

Intarea Working Group
Internet-Draft
Intended status: Best Current Practice
Expires: December 24, 2013

R. Bonica
Juniper Networks
C. Pignataro
Cisco Systems
June 22, 2013

A Fragmentation Strategy for Generic Routing Encapsulation (GRE)
draft-bonica-intarea-gre-mtu-02

Abstract

This memo documents a GRE fragmentation strategy that has been implemented by many vendors and deployed in many networks. It was written so that a) implementors will be aware of best common practice and b) those who rely on GRE will understand how implementations work. The scope of this memo is limited to point-to-point GRE tunnels. All other tunnel types are beyond the scope of this memo.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

Status of This Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

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."

This Internet-Draft will expire on December 24, 2013.

Copyright Notice

Copyright (c) 2013 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 Simplified BSD License.

Table of Contents

1. Introduction	2
1.1. How To Use This Document	3
1.2. Terminology	3
2. Candidate Strategies and Strategic Overview	5
2.1. Candidate Strategies	5
2.2. Strategic Overview	6
3. Generic Requirements for GRE Ingress Routers	7
3.1. General	7
3.2. Tunnel MTU (TMTU) Estimation and Discovery	7
4. Procedures Affecting The GRE Deliver Header	8
4.1. Tunneling GRE Over IPv4	8
4.2. Tunneling GRE Over IPv6	9
5. Procedures Affecting the GRE Payload	9
5.1. IPv4 Payloads	9
5.2. IPv6 Payloads	9
5.3. MPLS Payloads	9
6. IANA Considerations	9
7. Security Considerations	10
8. Acknowledgements	10
9. References	10
9.1. Normative References	10
9.2. Informative References	11
Authors' Addresses	11

1. Introduction

Generic Routing Encapsulation (GRE) [RFC2784] can be used to carry any network layer protocol over any network layer protocol. GRE has been implemented by many vendors and is widely deployed on the Internet.

[RFC2784], by design, does not describe procedures that affect fragmentation. Lacking guidance from the specification, vendors have developed implementation-specific fragmentation strategies. For the most part, devices implementing one fragmentation strategy can interoperate with devices that implement another fragmentation

strategy. Operational experience has demonstrated the relative merits of each strategy. Section 3 of [RFC4459] describes four fragmentation strategies and evaluates the relative merits of each.

This memo documents a GRE fragmentation strategy that has been implemented by many vendors and deployed in many networks. It was written so that a) implementors will be aware of best common practice and b) those who rely on GRE will understand how implementations work. The scope of this memo is limited to point-to-point GRE tunnels. All other tunnel types are beyond the scope of this memo.

This memo specifies requirements beyond those stated in [RFC2784]. However, it does not update [RFC2784]. Therefore, a GRE implementation can be compliant with [RFC2784] without satisfying the requirements of this memo.

1.1. How To Use This Document

This memo is presented in sections. Section 2 reviews four fragmentation strategies presented in [RFC4459] and provides an overview the strategy described herein.

Section 3 defines generic requirements for GRE ingress routers. These include compliance with the specifications of [RFC2784] and Tunnel MTU Estimation and Discovery.

Section 4 defines procedures affecting generation of the GRE delivery header. It is divided into two subsections. Section 4.1 is applicable when GRE is delivered over IPv4 [RFC0791] and Section 4.2 is applicable when GRE is delivered over IPv6 [RFC2460].

Section 5 defines procedures for handling payloads that are so large that they cannot be forwarded through the GRE tunnel without fragmentation. Section 5.1 is applicable when the payload is IPv4, Section 5.2 is applicable when the payload is IPv6 and Section 5.3 is applicable with the payload is MPLS.

Section 6 discusses IANA considerations and Section 7 discusses security considerations.

1.2. Terminology

The following terms are specific to GRE and are taken from [RFC2784]:

- o GRE delivery header - an IPv4 or IPv6 header whose source address is that of the GRE ingress and whose destination address is that of the GRE egress. The GRE delivery header encapsulates a GRE header.

- o GRE header - the GRE protocol header. The GRE header is encapsulated in the GRE delivery header and encapsulates GRE payload.
- o GRE payload - a network layer packet that is encapsulated by the GRE header. The GRE payload can be IPv4, IPv6 or MPLS. Procedures for encapsulating IPv4 and IPv6 in GRE are described in [RFC2784]. Procedures for encapsulating MPLS in GRE are described in [RFC4023]. While other protocols may be delivered over GRE, they are beyond the scope of this document.
- o GRE payload header - the IPv4, IPv6 or MPLS header of the GRE payload
- o GRE overhead - the combined size of the GRE delivery header and the GRE header, measured in octets

The following terms are specific MTU discovery:

- o link MTU (LMTU) - the maximum transmission unit, i.e., maximum packet size in octets, that can be conveyed over a link. LMTU is a unidirectional metric. A bidirectional link may be characterized by one LMTU in the forward direction and another MTU in the reverse direction.
- o path MTU (PMTU) - the minimum LMTU of all the links in a path between a source node and a destination node. If the source and destination node are connected through an equal cost multipath (ECMP), the PMTU is equal to the minimum LMTU of all links contributing to the multipath.
- o tunnel MTU (TMTU) - the maximum transmission unit, i.e., maximum packet size in octets, that can be conveyed over a GRE tunnel without fragmentation. The TMTU is equal to the PMTU associated with the path between the tunnel ingress and the tunnel egress, minus the GRE overhead
- o Path MTU Discovery (PMTUD) - A procedure for dynamically discovering the PMTU between two nodes on the Internet. PMTUD procedures rely on a router's ability to deliver ICMP feedback to the host that originated a packet. PMTUD procedures for IPv4 are defined in [RFC1191]. PMTUD procedures for IPv6 are defined in [RFC1981].
- o Packetization Layer MTU Discovery (PLMTUD) - An extension of PMTUD that is designed to operate correctly in the absence of ICMP feedback from a router to the host that originated a packet. PLMTUD procedures are defined in [RFC4821]

The following terms are introduced by this memo:

- o fragmentable packet - all IPv4 packets with DF-bit equal to 0
- o non-fragmentable packet - all IPv4 packets with DF-bit equal to 1. Also, for the purposes of this document, all IPv6 packets are considered to be non-fragmentable.

2. Candidate Strategies and Strategic Overview

2.1. Candidate Strategies

Section 3 of [RFC4459] identifies the following tunnel fragmentation strategies:

1. Fragmentation and Reassembly by the Tunnel Endpoints
2. Signalling the Lower MTU to the Sources
3. Encapsulate Only When There is Free MTU
4. Fragmentation of the Inner Packet

In Strategy 1, the tunnel ingress router encapsulates the entire payload, without fragmentation, into a single GRE-delivery packet. It then forwards the GRE-delivery packet in the direction of the tunnel egress. If the GRE-delivery packet exceeds the LMTU of any link along the path to the tunnel egress, the router directly upstream of that link fragments it. The tunnel egress router reassembles the GRE-delivery packet, de-encapsulates its payload, and processes the payload appropriately.

In Strategy 2, the tunnel ingress router performs PMTUD procedures or some variant thereof (e.g., PLMTUD). When the tunnel ingress router receives a non-fragmentable IPv4 packet so large that it cannot be forwarded through the tunnel, it discards the packet and sends an ICMPv4 [RFC0792] Destination Unreachable message to the packet source, with type equal to 4 (fragmentation needed and DF set). The ICMP Destination Unreachable message contains a Next-hop MTU (as specified by [RFC1191]) and the next-hop MTU is equal to the TMTU associated with the tunnel. If the ICMPv4 message reaches the packet source, and if the packet source executes PMTUD procedures, the packet source adjusts its PMTU for the packet destination and emits subsequent packets with size less than the TMTU.

In Strategy 3, the network is engineered so that all network ingress links have LMTU less than the TMTU of any tunnel contained by the network. In this case, all packets entering the network are small

enough to be forwarded through any tunnel contained by the network, without fragmentation. The entire issue is thus avoided.

In Strategy 4, the tunnel ingress router performs PMTUD procedures or some variant thereof (e.g., PLMTUD). When the tunnel ingress router receives a fragmentable IPv4 packet so large that it cannot be forwarded through the tunnel without fragmentation, it fragments the payload and encapsulates each payload fragment in to a complete, separate GRE-delivery packet. It forwards those complete packets to the tunnel egress router which de-encapsulates them and forwards each payload fragment, individually and without re-assembly, to the payload destination. The payload destination reassembles packet.

Strategy 3 is attractive because it avoids fragmentation. However, networks cannot always be designed to meet the requirements of Strategy 3. When this is the case, Strategies 1, 2 and 4 become applicable.

Strategy 2 is also attractive, because it avoids fragmentation. However, Strategy 2 requires the payload source and the tunnel egress to execute PMTUD procedures. PMTUD procedures require ICMP feedback from downstream routers and fail when the network blocks required ICMP messages. Therefore, Strategy 2 can cause blackholing in networks that block ICMP.

Strategy 1 is an attractive alternative to Strategy 1, because it does not rely on PMTUD. However, Strategy 1 may not be feasible in many operational environments because it assigns the task of reassembly to the tunnel egress router. When the tunnel supports high data rates, reassembly at the tunnel egress is not cost-effective.

Strategy 4 moves the task of packet reassembly from the tunnel egress to the payload destination. However, it is applicable only when the payload is fragmentable. Furthermore, it requires the tunnel ingress router to perform PMTUD procedures and fails when the network blocks ICMP messages from tunnel interior to the tunnel ingress.

2.2. Strategic Overview

The fragmentation strategy described herein, has two modes of operation. The default mode resembles Strategies 2 and 4, above. When a GRE ingress router runs in the default mode, and it receives a non-fragmentable packet that is too large to forward through the tunnel, it behaves as described in Strategy 2, above. When it receives a fragmentable packet that is too large to forward through the tunnel, it behaves as described in Strategy 4, above. In neither case will the GRE ingress router fragment the GRE-delivery packet.

When GRE is delivered over IPv4, the DF-bit on the delivery header is always set to 1 (Don't Fragment).

