of dynamically changed quota so that in each second, each RUAS has an upper limit on the number of mapping updates it can send. This is to ensure that that the number sent will not overload the Replicators. If an RUAS has a peak of updates to send, more than this limit, then some will need to be delayed for a few seconds.

The RUAS organizations collectively run a set of Launch servers - for

instance 8 in the above diagram. Launch servers are strategically located at major data centers around the world, such as one in China, one in Japan, one in North America etc. There is a pipelined process which ensures that each second, the Launch servers all send out an identical set of updates. This is to ensure that all QSDs get the same set.

In the first stage of the pipeline, for one second, the RUAS would collect incoming mapping updates and sort them according to which MAB they concerned. In the second stage, the RUASes would transmit their updates to every Launch server. In the third stage, also lasting a second, the Launch servers compare notes and form a quorum which agrees on exactly which sets of updates will be sent to the FMS. This will require some careful programming, but there may be existing protocols which can help with this. The outcome of the third stage of the pipeline is that all Launch servers will have the same set of updates ready to launch into the FMS in the next second - and all RUASes will know this set while preparing which updates to send in the next second.

In the final stage, each of the (for example) 8 Launch servers sends out the updates, as a series of UDP packets secured by DTLS (RFC 4347) to multiple Level 1 Replicators. Each Replicator is a COTS server with suitable software. The sessions with these Launch servers are reasonably stable, but the whole structure of the FMS cross-linked trees of Replicators must be adaptable to gradual change, while in continuous operation.

Each Level 1 Replicator will receive at least two sets of update packets each second, from different Launch servers. Ideally these will arrive from different links and will be from topologically diverse Launch servers. Each packet carries fields which specify which second it was sent in, and which number the packet is in the complete set of packets sent that second. While the packets of each feed are different, in terms of their DTLS protection and source addresses, the mapping change payload of similarly numbered packets from different Launch servers will be identical. So, for instance, as long as the Replicator receives a packet 23 for the current second from one of its upstream Launch servers, then it will be able to replicate this packet's information to multiple Replicators in the next level. If a Replicator does not receive any packets of a given number, it makes no attempt to obtain the information from another source.

As soon as a Level 1 Replicator receives the first copy of a given numbered packet for this second, assuming it is received completely and that DTLS ensures it is really the packet sent by the Launch server, this Replicator transmits copies of its payload to all its

downstream Replicators in the next level. Each such packet is handled by DTLS for the session previously established with each Downstream Replicator. In this example, there are 4 levels of Replicators, each receiving two streams and generating 20 streams to Replicators on the next level. So the amplification factor is 10. In this example, 80,000 Level 4 Replicators send update streams to 800,000 QSDs. It may not be strictly necessary for a Level N Replicator to receive packets from Level N-1 Replicators, but this is the easiest way explain the structure. There is no danger of loops since a Replicator will ignore a packet with second and sequence number it has already sent.

This resembles a multicast arrangement - and multicast is very subject to packet loss. However, each Replicator has a very good chance of receiving all the information, despite occasional packet losses and the failure or unreachability of any of upstream Replicators. The transmission of data should be nearly instantaneous at each Replicator, so the mapping data should be fanned out all over the world, through four or more layers, in less than a second.

Supprimé : unteachability
 Supprimé : one

At the branch-tips of this cross-linked tree-like structure of Replicators, the packets are received by QSDs. Each QSD should have two or more feeds of updates, ideally coming over different physical links from topologically diverse Replicators. Like Replicators, QSDs receive the update packets securely via DTLS, and discard all but the first of each sequence number each second.

The one server could run both QSD and Replicator functions. In principle one server could do this and be an ITR and ETR as well. In the initial stages of deployment, with update rates relatively low compared to the full-scale 10^10 mobile end-user network scenario, it will probably be attractive to run multiple Ivip functions on the one server.

