DRTM - Distributed Real Time Mapping for Ivip and LISP
draft-whittle-ivip-drtm-01.txt

Abstract

Distributed Real Time Mapping (DRTM) is intended for Ivip, but may be useful for other Core-Edge Separation solutions to the routing scaling problem, such as LISP. End-user networks - or other organizations they appoint - control the mapping of each or their one or more micronets (ranges of "edge" space with a single mapping) via sending commands to the organizations from whom they lease this space. This is conveyed to all ITRs which need it in "real-time" such as a second or two, globally. There is no need for any single set of servers to handle all the mapping updates, or to store all the mapping databases for all the MABs (Mapped Address Block) of "edge" space. Networks with ITRs use purely caching Resolving query servers (QSRs), in contrast to previous Ivip arrangements which required them to run a query server which contained the real-time updated full mapping database for all MABs.

Status of this Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that 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 September 7, 2010.
Copyright Notice

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

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

Table of Contents

1. Introduction ......................................................... 3
2. Terminology and overall structure ................................. 9
3. Development stages .................................................. 15
   3.1. Stage 1 - DITRs only ......................................... 15
   3.2. Stage 2 - Add ITRs in ISPs and EUNs, with purely
caching QSRs ......................................................... 17
   3.3. Stage 3 (optional) - ISPs/EUNs have non-caching QSRs .... 20
      3.3.1. X ............................................................ 22
      3.3.2. Y ............................................................ 22
      3.3.3. Z ............................................................ 22
3. (continuation from the previous sub-section) .................... 24
4. Please refer to the RRG message for the remaining material ... 25
5. Security Considerations .............................................. 26
6. IANA Considerations .................................................. 27
7. Informative References .............................................. 28
8. Author’s Address ..................................................... 29
1. Introduction

(This ID was written in a hurry - so it is lacks sub-headings and has rough edges. Nonetheless, it provides a good description of the new mapping distribution system for Ivip, which I think is also applicable to LISP.)

Distributed Real Time Mapping (DRTM) is a complete revision of the structure of Ivip’s mapping distribution system. Prior to DRTM’s announcement on the ITRF Routing Research Group mailing list (2010-02-26: http://www.ietf.org/mail-archive/web/rrg/current/msg06128.html) and a short-lived plan of 2010-02-07, Ivip relied on a single global inverted-tree structured system of Replicators, which fanned out all the mapping changes for all the MABs (Mapped Address Blocks) in the Ivip system. The Replicator system, and its accompanying Missing Payload Servers, enabled tens or hundred of thousands of full database query servers (QSDs) in ISP and other networks to maintain a full, real-time-updated copy of the mapping for all MABs. There were concerns about the practicality of running and depending upon this global, synchronised, full-database mapping structure.

DRTM makes this system unnecessary. Perhaps some elements of this older system, as described in "Fast Payload Replication mapping distribution for Ivip" [I-D.whittle-ivip-fpr] may be useful for optional purposes in Ivip as noted in that ID.

In DRTM, there is no need for any query server to store the full mapping database of all MABs. When an ISP or an end-user network wishes to install ITRs, it does not need to run any query server which stores the full mapping database of any MABs. Instead, the ISP or end-user network installs one - or ideally two or three - Resolver Query Servers (QSRs). QSRs are caching devices and which pass on queries which can’t be answered by the mapping they have cached to one of several typically nearby Authoritative Query Servers (QSAs) which are outside the ISP’s network. These QSAs are typically close enough (several thousand km, or less in densely developed areas) to send mapping replies without significant delays or risk of packet loss.

Companies which operate MABs are known as MABOCs (MAB Operating Companies). MABOCs run - or contract other organizations (DSOCs - DITR-site Operating Companies) to run - a number of widely dispersed sites for operate DITRs (Default ITRs in the DFZ). These DITRs are required to tunnel packets sent from hosts in networks without ITRs. DITRs are caching ITRs closely connected to QSAs which are sent a real-time feed of mapping updates for all the MABs the site supports.
At east DITR-site, this QSA - or one or more similar QSAs - are also able to accept and respond to queries from QSRs in (typically) nearby ISP networks. Thus, assuming there is no suitable cached mapping, a query originates in an ITR and goes directly (or indirectly, through one or more levels of optional caching query servers - QSCs) to one of the several QSRs in the ISP. The QSR creates a second map request and sends it to one of the typically nearby QSAs which is authoritative for the MAB the queried address is within. The QSA responds to the QSR and the QSR responds directly (or indirectly, through QSCs) to the ITR. Since the QSAs are typically close enough for all this to happen within a few tens of milliseconds, the ITR buffers the traffic packet and tunnels it to the ETR specified in the mapping as soon as the mapping arrives.

The ITR caches this mapping, which covers a micronet of address space - which may extend beyond the single IPv4 address or the single IPv6 /64 which was in the ITRs map request. Likewise, the QSR and any intermediate QSCs cache this mapping. This means that other ITRs in this ISP, or in EUNs (end-user networks) served by this ISP, will sometimes or often be able to obtain the mapping they need from this cached information, without the QSR needing to query a QSA.

The longer the caching time, the less often the ITRs, any intermediate QSCs and the QSR will need to send a query. The longer caching time increases the memory requirements of these devices - but memory and CPU power in COTS (Commercial Off The Shelf) servers is exceedingly inexpensive.

