Network Working Group
Internet-Draft
Intended status: Experimental
Expires: September 8, 2010

R. Whittle
First Principles
March 07, 2010

Ivip (Internet Vastly Improved Plumbing) Architecture
draft-whittle-ivip-arch-04.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 "edge" address space which is suitable for multihoming and inbound traffic engineering (TE) to end-user networks of all types and sizes - in a manner which imposes far less load on the DFZ control plane than the only current method of achieving these benefits: separately advertised PI prefixes. Ivip includes two extensions for ITR-to-ETR tunneling without encapsulation and the Path MTU Discovery problems which result from encapsulation - 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 or more IPv4 addresses (or IPv6 /64s) no matter what addresses or access network they are using, including behind NAT and on SPI addresses. TTR mobility for both IPv4 and IPv6 involves generally optimal paths, works with unmodified correspondent hosts and supports all application protocols.

Status of this Memo

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

This Internet-Draft will expire on September 8, 2010.

Table of Contents

1. Introduction

Version 03 (2010-01-13) of this Ivip-arch ID was 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.

This Version 04 includes significant changes to Ivip's mapping system. The DRTM (Distributed Real Time Mapping) system [I-D.whittle-ivip-drtm] removes the need for "Replicators", or for any server to carry the full Ivip mapping database. While DRTM is discussed in this Ivip-arch ID, please see the Ivip-drtm ID for a full description of this system, and how it enables the introduction of scalable routing solutions and global mobility with the initiative and investments being made by organisations which need not be ISPs.

The Ivip (pr. "Eye-vip") project began in June 2007 and in early 2010 is one of the four Core-Edge Separation (CES) architectures being considered by the RRG (IRTF Routing Research Group)

[I-D.irtf-rrg-recommendation] - the others being IRON-RANGER, LISP [I-D.ietf-lisp] and TIDR [I-D.adan-idr-tidr].

For my overall assessment of the proposals submitted to the RRG, and for my arguments for why Ivip is the most suitable for further IETF development, please see http://www.ietf.org/mail-archive/web/rrg/current/msg06162.html ("Recommendation suggestion from RW" 2010-03-04) My discussion of the other proposals can be found in the RRG Archives of January and February 2010.

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 ID is intended for readers who are broadly familiar with the routing scaling problem and RRG discussions and who have, ideally, familiarised themselves with LISP.

This ID 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:

The DRTM (Distributed Real Time Mapping) system

[I-D.whittle-ivip-drtm] describes the new approach to Ivip's real-time mapping system, which uses multiple typically "nearby" full database query servers provided directly or indirectly by MABOCs.

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] .

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

2. Brief description of Ivip

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 "edge" address space which is suitable for multihoming and inbound traffic engineering (TE) to end-user networks of all types and sizes - in a manner which imposes far less load on the DFZ control plane than the only current method of achieving these

benefits: separately advertised PI prefixes.

The new "edge" 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 single ETR (Egress Tunnel Router) address. Micronets have arbitrary starting points and integer lengths - in units of IPv4 addresses or, for IPv6, /64 prefixes.

When an ITR (Ingress Tunnel Router) receives a packets which are addressed to an SPI address. After looking up the mapping of the micronet which covers the destination address, the ITR tunnels the traffic packet to the ETR specified in that mapping - and the ETR delivers the packet to the end-user network.

A Mapped Address Block (MAB) is a DFZ-advertised prefix of global unicast address space which is typically divided up into many separate micronets - such as hundreds to hundreds of thousands of micronets, each of which can be used via any ISP. The total set of all MABs constitutes the "edge" (SPI) subset of the global unicast address range. The remainder is known as "core" space.

A MAB is managed by an MABOC (MAB Operating Company). MABOCs may be end-user networks and the micronets their MABs contain may be used solely for that end-user network - but each micronet can be mapped to any ETR in the world. More typically, MABOCs will lease the SPI space to large numbers of end-user networks on a commercial basis, rather than use it themselves.

The mapping of each micronet is controlled directly by the end-user network which owns or leases the portion of SPI space the micronet is within - or by another organization appointed by this end-user network. Multihoming end-user networks would typically contract a separate company to change the mapping of their micronets, in response to the reachability of their network through their two or more ETRs and according to the network's inbound TE requirements.

DITRs (Default ITRs in the DFZ) are required for handling packets sent to SPI addresses from hosts in networks without ITRs. The one or more DITRs at a DITR site advertise in the DFZ the MABs the site

supports, which is typically a subset of all MABs in the Ivip system.

ITRs other than DITRs request mapping for SPI addresses from local Resolving Query Servers (QSRs) in their own network or in their ISP's network. They may do this directly or through one or more levels of caching query servers - QSCs.

QSRs are caching query servers which query multiple, distributed, authoritative query servers (QSAs) which are typically "nearby", such as within a few thousand km. QSAs are located at a number of widely dispersed sites, such as 5 to 50, where DITRs are located and run by, or for, these MABOCs. Each QSA is authoritative for only a subset of all MABs - the set supported by that DITR site.

Each QSR uses a DNS-based mechanism and an additional protocol to discover two or more typically "nearby" QSAs for each MAB. Since each QSA handles mapping requests for multiple MABs, this means the number of such QSA's each QSR needs to discover is much less than the number of MABs. The number of MABs is much less than the number of end-user networks using SPI space - and the number of micronets is greater than this number, since each end-user network may have many micronets.

End-user networks or their appointees generate real-time mapping changes using facilities provided by the MABOC which manages the MAB the micronet is located within. Most mapping changes will be to change the ETR address of an existing micronet. Other mapping changes will redefine how an end-user network's SPI space is divided into separate micronets. MABOCs will typically charge their customers for each mapping change.