The RUAS companies will probably run the first two or so levels of Replicators - which would the mapping updates to major exchange points all over the Net. For levels 3 and beyond, ISPs could be relied upon to run their own Replicators, and to offer feeds to each other, as part of ensuring the mapping feed for all ISPs in their area will be robust.

There is significant complexity in the RUAS systems, but their interface to the Launch servers is the only thing which needs to be standardised. The Launch servers need to run a complex algorithm so they all decide, and all know the final decision, on which updates all Launch servers, or perhaps a majority of them, will send in the next second. This is to ensure that all level 1 Replicators are sent the same set of packets. This is primarily to avoid a problem where

one Launch server only sends its updates to few, or none, of the level 1 Replicators. Such a condition, combined with lost packets and dead links could result in the cross-linking of the system being insufficient to ensure all QSDs get the same set of updates.

An RUAS which was for some reason unsuccessful in having some or all of its updates accepted for launch in the next second would add the updates they contain to the list of updates to send in the second which follows. This failure may have been due to an insufficient number of active Launch servers receiving this RUAS's full set of update packets. To form a quorum which agrees on what updates to send, it is not necessary that every Launch server has every update. As long as the majority do, then due to the crosslinked nature of the Replicator system (and assuming some maximum number of dead links and dropped packets), it can be assured that every QSD will get the same set of mapping updates.

As noted above, QSDs will sometimes fail to receive the same numbered packet from both of its upstream Replicators. Each RUAS will provide HTTP servers by which any QSD, or indeed any other host, could download specific packets from the last minute or so. Each packet would be named according to its place in that second's sequence and according to the second it was sent in. There would be some arrangement by which QSDs can reliably ascertain from each packet how many packets there are in each second - and failing that, this information could be obtained from the RUASes HTTP servers.

At regular intervals, such as every few minutes, the RUAS will include a special type of "snapshot" message in the update stream. This will concern one of its MABs, and provide some kind of hash result or other compact number by which a QSD can check the validity of this MAB's section of the mapping database. This means that QSDs will need to maintain their internal data in a standardised way which is amenable to computing a local hash function to compare with that received in the snapshot message.

Each snapshot message will coincide with a compact dump of this MAB's mapping database, at a particular point in time. When a QSD is initialising, or if it loses sync with updates and needs to regenerate its database for one or more MABs, it waits for a snapshot message and records it and all subsequent updates for this MAB. It then downloads the snapshot from one of the RUAS's HTTP servers, unpacks it into RAM and applies all the updates received since the snapshot. This brings this MAB's mapping up-to-date, ready for answering mapping queries and ready to be updated as further updates arrive.

In the long-term future, if QSDs ever hold 50 gigabytes or more of

mapping information (that is around a billion or more IPv6 micronets) it would be good to have a protocol so one QSD could initialize itself by copying live data from another.

This is a challenging piece of infrastructure to design and run. However, the code for the Replicator, once written, will serve well for all Replicators. Parts of this code will be re-used in the RUAS and QSD code. There needs to be some kind of monitoring and management system for Replicators, and some localised arrangements for ISPs to collaborate on inter-linking their various levels of Replicators.

The FMS is not necessarily expensive in terms of hardware, since the whole RUAS, Launch server, Replicator, QSD and QSC system can be implemented on COTS servers. Replicators store no data and are not involved in complex communication protocols other than the relatively simple functionality described above. Nor is it necessarily expensive in terms of bandwidth for an ISP to receive multiple feeds of mapping updates from Replicators near and far.

The exact volume of updates depends on many factors, including how low the price per update is, and how many end-user networks find this price attractive for real-time inbound TE steering of traffic to maximise the utilization of their data links. Market forces will cause RUASes to set the fee per update at a level which encourages deployment of a fast Replicator system, and lowers fees to make updates more attractive for end-user networks - within the limits imposed by the actual cost of running the RUAS and Replicator systems.