The caching time does not impede the real-time ability of the EUN or their appointee to control the tunneling behavior of all ITRs which are handling packets addressed to one of the EUN’s micronets. The mechanism for this is as follows:

When a QSA receives a mapping update for a micronet whose mapping it recently ("recently" in this discussion means within the previous 10 minutes, or whatever the map reply caching time is) sent to one or more QSRs, it sends each such QSR a Cache Update message. This is secured by the nonce in the QSR’s original map request.

The QSR updates its cache and performs a similar algorithm - to send a similar Cache Update message to the one or more queriers to which it recently send the mapping for this micronet. The queriers may be ITRs, or caching QSCs. In the case of a QSC, the QSC receives the Cache Update and performs the same algorithm for its queriers, which may be ITRs or further QSCs.

Cache Updates are expected to be securely acknowledged. Cache Updates will be sent on the above basis for as long as the caching
time of the original map reply - they do not extend this caching
time. After the caching time expires - or just before it does - any
ITRs which are still handling traffic packets for the micronet will
request mapping for this micronet again. This will cause the QSR to
query the authoritative QSA again, and the process will continue as
before.

An additional mechanism is required to protect against the caching of
stale mapping which is not updated as just described due to the
upstream device (a QSC, the QSR or the QSA) becoming unreachable or
being rebooted. The querier (an ITR, QSC or QSR) must, within some
time period such as ten seconds or so, check that its source of
mapping is alive and has not been rebooted. If this cannot be
established, then all cached mapping from this upstream query server
should be flushed, forcing the ITR and upstream query servers to gain
fresh, up-to-date mapping again. This may involve using a different
QSC, QSR or QSA.

Since MABOCs and any DSOCs they use to run their DITRs and QSAs
ensure that multiple QSAs are always active (they need them for their
DITRs anyway) then the system has no single point of failure. A QSR
may need to access a more distant QSA, but in general it will use one
of several "nearby" QSAs and so gain mapping information in a
sufficiently short time, with high enough reliability that ITRs will
not delay traffic packets to a degree which is noticeable to users or
which materially affects the application protocols (except perhaps
NTP).

Most MABOCs will be in the business of leasing out the SPI space in
their MABs to thousands or perhaps millions or billions of EUNs - for
non-mobile portability, multihoming and inbound TE and for TTR
Mobility. The remainder of MABOCs will be large EUNs which run their
own MABOC for their own sites alone. Assuming this SPI space - these
micronets - is to be used at any ISP in the world, MABOCs will be
motivated to ensure that total path lengths from the sending host, to
the nearest (in BGP terms) DITR, and then to the ETR are kept as
short as possible. The best way they can do this is to have
multiple, widely dispersed, DITRs so that no matter where the sending
host and the ETR are located, there will be a DITR not too far away
from the shortest path between these two.

As described below, ISPs do not absolutely need ITRs, but as SPI use
becomes more common, they will be motivated to install their own ITRs
- which means they need to install one, or ideally two or three,
QSRs. It is not absolutely assured that each MABOC will run (or
contract a DSOC to run for them) widely distributed DITR-sites.
However, in general MABOCs will want to do this. This typically
widespread distribution of DITR-sites, each with a QSA which can be
used by any ISP’s QSR, plus the mechanisms by which QSRs identify the closest such QSAs, ensures that typically, the ITRs will not need to wait more than a few tens of milliseconds for their mapping, even if no QSC or the QSR it relies on has this mapping cached.

With DTRM, the MABOCs - the organizations which are providing SPI space to their customers (unless the MABOC is itself the only user of its space) are the ones deriving revenue from this new form of scalable portability, multihoming and inbound TE. It is the MABOCs who need to make the first investments, and who will hopefully derive the first profits - not the ISPs. (Of course a MABOC may be an ISP - but the key point is that MABOCs need not be ISPs.) MABOCs invest in the DITR infrastructure and with little extra effort provide QSAs at those sites for nearby ISPs (and EUNs too) to use if they want ITRs. MABOCs have a keen interest in ISPs and EUNs installing their own ITRs, since this takes the load off their DITRs.

ISPs (and EUNs with ITRs) need only install ITRs, a few QSRs and optionally some QSCs to get their own ITRs handling packets for all MABs. The MABOCs typically "go the distance" by reaching out across the Earth with their DITR-sites. ISPs do not need to pay anything to use each MABOC’s, or each DSOC’s QSAs. If the MABOCs are properly supporting global portability and mobility for the SPI space they lease to their customers, then the ISPs will find that there are QSAs close enough for there to be reliable and low-delay responses to the queries sent out by their QSRs.

QSAs get a robust, secure, real-time feed of mapping updates for all the MABs they handle. This is practical and scalable, since each DSOC has a finite number of sites, such as 5 to 30, or in principle perhaps a hundred or so - and the secure pushing of mapping to all these sites can be done however they like, including with the use of private network links.

The DSOC needs to receive real-time mapping updates from the one or more MABOCs whose MABs its DITR-sites handle. This too is a finite problem with practical solutions, since the DSOC only needs to work with its chosen number of MABOCs. Within the MABOCs, the same principles apply - they can build their own internal mechanisms for accepting mapping change commands from EUNs or whoever the EUN appoints to control the mapping of their micronets. They can then pass this to the DSOC, via secure, real-time and ideally redundant mechanisms. No MABOC or DSOC needs to handle the entire set of MABOCs, so their systems are not required to scale beyond what is practical and desirable for them. Very large numbers of DSOCs, and therefore of QSAs, can be securely, automatically and scalably handled by each QSR, using a DNS-based discovery mechanism described below.
To the extent that scaling difficulties present barriers to any one MABOC or any one DSOC handling large numbers of micronets, these can be overcome by there being larger numbers of MABOCs and DSOCs.