Default mode operation is desirable with the following conditions are true:

- o the payload source supports PMTUD procedures
- o the tunnel ingress supports PMTUD procedures
- o the network does not block ICMP messages required by PMTUD

Realizing that some devices do not support PMTUD and that some networks indiscriminately block ICMP messages, the fragmentation strategy described herein includes a non-default mode, which incorporates some characteristics of Strategy 1, above.

When a GRE ingress router runs in the non-default mode, and it receives a non-fragmentable packet that is too large to forward through the tunnel, it behaves as described in Strategy 2, above. When the it receives a fragmentable packet that is too large to forward through the tunnel, it behaves as described in Strategy 4, above. In neither case will the GRE ingress router fragment the GRE-delivery packet. In this respect, the default and non-default modes are identical to one another.

However, if the ingress router delivers fragmentable payload over IPv4, it copies the DF-bit value from the payload header to the delivery header. Therefore, the GRE delivery packet may be fragmented by any router between the GRE ingress and egress. When this occurs, the GRE delivery packet is reassembled by the GRE egress.

The non-default mode of operation is desirable in some scenarios where networks block ICMP messages required by PMTUD.

3. Generic Requirements for GRE Ingress Routers

This section defines procedures that all GRE ingress routers must execute.

3.1. General

Implementations MUST satisfy all of the requirements stated in [RFC2784].

3.2. Tunnel MTU (TMTU) Estimation and Discovery

Implementations MUST maintain a running TMTU estimate. The TMTU associated with a tunnel MUST NOT, at any time, be greater than the LMTU associated with the next-hop towards the tunnel egress minus the GRE overhead.

Implementations SHOULD execute either PMTUD or PLMTUD procedures to further refine their TMTU estimate. If they do so, they MUST set the TMTU to a value that is less than or equal to the discovered PMTU minus the GRE overhead.

However, if an implementation supports PMTUD or PLMTUD for GRE tunnels, it MUST include a configuration option that disables those procedures. This configuration option may be required to mitigate certain denial of service attacks (see Section 7). When PMTUD is disabled, the TMTU MUST be set to a value that is less than or equal to the LMTU associated with the next-hop towards tunnel egress, minus the GRE overhead.

The ingress router's TMTU estimate will not always reflect the actual TMTU. It is only an estimate. When the TMTU associated with a tunnel changes, the tunnel ingress router will not discover that change immediately. Likewise, if the ingress router performs PMTUD procedures and tunnel interior routers cannot deliver ICMP feedback to the tunnel ingress, TMTU estimates may be inaccurate.

4. Procedures Affecting The GRE Delivery Header

This section defines procedures that GRE ingress routers execute while generating the GRE delivery header.

4.1. Tunneling GRE Over IPv4

By default, the GRE ingress router MUST set the DF-bit in the delivery header to 1 (Don't Fragment). Also, by default, the GRE ingress router MUST NOT emit a delivery header with MF-bit equal to 1 (More Fragments) or Offset greater than 0.

However, the GRE ingress router MUST support a configuration option that invokes the following behavior:

- o when the GRE payload is IPv6, the DF-bit on the delivery header is set to 0 (Fragments Allowed)
- o when the GRE payload is IPv4, the DF-bit value is copied from the payload header to the delivery header

When the DF-bit on the delivery header is set to 0, the GRE delivery packet may be fragmented by any router between the GRE ingress and

egress and the GRE delivery packet will be reassembled by the GRE egress.

4.2. Tunneling GRE Over IPv6

The GRE ingress router MUST NOT emit a delivery header containing a fragment header.

5. Procedures Affecting the GRE Payload

This section defines procedures that GRE ingress routers execute when they receive a packet a) whose next-hop is a GRE tunnel and b) whose size is greater than the TMTU associated with that tunnel.

5.1. IPv4 Payloads

If the payload is non-fragmentable, the GRE ingress router MUST discard the packet and send an ICMPv4 Destination Unreachable message to the payload source, with type equal to 4 (fragmentation needed and DF set). The ICMP Destination Unreachable message MUST contain an Next-hop MTU (as specified by [RFC1191]) and the next-hop MTU MUST be equal to the TMTU associated with the tunnel.

If the payload is fragmentable, the GRE ingress router MUST fragment the payload and submit each fragment to GRE tunnel. Therefore, the GRE egress router will receive complete, non-fragmented packets, containing fragmented payloads. The GRE egress router will forward the payload fragments to their ultimate destination where they will be reassembled.

5.2. IPv6 Payloads

The GRE ingress router MUST discard the packet and send an ICMPv6 [RFC4443] Packet Too Big message to the payload source. The MTU specified in the Packet Too Big message MUST be equal to the TMTU associated with the tunnel.

5.3. MPLS Payloads

The GRE ingress router MUST discard the packet. As it is impossible to reliably identify the payload source, the GRE ingress router MUST NOT attempt to send an ICMPv4 Destination Unreachable message or an ICMPv6 Packet Too Big message to the payload source.

6. IANA Considerations

This document makes no request of IANA.

7. Security Considerations

PMTU Discovery is vulnerable to two denial of service attacks (see Section 8 of [RFC1191] for details). Both attacks are based upon on a malicious party sending forged ICMPv4 Destination Unreachable or ICMPv6 Packet Too Big messages to a host. In the first attack, the forged message indicates an inordinately small PMTU. In the second attack, the forged message indicates an inordinately large MTU. In both cases, throughput is adversely affected. On order to mitigate such attacks, GRE implementations MUST include a configuration option to disable PMTU discovery on GRE tunnels. Also, they MAY include a configuration option that conditions the behavior of PMTUD to establish a minimum PMTU.

8. Acknowledgements

The authors would like to thank Jagadish Grandhi, Jeff Haas, John Scudder, Mike Sullenberger and Wen Zhang for their constructive comments. The authors also express their gratitude to an anonymous donor, without whom this document would not have been written.

9. References

9.1. Normative References

- [RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, September 1981.
- [RFC0792] Postel, J., "Internet Control Message Protocol", STD 5, RFC 792, September 1981.
- [RFC1191] Mogul, J. and S. Deering, "Path MTU discovery", RFC 1191, November 1990.
- [RFC1981] McCann, J., Deering, S., and J. Mogul, "Path MTU Discovery for IP version 6", RFC 1981, August 1996.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.
- [RFC2784] Farinacci, D., Li, T., Hanks, S., Meyer, D., and P. Traina, "Generic Routing Encapsulation (GRE)", RFC 2784, March 2000.

- [RFC4023] Worster, T., Rekhter, Y., and E. Rosen, "Encapsulating MPLS in IP or Generic Routing Encapsulation (GRE)", RFC 4023, March 2005.
- [RFC4443] Conta, A., Deering, S., and M. Gupta, "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", RFC 4443, March 2006.
- [RFC4821] Mathis, M. and J. Heffner, "Packetization Layer Path MTU Discovery", RFC 4821, March 2007.

9.2. Informative References

- [RFC4459] Savola, P., "MTU and Fragmentation Issues with In-the-Network Tunneling", RFC 4459, April 2006.

Authors' Addresses

Ron Bonica
Juniper Networks
2251 Corporate Park Drive Herndon
Herndon, Virginia 20170
USA

Email: rbonica@juniper.net

Carlos Pignataro
Cisco Systems
7200-12 Kit Creek Road
Research Triangle Park, North Carolina 27709
USA

Email: cpignata@cisco.com

Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: January 16, 2014

T. Eckert, Ed.
R. Penno
A. Choukir
C. Eckel
Cisco Systems, Inc.
July 15, 2013

A Framework for Signaling Flow Characteristics between Applications and
the Network
draft-eckert-intarea-flow-metadata-framework-01

Abstract

This document provides a framework for communicating information elements (a.k.a. metadata) in a consistent manner between applications and the network to provide better visibility of application flows, thereby enabling differentiated treatment of those flows. These information elements can be conveyed using various signaling protocols, including PCP, RSVP, and STUN.

Status of This Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

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."

This Internet-Draft will expire on January 16, 2014.

Copyright Notice

Copyright (c) 2013 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 Simplified BSD License.

Table of Contents

1. Introduction	2
2. Background	3
2.1. Deep packet inspection	4
2.1.1. Benefits	4
2.1.2. Limitation	4
2.2. Explicit signaling methods	5
3. Proposed framework	6
3.1. Overview	6
3.1.1. Common, application independent, IPFIX registered, information elements	6
3.1.2. Cross-protocol information element encoding rules . .	7
3.1.3. Anticipated Usage Models	7
3.1.3.1. Informational	8
3.1.3.2. Advisory	8
3.1.3.3. Service Request	9
3.1.4. Considerations for signaling of common information elements	9
3.1.4.1. Proxy originated information	9
3.1.4.2. Authentication	9
3.1.4.3. Common encoding	9
3.1.4.4. Usage Model to Protocol integration	10
3.2. Proposed common information elements	11
3.2.1. Bandwidth Attributes	11
3.2.1.1. Maximum Bandwidth	11
3.2.1.2. Minimum Bandwidth	12
3.2.1.3. Bandwidth Pool	12
3.2.2. Traffic Class Attributes	12
3.2.2.1. RFC4594-DSCP	12
3.2.2.2. Traffic Class Label (TCL)	12
3.2.3. Acceptable Path Attributes	13
3.2.3.1. Delay Tolerance	13
3.2.3.2. Loss Tolerance	13
3.2.4. Application Identification	13
3.2.4.1. RFC 6759 style application identification	13
3.2.4.2. URL style application identification	14
4. Acknowledgements	15
5. Informative References	15
Authors' Addresses	16

1. Introduction

This document provides a framework for communicating information elements (a.k.a. metadata) in a consistent manner between applications and the network to provide better visibility of application flows, thereby enabling differentiated treatment of those flows. These information elements can be conveyed using various signaling protocols, including PCP, RSVP, and STUN.

