Ivip4 ETR Address Forwarding
draft-whittle-ivip4-etr-addr-forw-01.txt

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with Section 6 of BCP 79.

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

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

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

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

This Internet-Draft will expire on February 23, 2009.
ETR Address Forwarding (EAF) is a novel method by which an IPv4 Core-Edge Separation solution to the Internet’s routing scaling problem can tunnel packets from an ITR to an ETR. EAF involves using 31 bits of the IPv4 header for new purposes: bit 48, the More Fragments flag, the Fragment Offset field and the Header Checksum field. Consequently, packets in this format need to be handled by routers with modified functionality. EAF is an alternative to encapsulation (map-encap) or address translation (Six/One Router) and has advantages including: simpler ITRs and ETRs, direct support for conventional RFC 1191 PMTUD, no encapsulation overhead and full compatibility with IPsec AH and Traceroute.
1. Quick note on other uses of bit 48, Flag 0, Evil bit

Within hours of version 00 of this I-D appearing, Fred Templin alerted me to other proposals for using bit 48 of the IPv4 header.

Bob Briscoe and colleagues have an extensive I-D regarding "re-ECN" - a new approach to Explicit Content Notification. This I-D has been in active development since 2005: "Re-ECN: Adding Accountability for Causing Congestion to TCP/IP draft-briscoe-tsvwg-re-ecn-tcp-06": [I-D.briscoe-tsvwg-re-ecn-tcp].

Two other proposals regarding bit 48 are cited in briscoe-tsvwg-re-ecn-tcp. Firstly, a 2005 BT Journal paper by J. Adams and colleagues and secondly RFC 3514 which I discuss below. The aims of the J. Adams et al paper would apparently be met by the proposal of Bob Briscoe and colleagues.

Fred also mentioned that some ca. 2002 proposals from Jim Fleming proposed new uses of IPv4 header bits, and setting bit 48 to 1. Searching for "Jim Fleming" with "IPv8" and "IPv16" turns up a number of mailing list discussions, but I can find no sign of ongoing development of these proposals from ca. 1998 and 2002.

The re-ECN proposal is substantial and active. Please let me know of any other proposals concerning bit 48.
2. Introduction

The Internet’s routing and addressing scaling problem, as discussed in RAWS [RFC4984] has given rise to a number of Core-Edge Separation (CES) proposals which create a new form of address space for end-user networks. The term SPI (Scalable Provider Independent) space is used here to describe this form of address space.

In the absence of a successful CES system, most of demand driving the unsustainable growth in the number of advertised prefixes is likely to arise from hundreds of thousands or millions of end-user networks wanting their own conventional BGP advertised PI prefix - since this would remain the only method by which each such end-user network could be multihomed and by which its address space could be portable between ISPs.

SPI space is portable between different ISPs (providers) and can be used for multihoming - including the use of inbound Traffic Engineering (TE) to control the balance of ingress traffic over the links from multiple providers.

The purpose of the CES system is to make SPI space highly attractive to user networks of all sizes, and to ensure that their use of this space is highly scalable. Only if the new type of space is attractive to the great majority of end-user networks will it be possible to ensure that most such networks choose to use this space, rather than the conventional BGP approach to PI space, which is the main cause of the routing scaling problem.

To be scalable, each SPI end-user network’s address space must not be a separately advertised BGP prefix. More generally, the interdomain routing system with the CES extensions should be able to scale gracefully to millions and perhaps billions of SPI-based end-user networks.

CES systems achieve this goal be retaining provider prefixes in the existing core routing system, whilst excluding individual SPI (edge) prefixes. This gives rise to the term Core-Edge Separation scheme. An early use of this term was in a taxonomy by Jari Arkko in July 2008 [Arkko-2008a] [Arkko-2008b] [Arkko-2008c].

The first CES proposals were known as "Locator / ID separation" and/or "map-and-encaps" (map-encap) systems, and include LISP [I-D.farinacci-lisp], APT (A Practical Transit Mapping Service) [I-D.jen-apt], Ivip4e and Ivip6e [I-D.whittle-ivip-arch] and TRRP (Tunneling Route Reduction Protocol) [TRRP]. These involve ITRs (Ingress Tunnel Routers) tunneling a packet to an ETR (Egress Tunnel Router), typically in another provider network, where the ETR...
forwards the packet to the destination end-user network. Map-encap systems suffer from problems of encapsulation overhead, this overhead causing more PMTU problems, and with problems of complexity and extra management traffic in order that ITRs and ETRs can overcome the barriers encapsulation presents to RFC 1191 PMTUD [Ivip-PMTUD-Frag].

A second class of CES system is "Translation", in which the equivalents of ITRs send packets to the equivalents of ETRs (both known as Translation Routers) by rewriting their source and destination addresses. There is no encapsulation, and the packets with translated addresses are forwarded across the core by ordinary BGP routers. To date, the only instance of the Translation class is Six/One Router [SixOneRouter] - which is not practical for IPv4. Translation schemes have the great advantage that the packets are not made any longer. A translation scheme involves less problems with PMTU limits than an encapsulation scheme, and in principle may be able to support full RFC 1191 [RFC1191] PMTUD without excessive complexity in the translation routers.