As long as there are no more than 100k to perhaps 200k MABs, and assuming each MAB contains at least dozens of micronets (many will have hundreds of thousands or perhaps millions) then the system will be perfectly scalable and will serve very large numbers of EUNs without excessive load on the DFZ control plane.

A crucial part of the scalability of this system is the ability of the QSR to automatically discover the "nearest" two or three QSAs for each MAB. Since multiple MABs will generally be run by one MABOC, and multiple MABOCs will usually have their MABs supported by the DITR-sites of a single DSOC, it follows that there will be many less DSOCs than there are MABs. Whether a DSOC runs a handful of DITR sites or hundreds, the QSRs will still be able to reliably find the two or three "nearest" QSAs at these sites.

When fully developed, DRTM will contain some additional mechanisms to aid scaling, load balancing and robustness. Firstly, map replies and Cache Updates from QSAs will optionally contain additional information advising the querying QSR to what extent this QSA should be used for further queries - with suggestions on which other ones should be used as alternatives. This will enable one DSOC’s system to dynamically load-share the burden of QSR queries across its QSAs.

Sometimes, a QSR may need to query a distant QSA - such as one on another continent. This will be a failing of the DSOC/MABOC to properly reach out to the ISPs - not a failure of the architecture. MABOCs earn money from their MABs, so to remain competitive against other MABOCs, they will need to run a good DITR system and handle mapping queries from all QSRs as quickly and reliably as possible.

It will be technically possible for a MABOC to run its one or more MABs with a single DITR, at a single DITR-site - while providing a single QSA for all the QSRs in the world to send their queries to. This would not be a good service for its SPI-leasing customers compared to the best services of other MABOCs, but it does show how a company could start up in business as a MABOC with simply a DITR-site and some address space. DITRs are most likely to be, initially at least, COTS servers. However a DITR is not just a server in a data-center - it needs direct connection to other routers at an Internet exchange since it needs to advertise the one or more MABs it serves to the DFZ and be able to forward encapsulated packets to other DFZ routers.

DRTM is firstly a new approach to CES mapping distribution, secondly
a new approach for Ivip which is not subject to some important
concerns which applied to earlier approaches, and thirdly it is a
model of introducing a new form of address space and DITR service for
scalable routing without ISPs needing to make any initial investment.
2. Terminology and overall structure

DRTM involves new protocols, servers with new functions and organizations playing new roles — with some flexibility. In order to keep the IDs as short as possible, please refer to the Ivip Glossary [I-D.whittle-ivip-glossary] for fuller descriptions of the following terms: DITR - Default ITR in the DFZ; DSOC - DITR-site Operating Company; EUN - End-User Network; ITFH - ITR Function in sending Host; ITR - Ingress Tunnel Router; MAB - Mapped Address Block; MABOC - MAB Operating Company; Mapping; Mapping Distribution System; Micronet; Mobility; MN - Mobile Node; Multihoming; Portability; QSA - Authoritative Query Server; QSC - Caching Query Server; QSD (obsolete term); QSR - Resolving Query Server; Replicator (obsolete term); SPI - Scalable Provider Independent address space; TE - Traffic Engineering; TTR Mobility architecture; TTROC - TTR Operating Company; and, UAB - User Address Block. The Glossary also contains the history of Ivip’s mapping system, but this can be ignored by anyone who is not familiar with Ivip before mid-March 2010.

For a general overview of Ivip, including its goals and non-goals, please refer to the Ivip-arch ID [I-D.whittle-ivip-arch].

Figure 1 illustrates the minimal configuration for DRTM. All ITRs are assumed to be in an ISP network, and there are no QSCs. The ISP network has two QSRs. One is theoretically sufficient, but two, three or perhaps more should be used for robustness and load sharing. The query-response paths are shown between the ITRs and the QSRs, and between QSR-1 and the two nearest QSAs of two separate sets of DITR-sites, run by two separate DSOCs A and B. The links (perhaps via private networks) between the sites of DSOC-A are shown as "AA.." and likewise as "BB.." for DSOC-B.
Inside the ISP, two sets of DITR-sites A and B, showing the two closest QSAs in each set.

Multiple ITRs Two QSRs

[QSA-A4]

A

[QSA-B9]B

A

[QSA-B8]BBBB

B

[QSA-B7]BBBBBBB

B

Figure 1: Minimal structure for DRTM.

Figure 2 depicts the direction of flow of messages between the querier and the upstream query server. This is true for the pair "ITR -> QSR" and for the pair "QSR -> QSA". Not shown in Figure 1 is one or more levels of optional QSC caching query servers. The same pattern of messages applies to the pair "QSC-downstream -> QSC-upstream", and to the pairs "ITR -> QSC" and "QSC -> QSR". At present, there are no plans for any intermediate servers between QSRs and QSAs. Inside each DITR-site, the DITRs are assumed to either query directly a QSA, or if there are multiple DITRs, they may do so via QSCs.

The internal operations of DITR sites are not our primary concern, since these are inherently scalable and are a matter only for the DSOCs and MABOCs who run them. Our primary concern is with the scaling, robustness, performance and security by which ISPs (or any EUN which has its own ITRs and QSRs) can work with sufficient QSAs for the ITRs to be able to handle packets addressed to all the MABs in the entire Ivip system.

Note that an ITR has one or a few upstream query servers, and so does a QSC. In later work, some automatic discovery methods by which ITRs and QSCs can discover their upstream query servers will be proposed.