The framework is built around the definition of four key components:

1. A set of application independent information elements (IEs)
2. An encoding of these IEs that is independent of the signaling protocol used as transport
3. Usages of these IEs to support various transactional semantics
4. A mapping of one or more of these usages to an initial set of signaling protocols, including PCP, RSVP, and STUN

This document defines an initial set of IEs, a set of encoding rules, and initial usage model. The actual encoding is defined in [I-D.choukir-tsvwg-flow-metadata-encoding]. Additional documents define the mapping to specific signaling protocols (e.g. RSVP [I-D.zamfir-tsvwg-flow-metadata-rsvp], STUN [I-D.martinsen-mmusic-malice], and PCP [I-D.wing-pcp-flowdata])

2. Background

This section provides background on the motivation for the framework.

Identification and treatment of application flows are critical for the successful deployment and operation of applications based on a wide range of signaling protocols. Historically, this functionality has been accomplished to the extent possible using heuristics, which inspect and infer flow characteristics.

Heuristics may be based on port ranges, IP subnetting, or deep packet inspection (DPI), e.g. application level gateway (ALG). Port based solutions suffer from port overloading and inconsistent port usage. IP subnetting solutions are error prone and result in network management hassle. DPI is computationally expensive and becomes a challenge with the wider adoption of encrypted signaling and secured traffic. An additional drawback of DPI is that the resulting insights are not available, or need to be recomputed, at network nodes further down the application flow path.

The proposed solution allows applications to explicitly signal their flow characteristics to the network. It also provides network nodes

with visibility of the application flow characteristics and enables them to contribute to the flow description. The resulting flow description may be communicated as feedback from the network to applications.

2.1. Deep packet inspection

2.1.1. Benefits

Deep Packet Inspection (DPI) and other traffic observation methods (such as performance monitoring) are successfully being used for two type of workflows:

1. Provide network operators with visibility into traffic for troubleshooting, capacity planning, accounting and billing and other off network workflows. This is done by exporting observed traffic analysis via protocol such as IPFIX and SNMP.
2. Provide differentiated network services for the traffic according to network operator defined rule sets, including policing and shaping of traffic, providing admission control, impacting routing, permitting passage of traffic (e.g. firewall functions), etc.

Note: For the context of this document, we consider that DPI starts as early into packets as using ACLs with UDP/TCP port numbers to classify traffic.

2.1.2. Limitation

These two workflows, visibility and differentiated network services, are critical in many networks. However, their reliance on inspection and observation limits the ability to enable these workflows more widely.

- o Simple observation based classification, especially ones relying on TCP/UDP, ports often result in incorrect results due to port overloading (i.e. ports used by applications other than those claiming the port with IANA).
- o More and more traffic is encrypted, rendering deep packet inspection impossible or much more complex (e.g. needing to share encryption keys with network equipment).
- o Observation generally requires inspecting the control and signaling traffic of applications. This traffic may flow through a different network path than the actual application data traffic. Impacting the traffic behavior is ineffective in those scenarios.

- o Observation of control, signaling and data traffic with DPI will in general result in less insight into the applications intent than if the application was explicitly signaling its intent to the network.
- o Without explicit desire by the application to signal its intent to the network, it will also not consider to explicitly provide authentication to the network. DPI mechanism have a more difficult job in analyzing application traffic when authentication mechanisms are in use (if they even can)
- o Without explicit involvement of the application, network services leveraging DPI traffic classification impact the application behavior by impacting its traffic, but cannot provide explicit feedback to the application in the form of signaling.

2.2. Explicit signaling methods

There are a variety of existing and evolving signaling options that can provide explicit application to network signaling and serve the visibility and differentiated network services workflows where DPI is currently being used. It seems clear that there will be no single one-protocol-fits-all solution. Every protocol is currently defined in its own silo, creating duplicate or inconsistent information models. This results in duplicate work, more operational complexity and an inability to easily convert information between protocols to easily leverage the best protocol option for each specific use case. Examples of existing signaling options include the following:

- o RSVP is the original on path signaling protocol standardized by the IETF. It operates on path out-of-band and could support any transport protocol traffic (it currently supports TCP and UDP). Its original goal was to provide admission control. Arguably, its success was impacted by its reliance on router-alert because this often leads to RSVP packets being filtered by intervening networks. To date, more lightweight signaling workflows utilizing RSVP have not been standardized within the IETF.
- o NSIS (next Steps in Signaling) is the next iteration of RSVP-like signaling defined by the IETF. Because it focused on the same fundamental workflow as RSVP admission control as its main driver, and because it did not provide significant enough use-case benefits over RSVP, it has seen even less adoption than RSVP.
- o STUN is an on path, in-band signaling protocol that could easily be extended to provide signaling to on path network devices because it provides an easily inspected packet signature, at least for transport protocols such as UDP and SCTP. Through its

extensions TURN and ICE, it is becoming quite popular in application signaling driven by the initial use-case of automatically opening up firewall pinholes and determining the best local and remote addresses for peer-to-peer connectivity (ICE).

- o PCP is a protocol designed to support use cases similar to UPnP firewall traversal. It also can easily be extended to provide more generic application to network signaling for traffic flows. Unlike the prior protocols, it is not meant to be used on path end-to-end but rather independently on one "edge" of a traffic flow. It is therefore an attractive alternative (albeit with challenges under path redundancy) because it allows the introduction of application to network signaling without relying on the remote peer. This is especially useful in multi-domain communications.
- o In addition to these, depending on the devices where it is performed, different degrees of DPI may be used to achieve explicit signaling. For example, inspection of HTTP connections is often viable in high-touch network devices. Such inspection may provide explicit signaling if the application purposely keeps or inserts information elements that are meant to be signaled to the network in the clear, or knowingly uses an encryption scheme shared with the network.

Rather than encourage independent, protocol specific solutions to this problem, this document provides a protocol and application independent framework that can be applied in a consistent fashion across the various protocols.

3. Proposed framework

3.1. Overview

The proposed framework includes the following elements:

3.1.1. Common, application independent, IPFIX registered, information elements

An application media flow may be expressed as a set of information elements that are defined and registered like observation-based IPFIX attributes. We propose leveraging IPFIX as the information model (not necessarily as the transport signaling) for the following reasons:

- o As outlined above, export of traffic information is one of the two big workflows. IPFIX is arguably the most flexible, extensible

and best defined option for this. Leveraging the same information model for flow characteristics facilitates export of this information via IPFIX.

- o IPFIX allows for IETF/IANA standardized information elements, but also for unambiguous vendor-defined attributes by including the so-called PEN (Private Enterprise Number) into the information element type. Note that IPFIX has ongoing work to better disseminate vendor specific registration of attributes. The framework defined here expects to be able to leverage the output of that work.

3.1.2. Cross-protocol information element encoding rules

The majority of the protocols listed previously (RSVP, NSIS, STUN/ICE, PCP) require (or favor) compact binary encoding of information elements. This is natively supported by the information element registration of IPFIX.

The IPFIX registry defines each information element's data-type, and there is a native binary network encoding for each of these types. At a minimum, every protocol leveraging common information elements would need to use an encoding that identifies the information element's PEN and IE-ID, and that leverages network standard binary encoding of the value including the length of the value. Including the length of the value into the encoding is required for extensibility because otherwise new information elements could not be introduced without first having all network devices know the data-type, and therefore the length, of the information element. Leveraging network standard binary encoding is equally important to permit network elements to propagate information elements from one protocol to another protocol without understanding the information elements data-type.

In protocols that are not constrained to binary encoding, it is nevertheless highly desirable to include the equivalent information and therefore permit propagation between binary and non-binary transport of information elements without having to understand all information elements.

3.1.3. Anticipated Usage Models

The signaling of information elements may be from application to the network or from network to application. When signaled within a given protocol, the information elements may be interpreted independently of that protocol, or it may be used in combination with the given protocol.

3.1.3.1. Informational

The most simplistic usage model is one in which applications signal information elements describing their anticipated or existing flows into the network along the path of those flows without expecting or requiring anything back from the network. Network elements along the flow path may or may not do something with this information.

This "informational" usage model enables network elements along the path to support the workflows traditionally performed via DPI mechanisms, as described previously.

3.1.3.2. Advisory

This usage model extends the "informational" usage in that the application expects or requests some information back from the network. With this usage, the same information elements apply and may be communicated by the application into the network, but the application indicates its interest in receiving some feedback.

Default values are defined for each information element to unambiguously support cases in which an application does not have a valid value to communicate with the network; rather, it wants the network to provide a value back to it in response. In essence, this allows an application to ask a question and receive an answer from the network. Of course, a network element may provide similar feedback for cases in which an application communicated a non-default value as well. Network elements may also provide unsolicited advisory feedback.

In all cases, applications are not guaranteed to receive an answer or any specific service from the network. In the event an answer is provided, that answer is similarly not a guarantee of any specific service or treatment by the network. It is to be interpreted as advisory only.

As mentioned previously, the same information elements are used in the signaling from the application to the network as well as from the network to the application. The underlying transport protocol used to carry the information elements is expected to provide the necessary request/response semantics or some other mechanism by which the communication in both directions can be tied together.

3.1.3.3. Service Request

This usage model extends the "advisory" usage to operate as an explicit service request. Unlike the advisory usage, information elements signalled by the application are interpreted by network elements within the context of a service request, and information elements signalled by the network back to the application are interpreted within the context of a response to that request.

As with the advisory usage, the same information elements are used in the signaling from the application to the network as well as from the network to the application. The underlying transport protocol used to carry the information elements is expected to provide the necessary service request/response semantics.

3.1.4. Considerations for signaling of common information elements

3.1.4.1. Proxy originated information

The goal of this framework is to enable applications to explicitly signal common information elements about their traffic flows and optionally receive common information elements from the network as feedback. Nevertheless, it is clear that broad adoption of such technology is improved by enabling the use of proxies. The proxies can provide or amend the flow description information in the absence of Flow Metadata support by the application itself.

3.1.4.2. Authentication

Common information elements should provide for cryptographic authentication by the sender. In general the authentication provides some form of identification of the sender and proves that the common information elements covered by the authentication were originated from, or approved by, that identity.