The use of this system for 10^{10} micronets, most of them for mobile devices, should be technically and economically feasible by the time such levels of adoption are achieved. The TTR mobility system does not involve mapping updates when the MN changes its physical address, or moves to another access network. Mapping changes are desirable when the MN's point of connection is far enough away from the TTR to choose a closer TTR, in order to minimise path lengths. This would be infrequent, since MNs do not frequently move such distances of 1000km or more.

5.6. MHF - Modified Header Forwarding

5.6.1. EAF (~~ETR Address Forwarding for IPv4~~)

Supprimé :-

Please see [I-D.whittle-ivip-etr-addr-forw] and the discussion above in the ITR section.

EAF will not accept fragmented packets or fragmentable packets longer

than some globally agreed constant, somewhat below 1500 bytes. By the time Ivip is introduced, it will have been over 20 years since RFC 1191 PMTUD was introduced. There's no need for fragments or fragmentable packets - and IPv6 does fine without them.

EAF requires upgraded routers between ITRs and ETRs. This does not necessarily include every DFZ router, but it is reasonable to approximate the requirement to this. For instance, if a DFZ router never handles packets for networks which contain either ITRs or ETRs, then it does not need to handle EAF formatted packets. EAF ETR addresses contain only the 30 most significant bits. To avoid the need to change ETRs' addresses when encapsulation is transitioned to EAF, ETRs should not be placed closer than 4 IP addresses apart. Perhaps they should be placed on the 01 address of these four.

Since ITRs will commonly be placed deep within ISP and end-user networks, and ETRs may be deep within ISP networks (such as at an end-user site, at the end of the link from the ISP) any router between the DFZ and these devices also needs to handle EAF packets.

It will be straightforward to build this capability into new routers, and into firmware updates for many existing routers. The upgrade only concerns the FIB. All that is altered is that the FIB forwards the packet according to the 30 bits ETR address bits in the header, rather than using the destination address. There is no change to BGP functions, the RIB or how the RIB writes to the FIB.

Since it is going to take a few years before Ivip or the like is introduced, it is possible that by then, many or almost all of the installed DFZ routers will be able to do this with a firmware update.

With a year or two's notice, upgrading all the DFZ routers, and likewise many internal routers, would enable Ivip to be introduced in its final mode of operation - without encapsulation overhead or its PMTUD problems. This means all ITRs can be a lot simpler - and that ETRs can be trivially simple. Reducing the complexity of ITRs is perhaps the biggest challenge in designing a core-edge separation architecture, since we want ITRs to be cheap and plentiful, including them being easy to add to the stacks of sending hosts. Starting with EAF would also avoid the need for devising a transition mechanism from encapsulation.

5.6.2. ~~PLF (Prefix Label Forwarding) for IPv6~~

The current state of PLF design is described in [~~PLF for IPv6~~]. Please see this for more details, including why it is totally different from MPLS and how it could be extended to provide a similar 2^19 or 2^20 destination forwarding system within each ISP (or end-

Supprimé : -
Supprimé : ,
Supprimé : PLF
Supprimé : for

user) network. Please refer to the description and example of PLF operation at the end of ~~[[Ivip-Summary-and-Analysis]].~~

While EAF is pretty much a functional replacement of IPv4's encapsulation system, PLF is rather different in that it only takes the packet to a BR of one of 2^19 or 2^20 DFZ-advertised prefixes. This would be a regular, contiguous, set of prefixes used only by ISPs - for this and for potentially other purposes.

If Ivip for IPv6 began with encapsulation, then it would make sense for the ETRs to be already located in these special prefixes. Otherwise, they would need to be moved there before EAF could be turned on.

EAF may require a second lookup at the BR of the ISP's network - if there are more than one ETRs for that prefix. One way of forwarding the packet from the BR to the correct ETR would be to use these PLF bits for a similar system within the ISP's network, with 2^19 or 2^20 internal prefixes. How the ISP uses these bits is a private matter. This could be a very powerful way of directing traffic inside a large provider network. This would give rise to ePLF - the one system for the DFZ - and iPLF, as used inside an individual ISP network.