ETR Address Forwarding (EAF) is a separate class of CES proposal from map-encap and translation schemes. It is somewhat related to Prefix Label Forwarding (PLF) [Ivip6f] - an IPv6-only system which also involves using bits in the existing IP header in a novel fashion. As with Translation, there is no encapsulating header so there is no overhead or worsening of PMTUD difficulties. Unlike Translation, the source and destination addresses are unaltered.

A brief description of EAF is that the ITR sets bit 48 of the IPv4 header to 1 and 30 other bits of the header (currently used for the More Fragments flag, the Fragment Offset field and the Header Checksum) are then used to carry the 30 most significant bits of the ETR’s address. Routers (core BGP routers and potentially internal routers) with modified functionality recognise the EAF formatted header, and forward the packet to the ETR according to this address, rather than according to the destination address. The ETR converts the header back to its normal state and delivers the packet to the destination network. Routers sending ICMP messages, such as Packet Too Big or other messages in response to traceroute, perform a similar conversion on the modified header to reconstruct the IPv4 version of the packet - so PMTUD works directly for all routers between the ITR and the ETR.

2.1. Advantages

EAF has the following advantages over map-encap:

1. There is no transmission overhead - the packet is not made any longer.
2. Conventional RFC 1191 PMTUD is supported over the entire path, including between the ITR and ETR, without any involvement of the ITR.

3. Traceroute is expected to work over the entire path.

Avoiding encapsulation overhead is a major long-term benefit, especially for VoIP packets.

The retention of conventional RFC 1191 PMTUD through the ITR to ETR path removes the need for a great deal of complexity which would otherwise be required in map-encap ITRs and ETRs. While that complexity could be implemented in ITRs and ETRs, it will be highly desirable to avoid this, and the consequent need for the protocols, occasional probe packets, reply packets etc. A discussion of the arrangements necessary to handle PMTUD problems [Ivip-PMTUD-Frag] gives some idea of the extra complexity, state etc. required in ITRs and ETRs for any practical map-encap system.

A survey of higher level protocols’ references to bits within the IPv4 and IPv6 headers [Checksums] shows that none of these protocols depend on the bits which are changed in the EAF formatted IPv4 header.

2.2. Disadvantages

The major disadvantage of EAF is the need to upgrade the capabilities of most or all core routers, and likewise any internal routers between ITRs or ETRs and the border routers. However, since the modified functionality is relatively slight compared to the current functionality of these routers, it seems likely that many existing routers could be upgraded with a firmware update.

As discussed below, EAF ITRs do not accept fragmented packets, or fragmentable packets longer than some globally agreed constant, which would be somewhat below 1500 bytes. However, identical restrictions would apply to a map-encap scheme which properly handled PMTUD problems [Ivip-PMTUD-Frag].

RFC 1191 PMTUD will have been available for 20 years by the time any practical scalable routing system is deployed. The design of such a system should not be constrained by hosts which avoid RFC 1191 and instead burden the network with the task of fragmenting packets which exceed the PMTU to the destination host - and which likewise burden the destination host with the reassembly task.

Care must be exercised to ensure that packets with the new EAF format IPv4 headers are only forwarded to upgraded routers, or to an ETR.
Any other device will regard the header as malformed, and could be expected to drop the packet without generating any ICMP message.

EAF provides only the 30 most significant bits of an ETR address. This means that ETRs can be located no more densely than one every 4 IP addresses. This is not expected to be a problem, since it is common for routers to each use a /30 prefix.
3. Ivip4 with or without encapsulation

There are three notional approaches to Ivip for IPv4: Ivip4e (encapsulation only), Ivip4ef (encapsulation at first, transitioning to EAF in the long term) and Ivip4f (EAF only). Ideally, it would be possible to introduce Ivip4f - so avoiding the need for many complex functions in ITRs and ETRs. If this was not possible, then Ivip4ef would be introduced using encapsulation, but with all ITRs and ETRs ready to switch to EAF as soon as the routers have been upgraded. The long term benefits of no encapsulation overhead and straightforward PMTUD are very significant.

Since the EAF system is quite simple, it would make no sense to introduce Ivip4e - with encapsulation, but without all ITRs and ETRs having a full EAF functionality. In the long term, the cost of upgrading all routers approaches nil, and the cost of including EAF functionality in all ITRs and ETRs will be very low - so a properly designed system should be able to transition to using EAF exclusively whenever the core and internal routers are able to support it.
4. Fragmented and Fragmentable Packets

EAF involves some important restrictions on the packets which will be accepted by ITR for tunneling to an ETR.

Fragments will not be accepted. Neither will be fragmentable packets longer than some globally agreed limit - MinCoreMTU - likely to be set at about 1470 bytes. The same restrictions would be necessary for encapsulation in order to properly manage PMTUD problems and to avoid inefficient traffic patterns such as from fragmenting 9k packets into smaller than 1500 byte fragments [Ivip-PMTUD-Frag]. There are two reasons for these restrictions.

Firstly, it is undesirable for the network in general and the ITR-ETR system in particular to have to carry fragmented packets. All host operating systems support RFC 1191 (1990) PMTUD and 18 years later, applications can reasonably be expected to use it - rather than sending out packets as long as the next hop MTU and expecting routers and the receiving host to do extra work if the PMTU to the host is less than the packet’s length. In the mid-1990s, fragmentation in the network was judged unworthy of inclusion in IPv6. Since a core-edge separation system for IPv4 may be in operation indefinitely, since it would be much harder to engineer such a system to cope with fragmented and fragmentable packets, and since few hosts currently send fragmentable packets longer than the proposed ~1470 byte limit, it is best to simplify the system design by setting these limits on the types of packets it handles.