Although Figure 1 depicts QSR-1 as having just four upstream query servers, in reality, there may be dozens or even hundreds of DSOC systems such as A and B, and each QSR will automatically discover (before it needs to send any Map Request queries) the addresses of two or three of the nearest QSAs in each such system of DSOCs. QSR-2 would do the same in principle as QSR-1. Perhaps it would use
identical QSAs as QSR-1, or perhaps it would use somewhat or completely different QSAs. The latter would provide greater resilience in the event of some QSAs becoming unreachable or going down.

In Figure 2, the messages are shown in the order they would occur. All these messages are currently assumed to be UDP packets, but perhaps SCTP could be used as an option.

Future work will involve describing how ITRs automatically discover, via the one or more QSRs theirs queries go directly or indirectly to, what the current complete set of MABs is. As described in a separate section below, QSRs automatically discover this from a DNS-based mechanism. This could be quite a large body of information, but there needs to be a reliable and automated way that all ITRs know the complete set of MABs. They need this to advertise all these MABs in the local routing system, and so they only attempt to get mapping for, and tunnel to an ETR, those packets whose destination addresses match one of the MABs.

When an ITR receives a traffic packet whose destination address matches a MAB, but not any micronet it currently has cached mapping for, it buffers the traffic packet and sends a Map Request containing the destination address, which is an SPI ("edge") address. The Map Request contains the ITR’s address in the source field, and it also contains a nonce which the ITR generated and uses only for this Map Request.

This Map Request may go to the QSR directly (as shown in Figure 1) or it may go via one or more layers of QSCs before it reaches one of the network’s QSRs. If it goes directly to the QSR, this is a single Map Request. If it goes to a QSC-n, that is one Map Request. Assuming QSC-n does not have the mapping cached, then QSC-n will generate a separate Map Request, with a separate nonce, and send this to one of its upstream query servers, which may be either another QSC or one of the network’s QSRs. While the two Map Requests are for the same SPI address, they are separate Map Requests, with separate queriers and upstream query servers. The two Map Requests have separate nonces. Similarly, the Map Replies and the Acknowledgements are separate for the two legs of the path: "ITR --- QSC-n" and "QSC-n --- QSR".
Figure 2: Direction of flow of various messages.

The typical operation, for instance from ITR-1 to QSR-1 in Figure 1, would be:

ITR-1 sends Map Request to QSR-1.

The diagram shows a Map Request Ack being sent from QSR-1 to ITR-1. Whether this will be necessary is TBD. Normally, the Map Reply will come back to the ITR-1 soon enough that the retry time-constant can be short enough for rapidly trying to send a second request, perhaps to QSR-2. If QSR-2 has the mapping already cached, it will be able to send the Map Reply very rapidly — such as within a few milliseconds. Since the all the ITRs and QSRs are in the same ISP network (in this example at least) the delay times between them sending and receiving packets will be a sub-millisecond or a few milliseconds at most.

If QSR-1 already has the mapping cached, it will send back the Map Reply. If not, it will send a separate Map Request to one of the QSAs, and when that is replied to, it will send the Map Reply back to ITR-1.

It may not be necessary for the requester — ITR-1 in this example, to acknowledge the receipt of the Map Reply, but such a message is shown here. If this is part of the protocol, then QSR-1 would time-out after a few milliseconds of not receiving it, and resend the same Map Reply, at least for a few attempts before giving up.

In this exchange, the Map Request included a nonce unique to this Map Request, and this is used to secure the Map Reply, and any Map Request Ack. In a fully developed version of the protocol, the Map Request may also have a sequence number generated by ITR-1, which is
returned in the Map Reply and any Map Request Ack, so it can more easily index into the information it needs to relate these to. Likewise, the Map Reply and the Cache Update (to be described below) may have sequence numbers generated by the sender - QSC-1 in this example - to aid it in keeping a track of the Map Reply Ack and Cache Update Acks which result, and which would include copies of these sequence numbers.

At a later time, but within the caching time specified in the Map Reply message, if the upstream query server is notified of a mapping change to the micronet whose mapping was returned in the Map Reply message, then that upstream query server (QSR-1 in this example) will generate a Cache Update message.

The most common change to the mapping of a micronet is a change to its ETR address. In this case, the Cache Update specifies the micronet (its start and either its end, or its length) and then specifies the new ETR address. The Cache Update carries the nonce and any sequence number contained in the original Map Request which lead to the Map Reply with this micronet’s previous mapping.

The querier, ITR-1 in this example, simply updates its cache and from that point on, will tunnel any packets whose destination address matches this micronet to the new ETR address.

There are other changes to mapping than simply changing the ETR address of a micronet. The ETR address in the new mapping could be zero (0.0.0.0 in IPv4). That is processed as described above, but the ITR interprets it as an instruction to drop any packets whose destination address matches this micronet.

Another change is that the EUN or their appointee could command the MABOC to delete this micronet, and perhaps at the same time to make its existing range could be split into two, or used as part of a bigger micronet. Also, the new micronets may not have boundaries at the boundaries of the old one. While it is possible to imagine complex Cache Update commands to convey all this information, it is almost certainly the case that the best approach is to simply send a special Cache Update message to the effect that the cached mapping for this micronet should be deleted (AKA "flushed") from the cache of the querier. In this example, the querier is an ITR. If the querier was a QSC, or a QSR, then it would generate similar Cache Update messages to the one or more downstream devices which it recently (inside the current caching time) send mapping for this micronet.