3.1.4.3. Common encoding

A companion document [I-D.choukir-tsvwg-flow-metadata-encoding] covers recommended encoding rules that take the following aspects into account:

- o Compact binary encoding rules
- o Signaling for both sent and received traffic flows
- o Signaling of standard and vendor specific information elements

- o Minimizes protocol specific definition required to add informational or advisory common information elements into existing transactions
- o Signaling of feedback from the network
- o Identification of originator to support proxies and facilitate mitigation between common information elements from different originators
- o Signaling of authenticators

3.1.4.4. Usage Model to Protocol integration

There is a range of options for how this framework is integrated with a particular transport protocol. We describe two examples we consider useful:

3.1.4.4.1. Common transport informative integration

1. A transport protocol signaling method is defined to carry the common encoded information elements at least in signaling from application to network.
2. If the transport by itself does not already have a mechanism to indicate a purely informative protocol transaction, then a protocol specific indication for this is added.

In result, this integration achieves two option:

1. Informative common information elements can be sent from application to network by using the protocol's method to indicate the purely informational protocol transaction. This option effectively leverages the protocol as transport for additional informative attribute based services without impacting the services and transactions of the protocol otherwise.
2. Informative common information elements can be sent alongside an existing protocol transaction. In this case they may either be ships in the night (triggering informative attribute based services), or they may additionally be used by the policy rules of the protocol transaction itself which could be advisory or service request. All feedback of the transaction would still rely on protocol specific information element (common information elements only used from host to network).

This integration is for example defined in [I-D.wing-pcp-flowdata], [I-D.zamfir-tsvwg-flow-metadata-rsvp], and [I-D.martinsen-mmusic-malice].

3.1.4.4.2. Common transport advisory integration

In addition to the common transport informative integration, the transport encoding is extended to carry the common transport information element in feedback messages from the network to the host /application. The method to indicate informative only transaction, when sending to the network is used to indicate advisory only transaction when signaling from the network.

This option primarily enables informative and advisory usage models, but it can equally interact with pre-existing service-request options of the transport protocol and impact advisory feedback or the service request itself based on that interaction.

3.2. Proposed common information elements

The section defines an initial set of common information elements. These information elements are intended to be added to the set of IANA standardized information elements either by this or associated documents. Additional documents are expected to define additional attributes that can use either IANA or other vendor-PEN.

All information element definition must include the following:

1. Default value to be provided by an application when it does not have an informative value to provide to the network, but is interested in receiving an advisory value of the attribute from the network. If no advisory feedback is requested, and no informative value is known, the attribute may simply not be sent.
2. Conflict resolution in the presence of different values for the same information element (e.g. two peers signaling information elements for both the upstream and downstream direction of a flow include different values for the information element)

3.2.1. Bandwidth Attributes

3.2.1.1. Maximum Bandwidth

This attribute is used to convey the maximum sustained bandwidth for the flow. It is a 64 bit value and is specified in bits per second.

Default Value: 0

Conflict Resolution: Minimum for the set of non-default values

3.2.1.2. Minimum Bandwidth

This attribute is used to convey the minimum sustained bandwidth for the flow. It is a 64 bit value and is specified in bits per second. Not sending the Minimum Bandwidth is equivalent to sending the same value as for Maximum Bandwidth.

Default Value: 0

Conflict Resolution: Minimum of the set of non-default values

3.2.1.3. Bandwidth Pool

This attribute is used to convey that the traffic dynamically shares bandwidth with other traffic using the same Bandwidth Pool. Variable length GUID (Global Unique ID) of at least 48 bits. The Maximum Bandwidth used by the pool is the largest Maximum Bandwidth indicated by any member, the Minimum Bandwidth of the Pool is the largest Minimum Bandwidth indicated by any member.

3.2.2. Traffic Class Attributes

3.2.2.1. RFC4594-DSCP

This attribute is used to convey the DSCP value appropriate for the flow. It is an 8 bit value. Values signaled are assumed to be in compliance with [RFC4594] or backward compatible extensions thereof. Other values are undefined.

Default Value: 0xff

Conflict Resolution: tbd

3.2.2.2. Traffic Class Label (TCL)

The data type of this information element is a string. It carries the Traffic Class Label defined in [I-D.ietf-mmusic-traffic-class-for-sdp]. Depending on the outcome of that drafts standardization, the version carried as an information element may be slightly expanded over the its definition for SDP. The TCL is a structured string of the form:

<category>.<application>(.adjective)(.adjective)

category and application provide a base categorization of the traffic class that attempts to provide a simplified and extensible, framework

for the traffic class definitions in [RFC4594]. These base classifications can be refined with zero or more adjectives. Examples of a TCL is "conversational.video.avconf".

Default Value: Empty string

Conflict Resolution: tbd

3.2.3. Acceptable Path Attributes

3.2.3.1. Delay Tolerance

This attribute is used to convey the delay tolerance of an application with respect to the associated flow. When provided by a network element, it indicates the delay tolerance expected of the application with respect to the associated flow. It is a 16 bit field defined in terms of milliseconds.

Default Value: 0

Conflict Resolution: For application to network, the minimum of the set of non-default values. For network to application, the maximum of the set of non-default values.

3.2.3.2. Loss Tolerance

This attribute is used to convey the loss tolerance of an application with respect to the associated flow. When provided by a network element, it indicates the loss tolerance expected of the application with respect to the associated flow. It is a 16 bit field defined in terms of hundredths of a percent of dropped packets (e.g. 5 == 0.05%, 150 == 1.50%, etc.)

Default Value: 0

Conflict Resolution: For application to network, the minimum of the set of non-default values. For network to application, the maximum of the set of non-default values.

3.2.4. Application Identification

Application identification is clearly one of the more difficult classification goals. The proposals included here are as of yet not widely vetted:

3.2.4.1. RFC 6759 style application identification

[RFC6759] defines the IPFIX IE-IDs that permit both IANA and vendor specific application identification. Though defined for observation (a.k.a.: DPI), it could also be used with explicit signaling from applications.

Applications that use one of the protocols for which there is an IANA port allocation could explicitly indicate this port via the IANA-L4 engine-id in their application to network signaling. This would identify the application even if the application is not using the IANA assigned port for it. This covers cases in which applications use ports other than registered, such as HTTP servers running on other than 80, or when ports get mapped due to PAT.

To avoid collision with DPI exported IANA-L4 classification, it is necessary to assign a new engine-id for application-self assigned IANA-L4 classification (e.g. new engine-id for IANA-L4-SELF-ASSIGNED). If an application vendor has a PEN, the application can use a PANA-L7-PEN classification with the PEN of the originating application vendor. Likewise, if applications are in general made available via "market" type reseller mechanism (common in mobile device applications), then the application vendor could request an application identification from the market owner and leverage the market owners PEN.

3.2.4.2. URL style application identification

One problem with [RFC6759] style application identification especially non-IANA registered ones is the complexity in making all network elements learn the semantic of the numeric encoding of e.g. the PANA-L7-PEN information element in signaling protocols that only use the numeric encoding of information elements. The second problem may be to determine what PEN to use, because not every developer of an application may be a company that has a PEN or otherwise would intend to apply for one. Application identification via a URL encoded string information element is a way to overcome both issues. Today, almost all applications have some DNS domain associated with them through which they are being marketed or that belongs to the company developing the application. Therefore, one simple form of self assigned application identification is a new IPFIX information element: `UrlAppId`. The value of this information element is an abbreviated URL of the following form:

```
<fqdn> / <app-name> /[ <version> | <other-details> ]
```

The idea is that the owner of <fqdn> (fully qualified domain name) is assigning an <app-name>, and by signaling both <domain-name> and <app-name>, this information element provides a self-identifying, unambiguous application identification.

Example:

example.com/network-lemmings/sdn-edition

A game publishing house or application market operator with the domain name example.com is initially allocating the UrlAppId example.com/network-lemmings to that application. After 35 years, a new variant of the game is released, the SDN edition, and the app-developer decides that it would best like to distinguish this application variant by the above UrlAppId example.com/network-lemmings/sdn-edition.

In general, different traffic flows within a single application should best not be distinguished via the UrlAppId, but instead rely on attributes more specifically targeted for that purpose (such as the TrafficClassLabel). If there is no adequate better attribute defined, application developers may choose to use the other-details section of the UrlAppId to distinguish flows within the same application.

Formally, the only requirement against the UrlAppId is that the fqdn part is a DNS domain owned by the assigner, and that the rest of the string after the first / is as self explanatory as possible.

It should be noted that in the context of DPI, classification of web-based application traffic is very often performed by URL inspection of HTTP traffic. This proposed intent based information element leverages that model and makes it usable where it can not be currently used with just DPI: encrypted HTTP, non-HTTP applications, HTTP applications with non-descriptive URLs, etc.

4. Acknowledgements

The authors would like to thank Dan Wing, Anca Zamfir, Paul Jones, and Tirumaleswar Reddy for their valuable contributions to this document.

5. Informative References

- [I-D.choukir-tsvwg-flow-metadata-encoding]
Eckert, T., Zamfir, A., Choukir, A., and C. Eckel,
"Protocol Independent Encoding for Signaling Flow
Characteristics", draft-choukir-tsvwg-flow-metadata-
encoding-00 (work in progress), July 2013.
- [I-D.ietf-mmusic-traffic-class-for-sdp]
Polk, J., Dhesikan, S., and P. Jones, "The Session
Description Protocol (SDP) 'trafficclass' Attribute",

draft-ietf-mmusic-traffic-class-for-sdp-04 (work in progress), July 2013.

[I-D.martinsen-mmusic-malice]

Penno, R., Martinsen, P., Wing, D., and A. Zamfir, "Meta-data Attribute signaling with ICE", draft-martinsen-mmusic-malice-00 (work in progress), July 2013.

[I-D.wing-pcp-flowdata]

Wing, D., Penno, R., and T. Reddy, "PCP Flowdata Option", draft-wing-pcp-flowdata-00 (work in progress), July 2013.

[I-D.zamfir-tsvwg-flow-metadata-rsvp]

Eckert, T., Zamfir, A., and A. Choukir, "Flow Metadata Signaling with RSVP", draft-zamfir-tsvwg-flow-metadata-rsvp-00 (work in progress), July 2013.

[RFC4594]

Babiarz, J., Chan, K., and F. Baker, "Configuration Guidelines for DiffServ Service Classes", RFC 4594, August 2006.