Secondly, it is technically impossible to implement EAF with ITRs accepting fragments, or with the packets sent by ITRs being fragmented en-route to the ETR.

The EAF system will carry fragmentable packets which are no longer than MinCoreMTU. After these packets emerge in their original form from the ETR, they may be fragmented en-route to the ETR.

Currently it is common for fragmentable packets to be sent across the Internet’s core. For instance, Google servers typically send TCP packets as long as the mutually negotiated MSS value allows, with the Google server offering an MSS of 1430. This can result in the servers sending DF=0 packets of up to 1470 bytes long.
5. Choice of MinCoreMTU value

Ivip ITRs will drop incoming fragmentable packets which exceed a particular length – MinCoreMTU. The value of this global constant must be chosen with some care.

The primary or sole reason for making it a high value is the desirability of rejecting fewer fragmentable packets.

The primary reason for making it lower is because all ITRs and ETRs must be located such that every ITR has a PMTU to every ETR of at least MinCoreMTU bytes.

At present, it is common for there to be 1500 byte MTU limits at many locations in the core of the Internet. It is also common for servers to be connected with 100Mbps Ethernet switches, even when they have Gigabit Ethernet interfaces. One reason for this is the simple expedient of limiting each server’s ability to create peak outgoing packet rates which would overload upstream links. 100Mbps Ethernet switches typically impose a 1500 byte MTU, while an Ethernet switch operating with all ports running at 1Gbps typically enables an MTU on all links of ~9k bytes.

It is highly desirable to maximise the flexibility with which ITRs and ETRs can be located. The deeper they can be located in networks – the further from the border routers – the more numerous they can be and so the better the total load can be shared amongst them. Furthermore, it will be possible to implement the ITR function in the sending host. This is likely to involve zero incremental cost, assuming the host has sufficient CPU and memory capacity to spare. Such ITFH (ITR Function in sending Host) hosts cannot be behind NAT. They may be on conventional addresses or on the new kind of SPI address.

Even in the long-term future, it is safe to assume that there will be 1500 byte MTU limits in some part of the Internet or edge networks through which ITR to ETR packets will travel. For the foreseeable future, MinCoreMTU must be at or more likely somewhat below 1500 bytes. It is unlikely that a higher value could be contemplated in the future, and by then, hopefully, there would be no impetus to alter it, since (hopefully) there would be no remaining applications still sending fragmentable packets across the core.

Choosing a lower value, such as 1400 bytes, would make for few restrictions in the location of ITRs and ETRs, but would cause more trouble with hosts which currently send DF=0 packets longer than this. The task is to find a value which is as high as possible without unreasonably restricting the potential locations of ITRs and
ETRs. Ideally, the value would not constrain existing widespread use of fragmentable packets.

The final decision about MinCoreMTU’s value will be made some years in the future. For now, it will be assumed to be 1470 bytes.
6. Changed bits in the IPv4 header

The IPv4 header is defined in RFC 791 [RFC0791].

<table>
<thead>
<tr>
<th>Version</th>
<th>IHL</th>
<th>Type of Service</th>
<th>Total Length</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Identification</td>
<td>Flags</td>
<td>Fragment Offset</td>
<td></td>
</tr>
<tr>
<td>Time to Live</td>
<td>Protocol</td>
<td>Header Checksum</td>
<td></td>
</tr>
<tr>
<td>Source Address</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Destination Address</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Options</td>
<td>Padding</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 1: IPv4 header format.

When the header is modified to the EAF format, only bit 48, bits 50 to 63 and bits 80 to 95 are altered.

<table>
<thead>
<tr>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>6</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>M</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>c</td>
<td>c</td>
</tr>
<tr>
<td>Header Checksum</td>
<td></td>
</tr>
</tbody>
</table>

Figure 2: IPv4 header bits altered in EAF format.
Bit 48 is set to 1. Bits 50 to 63 and 80 to 95 are set to the 30 most significant bits of the ETR address.

```
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   | D |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 1 | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| 0 1 | Flags | 14 Bits of address to which packet will be Forwarded |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| 16 Bits of address to which packet will be Forwarded |
```

Figure 3: IPv4 header bits as used in EAF format.

ITRs will write bits 0 to 29 of the ETR address into the F bits shown in Figure 3 in an order TBD.

6.1. The evil bit 48

RFC 791 defines bit 48 as Flag Bit 0: "reserved, must be zero".

RFC 3514 [RFC3514] redefines this bit as the "evil" bit "E": "Benign packets have this bit set to 0; those that are used for an attack will have the bit set to 1."

However - while the Internet is awash with packets which upon close inspection are found to be of questionable moral integrity - RFC 3514 has not been widely implemented.

We therefore retain RFC 3514’s nomenclature of "E" and define new semantics for bit 48 (Flag 0) as the EAF flag: setting it to 0 to denote the conventional IPv4 header format and to 1 to denote the EAF usage of the abovementioned bits.
Currently-assigned values are defined as follows:

0  The packet’s header bits conform to RFC 791 semantics.

1  The packet’s header bit 50 to 63 and 80 to 95 contain the 30 most significant bits of an address which modified routers will use to determine forwarding, rather than the destination address. In addition, the header will be subject to special processing as described below to restore its state to that of a conventional IPv4 header, in the event that it is either received by an ETR or to be included in an ICMP message.