When a micronet is deleted from the cache of an ITR, the next matching packet (and there may be none, for a long time, or perhaps
ever) will cause the ITR to start afresh and request mapping for the
destination address of the new traffic packet. Since its upstream
query servers will just have had their cached mapping for this
micronet deleted, then unless one of them has already requested and
received mapping which matches this new packet's address, then a Map
Request message will be sent by a QSR to a QSA, and the reply will
soon come back, as described above, giving the ITR the up-to-date
mapping for whatever new micronet this destination address matches.
To implement this "delete cached mapping" command without any special
flags in the message, it would suffice to define an ETR address such
as 0.0.0.1 to have this specific meaning.

The fully developed DRTM protocol will specify timeouts for retrying
these communications. For instance, if an ITR sends a packet to its
first choice of three upstream query servers and doesn't receive a
response within 50ms, it would send a separate Map Request, with a
separate nonce, either to the same upstream query server, or probably
to another one. For instance, if, as in Figure 1, the
upstream query servers were QSR-1 and QSR-2, and if the first query
went to QSR-1, then the retry would go to QSR-2. Similarly, if QSR-1
sent a Map Request to QSA-A6 and didn't get back a reply in some
predetermined time, it would send a fresh Map Request to QSA-A5.

The time constants for these retries need to be set with care. Also
there needs to be some limit to the number of retries, with error
reporting if no mapping can be found. In the case of a QSR which
can't get a reply from any of the QSAs it has currently chosen for a
given MAB or set of MABs, this would be a reason for it to recheck,
via the DNS mechanism, which QSAs are authoritative for this MAB and
to choose some other ones to try instead.

Future work will include describing how a QSR chooses generally
"nearby" QSAs - such as by sending a null Map Request and timing how
long after this the Map Reply arrives - whilst also considering any
instructions in the Map Reply about whether to continue to send
requests to this QSA, and perhaps with suggestions as to which
alternative QSAs to use. (The suggested QSA addresses or FQDNs must
match one already returned by the DNS mechanism.) As part of this,
the QSR would determine the typical time it takes this QSR to
respond. If a significantly longer time elapsed after a real Map
Request, without a Map Reply arriving, then this would constitute a
time-out and be cause for sending a second Map Request to an
alternative QSA.
3. Development stages

3.1. Stage 1 - DITRs only

For non-mobile services, one or more MABOCs set themselves up in business, with one or more MABs, some method of accepting map change commands from their EUN customers (or whoever the EUN appoints to control the mapping of the SPI space they lease, divide into micronets etc.) and with at least one, but probably half a dozen or more DITRs at DITR-Sites around the world. For simplicity in this example I will assume the MABOC runs its own DITR sites, so it is its own DSOC.

A MABOC doesn’t absolutely need more than one DITR or multiple DITRs at sites all over the world. They can still make their MAB work with a single DITR or DITRs only in a given region. If, for instance, the MABOC’s SPI-using EUN customers for some reason always use their SPI space via ISPs in Europe, then for the purely DITR purposes, it would be fine for the MABOC to run a handful of DITRs just in European sites.

If a Sending Host (SH) was in Adelaide (Australia) and was sending packets to a host in a micronet which is mapped to an ETR accessible via an ISP in Dusseldorf, then it will be fine if these packets traverse most of the DFZ in their raw SPI-addressed state, to be tunneled to the ETR by a DITR in Amsterdam, London or Zurich.

However, later in the development phase when the MABOC wants to encourage ISPs and EUNs all over the world to run their own ITRs, then it really needs to do better than just have DITR-Sites in Europe. Also, its a second-rate service to only have DITRs in a given region - since at least some of its European companies might want to use their space in branch offices in Asia, North and South America etc. If the sending host in Adelaide was in an EUN or ISP with ITRs, then it would be best if the caching MR those ITRs depend on can send queries to a DITR-Site-QSD a lot closer than Europe.

DRTM doesn’t absolutely ensure that the closest QSA or closest several QSAs to any QSR in the world will be "nearby". Firstly, as just noted, there may only be a single DITR site with a single QSA. This would be a pretty lousy service by the MABOC, but perhaps it would suit a start-up company, or be suitable for a smaller EUN who has converted its own PI prefix into a single MAB and is now using it as SPI space, being its own MABOC.

If the MABOC company is providing a good service to its customers (or to itself, it this is a EUN only using the space for itself), it will ensure it has DITRs widely scattered around the Net and the nearest
DITR-site – and therefore the nearest QSA – will be "nearby" or "near enough to generally provide a fast response with little chance of the query or response packets being lost" to QSRs all over the world.

If a MR in Dallas Fort-Worth finds that the closest DITR-Site is in London, then it's not disastrous. When I wrote the original version of this DRTM material on 2010-02-26, there was a 104ms RTT to London via Houston (I think "IAH"), LA and Washington DC. As long as the query or response packet isn’t dropped, this shouldn’t cause much complaint about slow starts to communications. But it is not ideal, and it would be much better if the nearest DITR-Site-QSD was in San Jose, which should be a RTT of 40msec or probably much less (though on 2010-02-26 I got a traceroute from DFW to San Jose via Amsterdam and London with a RTT of 172ms!).

For Stage 1 – DITRs only – as long as the DITR-Sites are reasonably close to wherever the SPI space is being used, and as long as they can handle the traffic loads, then the only other things which are needed are:

1. The DITRs need to have a tunneling and PMTUD protocol which is compatible with the ETR functionality of whatever the SPI-using EUNs (SPI-leasing customers of this MABOC) are using on their PA addresses. For now, I am assuming that the ETR functionality can be provided by the MABOC for free – such as being downloaded from somewhere and run on a COTS server of the SPI-EUN – or, ideally, be implemented on a router which the SPI-using EUN already owns.