[RFC6759]

Claise, B., Aitken, P., and N. Ben-Dvora, "Cisco Systems Export of Application Information in IP Flow Information Export (IPFIX)", RFC 6759, November 2012.

Authors' Addresses

Toerless Eckert (editor)
Cisco Systems, Inc.
San Jose
US

Email: eckert@cisco.com

Reinaldo Penno
Cisco Systems, Inc.
170 West Tasman Drive
San Jose 95134
USA

Email: repenno@cisco.com

Amine Choukir
Cisco Systems, Inc.
Lausanne
CH

Email: amchouki@cisco.com

Charles Eckel
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134
US

Email: eckelcu@cisco.com

Network Group
Internet-Draft
Intended status: Informational
Expires: January 17, 2014

S. Matsushima
Softbank Telecom
R. Wakikawa
Softbank Mobile
July 16, 2013

Stateless user-plane architecture for virtualized EPC (vEPC)
draft-matsushima-stateless-uplane-vepc-01

Abstract

We envision a new mobile architecture for the future Evolved Packet Core (EPC). The new architecture is designed to support the virtualization scheme called NFV (Network Function Virtualization). In our architecture, the user plane of EPC is decoupled from the control-plane and uses routing information to forward packets of mobile nodes. Although the EPC control plane will run on hypervisor, our proposal does not modify the signaling of the EPC control plane. The benefits of our architecture are 1) scalability, 2) flexibility and 3) Manageability. How to run the EPC control plane on NFV is out of our focus in this document.

Status of This Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

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."

This Internet-Draft will expire on January 17, 2014.

Copyright Notice

Copyright (c) 2013 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 Simplified BSD License.

Table of Contents

1. Introduction	2
1.1. The Benefits of NFV	3
2. Motivations and Requirements, - Why IETF? -	4
2.1. Motivations	4
2.2. Requirements	5
3. Stateless user-plane architecture for virtualized EPC	7
3.1. Architecture Overview	8
3.2. Protocol Overview	9
3.3. Control-plane awareness of stateless user-plane	13
3.4. Routing mechanism	13
3.5. IPv4 Support	16
4. Operational Considerations	17
4.1. Scalability	17
4.2. Backward Compatibility	18
5. IANA Considerations	18
6. Security Considerations	18
7. References	18
7.1. Normative References	18
7.2. Informative References	19
Authors' Addresses	20

1. Introduction

3GPP introduces Evolved Packet Core (EPC) that is fully IP based mobile system for LTE and -advanced in their Release-8 specification and beyond. Operators are now deploying EPC for LTE services and encounter rapid LTE traffic growth. There are various activities to offload mobile traffic in 3GPP and IETF such as LIPA, SIPTO and DMM. The concept is similar that traffic of OTT (Over The Top) application is offloaded at entity that is closer to the mobile node (ex. eNodeB or closer anchor).

Likewise, overload of signaling (control plane) is also increasing day by day. Network operators expect recent innovation and trends of NFV (Network Function Virtualization) to solve this overloaded control plane. NFV is discussed at the ETSI NFV ISG and is introduced in [NFV-WHITEPAPER]. Mobile operator's network is built with variety of proprietary hardware appliances today. If we can get rid of these physical appliances and could shift to a cloud-based

service, we will have a lot of benefits explained in the next section. This document assumes that NFV will push networking functions currently run on dedicated hardware onto a cloud network. Expected network functions are Mobility Management Entity (MME), Serving Gateway (SGW) PDN Gateway(PGW), etc. With NFV, EPC can be operated onto servers/hyper-visors. We name it virtualized-EPC (vEPC) in this document.

This document uses a lot of 3GPP specific terms. These terms can be found mostly at [RFC6459].

1.1. The Benefits of NFV

This section briefly explains the benefits of NFV. The detailed benefits can be found in [NFV-WHITEPAPER]. Although today's eco-system of EPC appliances might be affected, we believe there are various approaches to enhance current eco-system and migrate to new NFV approaches. For example, operators could pay monthly recurring charges for the NFV services and operations to vendors, instead of one-time purchase and a little maintenance cost.

- o [Flexible Network Operations]: The control functions of EPC are no longer in appliances deployed widely in operator's network and can be run at hypervisor (cloud). It is easier to add and/ or delete functions from the services, because no physical construction is needed. Network operations will be much simpler and easier because complications of today's network are pushed to NFV (i.e. hypervisor).
- o [Flexible Resource Managements]: The EPC functions can be run on hypervisor and are now less dependent on proprietary hardware. Adding additional resources is easier in hypervisor, while adding or replacing physical appliances require installation, construction, configuration, and even migration plan without service cutoff. A hypervisor can be also shared across various functions such as PGW, SGW and MME. NFV also brings multi-tenancy and allows a single platform for different services and users. The operator can optimize resources and costs to share a NFV platform for multiple customers (ex. MVNO customers) and services (ex. multiple APNs).
- o [Faster Speed of Time to Market]: When an operator wants a new function to its network and services, the operator needs to negotiate appliance vendors to implement the new functions or to find alternative equipment supporting the new function. It takes a longer time to convince the vendors, or to replace existing hardware. However, if functions can be implemented as a software, it is much faster to implement the functions on NFV. Even the

operator may implement them and try the new functions by themselves. Field trial is also getting easier because of no physical installation or replacement. You may turn on a new function in NFV and observe how the new function behaves in your network. NFV can save preparation time and tuning time of the new function.

- o [Cost Optimization]: Last but not least, Cost is the most important motivation for operators to realize NFV. Operators can remove many of proprietary appliances from its network and replace them with industry standard servers, switches and routers. In addition, it is easy to scale up and down operator's services so that resources can be always tuned to the size of services. In addition, operational costs led by any physical hardware such as power supply, maintenance, installation, construction and replacement can be minimized or even removed. The network design can be simpler, because complicated functions could be handled by NFV. That simple operation may enable automatic configurations and prevent unnecessary trouble-shooting. As a result, CAPEX and OPEX can be always optimized and lowered.

2. Motivations and Requirements, - Why IETF? -

2.1. Motivations

What is a role of IETF to realize vEPC in the future? IETF is not the right place to discuss, for instance, how to run MME on hypervisor. An important IETF activity must be to decouple the control- and user- planes of mobility protocols used in EPC. In doing so, NFV-enabled solutions can be easily designed and implemented with interoperability across multiple vendors and platforms. Otherwise, NFV solutions can be easily fragmented due to many proprietary solutions for the protocol separations. As stated in [NFV-WHITEPAPER], interoperability is highly important.

In the past, IETF has developed tunnel based mechanisms for mobile nodes such as Mobile IPv6 [RFC6275][RFC5555], Proxy Mobile IPv6 [RFC5213][RFC5844] and NEMO [RFC3963]. Similarly, 3GPP has developed tunnel protocols called GPRS Tunneling Protocol (GTP). These tunnel-based protocols establish a data path for a mobile node between the mobile node and an anchor point (s). There is a case where an access router terminates a tunnel instead of a mobile node (ex. Proxy Mobile IP). In 3GPP, a tunnel is established between SGW and PGW per a mobile node by either Proxy Mobile IPv6 or GTP. The control and the user planes of these mobility protocols are tightly related and cannot be decoupled. The signaling like Binding Update and user's packets are routed along a same path in EPC. It might be necessary to extend these mobility protocols for the user- and control- planes

separation. The protocol separation of Mobile IP is discussed in [I-D.yokota-dmm-scenario].

Alternatively, if vEPC was realized, we should have an opportunity to re-visit the basic architecture of mobility system. Instead of tunneling packets on today's EPC, why can't we just route packets to a mobile node? Since a role of the user plane is "routing", BGP and other routing protocols could be used to forward UE's traffic. This document introduces a BGP-based solution. Software Defined Networking (SDN) can be an alternative solution. Open Flow and other relevant protocols can setup the forward path dynamically according to UE's states available in the control plane.

We have to remember that there is a good reason of adapting tunneling in Mobile IP based solutions, that is global mobility and signaling. A mobile node should be able to move anywhere on the Internet and be reachable from anyone on the Internet. There were routing based global mobility solutions like Boeing global mobility [Boeing-BGP] and WINMO [RFC6301]. In these proposals, BGP was used to propagate forwarding information of mobile nodes to the Internet. Whenever a mobile node changes its point of attachment, the route must be updated. Due to scalability and stability issues of the Internet, this solution was not recommended by IETF [Boeing-BGP]. However, as Boeing showed, it is doable to support global mobility by using BGP routing update. If scalability is not your concern, a routing based approach becomes a candidate of the mobility solution.

While global mobility is important, the "reality" is that your cell phones (i.e. UE/mobile node) are moving just within an operator's network and fully controlled in your local EPC. If mobility is limited within an operator, we believe a routing based approach is feasible and practical for today's mobile system. Instead of dedicated proprietary equipment like SGW and PGW to manage a tunnel path for a mobile node, multiple industry standard routers and switches are configured in the user plane. These switches and routers receive mobile nodes' forwarding information from the control plane of vEPC by routing update.

2.2. Requirements

Requirements of our stateless user plane for vEPC are followings.