6.2. More Fragments flag and Fragment Offset

As mentioned above, ITRs will drop all packets which are fragments. Consequently, the states of the More Fragments flag (bit 50) and the Fragment Offset bits (bits 51 to 63) are all zero when the packet is accepted to be tunneled to an ETR. Whenever the packet is converted back to the conventional IPv4 header format, all these bits will be set back to zero.

6.3. Header Checksum

Since the EAF format is used for the packet’s travel from the ITR to the ETR, and since the IPv4 Header Checksum bits are used for another purpose, the routers in that path - all of which must have the modified functionality which recognises the EAF formatted header - will not check the header’s integrity via any checksum facility.

Significant problems due to transmission processing errors are not anticipated, primarily due to the low error rate in most L2 layer implementations. These low error rates were presumably considered when the IPv6 header was designed without any header checksum. The possible bit error scenarios are discussed in Appendix 1.

The checksum will be recalculated by the ETR when it converts the packet to normal IPv4 format, or by a router en-route to the ETR which needs to convert the packet to this format in order to send an ICMP message to the sending host.
7. Reconstructing the normal IPv4 header

ITRs are the only devices which are required to convert an ordinary IPv4 header into an EAF formatted IPv4 header. We discuss that functionality in a section below. Here we discuss a function which needs to be performed by ETRs, and in some circumstances by any of the modified routers between the ITR and ETR. Since most ITRs also have the additional capabilities of a modified router, most ITRs need the following functionality as well.

ETRs perform this operation when they accept a packet forwarded from an ITR - to convert the packet into an ordinary IPv4 packet to be forwarded to the destination network. Modified routers (including ITRs) may need to convert the packet back to its IPv4 format in order to include the packet, or part of it, in an ICMP message.

The first step in converting the header is to set bit 48 and bits 50 to 63 to 0. As noted previously and in the following section on converting the packet to EAF format, all packets which are accepted by an ITR have the MF flag and all Fragment Offset bits set to 0.

The second step is to calculate a new Header Checksum from the current state of the header, after bits 80 to 95 have been set to 0. Once this is value is written into bits 80 to 95, the packet is identical to that accepted by the ITR, except for the lower value of TTL which results from the EAF format packet having passed through routers, and for the consequently different Header Checksum.
8. ITR Functionality

This section formally defines the functions the ITR performs when EAF forwarding a packet, which are additional to its basic functionality documented elsewhere regarding deciding which packets need to be tunneled to an ETR, which ETR to tunnel them to etc.

In a map-encap system, the ITR encapsulates the packet in an IP header with the outer header’s destination address being that of the ETR. (The details vary - for instance Ivip only uses an IP header, while LISP uses IP, UDP and then a LISP header.)

8.1. Rejecting fragments or fragmentable packets which are too long

The first step of EAF processing involves rejecting packets which are fragments, or which are fragmentable but longer than MinCoreMTU. In both cases, the packets are dropped and an ICMP message sent to the sending host, including some number of bytes of the original packet.

There is an unresolved question about what type of ICMP message to send in these circumstances. The sending host will not be expecting an ICMP PTB (Packet Too Big: Type 4 Fragmentation required, and DF flag set) message. There is no point defining a new ICMP message, since the problem only arises from hosts which are not using RFC 1191 PMTUD or which are ignoring future BCP guidance not to send fragmentable packets which are longer than MinCoreMTU to any host which is reachable only via the ITR-ETR network. If the host OS and application could be upgraded to accept a new kind of ICMP message, it would be better to upgrade them never to send fragments or fragmentable packets.

8.2. Setting bit 48

The router sets bit 48 to 1.

8.3. Writing the ETR address

The router writes the 30 most significant bits of the ETR address into bits 50 to 63 and 80 to 96. The packet’s header is now valid according to the EAF format.

8.4. Forwarding according to the ETR address

Generally, ITRs are conceived of as additional functions built into a conventional internal or border router. However, there are two instances where this is not the case. Firstly, the ITR functions can be built into a sending host, which has no routing functions, due to it having a single link to its network. Secondly an ITR could be
implemented as a device which advertises prefixes in the local (or interdomain, in the case of an OITRD) routing system, and which converts the packet to the EAF format (or encapsulates it, with map-encap) - however the ITR has a single link to its network and therefore makes no forwarding decisions about the resulting EAF (or encapsulated) packet.

Where the ITR function is in a device with no routing functions, the next step is to forward the packet out of the one interface which leads to the upstream router. This router must have modified functionality so it recognises the EAF formatted packet.

Where the ITR functions are performed within a router (necessarily with EAF modified functionality), the next step is to present the EAF formatted packet to the FIB - which will forward the packet to or towards the ETR address specified in bits 50 to 63 and 80 to 95. This modified forwarding functionality is described in the next section.

Also, when the ITR function takes place in a router, the router’s FIB must be able to generate an ICMP message as described in the next section if the EAF formatted packet cannot be forwarded.

Converting the packet from conventional IPv4 format to EAF format does not involve decrementing the TTL value. The subsequent forwarding step in the ITR itself, or in whichever upstream router the ITR sends the packet to, will decrement the packet’s TTL.
9. Upgraded Router Functionality

The additional functionalities described in this section must be present in all internal and core routers through which the EAF formatted packet travels between the ITR and the ETR. These additional functionalities must also be present in ETRs and in any ITR which is itself a router.