Rapid adoption of IPv6 is still somewhere beyond the immediately foreseeable future. So there's no hurry about deploying a scalable routing solution for IPv6. I think the most likely scenario for widespread adoption of IPv6 is one or more large 3G systems using it to give each phone (or whatever) its own global unicast address. This in itself will not cause a scaling problem, since these will be large systems with few new prefixes to add to the DFZ. However, there would then be a strong need for mobility - and the TTR approach has advantages over traditional MIP techniques, including seamless, generally optimal path mobility with the MN on any address, in any access network.

Perhaps by the time Ivip is deployed for IPv6, all the IPv6 DFZ routers will be upgradable to PLF with firmware updates - so scalable routing could be done without encapsulation. PLF involves small changes to the FIB and to the RIB. It does not involve any new BGP functionality.

5.7. TTR Mobility

TTR Mobility is fully described, with diagrams, in ~~[[TTR-Mobility]].~~ This architecture will work equally well for IPv4 and IPv6. The MN can be on any kind of address, including behind multiple layers of NAT, on DHCP addresses and on addresses provided by conventional

- Supprimé : Ivip-summary.pdf.
- Supprimé : Ivip
- Supprimé : Summary
- Supprimé : ¶

- Commentaire [Med36] : arguable
- Commentaire [Med37] : agreed

Supprimé : TTR

Mobile IP protocols. The MN can even be on an SPI address which is within another MN's micronet. No stack or application changes are required and the hosts communicate normally with all other hosts, including of course others using TTR mobility. There is no home-agent and paths to correspondent hosts are generally optimal.

Mapping changes are not required when the MN gains a new address. They are not actually required at all, but are desirable if the MN moves to a part of the network which is far from its current TTR. This may be a distance of 1000km or more. Then, it should establish a tunnel to a nearby TTR so the TTR company can change the mapping of its micronet to this new TTR. With Ivip's real-time control of mapping, this means the MN could close the tunnel to the old TTR within five or so seconds of the mapping change being sent. Changing the mapping does not cause any glitch in connectivity, since the MN gets packets from both TTRs during the changeover.

The MN needs some additional tunneling software - which is controlled by the TTR company. This could be added alongside existing stacks, or integrated into the stack. Ideally the MN to TTR interface would be standardised in RFCs, but this is not strictly necessary, since the MN only needs to interoperate with TTRs of the TTR company chosen by the MN's owner. RFC-standardised MN and TTR functionality would be desirable, by allowing easy choice between TTR companies without the need to install software. However, there is a lot of scope for innovation in this area, and it might be difficult to adequately develop a full range of desirable protocols soon enough for the expected rapid uptake of mass-market Mobility.

I think this approach to mobility, for IPv4 and at some stage for IPv6, is so attractive that there would be a business case for a company setting up its own Ivip-like system just for this purpose - irrespective of the need for a scalable routing solution. Such a system would need to use encapsulation. Multiple such systems could exist at the same time - and a MN in one system A would be able to communicate directly with a MN in another system B via the following paths (->) or tunnels (==>): MN-A ==> TTR-A -> (via DFZ) -> TTR-B ==> MN-B.

Any such systems should be designed to be upgraded in the future to comply with future RFCs for an Ivip-like system, including initial or long-term adoption of Modified Header Forwarding rather than encapsulation.

6. Security Considerations

Security analysis can only be done in the years to come, once the protocols are designed in some detail.

Ivip ITRs and ETRs are much simpler than those of LISP. The fast-push mapping system is a tightly structured, easily secured, system for pumping data from one set of sources out to many QSDs. It is unusual, but it is simpler than global query server systems such as LISP-ALT or DNS, and a lot simpler than BGP, with its routing information percolating through the DFZ one router to the next, via complex individual messages.