2. The ISPs these EUNs connect with must allow them to emit packets using these SPI addresses as source addresses.

In this simple arrangement multiple EUNs EUN-0000 to EUN-0999 are customers of this MABOC-X and are using micronets in one or more of MABOC-X’s MABs. Maybe another set of EUNs EUN-1xxx are leasing space from another MABOC-Y. Assuming any EUN-0xxx is only using SPI space from MABOC-X, then their ETR functions only have to be compatible with the DITRs run by MABOC-X. So far, there’s no absolute need for standardization.

Ideally there would be RFC standards for ITRs and ETRs and all the DITRs in the world would support this one standard. Then EUNs could lease SPI space from multiple MABOC companies and know that the one ETR function could handle packets tunneled by the ITRs run by their two or more different MABOCs.
The real need for standardization of ITR and ETR functions comes when parties other than MABOCs are running ITRs, and/or if parties other than MABOC customers are running ETRs. In the latter case, perhaps an ISP runs an ETR which connects to the networks of multiple SPI-using EUNs. They don’t want to be mucking around with different ETRs for customers using different MABOCs, and therefore relying on different sets of technically different DITRs.

3.2. Stage 2 - Add ITRs in ISPs and EUNs, with purely caching QSRs

This Stage 2 is the main part of DRTM. As far as I know, it should suffice for Ivip scaling to the largest imaginable numbers of MABs, SPI-using EUNs (including billions of TTR Mobile MNs) and so billions of micronets.

Stage 3 goes beyond the primary purpose of DRTM to allow for arrangements which have something in common with previous approaches to Ivip’s mapping system, but not the global inverted tree of Replicators. Specifically, Stage 3 in some ways resembles the so-far only partly documented, short-lived "Plan-C" arrangement of 2010-02-17 as described in the History of Ivip mapping systems at the end of Ivip-glossary. There may be no reason to go beyond this Stage 2 arrangement, since as far as I know it will scale very well. Stage 3 is documented because it is possible and there may be some benefits for certain ISPs or other networks to pursue it. The Stage 3 arrangements are absolutely not required of any participant in Ivip - and perhaps they will never be developed or deployed.

In all stages the MABOCs charge their SPI-leasing customers for the use these customers make of their DITRs - or rather, the use of these DITRs by packets sent to the SPI addresses of each of their SPI-leasing customers. If the MABOC uses another company - a DSOC - to run its DITRs, then the DSOC needs to monitor DITR usage and present statistics to the MABOC, both for the total uses of DITRs by packets addressed to MABs run by this MABOC and in greater detail by micronet, so the MABOC can charge its SPI-using EUN customers accordingly.

The MABOCs will also charge their customers for each mapping change - probably a few cents or similar. This is in part to deter EUNs from making frivolous and repeated mapping changes, and also to help pay for the cost of running DITR sites and QSRs, and for fanning out the mapping change to all the DITR sites which support this MABOC’s MABs.

The MABOCs would be happier if some, most or ideally all of the tunneling of the traffic addressed to their SPI-leasing customer EUNs was done by someone else’s ITRs: the ITRs of the EUNs or ISPs of the sending hosts (SHs). To this end, “perhaps” the MABOCs would want to
pay ISPs and large EUNs to run ITRs covering their MABs.

(There is an unresolved question regarding a scenario in which an EUN very frequently changes the mapping of its micronets, and this results in very frequent mapping updates being sent to QSRs and ITRs in ISPs whose ITRs are tunneling packets to this micronet. The ISP may be unhappy about this high level of updates giving their QSR and ITRs a workout. Should the MABOC, which is charging money for these updates, use some of that revenue keep ISPs happy to receive and act on these frequent mapping updates?)

As the use of SPI space becomes more widespread, the ISPs themselves would want to have their own ITRs. If an ISP has one or more customers with SPI space (either with their own ETRs, or using an ISP-supplied ETR) and there are other customers of this ISP sending packets to these SPI addresses, then the ISP would prefer to have its own ITR to tunnel these directly, rather than let the packets go out the upstream link, to a DITR, and return in encapsulated form via that link.

If the ISP had its own ITR, at least to cover the MABs of these SPI-using customers, it could reduce the traffic on its expensive upstream links – and provide faster packet delivery times.

For this discussion, I will assume that an ISP installing an ITR will make that ITR advertise, in its internal routing system, all the MABs of the complete Ivip system. This is not necessarily the case, but it makes the discussion less complex to assume this.

So the ISP wishes to run one or more ITRs which cover all MABs.

Also, individual EUNs using this ISP may wish to run their own ITRs so their outgoing packets addressed to SPI addresses will definitely takes the shortest path to the ETR, rather than going by some potentially longer path to the "nearest" DITR. (After seeing a Dallas Fort-Worth to San Jose traceroute go via Amsterdam and London, I am keen to put the word _nearest_ in inverted commas if I mean "nearest in terms of the current state of the DFZ"!

This is for EUNs on conventional PI space, PA space or SPI space. The question of an EUN having its own ITRs, or wanting its ISP to have ITRs, is independent of whether the EUN is using conventional PI or PA space, or is using SPI space via an ETR. In Ivip, ITRs can be on SPI addresses. They can also be implemented in sending hosts on any global unicast address (PI, PA or SPI). (At present, I don’t have arrangements for ITRs to be behind NAT, but it could be done with a different protocol between the ITR and the upstream query server - QSC or QSR – it queries.)
In all cases, these ITRs need a "local" QSR to send their queries to.  
(I don't plan for ITRs to directly query QSAs, which are typically 
outside the ISP's network, at DITR-sites.  It would probably be 
technically possible, since the ITR -> QSC/QSR query protocol would 
similar or identical to the QSC -> QSC, QSC -> QS and the QSR -> QSA 
protocols.)