9.1. Recognizing the EAF format

The router must recognise bit 48 being set as indicating that the packet should be handled somewhat differently, as described below, rather than discarded.

The packet can be treated as an ordinary IPv4 packet in all respects apart from those mentioned in this section.

9.2. No checksum calculations

The router still decrements TTL and sends an ICMP message if it reaches zero. Therefore, Traceroute will work over the ITR to ETR part of the path. However, there is no checking of the header’s checksum, or altering the former checksum bits to account for the change to the TTL bits.

9.3. Forward according to 30 bit ETR address

Instead of forwarding the packet according to its destination address, the 30 bit ETR address in bits 50 to 63 and 80 to 95 is used, and extended to 16 bits with two zeroes in the least significant bits.

Any filtering, statistics gathering etc. regarding source and destination address can proceed as usual.

The EAF formatted packet should not be forwarded to any router which is not upgraded to handle it - therefore it must be forwarded to a modified router or to an ETR. (An EAF-formatted packet may also be forwarded to a modified router with ITR capabilities, but the ITR function in that router will ignore it, since it is already in EAF format.) However it is not a task of the router to ensure that the next hop router is upgraded. This requirement must be achieved by ensuring that only upgraded routers advertise prefixes by which ETRs can reached.
9.4. ICMP generation

There are a variety of circumstances in which an ordinary router may drop a packet and send an ICMP message to the sending host, including TTL reaching zero, the packet being DF=1 and too long for the next-hop MTU, and there being no path to forward the packet on.

Modified routers use the same criteria for these actions as with normal packets, except that the address for determining next hop is the ETR address bits, not the destination address. (An ICMP destination network unreachable packet which results from this contains the reconstructed Internet header, which contains the destination address and no mention of the ETR address.)

In order to create an ICMP message which includes the start of the original packet, the router first reconstructs the header into ordinary IPv4 format – as described in a section above.

Packets will also be dropped if their length exceeds the next hop MTU, with an ICMP PTB message being sent to the sending host. Again, in the resulting ICMP message, the attached start of the packet will contain no mention of the ETR address or of the EAF format of the packet at the router where this MTU limit was exceeded.

9.4.1. RFC 1191 PMTU with much less complexity than with map-encap

In a map-encap system where the outer source address is that of the ITR, a router generating a PTB would send it to the ITR, requiring arguably impossible amounts of state and processing in the ITR to securely transform it into an appropriate PTB message addressed to the sending host [ITR-complexity]. (Authenticating the original PTB involves storing huge quantities of state of recently tunneled packets, and altering the MTU value in the PTB to a lower value, to account for the encapsulation overhead, so the sending host will send packets short enough not to exceed the MTU limit, once encapsulated.)

In Ivip’s encapsulation approach, the outer source address is that of the sending host. The PTB would go back to the sending host, but be ignored, since it resulted from an encapsulated packet. There are methods of ensuring that PMTUD works with Ivip’s encapsulation approach [Ivip-PMTUD-Frag], but they involve undesirable complexity in the ITR and ETR.

One of the most important benefits of EAF over encapsulation is that the router which generates a PTB message as just described generates a PTB message which will always be recognised by the sending host. The MTU value it contains is the one the sending host will comply with – which ensures that further packets will pass through this
limiting next-hop without any problems. This is because there is no encapsulation overhead, and because the router is able to reconstruct the packet into a form identical to that which would be expected by the sending host after it has passed through any routers before reaching the router where the MTU limit was exceeded.

The above-mentioned method of reconstructing the ordinary IPv4 version of the packet does not alter the DF flag. It sets the More Fragments flag and the Fragment Offset bits to zero - which is in accordance with the fact that the ITR only accepts packets with those bits set to zero.

It is a requirement of the whole system - in terms of the placement of ITRs and ETRs, the placement of upgraded routers, and the MTUs of the paths between them all - that an EAF packet can be sent from any ITR to any ETR without encountering a non-upgraded router. An additional condition is that the MTUs of all these paths must be at least MinCoreMTU bytes.

Consequently, no modified router will handle a fragmentable packet (DF=0) which exceeds the next hop MTU. (Once the EAF packet has been converted to an ordinary IPv4 packet at the ETR, it may hit an MTU limit - at the ETR’s next-hop or at the next-hop of subsequent routers - and so be fragmented by that router according to established principles.)

Therefore, PTB messages will only be generated for DF=1 packets. Conventional RFC 1191 PMTU will function correctly and the sending host will send subsequent packets with lengths which do not exceed the current router’s next-hop MTU.
10. ETR Functionality

We assume that the ETR has sufficient functionality to recognise a packet whose destination address matches an end-user network it connects to, and that the ETR is able to transport an ordinary IPv4 packet with such a destination address to that destination network. This may be achieved by forwarding in the local routing system, by tunneling or by forwarding the packets out of a specific interface which leads directly to that destination network.

We assume that the ETR also has the extra functionality of modified routers, but this is minimal and would only in fact be needed if the ETR was functioning as a router handling packets with EAF addresses of other ETRs. This would be the case if the ETR was a CE or PE router, and there were any ITRs inside the end-user network it serves — those ITRs or ITR functions in hosts would be sending EAF packets upstream, to be forwarded to other ETRs.