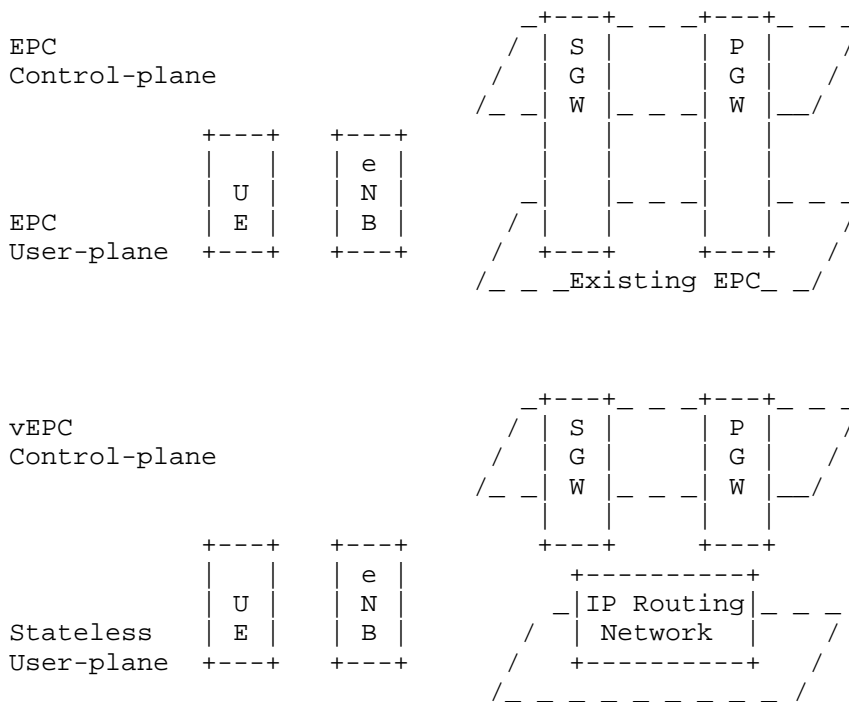
NFV Support

The future EPC architecture must support NFV capability. The control plane of EPC operated on NFV framework is named "virtualized EPC (vEPC)" in this document. The control plane of vEPC should keep backward compatibility with the today's EPC's control plane. It means this document doesn't modify

the control plane at all. It only assumes software-based MME, SGW, and PGW run on hypervisor.

Separation of Control- and User- Planes

Due to tight relationship of the control- and user- planes in today's EPC, resource increase is always provisioned to both planes at once. It prevents flexible resource arrangement and introduces high capital investment and over-provisioned resources to one of planes. If NFV is deployed, it is expected that computing resources can be independently allocated to the control planes of the vEPC in a flexible manner.



NFV enabled EPC architecture

Figure 1

Figure 1 shows a possibility that the entities of EPC Control- plane are virtualized in generic cloud environment, however user packets won't go through those virtualized EPC

nodes. Decoupling User-plane from the Control-plane entities will be made virtualized Control-plane nodes relax hypervisor data- path capacity requirements. On the other hand, decoupled User-plane into IP routing network will be agnostic from sessions and bearers states, of which are generated and maintained in the Control-plane. In terms of IP routing, forwarding packets through the networks is based on the destination address of the packets evaluated with network reachable information in the routing table that accommodated in the routing nodes. To forward EPC User-plane packets correctly, those states must be indicated by network reachable information.

Flat Design for Distributed Operation

Today's 3GPP architecture introduces PDN gateway (PGW) as a gateway to external networks like the Internet. PGW manages all traffic from and to UEs and could be a bottleneck and single point of failure of network connectivity. In addition, due to recent rapid traffic increase, it is important to perform traffic engineering and to offload traffic to multiple locations (ex. SGW, PGW, eNodeB). For enhancements of traffic engineering capability, more flat design with multiple gateways is expected so that traffic can be distributed to all these gateways. There were proposals how to enable flat design to (Proxy) Mobile IP such as [I-D.wakikawa-mext-haha-interop2008] in IETF. Distributed Mobility Management (DMM) Working Group has also discussed how to extend Mobile IP-based solutions to support traffic distribution in an optimal way by removing centrally deployed anchors that is like a Home Agent.

Stateless in User Plane

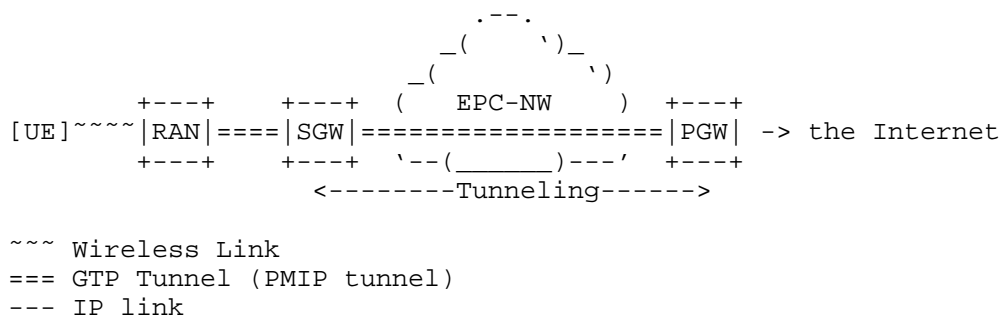
Ultimate goal of vEPC is to remove all mobility specific states from the forwarding nodes in the user-plane of vEPC. If we succeed in this, industry standard routers and switches can be used to forward mobile nodes traffic in the user plane of vEPC. A mobile node's specific states are kept in both an IP header of the mobile node's packets and a routing entry of the mobile node. The detail is described in Section 3.2

3. Stateless user-plane architecture for virtualized EPC

This section explains our solution that is the stateless user-plane architecture for vEPC. This solution is basically a combination of existing protocols defined in IETF. A minor extension might be needed but it should be easily addressed in IETF. We first introduce our architecture and then protocol overview.

3.1. Architecture Overview

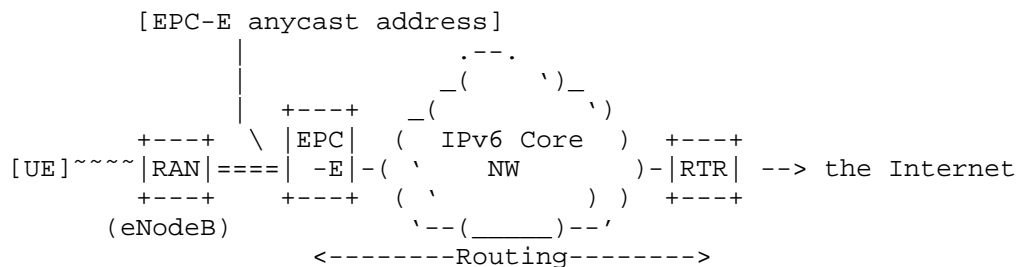
Figure 2 shows the user plane of the current EPC architecture. A tunnel is established between SGW and PGW by either Proxy Mobile IP or GTP. PGW is an anchor point of UE for incoming packets. All the packet destined to UE is routed first to PGW. The UE's packets are intercepted by PGW and tunneled to SGW. SGW then forwards the packet to UE via access points (i.e. eNodeB) over Radio Area Network (RAN).



User plane of the current EPC

Figure 2

Figure 3 is our proposed user plane of vEPC. The control plane is not shown in this figure.



User plane of vEPC

Figure 3

We introduce two new entities such as

EPC Edge Router (EPC-E)

EPC-E is located at the same place of today's SGW and terminates GTP tunnel established with eNodeB (RAN). EPC-E supports the user plane functions of SGW and PGW. EPC-E is configured an anycast address to the network interface facing to eNodeB. The eNodeB establishes a GTP tunnel per UE with this anycast address. Thanks for anycast address, UE's traffic forwarded by eNodeB is always routed to the closest EPC-E of UE. EPC-E is a router and maintains routing information of every UE that is notified by the control plane. Detail of routing mechanism can be found in Section 3.4.

Router (RTR)

It is a regular IP router. The control plane of vEPC distributes routing information of every UE by a routing protocol like BGP. Therefore any additional protocols other than routing protocols are not needed for RTR. Multiple RTRs can be configured anywhere in the user plane of vEPC. RTRs announce UE's routing information to the external network (ex. The Internet).

As you see in Figure 3, we omit a tunneling mechanism originally established between SGW and PGW for routing UE's packets in the user plane. By removing this tunnel, UE's packets are forwarded to and from the Internet according to routing tables on routers in the core network. Note that, although we remove the tunnel for UE's traffic in the user plane, the control-plane signaling stays same in the control plane. If Proxy Mobile IP is used for this tunnel, Proxy Binding Update and Acknowledgment are exchanged between PGW and SGW that are managed by NFV on servers/hyper-visor. Instead of a tunnel setup, states created by Proxy Mobile IP are distributed to all routing entities (EPC-E and RTR) by a routing protocol. From the user plane point of view, these states are just seen as routing entries. EPC-E and RTR are not involved in any signaling of the control plane. The control plane just injects routing information to EPC-E and RTR to setup routing paths to and from UEs.

Although this architecture just uses IPv6 core network, it supports both IPv4 and IPv6 packets. The detailed operation of IPv4 support will be discussed in Section 3.5.

3.2. Protocol Overview

This section gives an example of protocols used for vEPC. Figure 4 is the procedure of the PDN connection setup in vEPC. This figure is copied from the section 3 of [RFC6459]. All the steps from (1) to (13) are same as the original except for NFV-based MME, SGW, PGW, HSS, and AAA.