So in all the above circumstances, to accommodate one to hundreds or 
thousands of ITRs in an ISP's network (including in the EUN customers 
of this ISP) then the ISP should install two or more QSRs.  (I will 
refer to ISPs doing this, but the same principles apply to any EUN 
which wants to run a QSR itself.)

These ITRs need to be configured - or ideally automatically discover 
- the two or more QSRs of this ISP, or the upstream QSCs if these are 
used.  I haven't worked on how to do this, but I am sure it will be 
possible.

In some networks the ITRs would query the QSRs directly.  In others, 
they would query a QSC first, which handles a bunch of ITRs and which 
queries either the QSR directly, or the QSR via one or more other 
QSCs.  One way or another, each ITR needs at least two upstream QSCs 
or QSRs to query.  It would typically send a query to one, and if 
nothing came back within some time like 100ms, it would send a fresh 
Map Request query for the same address (with a different nonce) to 
the other upstream query server.

Then the task of designing the DRTM system is to devise a method by 
which these two QSRs know at least two (ideally nearby) QSAs to query 
for each of the MABs in the entire Ivip system.  This can be done by 
a new DNS-based system I describe in a section below.  There could be 
other, better, ways - but this will do for now.

This is a highly scalable arrangement.

The MABOCs directly or indirectly push their own mapping, for their 
own MABs, in real-time, highly reliably, to all their DITR-Sites.  
They need to do this to have full-database query servers in the same 
rack (or even the same server) as their DITRs.  Theoretically, DITRs 
could rely on a distant full database query server, but this would be 
pretty sloppy - and the DITR is surely going to be getting a lot of 
packets, so it makes sense for the MABOC to push its full mapping for 
each MAB to a QSA query server at each DITR-Site which is full-
database for all the MABs covered by that DITR-Site.

So it is not much extra work to use this fresh, reliable, feed of 
real-time mapping to drive one or more a publicly accessible QSAs at 
each DITR-site.
For the MABOC or whoever runs the DITR-Site (the DSOC), it will be a lot less effort easier answering queries and so allow other organizations’ ITRs to tunnel a bunch of packets, than for this DITR-Site’s DITRs to tunnel the same packets.

The ITRs, QSCs and QSRs in the ISP and its EUNs do not store the entire mapping database, or the full mapping database of any of the MABs. The QSRs can boot up very quickly, as described below - they only need to discover the current set of MABs and the DITR-Site-QSDs they will query for each MAB. (If the QSR was full-database for one or more MABs - as mentioned below in Stage 3, which is entirely optional - it would need to download snapshots as described in Ivip-fpr. These snapshots could be quite bulky if there were billions of micronets, as there will be with widely deployed TTR Mobility.)

The DITR-Site can scale well by spreading the load of traffic for multiple MABs (including potentially every MAB in the whole Ivip system, if for some reason one DITR-Site was working for all the MABOCs) over multiple separate DITRs, each of which advertises to the DFZ a subset of these MABs. A single MAB could, in principle, be split between two DITRs if necessary, but each advertising half, or a quarter of it. The DITRs would presumably be either acting as DFZ routers, advertising MABs and emitting packets back over the same link, or perhaps another link, to be forwarded by other DFZ routers - or they could be behind a single DFZ router. In the latter case, if four DITRs advertised a quarter of a MAB, then their common router should aggregate these into a single shorter prefix of the original MAB for its DFZ neighbours.

The overall load of traffic can be shared by creating more DITR-Sites in the areas where they are most needed. Larger DITR-Sites - such as those which are not operated by a single MABOC, but which serve the MABs of multiple MABOCs - including perhaps every MAB in the Ivip system - would also offer scaling benefits by sharing the various peaks in traffic for particular MABs in the system.

It is possible that the ITRs of a given ISP and its dependent EUNs might not advertise, into their local routing systems, all the MABs - but in this discussion, I assume that they do. Then, the QSRs they use need to know how to send queries to (ideally) nearby QSAs for every MAB in the system. More on this in the section below on the new DNS-based mechanism.

3.3. Stage 3 (optional) - ISPs/EUNs have non-caching QSRs

This stage may never be developed or implemented - because I suspect it may never be needed. Nonetheless, it exists as a technical possibility and is described here in case it turns out to be useful.
This involves some things which people have concerns about - real-time pushing of mapping changes to query servers which are full-database for some or all MABs. In Stage 2, which is what DRTM is intended to achieve, there is a need to push real-time mapping to query servers. However, these are the QSA query servers at DITR sites, and the pushing only happens within or between the DITR-sites operated by a single DSOC - though the DSOC has to get the real-time mapping changes from as many MABOCs as it chooses to run DITRs for. The fact that the number of DITR-sites is limited, such as to dozens or at the very most a few hundred, makes the system much less open to objections about scaling problems. Also, the fact that the real-time pushing largely or only occurs in a single organizations system, albeit a globally distributed system, means that it is more scalable and secure than trying to build a single system to be shared by many organizations, which was the plan for Ivip until late February 2010.

A further reason this Stage 2 pure DRTM arrangement is more scalable is that no DITR-site and so no DSOC has to carry the whole set of MABs. To whatever extent there are scaling problems or other objections to this real-time distribution of mapping to multiple QSAs, they can be avoided in total by the natural result of each system operating to its limit, and there being multiple independent systems operating in parallel to meet the total demand.