The ALT network carries queries which could be used to infer communication patterns of users, with considerable detail. The global part of Ivip simply pushes mapping data - and so does not handle any such queries. Queries are made, but typically the QSD is in the ISP's network, where security can be well assured - rather than having to trust that every ALT router which handles a mapping query is secure against compromise.

Ivip ETRs easily enforce ISP BR source address filtering. For LISP ETRs to enforce this would be administratively complex and very expensive for large numbers of filtered prefixes.

7. IANA Considerations

[To do.]

8. Informative References

[C-E-Sep-Elim]

Jen, D., Zhang, L., Lan, L., and B. Zhang, "Towards a Future Internet Architecture: Arguments for Separating Edges from Transit Core", September 2008, <<http://conferences.sigcomm.org/hotnets/2008/papers/18.pdf>>.

[Constraints-Voluntary]

Whittle, R., "List of constraints on a successful scalable routing solution which result from the need for widespread voluntary adoption", April 2009, <<http://www.firstpr.com.au/ip/ivip/RRG-2009/constraints/>>.

[Critique of draft-jen-mapping-00]

Whittle, R., "draft-jen-mapping does not apply to the TTR Mobility architecture", January 2010, <<http://www.ietf.org/mail-archive/web/rrg/current/msg05605.html>>.

[DFZ-unfrag-1470]

Whittle, R., "Google sends 1470 byte unfragmentable packets", August 2008, <<http://www.firstpr.com.au/ip/ivip/ipv4-bits/actual-packets.html>>.

[Deering-1996]

Deering, S., "The Map & Encap Scheme for scalable IPv4 routing with portable site prefixes", March 1996, <<http://irl.cs.ucla.edu/references/Deering-encap.pdf>>.

[Host-Responsibilities]

Whittle, R., "Objections to burdening hosts with more Routing and Addressing responsibilities", December 2009, <<http://www.firstpr.com.au/ip/ivip/RRG-2009/host-responsibilities/>>.

[I-D.adan-idr-tidr]

Adan, J., "Tunneled Inter-domain Routing (TIDR)", draft-adan-idr-tidr-01 (work in progress), December 2006.

[I-D.ietf-lisp]

Farinacci, D., Fuller, V., Meyer, D., and D. Lewis, "Locator/ID Separation Protocol (LISP)", draft-ietf-lisp-05 (work in progress), September 2009.

[I-D.irtf-rrg-recommendation]

Li, T., "Recommendation for a Routing Architecture", draft-irtf-rrg-recommendation-03 (work in progress), December 2009.

[I-D.jen-apt]

Jen, D., Meisel, M., Massey, D., Wang, L., Zhang, B., and L. Zhang, "APT: A Practical Transit Mapping Service", draft-jen-apt-01 (work in progress), November 2007.

[I-D.jen-mapping]

Jen, D. and L. Zhang, "Understand Mapping", draft-jen-mapping-00 (work in progress), October 2009.

[I-D.lear-lisp-nerd]

Lear, E., "NERD: A Not-so-novel EID to RLOC Database", draft-lear-lisp-nerd-06 (work in progress), December 2009.

[I-D.lewis-lisp-interworking]

Lewis, D., "Interworking LISP with IPv4 and IPv6", draft-lewis-lisp-interworking-00 (work in progress), December 2007.

[I-D.meyer-lisp-cons]

Brim, S., "LISP-CONS: A Content distribution Overlay Network Service for LISP", draft-meyer-lisp-cons-04 (work in progress), April 2008.

[I-D.rja-ilnp-intro]

Atkinson, R., "ILNP Concept of Operations", draft-rja-ilnp-intro-02 (work in progress), December 2008.

[I-D.whittle-ivip-db-fast-push]

Whittle, R., "Ivip Mapping Database Fast Push", draft-whittle-ivip-db-fast-push-02 (work in progress), January 2010.