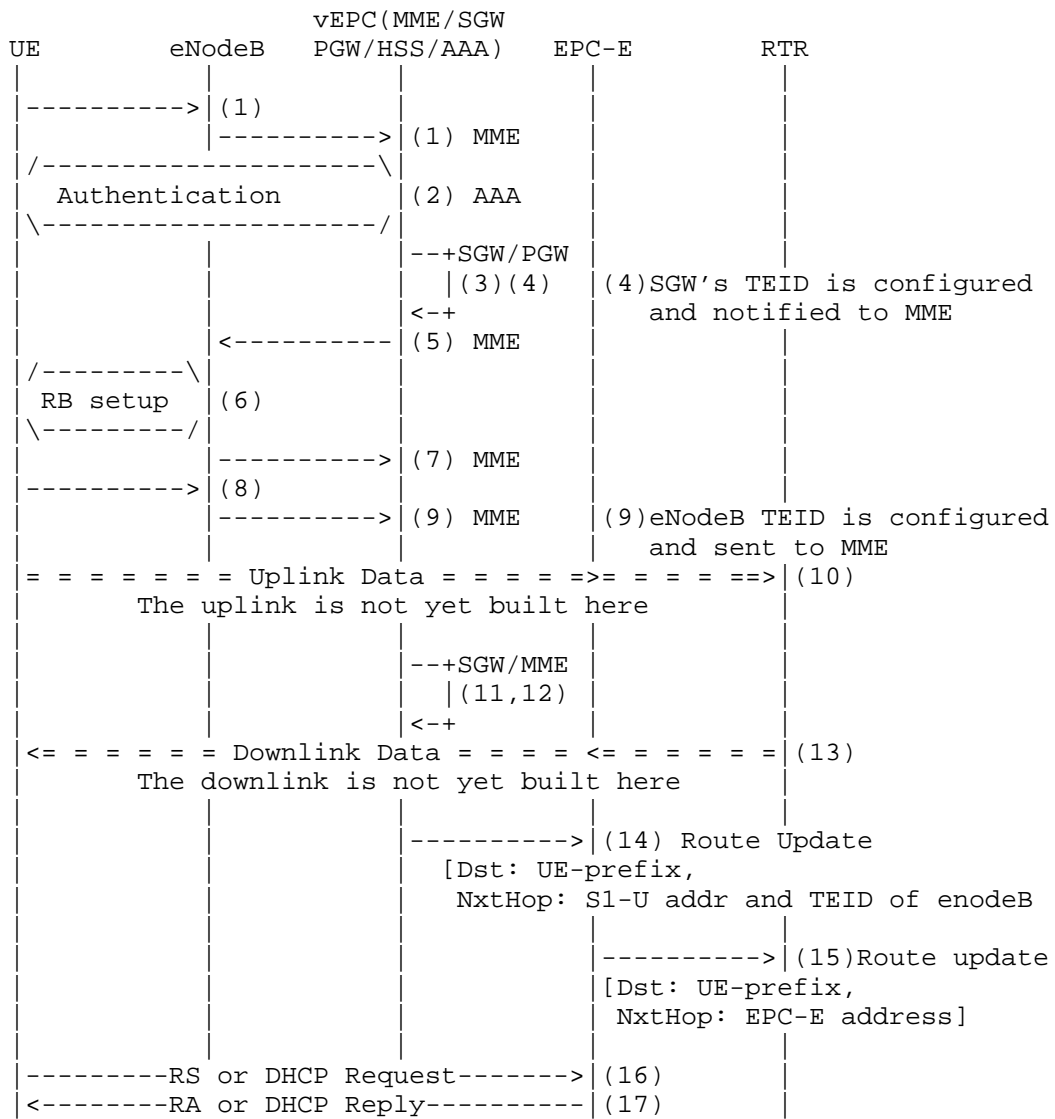
The vEPC introduces two new steps, (14) and (15), to setup paths in the user-plane after finishing all the signaling on the control-plane. (16) and (17) are the steps to assign IP address to the mobile node.

In (14), vEPC advertises a routing information of UE whose next hop is set to GTP tunnel between eNodeB and UE in RAN. In order to distribute a route entry of the UE's prefix from vEPC to all EPC-E, [I-D.vandeveldede-idr-remote-next-hop] enables BGP to carry the tunnel information as Network Layer Reachability Information (NLRI) within BGP route. In our document, we use GTP tunnel endpoint information as a next-hop of the route entry of UE's prefix. The GTP information is a combination of SI-U address and TEID of UE's attaching eNodeB.

BGP has already been extended for BGP speakers to advertise tunneling information to its peers [RFC5512], but [RFC5512] does not support GTP as a tunnel type. Moreover, it assumes only a single tunnel between a pair of BGP speakers although there are multiple tunnels between RAN and vEPC in cellular system. To keep compatibility to RAN architecture, it is not possible for eNodeB to act as a BGP speaker. Some mechanism will thus be required to advertise and to reflect tunnel endpoint routes to EPC-E instead of eNodeB. BGP remote next-hop will be dealt with these issues.

In step (15), EPC-E can aggregate multiple UE's prefixes into less BGP routes as a part of normal routing operation within operator's network.

When tunnel endpoint is updated by UE hand-over between eNodeBs, vEPC must refresh the route of UE with the updated tunnel endpoint as new remote next-hop. The updated route should be immediately advertised to all the EPC-Es. In the case of UE detachment, vEPC simply removes the route of the detached UE.



Extended PDN Connection Setup Procedure (copied Figure 8 of RFC6459)

Figure 4

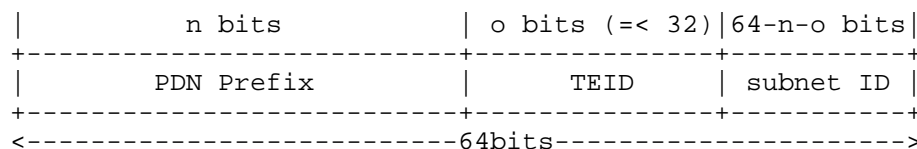
UE requests an IPv6 prefix for its address assignment in the step (16). In our architecture, an IPv6 prefix is still assigned by vEPC in the control plane, as PDN-GW does in the legacy EPC. However, EPC-E is responsible to deliver the IPv6 prefix to UE by DHCP or Stateless address autoconfiguration (SLAAC).

We now explain how EPC-E can know the prefix assigned to UE from vEPC for address configuration steps (16 and 17). When (1) to (15) are completed, vEPC has already advertised the UE's prefix as route information to all the EPC-E. Therefore, when EPC-E receives a packet of either Router Solicitation or DHCPv6 request message, it just looks up the remote next-hop field of its routing information base (RIB) with the source IP address and the TEID of the received packet. A route entry matched for this search is the prefix delegated to the requesting UE. Therefore, EPC-E simply uses the prefix of the route entry as an assigned UE's prefix.

In (17), EPC-E returns the found prefix to UE by either Router Advertisement or DHCPv6 reply message. UE now creates an address(es) from the received prefix. It is important to highlight that UE can obtain the same prefix information from any EPC-E all the time because the same UE's route information is available on all the EPC-E.

It would be convenient to use automatic UE's prefix creation rule or algorithm for vEPC. There are various mechanisms to create UE's prefix. As an example, Stateless IPv6 Prefix Delegation [I-D.savolainen-stateless-pd] is introduced as an algorithm to create UE's prefix in vEPC below. It is important to mention that our architecture of the stateless user plane does not rely on any particular prefix creation mechanisms like [I-D.savolainen-stateless-pd] and can be run with any of them.

In the case of an UE's prefix length is equal, or shorter than /64, the generated prefix is consisted as shown in Figure 5. Each PDN is assumed to have single or several prefixes (named PDN prefix) used to generate UE's address. Followed by the PDN prefix, there is TEID field assigned for a UE's session on S1-U interface of vEPC. TEID is 32 bits identifier in GTP header to distinguish each bearer. The remaining bits are filled by subnet ID.



Stateless-pd Prefix

Figure 5

3.3. Control-plane awareness of stateless user-plane

Nodes in the control-plane in vEPC must be aware that the anycast address assigned to EPC-E is a S1-U address of vEPC. The vEPC must use the anycast address in signaling between vEPC and RAN. By doing this, packets from RAN are correctly forwarded to an appropriate EPC-E. Due to anycast nature, it means there is no hand-off procedure between SGWs because all eNodeB in the RAN send packets to the same anycast address.

When an operator needs to increase virtualized instances to cope with just signaling overload, the operator should use the existing S1-U address (i.e. EPC-E anycast address) for the new instances. If the operator would increase the capacity of the user plane, it can add additional EPC-Es in the core network. The operator can group the new EPC-Es as a set and increase scalability and performance of the user plane. In this case, the operator uses a new anycast address to the new set of EPC-E. We will discuss operational consideration in detail in Section 4.

3.4. Routing mechanism

Figure 6 shows a packet forwarding mechanism of our stateless user plane. As an example, there are four eNodeB (illustrated as eNB-x) , three EPC-Edge routers (shown as EPC-Ex) and two routers (RTRx) in Figure 6. UE is first connected to eNB-C and then moves to eNB-D. The UE at the new location is illustrated as UE'. Routing entry for UE is also illustrated at the right side in Figure 6.

EPC-E has two interfaces facing either RAN or CORE networks. An anycast address (shown as X) is configured to the interface facing RAN of all EPC-E. EPC-E assigns an individual IPv6 address to another interface (illustrated "a" to "d" in the figure). It is important to mention that the anycast address X can be treated as the SGW's S1-U address.

Since RTRs are a gateway to the Internet, they advertise routes of an operator's prefix to the Internet. After one of RTR receives a packet of UE from the Internet, it needs to routing it to UE in the user plane. RTR has a simple routing entry for PDN prefix whose next hop points to the EPC-E. One of RTR (let's say RTR2 in this case) looks up a routing table with UE's address and matched it with a routing entry of PDN prefix. Since multiple EPC-Es advertise a route for the same PDN prefix, RTR2 should forward the packet to one of