These mapping changes are transmitted in real-time from the MABOC to the organisation which runs the DITR-sites with DITRs which advertise this MAB. The mapping changes are received and incorporated into a real-time updated full mapping database for this MAB, in one or more QSAs at each site. One or more of these QSAs handle mapping queries from the DITRs at the site and one or more handle mapping queries from QSRs in typically nearby ISP and end-user networks. Any QSR can send queries to any QSA, but would normally choose nearby ones. QSAs can give feedback in mapping replies concerning how busy they are, with suggestions of other QSAs to use instead. So there is natural load-sharing with multiple QSAs being spread around the world and dynamic load-balancing between them according to actual loads.

Since no one QSA or DITR-site is required to handle the full set of MABs, since each DITR-site organization controls its own real-time push of mapping to its sites, and since there can be any number of DITR-sites and any number of DITR-site operating companies, there are

no obvious scaling limits on the number of micronets the entire system can handle, or the frequency of mapping updates to those micronets. If a given global set of DITR-sites hits some kind of scaling limit in these respects, then the total load can be handled by more such systems of DITR-sites.

QSRs too can be installed in larger numbers in a busy ISP or end-user network if the query demand exceeds the capacity of one such server. Each QSR can automatically discover very large numbers (tens to hundreds of thousands) of QSAs, and each QSA will typically handle dozens to hundreds or perhaps thousands of MABs.

While there is no assurance of nearby QSAs, MABOCs will generally want to have numerous widely dispersed DITR sites, each with QSAs for two reasons. Firstly to ensure the DITRs tunnel packets without been too far from the path between the sending host (in a network without ITRs) and the ETR. Secondly to encourage ISPs and larger end-user networks to install ITRS and use the QSAs - since this will result in shorter paths for packets and less load on the DITRs.

ISPs and end-user networks do not absolutely need to install ITRs. However ISPs will be motivated to install them (and therefore install several QSRs for them to send mapping queries to) for two reasons. Firstly, to ensure the ISP's customer's SPI-addressed packets are tunneled reliably, rather than relying on DITRs. Secondly, when their customers send SPI-addressed packets to SPI-using end-user networks which are also customers of the ISP, if the ISP has its own ITRs, then these packets do not leave the ISP's network. Without ITRs, they would leave the network via an expensive upstream link, be tunneled by a DITR and return via the same or a different upstream link.

Since end-user networks can run their own ETRs on existing PA address space they get from their ISP, the only thing an ISP needs in order to allow such a network to use SPI space is to accept outgoing packets for forwarding when they have SPI source addresses. All other initiatives and investments - including the provision of multiple widely dispersed DITRs, QSAs and the real-time push of mapping changes to these - is undertaken by the MABOCs who profit by renting SPI space to their end-user customers. A MABOC need not be an ISP.

Ivip includes two extensions for ITR-to-ETR tunneling without encapsulation and the Path MTU Discovery problems which result from encapsulation - 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 or more IPv4 addresses (or IPv6 /64s) no matter what addresses or access network they are using, including behind NAT and on SPI addresses. TTR mobility for both IPv4 and IPv6 involves generally optimal paths, works with unmodified correspondent hosts and supports all application protocols. TTR Mobility is described in: [TTR Mobility]

3. The routing scaling problem and other goals for an architectural enhancement

For a fuller account of my understanding of the routing scaling problem, and other problems which should be considered when devising an architectural enhancement to the Internet, please see http://www.ietf.org/mail-archive/web/rrg/current/msg06099.html ("Scalable routing problem & architectural enhancements" 2010-02-23) and http://www.ietf.org/mail-archive/web/rrg/current/msg06162.html ("Recommendation suggestion from RW" 2010-03-04).

The most visible aspect of the routing scaling problem can be summarised as there being practical problems and unfair cost-burdens due to the growth in the number of PI prefixes end-user networks advertise in the DFZ. Advertising PI prefixes is currently the only method of providing portability, multihoming and inbound traffic engineering (TE) for end-user networks. The same problem exists in principle for IPv4 and IPv6, but only IPv4 has a problem at present.

The less visible part of it is the large number of end-user networks who are unable to gain these benefits due to the costs and other barriers to obtaining their own address space and advertising it in the DFZ. Part of the reason for these costs and barriers is the push-back against this practice, due to concerns about the burden each PI prefix places on the DFZ control plane. Another part is the cost and other difficulties of obtaining the minimum amount of space which can be advertised in the DFZ - currently 256 IPv4 addresses as a /24 prefix.

The burden placed on the interdomain routing system (often referred to loosely as the Default-Free Zone - DFZ) by the prefixes advertised by ISPs 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 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.) Please see http://www.ietf.org/mail-archive/web/rrg/current/msg06163.html ("Geoff Huston's BGP/DFZ research" 2010-03-05) for up-to-date analysis of trends in the number of prefixes and in the problems caused by changes to those prefixes.

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.

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 reduce the number of prefixes the FIB must handle is not really a solution to the problem.

The number of prefixes advertised in the DFZ is the most obvious and directly costly part of the routing scaling problem - analogous to the tip of an iceberg. The larger, harder-to-measure, part of the problem is the unknown number of end-user networks which want or need portability, multihoming and/or inbound TE but which cannot obtain it at present, due to the costs and other barriers to gaining address space and advertising it as PI prefixes.

In order to provide portability etc. 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 in their RIB or FIB. Consequently, the Core-Edge Separation class of scalable routing architectures work by providing a special subset of the global unicast address space, 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. (Core-Edge Elimination architectures have a different approach, which is discussed below in "Architectural Choices - Core-Edge Separation rather than Elimination".

Support for mobility has not generally been considered part of the routing scaling problem. However, mobility is prominently mentioned in the RRG Charter. With the proliferation of cellphones, VoIP, other IP applications it is reasonable to assume that in the future - such as by 2020 - 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.

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. While provision of mobility is frequently assumed not to be related to interdomain routing, it is prominent in the RRG's Charter. The TTR (Translating Tunnel Router) Mobility architecture [TTR Mobility] is a new approach to global mobility, for both IPv4 and IPv6 - and is an extension of a CES architecture such as Ivip.