With this assumption, the functionality required of an ETR is very minimal.

When an ETR receives a packet in EAF format, it must recognise if the ETR address in the header matches its address (or one of its addresses). If not, then if the ETR device functions as a router (that is, it has two outgoing interfaces to the rest of the network) then it must have the upgraded router functions. In that case, it will forward the packet to whichever interface has a path for that ETR’s prefix — or drop the packet and generate an ICMP destination network unreachable message as described above.

Assuming the ETR recognises the ETR address in the EAF formatted header as corresponding to itself, then the ETR converts the header to ordinary IPv4 format as described above and then processes it normally — which includes transporting the packet to the destination network.
11. ITR and ETR placement: MTU and upgraded routers

EAF provides a much simpler method of transporting packets from ITR to ETR than map-encap. This can best be appreciated by considering what has to be done in a map-encap system to support RFC 1191 PMTUD. It is also 100% efficient, with no overhead from encapsulating headers.

However, EAF requires that all routers between ITRs and ETRs - in the core and in internal networks - be upgraded to handle the new header format. Hopefully these changes can be implemented with existing routers via a firmware upgrade by the time Ivip4 or any comparable core-edge separation system is deployed.

There are actually two requirements on routers between any ITR and any ETR - which in turn sets limits on where ITRs and ETRs can be located. Firstly, the intervening routers must be upgraded to handle EAF format. Secondly, the PMTU between all these routers, and between the ITRs and ETRs and their neighbouring routers, must be at least MinCoreMTU bytes.

11.1. PMTU and MinCoreMTU

In practice, with a wisely chosen value of MinCoreMTU, the PMTU requirement should not prove too constraining. For instance, MinCoreMTU = 1470 enables fragmentable packets to be carried without fragmentation over a DSL service, with 8 bytes of overhead due to PPPoE, and for instance 28 bytes of IPv4 + UDP or GRE tunnel overhead.

The sole purpose of MinCoreMTU is to support hosts currently sending DF=0 packets. It is not a constraint to how long DF=1 packets can be. As the core of the Net changes so that PMTUs of 9k or so become commonplace, hosts with 9k next-hop MTUs will naturally send 9k packets to destination hosts all over the world, where possible. EAF’s uncomplicated support of RFC 1191 enables all sending hosts to adapt quickly to any lower PMTU it must comply with on the path to any particular destination host.

11.2. Core and internal router upgrades

The biggest hurdle an EAF-only system would need to overcome is the task of upgrading essentially all core routers to handle EAF-formatted packets. "Core" routers in this context includes provider border routers. In this discussion we refer to four kinds of network: ordinary provider networks, upgraded provider networks, ordinary end-user networks and SPI end-user networks.
Ordinary provider networks and ordinary end-user networks do not contain any ITRs or ETRs. Therefore, these ordinary provider networks cannot be used by an SPI network. By definition, an ordinary end-user network is not used by any SPI or any other end-user networks for connectivity to the Net, because then it would be functioning as a provider network.

SPI end-user networks do not have any routers connected to the BGP interdomain core. All their border routers (CE - Customer Edge) either link to ETRs in one or more upgraded provider networks - or the upgraded provider network provides a link (from a PE - Provider Edge - router) to the CE router, which itself functions as an ETR.

With map-encap, it is possible to locate ITRs in a provider network which has no ETRs. Likewise, ITRs could be located in conventional end-user networks which connect to provider networks which lack ITRs or ETRs. In these cases, the purpose of the ITRs is to encapsulate packets from local sending hosts, rather than send those packets to the core and rely on OITRDs to do the encapsulation.

However, with EAF, this flexibility of locating ITRs deep within provider and end-user networks is restricted by the need to ensure all internal routers, (provider) border routers and connected core routers are upgraded to handle EAF packets - and that the PMTUs of all links equal or exceed MinCoreMTU bytes.

Within all networks, and the core, there is no need to upgrade a router if it can be assured that it never advertises a prefix which ETRs are located within. For networks with internal ITRs, including ITR functions in the sending hosts, this means that most or all internal routers need to be upgraded. Likewise, for any upgraded provider network which houses ETRs.

With the exception of their CE routers (which have direct links to provider networks), SPI end-user networks never contain ETRs. ETRs must be located on conventional address space, and so can only be in upgraded provider networks. (Again, a conventional end-user network with an ETR is not an end-user network for the purposes of this discussion, since it is providing connectivity to some other network, including perhaps parts of its own network which use SPI space.)

Any SPI end-user network, upgraded provider network or upgraded conventional end-user network may contain ITRs - in which case these networks need upgraded internal routers to handle all EAF packets emitted by these ETRs.

There may be routers in the core which do not need to be upgraded: any border router of any network which contains no ITRs or ETRs, and
(the next condition may be impossible to assure) any transit router which never handles prefixes which contain ETRs. This means that networks which do not want to use ITRs or ETRs need take no action while other networks upgrade their routers to support EAF. Hosts within these non-upgraded networks will still have full connectivity to hosts in SPI end-user networks, via OITRDs.

There may be some techniques within the BGP control plane to restrict the advertisement of ETR-containing prefixes only to routers which have been upgraded. This may be a fruitful technique in the event that some core routers remain without the EAF upgrades. However, this does not alter the requirement that there be enough core routers with EAF capabilities to ensure robust connectivity between all ITRs and ETRs.