Stage 3 resembles - or could be the same as - what I described on the RRG list on 2010-02-07 ("Plan-C" in the history at the end of Ivip-glossary). With Stage 2 looking so powerful and scalable, with no need for QSRs to get a feed of mapping updates for any of the MABs - there may still be some situations I have neither foreseen or ruled out the existence of yet where it would be desirable to have the QSR get a full feed for real-time mapping changes for some MABs. This means a QSR, or probably all the QSRs in a given ISP or EUN with ITRs, would then be "full-database" for one or more MABs. In principle, this could be extended to the QSRs being full database for all MABs. This would be "Plan C" as noted above.

Maybe there would be a scenario where it would make sense for all MABs - I can’t rule it out, but I can’t think of one either, other than as noted above with networks being on passenger jets, ocean liners, Antarctica or the Moon.

In Stage 3, each QSR could use three methods to get its mapping. Of the three methods mentioned below, it could use X alone, X and Y, Y alone, or Z alone. (I am using X, Y and Z to avoid confusion with Plans A to D or Stages 1 to 3. Sorry this is complex - but I am are trying to anticipate a variety of usage situations.)

X and Y are as described in draft-whittle-ivip-fpr-00 (Plan-B, 2010-
01-18) and msg05975 (Plan-C - 2010-02-06). Z is newly described here.

3.3.1. X

The QSR (initially described as a QSD) could get direct feeds of "Replicator format" mapping update packets directly from servers which generate this at various (ideally) nearby DITR-Sites. The QSR will need to get two feeds for each MAB, so this will involve a bunch of feeds for a set MABs from one DITR-Site, and a similar bunch of feeds for the same MABs from another DITR-Site. This is assuming that the local DITR-Sites are in similar sets, each set serving the MABs of one or more MABOCs. Then there will need to be pairs of feeds from other DITR-Sites, a pair for each other set of MABs.

3.3.2. Y

As per Plan-B and Plan-C, the QSR could get feeds via a system of Replicators which form a fully or partially meshed flooding system, accepting feeds from multiple DITR-Sites, and fanning the sum of the mapping updates they contain. This should be the full set of updates for all MABs, unless all packets with a particular payload somehow don’t make it to any of the Replicators.

In either X or Y, the QSRs will occasionally need to query one or more "Lost Payload" servers to get mapping update payloads they somehow missed. With larger sets of missing payloads, the QSR would need to re-sync its databases for one or more MABs, which involves downloading a snapshot file and bringing it up to date, as described in draft-whittle-ivip-fpr-00.

3.3.3. Z

The MR sets up some kind of secure, two-way, link to one or probably better two (ideally) nearby DITR-sites, for a given MAB or set of MABs. Assuming, for the purpose of discussion, that the entire set of MABs is covered by five DITR-Sites, then this means the MR will set up ten such sessions - using two DITR-sites per subset of MABs, for redundant supply of mapping changes for a given subset of the MABs.

Each such link should enable a suitable server at the DITR-site to quickly and reliably push all mapping changes to the QSR, and for the QSR to be sent any changes again, if it somehow didn’t get some of them. I guess TLS-protected SCPT might be a good protocol for this - RFC 3436. TCP would be OK, but it would be blocked for a moment by the loss of a single packet.
Theoretically, the QSR needs a single two-way link from a DITR-site server which handles a set of MABs, but it should have two such links, with two such DITR-sites for each subset of MABs - for Justin (Just In Case).
4. (continuation from the previous sub-section)

(Due to time constraints I haven’t figured out how to make this part be at the end of the previous sub-section 3.3.)

Also, it would be possible for a QSR to use the Z arrangement to pass on the mapping information to other QSRs.

In this way, an QSR could get real-time mapping feeds for one or more MABs and so be "full database" for these, while still sending queries to QSAs for addresses matching other MABs.

This enables the operators to have some flexibility. Say the QSR was in New Zealand, and there were some MABs G, H and I which for some reason it wanted to be full-database for. Maybe these MABs are run by MABOCs who lease the space to SPI-using EUNs for which this ISP wants to respond very quickly to for each new communication session. There may be some other MABs J, K and L for which the ISP doesn’t mind so much if the session establishment takes a few tens of milliseconds longer, by relying on some reasonably "nearby" DITR-Site-QSDs. Maybe some MABs are used primarily or solely for SPI-using EUNs who are almost always in a distant country, and for which this ISP’s customers hardly ever send packets. There’s no problem having these MABs as usual, on a query-basis, like J, K and L – but the ISP can still, for whatever reason, choose to have the its QSRs running full mapping databases for selected MABs G, H and I.
5. Please refer to the RRG message for the remaining material

Due to time constraints this ID does not yet have all the material it should have. Please refer to the latter parts of this RRG message: http://www.ietf.org/mail-archive/web/rrg/current/msg06128.html for several sections, there labelled: "Stage 2 needs a DNS-based system so TRs (QSRs) can find DITR-Site-QSDs (QSAs)"; "4 - With TTR Mobility" and "5 - DTRM for LISP".

In this RRG message "DITR-Site-QSD" means the same thing as "QSA" in this ID. Likewise, in the message, "MR" means the same thing as "QSR" in this ID.
6. Security Considerations

TBD.
7. IANA Considerations

TBD.
8. Informative References

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

[I-D.whittle-ivip-arch]
Whittle, R., "Ivip (Internet Vastly Improved Plumbing)
Architecture", draft-whittle-ivip-arch-03 (work in
progress), March 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.