In the TTR Mobility architecture, 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 whish are currently able to gain and advertise PI space.

4. Summary of Ivip's architectural choices

Ivip is based on some unique architectural choices, including: ITRs (Ingress Tunnel Routers) receiving mapping changes in real-time; typically "nearby" QSA authoritative query servers which are "full-database" for at least one MAB, but typically a significant fraction of all MABs; 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 decapsulated

packets.

The following description assumes that Ivip will be introduced with encapsulation, with long-term migration to MHF. 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.

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.

The adoption of an architectural enhancement to improve routing scalability is frequently assumed to depend largely or entirely on ISPs making the initial investment. However, with DRTM, this need not be the case.

DRTM enables SPI space to be leased to end-user networks - with full support for portability, multihoming and inbound TE for all their communications - with the investment and initiative being taken by organisations which may not be ISPs. These are the MABOCs - MAB Operating Companies - who lease the space in each MAB they control to typically thousands to hundreds of thousands separate end-user networks. An SPI-adopting end-user network can run its own ETR on the existing PA space it obtains from each of its one or more ISPs. ISPs need make no investment to allow this to proceed - but they must forward the outgoing packets from these SPI-adopting networks which have SPI source addresses.

DRTM removes the need for the Replicators, full-database query servers in ISP networks and "Missing Payload Servers" which are described in the more recent ID: Ivip Fast Payload Replication

[I-D.whittle-ivip-fpr]. However, within DRTM, there remains an option to have the caching QSR (Resolving Query Servers) be full database for one, multiple or perhaps all MABs, and to use a small (such as between several nearby ISPs) Replicator system as part of

fanning out mapping updates from DITR-sites to these "full-database" Map Resolvers. DRTM does not currently specify how the organisations which run DITR sites reliably and securely deliver the real-time mapping to each such site. This is an internal matter for these organisations and the potentially multiple MABOCs they receive this mapping information from. It is possible that Replicators could be part of these arrangements too.

With TTR mobility, the MN (Mobile Node) can be in any access network at all, including behind one or more layers or NAT and including being on SPI space in an end-user network which has adopted SPI space. In all cases, the MN needs no support from the network it is currently connected to, since the MN establishes a two-way tunnel to the TTR and sends its SPI source address outgoing packets to the TTR for forwarding. So TTR mobility is a scalable routing solution which requires no investment or support from ISPs, and in which the initiative and investment comes from TTR Mobility companies, which need not be ISPs.

5. Goals

5.1. 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 - where this includes a single MN (Mobile Node) within the definition of "end-user network".

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.

5.2. 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.

There is an assumed upper bound of order 10^7 on the number of non-mobile end-user networks. This is on the basis of a population of 10^10 and there being typically no more than one organization per 10^3 people which needs portability, multihoming and/or inbound TE enough to invest in a second ISP service and whatever else is required to achieve these goals. Brian Carpenter suggested the same thing (http://www.ietf.org/mail-archive/web/rrg/current/msg05801.html 2010-01-27).

Given the growing ubiquity of cell-phones and the desire to give them IP connectivity with mobility, including session survival when changing access networks, it is reasonable to assume an upper bound of order 10^10 on the number of "mobile end-user networks". This order 10^10 upper bound has been discussed on in the RRG and no-one has suggested a routing scaling solution with mobility should aim for any greater number of end-user networks.

In Ivip's mapping system and in its ITRs, no distinction is made between end-user networks which are mobile or non-mobile, so the total number 10^10 is the upper bound on number of micronets for the Ivip system to handle. In IPv4, since the smallest micronet is a single IPv4 address and there are only 3.7 billion global unicast addresses in total, from which the "edge" SPI addresses can be drawn, it follows that for Ivip in IPv4, there can be no more than probably

3 x 10^9 micronets.

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, config files of VPNs and hard-coded (however questionably) into firmware and software. Another example of the need for portability is end-user networks which host services for other organisations - typically their customers - in a way that the IP addresses of the network's hosts appear in the DNS zone files of these other organizations. For the network to have to renumber its network, such as to use PA space from another ISP, would require costly, error-prone and carefully timed updates to zone files of all these other organizations.

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 ISPs. (Outbound TE is simply a matter of sending outgoing packets out whichever ISP link is desired.) Ivip's approach to TE differs from that of other CES architectures. It is potentially finer-grained, more flexible and more able to respond to rapid changes in traffic patterns.

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 concerned about portability when choosing another ISP.

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.

5.3. Modular separation of the control of mapping from the CES architecture itself

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 Ivip ITRs do not need to make any choices between multiple ETRs for the purposes of multihoming service restoration or TE. The non-Ivip CES architectures do not provide real-time mapping to ITRs, and therefore need to have the ITRs perform their own multihoming reachability testing and decision-making, to choose which of several ETRs to tunnel packets to.

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

This appointment of another organization to control the mapping of one or more of an end-user network's micronets would involve a private, flexible, arrangement between an end-user network and the 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 RFCs which define Ivip. With TTR Mobility, the mapping of the micronet which the MN uses would be controlled by the TTR Company, rather than the end-user or the MN itself.

This modular separation of the detection and decision-making functions from the CES architecture is 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 regarding: reachability testing to ETRs, or of networks through ETRs, or with ETRs reporting reachability of end-user networks to ITRs by some means; multihoming failure detection based on these; decisions about how to choose between ETRs to restore service; and 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.

5.4. 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 the 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. The ETR then responds to the ITR and the ITR acknowledges this.

Apart from this, ITRs do not communicate with anything but their local query servers - directly with their local QSRs (Resolving Query Servers) or indirectly with these, via one or more levels of QSC caching query servers. ETRs do not communicate with any part of the Ivip system except for ITRs, and then only for this PMTUD management function.