In a transitional arrangement, where map-encap Ivip is progressively converted to EAF, any such BGP techniques could prove valuable in ensuring that for each prefix in which all ETRs are reachable via EAF upgraded internal routers, such prefixes are only advertised to upgraded core routers. However, the most attractive option remains to upgrade the core routers to EAF capabilities from the outset of operations, and therefore avoid the need to implement encapsulation with all its attendant complexities to handle PMTUD.

Why should ISPs want to upgrade their core routers? Perhaps because they could do so for minimal cost with a firmware upgrade, and because by doing so they are making it more possible to implement a core-edge separation system without the inefficiencies and complexities of encapsulation.

Why would router vendors want to develop such upgrades for their core routers - and make available, at minimal cost? Perhaps because they want to contribute to the establishment of the most efficient scalable routing solution, and/or because they would not want their installed routers to be viewed as being incapable of supporting the modest enhancements this entails.

Upgrading sufficient (most, or ideally all) core routers remains the biggest challenge to deploying a scalable routing system based on EAF from the outset. Quite a few years development needs to be done on these proposals before a sufficiently strong global consensus could be reached to choose and implement one scheme. By then, it may be relatively easy to upgrade the core routers within the desired timeframe.

Similar principles apply to internal routers. These are likely to include a larger variety of models, including more older, non-upgradable, models than in the core. However, to the extent that it
is harder or impossible to upgrade these older, smaller, routers, it is also true that they are more due for replacement - so the incremental cost of replacing them in order to support an EAF-only scalable routing solution is not necessarily prohibitive.
12. Security Considerations

Other than the extremely unlikely possibility of bit errors causing a packet to be presented to an upper layer protocol at some host other than the desired destination (Appendix 1) the EAF techniques do not appear to create any security problems beyond those inherent in the routing system or in map-encap.

IPsec Authentication Header is unaffected by EAF.

Where a border router – or any other router – handles the EAF format packet (this is necessarily a router with upgraded functionality which recognises the EAF format) the router can still perform filtering on the packet according to its source and destination addresses, and any other aspects of the packet – since only bit 48, the MF flag, fragment offset bits and the header checksum have been altered. Consequently EAF should not reduce the effectiveness of existing filtering arrangements.
13. IANA Considerations

[To do as more detail is developed about data formats and communication protocols.]
14. Informative References

[Arkko-2008a]

[Arkko-2008b]

[Arkko-2008c]

[Checksums]

[I-D.briscoe-tsvwg-re-ecn-tcp]

[I-D.farinacci-lisp]

[I-D.jen-apt]

[I-D.whittle-ivip-arch]

[ITR-complexity]
[Ivip-PMTUD-Frag]


[SixOneRouter]

Appendix A. Scenarios where bit errors alter the EAF formatted header

Bit errors are unlikely to go undetected by L2 protection mechanisms. In the unlikely event that such errors are undetected, it is highly unlikely that the packet would be forwarded to a node which accepts it. Silent packet loss is an acceptable outcome in the event of an bit errors in the header. The only outcome of concern would be the packet arriving at the wrong node and being recognised, so that it is accepted by an upper level protocol. This would disrupt that node, and potentially lead to information being divulged to inappropriate recipients. However, the chances of this occurring are remote, as the following exhaustive discussion indicates.

A.1. Error in bit 48

An error in bit 48 (setting it to zero) will cause the packet to be recognised by the recipient router as an IPv4 packet. That router will test the header against the presumed checksum in bits 80 to 95 - which are in fact part of the ETR's address. There is a $2^{-16}$ chance that the header would pass this test. Of those which do, the great majority can be expected to have at least one of bits 51 to 63 set, indicating to the receiving host that this packet is a fragment. The resulting packet would be forwarded towards whichever router advertises a prefix which matches the destination address.

Assuming the destination address is unaltered, since the destination address is an SPI address and is part of an Ivip MAB (Mapped Address Block) the packet would be forwarded to the nearest OITRD (Open ITR in the DFZ) which advertises this MAB. There it would be dropped, unless either: (1) all bits 50 to 63 were zero; or (2) bits 51 to 63 were zero, bit 50 was 1 and the packet was shorter than MinCoreMTU bytes. In the unlikely event the bit was not dropped by the OITRD, it would be turned back into a valid EAF format packet and forwarded to the ETR specified in the mapping for the packet's destination address - and so arrive correctly at the destination host in spite of the bit errors!

If there were also errors in the destination address bits, then the packet would be forwarded to some host which is not its intended destination. (If the altered destination address was an SPI address, then it would be forwarded to an OITRD and most likely be dropped, as just described.) Assuming the damaged packet was delivered to some host other than its intended recipient, the IP layer of the recipient host would fail to pass the packet upwards unless it was received with all bits 50 to 63 set to zero. If bit 50 was set, this would indicate that more fragments were to follow - yet no more fragments would be received. If any bits 51 to 63 were set, this would indicate this packet was a second or subsequent fragment - yet no
other fragments would be received.

Even if the packet was passed upwards, the changed destination address bits would have a 65,535 / 65,536 chance of being detected by the checksum of the higher level protocol header: UDP (if a checksum was used), TCP, DCCP and UDP-Lite. IPsec AH would also detect the altered destination address with one of its hash functions. ICMP, IGMP, RSVP, GRE, ESP, OSPF and SCTP would not detect the altered destination address [Checkums]. However this would only occur after a series of highly improbable - and unfortunate - events.