[I-D.whittle-ivip-etr-addr-forw]

Whittle, R., "Ivip4 ETR Address Forwarding", draft-whittle-ivip-etr-addr-forw-00 (work in progress), January 2010.

[I-D.whittle-ivip-glossary]

Whittle, R., "Glossary of some Ivip and scalable routing terms", draft-whittle-ivip-glossary-00 (work in progress), January 2010.

~~[Ivip-Summary-and-Analysis]~~

Whittle, R., "Ivip Conceptual Summary and Analysis", December 2008, <<http://www.firstpr.com.au/ip/ivip/Ivip-summary.pdf>>.

Supprimé : Ivip
Supprimé : Summary
Supprimé : and

[Ivip-2007-06-15]

Whittle, R., "ViP: Anycast ITRs in the DFZ & mobile tunnels", June 2007, <<http://www.ietf.org/mail-archive/web/ram/current/msg01518.html>>.

[LISP-ALT-Critique]

Whittle, R., "How can the ALT structure scale to 10^8, 10^9 or 10^10 EIDs with minimal delay times and robustness against single points of failure?", December 2009, <ALT structure, robustness and the long-path problem>.

Supprimé : "

Supprimé : "

[Namespace]

Whittle, R., "The meaning of the term *namespace* in addressing, computer networking etc.", April 2009, <<http://www.firstpr.com.au/ip/ivip/namespace/>>.

~~[PLF-for-IPv6]~~

Whittle, R., "Prefix Label Forwarding (PLF) - Modified Header Forwarding for IPv6", August 2008, <<http://www.firstpr.com.au/ip/ivip/PLF-for-IPv6/>>.

Supprimé : PLF

Supprimé : for

[PMTUD-Frag]

Whittle, R., "IPTM - Ivip's approach to solving the problems with encapsulation overhead, MTU, fragmentation and Path MTU Discovery", April 2008, <<http://www.firstpr.com.au/ip/ivip/pmtud-frag/>>.

[TRRP]

Herrin, W., "Tunneling Route Reduction Protocol (TRRP)", August 2007, <<http://bill.herrin.us/network/trrp.html>>.

~~[TTR-Mobility]~~

Whittle, R. and S. Russert, "TTR Mobility Extensions for Core-Edge Separation Solutions to the Internets Routing Scaling Problem", August 2008, <<http://www.firstpr.com.au/ip/ivip/TTR-Mobility.pdf>>.

Supprimé : TTR

[Vogt-2009]

Vogt, C., "Simplifying Internet Applications Development With A Name-Based Sockets Interface", December 2009, <<http://christianvogt.mailup.net/pub/vogt-2009-name-based-sockets.pdf>>.

[loc-id-sep-vs-ces]

Whittle, R., "Loc/ID Separation is different from Core-Edge Separation", January 2010, <<http://www.firstpr.com.au/ip/ivip/loc-id-sep-vs-ces/>>.

Appendix A. Acknowledgements

Thanks to the following people for their help and encouragement: Juan Jo Aden, Noel Chiappa, Olivier Bonaventure, Brian Carpenter, Dino Farinacci, Vince Fuller, Joel M. Halpern, Geoff Huston, Ved Kafle, Eliot Lear, Simon Leinen, Tony Li, Jeroen Massar, Dave Meyer, Chris Morrow, Dave Oran, Robert Raszuk, Jason Schiller, John Scudder, K. Sriram, Markus Stenberg, Letong Sun, Christian Vogt, Kilian Weniger and Xiaoming Xu.

This is not to imply that these people support Ivip.

I especially thank Steve Russert, formerly of Boeing, for collaborating on the TTR Mobility paper for MobiArch '08. The original draft wasn't accepted and by the time we revised it to the point of being happy with it, the paper was 2.5 times as long as the conference page limit.

Author's Address

Robin Whittle
First Principles

Email: rw@firstpr.com.au
URI: <http://www.firstpr.com.au/ip/ivip/>