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

Consequently ETRs and ITRs can be simple functions in existing routers or in standalone servers. 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, unreliable, link such as a wireless link. ETRs must be on conventional global unicast addresses ("core"

addresses) - not on SPI ("edge") addresses. ITRs can be on both kinds of address. Ivip may in the future 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 or SCTP from the ITR, or ITFH, to each query server, with keepalive packets.

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 decapsulate 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 CES 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.

5.5. Maximise the flexibility with which ITRs and ETRs can be located

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 Net through. If there is no such ITR, the packet 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 each end-user network they serve. 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 "core" global unicast address (usually from one of the ISP's prefixes) - not an SPI ("edge") address or behind NAT.

5.6. Mobility

One of Ivip's goals is to support mass adoption of IP mobility, since this will 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.

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

It is frequently assumed that in order for a CES architecture to support mobility, the Mobile Node (MN) must be its own ETR. LISP-MN makes this assumption. So does draft-jen-mapping-00 [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 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 when the MN uses a new TTR. This is desirable after the MN moves a large distance, such as 1000km or more, but it not absolutely needed. An MN can still work with a TTR which is on the other side of the world - albeit with longer latency and greater chance of packet loss.

Although the TTR approach to mobility could be used with other CES architectures, Ivip is a better basis for TTR mobility than other CES architectures such as LISP. None of these other proposals provide a method of ITRs gaining updated mapping within a few seconds, as Ivip does. With Ivip's real-time mapping system, the Mobile Node (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 might take 10 to 30 minutes or more for the non-Ivip CES architectures.

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 CES architectures. However, it is best to consider

TTR Mobility as a natural extension of the basic Ivip 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.

5.7. Elimination of encapsulation and PMTUD problems

When ITRs use encapsulation to tunnel traffic packets to ITRs, 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 a 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. The PTB generated by the router in the tunnel path will not be suitable for the sending host.

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) makes it desirable to either avoid encapsulation entirely, or to introduce Ivip with encapsulation, but in the long-term change 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:

1. ETR Address Forwarding (EAF) - for IPv4. [I-D.whittle-ivip-etr-addr-forw]

2. Prefix Label Forwarding (PLF) - for IPv6. [PLF for IPv6].

If Ivip is introduced with encapsulation, all ITRs and ETRs should 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.

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, there is an increasing probability that that most or essentially all DFZ routers could be upgraded in this way, for very little cost. Initial deployment with MHF is a goal, with the alternative goal being eventual transition to MHF.

For any near-term introduction of Ivip, such as to introduce TTR mobility services or simply to provide SPI space to non-mobile end-user networks, the organizations initiating these services will be unable to have all or perhaps any DFZ routers upgraded in time to start their services. Since these services, especially TTR mobility services, appear to be commercially attractive in the near-term, the most likely outcome is that Ivip will be introduced with encapsulation. If so, it is vital that all ITR and ETR software be updatable so that a future transition to MHF can be performed reliably and completely.

MHF involves some restrictions on the location of ETRs. For IPv4, only 30 bits are available for specifying the ETR address. However, an alternative which I have not yet fully explored is to define a new protocol type with its own header to replace the IPv4 header. In the new header at least 31 bits could be found - and probably 32. If 32 could be found , then the following paragraph would become irrelevant.

This 30 bit MHF ETR address forwarding arrangement is incompatible with the initially desirable arrangement where any "core" address can be used for an ETR. There is further work to do on this problem - but the solution is probably to avoid it with a new header format as noted above. If a large number of end-user networks established their ETRs on a variety of addresses, such as the IP addresses of their existing single PA address services, then it may not be possible to have them alter these addresses in time for the

transition to MHF. For instance, in an extreme case, four separate end-user networks may run four separate ETRs on four contiguous addresses 11.22.33.16, 11.22.33.17, 11.22.33.18 and 11.22.33.19. Yet the current 30 bit IPv4 MHF technique can only tunnel packets to addresses specified with 30 bit precision - which covers all four addresses. A workaround would be for a router at this ISP to perform a second lookup on the destination address of these tunnelled packets and to forward them to the correct service directly.

5.8. No requirement for new host functionality

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 current IPv4 and IPv6 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 foreseeable 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] See also http://www.ietf.org/mail-archive/web/rrg/current/msg06162.html ("Recommendation suggestion from RW" 2010-03-04).

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.

5.9. Full benefits to all adopters irrespective of level of adoption

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

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 (Default ITRS in the DFZ) and were previously known as OITRDs (Open ITRs in the DFZ). 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 - those sent in networks with ITRs. Yet the goal is for all adopters to use the new form of addressing entirely, and so not to have to use the existing unscaleable "advertise PI prefixes in the DFZ" approach to portability, multihoming and TE.

5.10. 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" (CES) architectures. [C-E-Sep-Elim]

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 motivation 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 unscaleable 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.

A CES 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.

CES architectures do not require any changes to hosts - to stacks or applications. They do however involve the creation of at least two items of infrastructure which are typically global in reach, before any end-user network can use the system. Before DRTM, Ivip required a single coordinated global mapping distribution system, though the DITR systems could be operated by or for particular MABOCs (MAB Operating Companies) and need not cope with all the MABS in the Ivip system. Now, with DRTM, there is no need for a single global mapping distribution system. There will be multiple such systems, each handling a subset of the MABs.

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. The DRTM arrangements [I-D.whittle-ivip-drtm] involve MABOCs (MAB Operating Companies) or TTR mobility companies establishing and running (or contracting other organizations to establish and run) multiple DITR-sites. At each DITR-site there are DITRs and QSAs supporting the subset of MABs which are run by the one or more MABOCs the DITR-site is run for.

Please see the DRTM ID for more information on how this system can develop without direct investment by ISPs, with MABOCs taking the

initiative and making the investment in reaching out with DITR-sites to sending hosts in networks without ITRs, and with QSAs at those sites to help nearby ISPs run their own ITRs and QSRs.

5.11. 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.