A.2. Errors in bits 50 to 63 and 80 to 96

Assuming bit 48 unaltered was (still set to 1), an error in any of the other modified bits 50 to 63 and 80 to 95 would cause the packet to be forwarded to a different 30 bit ETR address than the one set by the ITR.

There is a remote chance, this may result in the packet arriving at another ETR which accepts the packet, due to this ETR recognising the destination address as one belonging to an end-user network it connects to. More likely, the packet would be forwarded to the wrong address. If there was an ETR at that address - which is unlikely - the packet would most likely be dropped because that ETR does not recognise the destination address as matching one of its end-user networks. Alternatively, the packet may be recognised by an upgraded router as having a forwarding address which did not match any prefix for which it has a path - in which case the router would drop the packet and generate an ICMP destination network unreachable message. Alternatively, the packet would arrive at a host or at a non-upgraded router - in which case a host would be expected to view the header as invalid, due to bit 48 being set.

In the scenarios mentioned in this subsection and the one before - all involving errors in the bits which have new functions in the EAF format - there is an extremely low probability that the packet would arrive at a node other than its intended destination in a form which caused it to be presented to any upper layer protocol.

A.3. Errors in bits other than 50 to 63 and 80 to 96

Initially, we discuss changes to bits in the header other than bit 48, or 50 to 63 and 80 to 96 - assuming them to be unchanged. It is assumed the packet arrives at the correct ETR, which converts the header into an ordinary IPv4 format, calculating a fresh checksum. In this scenario, the resulting packet will contain any altered bits we consider below, with a checksum which matches the altered state of those bits. Consequently, we may expect the altered packet to be
sent to the correct recipient host, where the altered bits may have a variety of impacts.

Changes to bits 0 to 3 would result in the packet being dropped by any router, including the ETR or any router between the ITR and the ETR, due to it failing to have the proper IP version bits.

Changes to bits 4 to 7 (header length) would probably result in the packet being dropped by a router – due to the value being too low, or indicating that there were options in the header which the router does not accept or recognise. If such a packet was forwarded to the recipient host, it is unlikely that the altered header length would allow the packet to be accepted as valid by the IP layer of any receiving host.

Changes to the DiffServ and ECN bits would go undetected, but since the error is a random event affecting a single packet, it is unlikely that any serious negative consequences would arise.

Changes to the Identification field would go undetected in most instances. The only exception which comes to mind are ping packets. Identification is used for reassembling fragmented packets, but the ITR does not accept fragmented packets. If a fragmentable packet (less than MinCoreMTU in length) was emitted by an ETR with an altered Identification field, and then fragmented en-route to the destination host, there would almost certainly be no ill effect, since the value of the Identification field in all the fragments would be the same. Only in some extreme circumstances might a changed Identification field prove problematic: when it is changed to the value of another packet which is also fragmented, and where there is some out of order delivery, so disrupting the host’s ability to correctly reassemble the two packets. This would result in data loss of one packet, and potential data loss in the packet which was reassembled.

Changes to the TTL bits 64 to 71 would generally go unnoticed, but the consequences are at most the loss of the packet and the generation of a spurious ICMP Time Exceeded message.

Changes to the Next Protocol bits 72 to 79 would probably result in the packet not being accepted due to lack of an appropriate protocol handler in the destination host. In the event that the new value was accepted, it would not match the first four bits of the next header and so the packet would be dropped.

Changes to the source or destination address would be detected by many protocols, as noted above at the end of the subsection "Error in bit 48".
Bit errors in both the EAF bits and in others would result in more complex scenarios, but for the present discussion, it suffices to conclude that this is extremely unlikely to result in anything worse than data loss, and that comparable problems are considered acceptable in IPv6.
Appendix B. Applicability to core-edge separation schemes other than Ivip

There is nothing in EAF which depends on any other aspect of Ivip, such as the fast hybrid fast-push mapping distribution system. However, there would be little point in applying EAF to any system which required extra bits to be sent with each traffic packet.

LISP requires such data to be included with each traffic packet: The encapsulation involves an IP header, a UDP header and then a LISP header. EAF could not be applied to LISP without significant modification. Also, the vision of simplicity of Ivip4f - EAF-only Ivip - involves no communication between ITRs and ETRs. LISP involves messages going back and forth between ITRs and ETRs for purposes including testing reachability.

APT also uses a three layer encapsulation technique involving IP, UDP and an APT header. As with LISP’s header, the APT header carries information which is vital to the functioning of the system. (Ivip does not need these complex functions, because it does no multihoming reachability testing or service restoration decisions - end-users are required to do this in their own chosen manner, and send mapping changes to change the behaviour of ITRs in real-time.)

APT relies on its encapsulation technique for an ITR to signal to a Default Mapper that the ITR needs some mapping information for whatever EID prefix the current packet’s destination address is within. Converting this to EAF would not be possible, since there is no record of the ITR address in the packet.

TRRP uses GRE encapsulation - an IP header followed by a GRE header. Perhaps this could be replaced with EAF, as long as there was no need to convey data from the ITR to the ETR, or a need for the ETR to know the address of each ITR which was tunneling packets to it.
Author’s Address

Robin Whittle
First Principles

Email: rw@firstpr.com.au
Full Copyright Statement

Copyright (C) The IETF Trust (2008).

This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.