It is a goal of Ivip 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 CES architecture 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.

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 decapsulated packets where the two differ. (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.)

This also works well with packets tunneled from ITRs inside the ISP network. Please see the section "ETR support for ISP border router source address filtering" in "Recommendation suggestion from RW" (http://www.ietf.org/mail-archive/web/rrg/current/msg06162.html or any later version of this) for a discussion of why it appears to be impossible for LISP ETRs to enforce this BR source address filtering.

This approach - of the ETR dropping inner packets whose source address does not match the source address in the outer header - 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.

5.12. Avoiding the need for any one server to store or receive the complete mapping database

With DRTM, QSAs store the complete mapping database for one or typically many MABs, and so require real-time feeds of mapping updates for those MABs. At boot time, they need to be able to download snapshots of the databases and bring that information up-to-date with the updates sent since the snapshot was made, before the database can be used to answer mapping queries. The same procedure would be executed if the QSA ever lost sync with the feed of mapping updates.

However, there is no requirement that any one QSA handle all the MABs. There is no prohibition of this - for instance if a DITR-site handles every MAB in the Ivip system, this will be perfectly allowable. Its just that the system is intended to work with multiple sets of DITR-sites, with the DITR-sites of each set handling a subset of the MABs. To whatever extent there are scaling limits to the number of micronets a DITR-site and its one or more DITRs and QSAs can handle, this does not pose a problem for the scaling of the entire Ivip system, since the total load can be handled by multiple such DITR-sites. QSRs can handle many sets of DITR sites - so there is no obvious limit to the scaling of the entire system.

Before DRTM, each ISP with ITRs had to install two or more "QSDs" (full database query servers - the term is no longer part of Ivip). These were full-database for all MABs and so required real-time feeds of all mapping updates for all MABs. This presented a scaling problem and an unfair burden on the ISP if its customers rarely or never sent packets to micronets for which a large number of updates were sent, or never sent packets to whole MABs which the QSD still had to store and receive updates for. (These statements about ISPs also apply to any end-user network with ITRs which chooses to install

its own QSR, or previously QSD, rather than use those of its one or more ISPs.)

With DRTM, QSDs are replaced by QSRs - caching Resolving Query Servers. So there is no need for ISPs to maintain a server which is full-database for any MAB. This greatly reduces scaling problems. It will remain an option for a QSR to be full-database for one or more MABs - and in principle for it to be full-database for all MABs, in which case it would function just like the now-obsolete QSD. However, AFAIK, there will be no need to do this - since caching-only QSRs should scale well and cope with the largest imaginable numbers of micronets.

5.13. Eliminating unfair burdens

Prior to DRTM, Ivip had a "non-goal" of eliminating unfair burdens. This was because with full-database QSDs (as discussed above) it could not be ruled out that an ISP would face expenses running its one or more QSDs which in part depended on there being some large number of micronets, or large number of changes to micronets, which the ISP never gained any benefit from - because these did not affect packets sent by its customers.

This unhappy situation is no longer a part of Ivip.

Ivip's goal is to eliminate "unfair burdens", but no scalable routing system is likely to achieve this entirely.

An example of an "unfair burden" which remains with DRTM is that each QSR needs to automatically discover two or more typically "nearby" QSAs for every MAB in the Ivip system. Yet perhaps the QSR and the ITRs which depend upon it will never send packets to some of these MABs. This is unfair, but it is much less of a problem than before DRTM, where the QSD would need to store all the micronets of such MABs and receive all the updates to them as well.

Ivip's goal is to minimise unfair burdens and to eliminate them where possible. It should be able to achieve a huge improvement over the problem which lies at the heart of today's routing scaling problem - the unfair burden imposed on all DFZ router operators by the addition of each PI prefix by any end-user network in the world which is able to obtain the space and advertise it in the DFZ.

6. Non-goals

6.1. Isolation between core and edge networks is not required

At least one CES architecture - APT (which is no longer being developed) - appeared to have a goal of completely separating (really "isolating") core networks from edge networks. 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 has no such goal. For a discussion of my attempt to understand this aspect of APT, and how this may have affected the ways in which some people use and think about the term "Core-Edge Separation", please see: "CES & CEE: GLI-Split; GSE, Six/One Router; 2008 sep./ elim. paper (v3)" (http://www.ietf.org/mail-archive/web/rrg/current/msg06110.html 2010-02-24, or any later version).

6.2. Full adoption not required

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 space, 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.

6.3. Mapping changes need not be free of financial cost

It appears that the designers of other CES architectures have a goal of mapping changes being free of financial cost. This is not a goal of Ivip.

Ivip is the only CES architecture to contemplate or assume that mapping changes will 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.)

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 achieve two outcomes.

Firstly, the payment - which goes to the MABOC - helps the MABOC cover its costs of maintaining multiple DITR-sites, each with their QSA authoritative query servers. Each such change involves data transmission to these sites and may involve QSAs sending Cache Update commands to queriers (QSRs) to which mapping for the micronet has "recently" been sent in a map reply message, or in a Cache Update message. This is fully described in the DRTM ID.

It is also vaguely possible that if there are really large numbers of updates, ISPs and other networks with ITRs and QSRs may object to handling all these Cache Updates without some payment by the MABOC from whose QSAs they are sent. So it is vaguely possible that MABOCs may need to use some of these fees to encourage ISPs to accept these Cache Updates. Such frequent updates are most likely to arise from end-user networks doing short-timescale inbound TE changes - and they will do this as long as the cost of the mapping changes is lower than the benefit they derive from the inbound TE, which may be substantial.

Secondly, this fee per mapping change inhibits end-user networks from making so many mapping changes unless they have a suitably strong reason to do so. This will lighten the load on the MABOC's systems, including especially the DITRs and QSAs it either runs, or pays another organization to run.

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

6.4. 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 CES architectures. 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 message from their QSC or QSR 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 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 when the end-user network is reachable via both ETRs - 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 the end-user network via ETR-A, then Ivip cannot ensure complete connectivity, since the end-user network is not reachable to all hosts in all networks through just one ETR or the other. (Actually, practical connectivity only concerns the fraction of DFZ routers and other networks with hosts 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 CES architectures such as LISP 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 (or, if LISP is ever able to do this: determine the reachability of the end-user network through the two ETRs) and adapt their tunneling by choosing an ETR which enables the packets to get to the end-user network. In this circumstance, the non-Ivip CES architectures 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 so from anywhere in the Net, at least one was reachable - is likely to be a 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 such a situation.

Nonetheless, 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. This is because with Ivip, multihomed end-user networks will typically contract another company to continually probe the reachability of their network through their two or more ETRs - and that company will do so from a finite number of servers in particular

parts of the Net. There may be an outage affecting ITRs which are handling packets addressed to this end-user network which does not affect the set of servers the multihoming monitoring company is using - so that company will not detect the problem affecting these traffic handling ITRs. In that case, the non-Ivip approach would be superior - if the non-Ivip ITR could detect the outage and correctly chose another ETR through which the end-user network was reachable.

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 of the end-user network via various ETRs. This is particularly the case if tens or hundreds of thousands of ITRs are tunneling to one ETR. Such non-Ivip ITRs may not actually probe 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 (again for non-Ivip ETRs) is that ICMP messages may be lost or may not always be generated if there is an outage. Furthermore, it would be costly for these ITRs to be able to securely distinguish genuine ICMP messages from spoofed ICMP messages.

6.5. 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 DITR-site systems of DITRs, QSAs and the mapping distribution systems inside each system or DITR-sites for IPv4 and IPv6 are intended to be separate and independent - but there's nothing to prevent one server being used for both the IPv4 and IPv6 systems.

6.6. 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 objects of different types, usually called Locator and Identifier, which therefore are in different namespaces. The Locator is usually regarded as an "address" but the Identifier is not.

If both types of object 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 (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 CES 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 Core-Edge Separation. ITRs are the only devices which treat packets differently if their destination address is in the "edge" subset of the global unicast address range.

The full arguments about why Core-Edge Separation cannot correctly be construed as "Locator / Identifier Separation" are at: [loc-id-sep-vs-ces]. For further discussion and why LISP is misnamed, please see the following RRG messages from early 2010: msg05864, msg05865, msg06110 and msg06190.

7. Architectural Choices

7.1. Core-Edge Separation rather than Elimination

7.1.1. Core-Edge Elimination (CEE) architectures

Core-Edge Elimination (CEE) involves hosts dealing with two kinds of entity for dealing with other hosts and to write into packet headers in order that they will get to their desired destination: Identifiers and Locators. The simplest adaptation of existing protocols is to retain IP addresses as Locator addresses and develop a separate namespace for the Identifier addresses. Some CEE architectures only modify the stack of each host, and use unmodified IPv6 applications. Other require modified stacks and applications.

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 CEE architecture - the number of PI prefixes advertised in the DFZ by multihoming 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 (or perhaps its applications) 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 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 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 - unless the system supports unmodified applications in some way.

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 - so substantial benefits to adopters only occur when all, or essentially all, hosts have been upgraded to 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. Core-Edge Elimination architectures typically do not apply to IPv4 and so are based on IPv6 or on entirely new arrangements. If CEE was used for IPv4, it would not be practical due to the inherent inefficiency of its use of global unicast address

space. In IPv4, any end-user network which needs a /24 of

address space for its hosts would require a /24 from each of its upstream ISPs. So all multihomed end-user networks would consume at least twice the space they need - which is not practical with IPv4's address shortage.

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 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 the full set of Locators which can be used for a host with a given Identity - will contribute to delays in sending application packets. (All CEE architectures to date involve global query server systems with just one or a few authoritative query servers. None involve "nearby" or "local" authoritative query servers, which is the only way to avoid excessive delays and risks of packet loss.)

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 Core-Edge Separation approach.

7.1.2. Core-Edge Separation (CES) architectures

Ivip uses a Core-Edge Separation (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. The addresses which remain once this new, scalable, "edge" subset of the global unicast space is separated out is known as "core" address space.

(In LISP, the "edge" subset is known as EID (Endpoint Identifier) and the remainder is known as RLOC (Routing Locator). However it is a mistake to think of these as being "Identifiers" and "Locators" or to think that LISP has anything to do with the Locator / Identifier Separation naming model.)

This subset will consist of a growing number of prefixes, each of which is known as a MAB (Mapped Address Block). Each MAB is advertised in the DFZ by as many DITRs as are at DITR-sites which support this MAB. (The QSAs at those sites are also authoritative query servers for the MABs the site supports.)

Within each MAB, 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 - 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" space will grow to become a significant fraction of the total global unicast space. There must always be some conventional, "core", 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.

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 (unscaleable) 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. "Separation" in "Locator/Identifier Separation" means separate namespaces. Only CEE architectures implement "Locator / Identifier Separation".

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 applies 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.

7.2. Nearby authoritative query servers

Probably the greatest challenge for a CES architecture 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 four 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. ITRs in an ISP network (or in an end-user network) send queries to local caching query servers - directly to a QSR or indirectly via one or more levels of QSC. Both these types of server are caching query servers and are "local" in that they are in the same ISP network, or if the ITR is in an end-user network are either in that end-user network or in the networks of its one or more ISPs. QSRs are the interface between the ITRs and the authoritative query servers which are not local - but which are typically "nearby". (See following text of a definition of "nearby".)

4. 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.

The only architecture to propose option 1 was 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]

APT used option 2. Ivip before DRTM (that is, before March 2010) also used option 2 - the local full database query servers were called QSDs. In APT, they were also called Default Mappers, and also handle the encapsulation of some packets.

Ivip with DRTM uses option 3. The definition of "nearby" follows shortly.

All other CES architectures to date use option 4. The most prominent examples are LISP-CONS [I-D.meyer-lisp-cons], LISP-ALT and TRRP.

In option 3, "nearby" means something like within a few thousand km. In fibre, 200km involves approximately 1ms delay. So if the authoritative query server is 2000km away, the propagation delay in SiO2 sets a lower bound to the response time of 20ms. It is assumed that if the ITR buffers any packets it has no mapping for but gets the mapping within some time like 50ms or perhaps 100ms, then this constitutes an insignificant delay in the establishment of initial communications for all applications and human users. Therefore, "nearby" means close enough not to involve significant delay or risk of packet loss. "Typically nearby" means that except for unusual error conditions - assuming MABOCs are looking after the interests of their SPI-leasing customers well, by placing multiple DITR-sites with

their QSAs in widely spread locations around the Net - that ITRs will usually be able to send packets within 50ms or so, which is assumed to be an insignificant delay.

QSC, QSR and QSAs will all have response times, but it is reasonable to assume these will normally be a few ms, considering the enormous four-core 3GHz clock CPU power which inexpensive COTS servers now possess.

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.

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]

Typically "nearby" full-database query servers is the clear choice for Ivip because ITRs will normally not delay any packets to a

significant degree and because this system avoids the avoid scaling problems which arise from any server being required to store the full mapping database of all MABs, and the need for a single, coordinated, mapping distribution system to drive these servers.

7.3. Real-time mapping distribution

By getting mapping changes to all ITRs which need it (all ITRs handling packets addressed to the micronet whose mapping just changed) 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 externalises the reachability, multihoming service restoration and TE decision making systems from the CES architecture itself.

7.4. SPI address 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 IPv6/64s.

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 at DITR-sites which support this MAB. These DITRs attract 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. In this case, the end-user network is the MABOC of this MAB, and it does not lease any of the space to any other organizations.

Most MABs will be operated by MABOCs which are specialised companies - perhaps ISPs but not necessarily. The MABOC typically acquires rights to multiple prefixes of global unicast space, advertises each of them in a global system of DITRs and then leases out smaller portions of the MABs, on an annual basis, to a large number of end-user networks.

Each end-user network leases a section of the MAB - a User Address Block (UAB). One end-user network might lease 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 (MM) company and give them the credentials required to control the mapping via whatever mechanism the MABOC provides.

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 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 MABOC 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 not described at present, but as the Ivip system comes closer to being standardized, it would be desirable to provide a standard protocol or interface by which end-user networks or their appointees could issue mapping changes, rearrange the division of UABs into micronets etc. Also, it would be desirable to have a standardised way that an end-user network could allow its appointee to control the mapping for individual micronets within its UAB. If this was universally adopted by all MABOCs, then multihoming monitoring systems would only need to work with this one system for controlling the mapping of micronets.

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

The MAB company would charge fees for leasing the UAB space, and for the load placed on the DITRs which cover this MAB. The MAB company may run its own DITRs - and their associated QSAs - or may contract this out to another company which specialises in this service. 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. Likewise, the load-sharing between the QSAs at these sites, and the desirability of having QSAs "nearby" to the QSRs in ISP and other networks all over the world.

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.

7.5. 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.

7.6. 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.

7.7. 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 address and the outer header having a source address which is allowable. This enforcement is achieved by the ETR performing simple logic on each decapsulated 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.

7.8. IPTM (ITR Probes Tunnel MTU) PMTUD 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 CES 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, CES 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.

8. Architectural Elements

8.1. ITRs

8.1.1. 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. 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.

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 true router (hardware or software) it my 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.

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 (QSCs or QSRs) and for receiving PTB messages.

Here is a description of what happens when a sending host in an ISP network, such as a QSC or QSR, 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 QSC or QSR doesn't have an ITFH installed) 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. As just noted, this can be on conventional "core" space or "edge" (SPI) space.

8.1.2. DITRs - Default ITRs in the DFZ

DITRs are "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: "Default ITRs in the Default-Free Zone".)

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, meaning the packet 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 ultimate beneficiaries of DITRs are the end-user networks which the packets are addressed to - and these are the customers of the MABOCs who lease the space to them (except where the one end-user network runs a whole MAB for itself, and so is its own MABOC).

The most likely arrangement for DITRs is that the MABOCs who lease SPI space to end-user networks 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 MABOCs who are paying the operator for this service. MABOCs 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. The one or more QSRs they rely on will determine the current set of MABs by the DNS-based mechanism described in the Ivip-

drtm ID. Changes to this set will also need to be propagated to all QSCs and ITRs in the local system, by a mechanism which is yet to be designed.

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 or Internet exchanges, close to one or more full database query servers. DITRs advertise MABs to their neighbouring BGP routers, and have a default route to either one of these routers, or have the full set of DFZ routes with links to multiple neighbouring routers. So (unless they are implemented behind a suitable BGP router) DITRs are BGP routers and may or may not be "DFZ" routers, depending on how they forward their outgoing packets.

8.1.3. 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.

8.1.4. Encapsulation and PMTUD 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.

8.1.5. 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 IPv4 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.

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.

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.

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 CES architectures.)

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 thousand for a busy web-server. 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 - a QSC or a QSR. 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 may be 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.)

In the future, caching times will be discussed fully in the Ivip-drtm ID.

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. The ITR can also be sent a Cache Update message to the effect of flushing the cached mapping for a given micronet - by a QSC or QSR which previously sent mapping for this micronet in a Map Reply message. This is needed if the micronet has been deleted from the system, such as due to the end-user network changing the way their UAB is divided into micronets. If the ITR receives a packet with a destination address which matches this micronet, then there will be no cached mapping and the ITR will request mapping - and so gain the mapping for whatever micronet this address is now within.

At any time when a mapping is cached, the ITR may receive a Cache Update message from a QSR or QSC which previously sent it mapping for this micronet. The Cache Update message, like the Map Request query and the Map Reply message, will be a UDP packet. The Ivip-drtm ID has more information on these messages and their acknowledgement.

The Cache Update will be secured by the nonce sent by this ITR in its original Map Request query which resulted in the QSC or QSR sending a Map Reply (also secured by that nonce) which specified the start and end address of this micronet, and its mapping (the ETR address).

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. As noted above, the final kind of Cache Update is a command to flush the mapping cached for this micronet. This could be encoded via special flags, but it may be simpler, for instance with IPv4, to define a particular

ETR address such as 0.0.0.1 as meaning the mapping should be flushed.

None of these Cache Updates reset the caching time. So ITR's cached mappings will time out as usual, no matter how many Cache Updates have arrived to alter the ETR address stored in this mapping. If the Cache Update message from the query server reset the caching timeout process, then continued Cache 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 nearby QSA.

8.1.6. 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 QSCs. I do not suggest that QSRs should have to maintain sessions with such ITFHs. TCP with a keepalive might do, but SCTP would probably be much better. Then, instead of UDP mapping queries, replies and updates, the same

messages would be sent over SCTP. It is not out of the question to link all ITRs, QSCs and QSRs with SCTP, rather than use UDP packets, since the SCTP will ensure reliable delivery of messages, and so reduce the complexity of the code for sending receiving and acknowledging messages.

8.1.7. 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 - whichever two or more QSCs or QSRs the ITR or ITFH is supposed to send its Map Request queries to. 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.

Likewise, it would be desirable for QSCs to be able to automatically discover the upstream QSCs or QSRs they should send their Map Request

queries to.

8.2. ETRs

8.2.1. In servers or dedicated routers

The ETR function can be performed in a dedicated router or in a server 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.

8.2.2. 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.)

8.2.3. 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 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 of some kind, even if implemented on a server, since it would also be deciding which link to send outgoing packets on.

8.2.4. MHF ETR functionality - EAF and PLF

If Ivip is introduced with encapsulation, its ITR and ETR functions will contain Modified Header Forwarding 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. (But see previous discussion about how this could probably be redesigned to involve a new header type which would allow 31 or 32 bits to be used.) 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 - 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 (Border Routers, facing other ISPs and transit networks) 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.

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.

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.

8.2.5. 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 decapsulate 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 involves significant complexity in the ETR, but does not involve storing state for more than a few seconds.

Once the traffic packet has been decapsulated, 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 RFC 1191 PMTUD will proceed just as it would if there had been no ITR to ETR encapsulation.

8.3. QSRs - Resolving Query Servers

Please see the Ivip-drtm ID for a description of QSRs.

8.4. 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, QSRs and QSAs would share some common components. A QSC receives and responds to Map Request queries from ITRs or other QSCs in the same manner as a QSR. The QSC sends Map Request queries to a QSC or QSR in just the same way as was described above for an ITR - and likewise receives Map Reply and Cache Update messages the upstream QSC or QSR 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 whose ITRs and QSCs use the QSCs and QSRs 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 Map Request 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 QSRs. 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 QSR, 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 investment in expanding the capacity of a single QSR 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 QSR which both QSC1 and QSC2 query will be significantly reduced. This is because it will sometimes or 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 QSR - and its querier cache (where the QSR retains records of the mappings it sent in Map Replies to various queriers, along with the caching time variables and the nonce which it was sent in the initial Map Request) would be correspondingly larger. Furthermore, if more than one of QSC1's ITRs is caching mapping for a micronet for which the QSR receives a Cache Update, then the QSR 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.

There needs to be coordination between the caching times of Map Replies sent out by a QSA and those sent out by dependent QSRs and QSCs.

Also, each querier needs to periodically check that any upstream query server it is caching any mapping from is still alive and has not been rebooted. If the upstream server has died or been rebooted, there is a danger that the cached mapping in the querier should have been changed or flushed due to a Cache Update message which the upstream server would have sent if it had not died or been rebooted. This is for further work.

While the exact details are TBD, it is clear that it will be possible to define relatively straightforward protocols by which ITRs, optional QSCs, QSRs and QSAs can be combined to efficiently support the mapping needs of many ITRs per QSR.

8.5. MHF - Modified Header Forwarding

8.5.1. EAF - ETR Address Forwarding for IPv4

Please see [I-D.whittle-ivip-etr-addr-forw] and the discussion above in the ITR section. To-do - rationalise the various mentions of MHF and especially EAF in this ID.

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. (But see previous notes on how with a new protocol number a new header could

carry 31 r probably 32 bits of ETR address.) 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 (32?) 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.

If it takes 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 CES 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.

8.5.2. 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-user) network.

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, as discussed below.

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.

8.6. 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 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) -> ITR-B ==> TTR-B ==> MN-B.

Any such systems should be designed to 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.

9. 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.

Ivip ETRs easily enforce ISP BR source address filtering. For LISP ETRs to enforce this would be at least administratively complex and very expensive for large numbers of filtered prefixes - and it may be impossible to do while allowing for ITRs in the local ISP network tunneling to this ETR.

10. IANA Considerations

[To do.]

11. 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-06 (work in progress), January 2010.

[I-D.irtf-rrg-recommendation]
Li, T., "Recommendation for a Routing Architecture", draft-irtf-rrg-recommendation-05 (work in progress), February 2010.

[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-drtm]
Whittle, R., "DRTM - Distributed Real Time Mapping for Ivip and LISP", draft-whittle-ivip-drtm-01 (work in progress), March 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-fpr]
Whittle, R., "Fast Payload Replication mapping distribution for Ivip", draft-whittle-ivip-fpr-01 (work in progress), March 2010.

[I-D.whittle-ivip-glossary]
Whittle, R., "Glossary of some Ivip and scalable routing terms", draft-whittle-ivip-glossary-01 (work in progress), March 2010.

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

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>.

[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/>.

[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/>.

[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>.

[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/