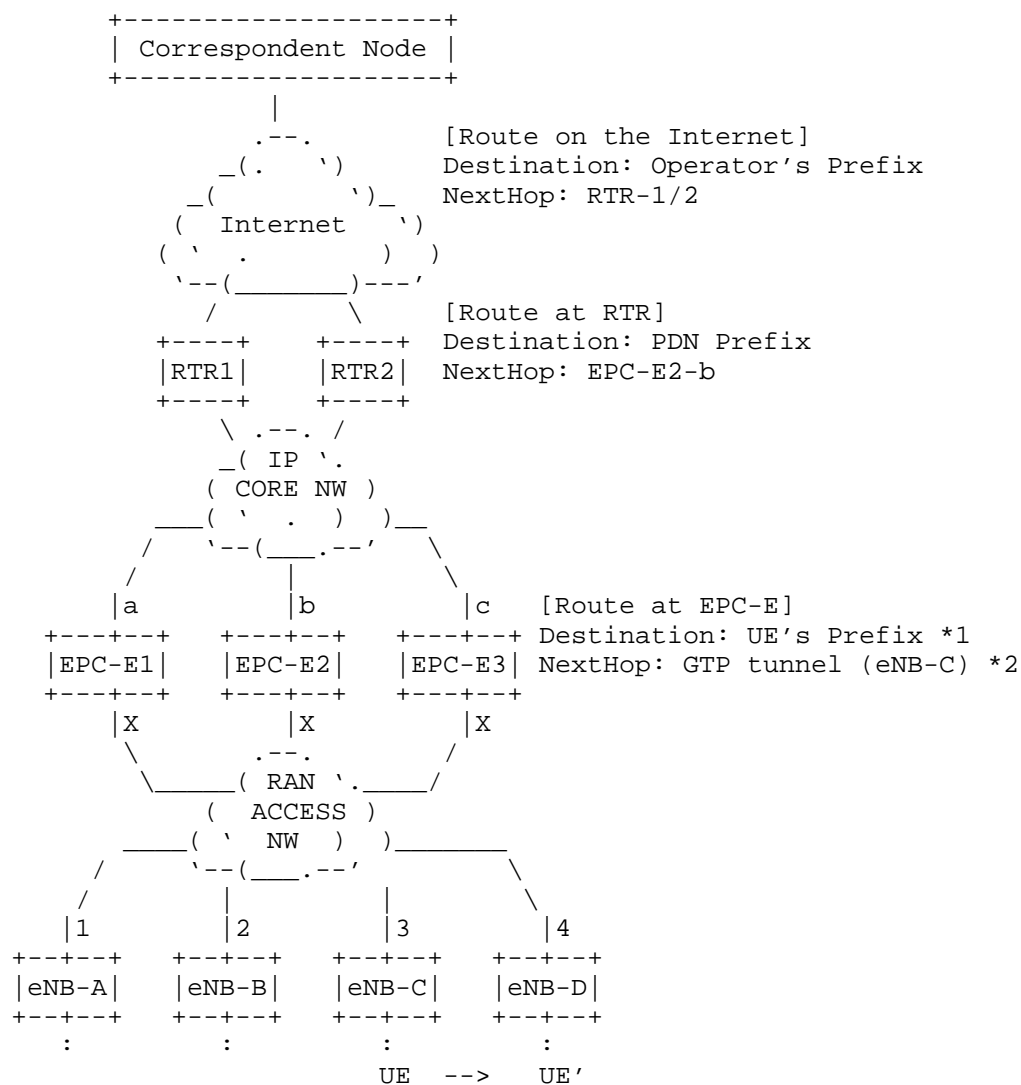
EPC-E according to the routing entry. This routing is known as hot-potato routing. In this example, the RTR2 uses EPC-E2-b as a nexthop of PDN prefix.

When the UE's packet is arrived at EPC-E2, EPC-E2 needs to forwards them to UE via eNodeB to which UE is connecting by using GTP tunnel. For this operation, EPC-E2 has a routing entry that destination is UE's prefix and that next hop points to GTP tunnel between eNB-C and the EPC-Es. In order to identify the GTP tunnel for UE, EPC-E needs S1-U address and Tunnel Endpoint ID (TEID) of eNB-C that is eNB-C-3 in Figure 6. The eNB-C TEID for UE is illustrated as TEID[eNB-C]. The SGW assigned TEID is utilized to generate the UE's prefix as we explained in Section 3.2. These TEID are assigned per UE. The TEID and S1-U address of eNodeB are retrieved from the next hop field of the routing entry of the mobile node. By using the GTP information, every EPC-E can now forward the UE's packets to right eNodeB.

Routing outgoing packets from UE is much simpler. The packets from UE are always routed to the closest EPC-E to UE because of anycast routing. In Figure 6, when UE sends a packet to a destination, the packet is reached to eNB-C and tunneled to EPC-E's anycast address. The GTP-tunneled packet is routed to the closest EPC-E that is EPC-E2 in this case. The packet is decapsulated by EPC-E2 and then forwarded to one of RTR according to the routing table. Since the decapsulated packet is regular IPv6 packet, no extra control other than routing is necessary.

When UE moves to a new location (UE'), it updates its location on the control plane. After signaling completion for location update, vEPC needs to update the UE's routing entry of all EPC-E so that vEPC advertises updated route with new location to all EPC-Es by a routing protocol. The routing entry should be updated with the new eNodeB's address that is eNB-D-4. During handover, there might be some traffic arriving to the older eNodeB (eNB-C). These packets can be re-routed to the new eNodeB (eNB-D) via X2-U interface in RAN.

The UE's address isn't changed when UE changes its attachment. In our scenario, SGW run on hypervisor and is independent from network topology. Therefore, logically we don't have handover across different SGWs. UE can stay connected with the same SGW all the time and can keep using the same TEID after handover. Thus, UE's address is unchanged even after handover.



*1 TEID used at EPC-E for the UE is included in this UE's prefix. see Figure 4.

*2 GTP tunnel state is stored in the next hop field. The state information is the combination of eNB-C S1 address that is eNB-C-3 and TEID(eNB-C) assigned for the UE.

Routing Mechanism Overview

Figure 6

3.5. IPv4 Support

Recent IPv6 transition mechanisms enable IPv6-only network to forward IPv4 packet with encapsulation or translation techniques. By using one of mechanisms, we can use IPv6 for our stateless user-plane network for transporting both IPv4 and IPv6 packets. Figure 7 shows available solutions of IPv4 support for each bearer type to deal with that requirement.

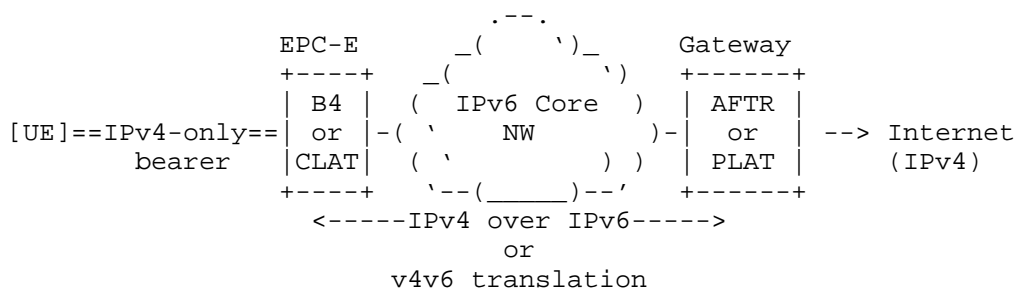
Bearer type	UE function	EPC-E function	Gateway function
IPv4	-	B4	AFTR
IPv4	-	CLAT	PLAT
IPv6	MAP-CE	-	MAP-BR
IPv6	B4	-	AFTR
IPv6	CLAT	-	PLAT

Solutions and functions for IPv4 support

Figure 7

In the case of a UE only support IPv4 bearer, B4 function of DS-Lite [RFC6333] or CLAT function of 464XLAT [RFC6877] may be implemented in a EPC-E. Both functions are stateless therefore EPC-E isn't required to maintain any tunneling or translation state.

Figure 8 shows how to support IPv4 on IPv6 core network in our vEPC. Instead of using RqTR as a gateway to the Internet, DS-LITE AFTR or 464XLAT PLAT is installed as a gateway to the IPv4 Internet.



IPv4 User plane of vEPC

Figure 8

If UE supports IPv6 capable bearer, IPv6 transition function may be implemented in the UE such as MAP-CE [I-D.ietf-softwire-map], B4 or CLAT. That means an EPC-E receives IPv6 packets from UE in this case so that the EPC-E does not need to be involved in the part of IPv4 support functions.

4. Operational Considerations

4.1. Scalability

Virtualization allows vEPC to be elastic for steep demand of requests to create and update for sessions. In our architecture, that makes routing update fluctuation from vEPC to EPC-E. This is the reason why we select BGP as a protocol between vEPC and EPC-E. BGP is scalable and stable routing protocol today.

BGP is an incremental update protocol so that once BGP peer established, millions of routes can be easily updated in stable manners. Operators can appropriately design BGP peering between vEPC and EPC-E to secure convergence time within appropriate period.

Granularity of the peering should be aware EPC-E capacity because it is assumed that EPC-E has upper limit of routing entries. BGP peering design should make sure that total number of routes does not exceed EPC-E capacity.

During the network planning, operators must understand EPC-E's capacity such as # of routes, bandwidth, etc. An example of estimation, if a EPC-E has 1Gbps throughput and each UE's bandwidth consumption is 10Kbps in average, the EPC-E should have 100K routes capacity.

This is an operational approach to minimize the risk of routing update fluctuation. If it is hard to support all the UEs by a single set of EPC-E in an operators network, different set of EPC-E can be introduced and configured in a network. The UEs are distributed to one of the set and handled by the EPC-Es in the set. We don't need to support millions of UEs by a single set of EPC-E. This is another advantage of using routing mechanism in the user plane. We already explain how to handle different set of EPC-E in our scheme in Section 3.3.

The notion of multiple EPC-E sets is easily fitted into our today's network. The operator's network is often separated into several

regional network for geographical scalability. Therefore, the operator can assign different EPC-E set to different region for better scalability.

In addition, routers and EPC-E in the IPv6 core network are required to process just "route", they naturally aggregate those routing entries. It helps limiting the total number of routing entries in our core network.

4.2. Backward Compatibility

vEPC should be able to fall back to the legacy EPC based packet forwarding to secure backward compatibility which is required to connect existing system, or to connect roaming partners through legacy S5/S8 interfaces. When fallback happened, all the packets are not routed on our stateless user plane, but forwarded to vEPC (i.e. SGW and PGW instances on hypervisor). vEPC must use a S1-U address that is different from anycast address assigned to EPC-Es. This address is assigned to SGW instances in vEPC and used to terminate tunnels in vEPC servers (i.e. hypervisor).

5. IANA Considerations

This memo includes no request to IANA.

6. Security Considerations

There are no security considerations specific to this document at this moment.

7. References

7.1. Normative References

- [I-D.vandeveldede-idr-remote-next-hop]
Veldede, G., Patel, K., Rao, D., Raszkuk, R., and R. Bush,
"BGP Remote-Next-Hop", draft-vandeveldede-idr-remote-next-hop-03 (work in progress), October 2012.
- [RFC5512] Mohapatra, P. and E. Rosen, "The BGP Encapsulation Subsequent Address Family Identifier (SAFI) and the BGP Tunnel Encapsulation Attribute", RFC 5512, April 2009.

7.2. Informative References

[Boeing-BGP]

Andrew, ., "Global IP Network Mobility using Border Gateway Protocol (BGP)", IAB Plenary IAB Plenary of IETF 62nd, March 2005.

[I-D.ietf-softwire-map]

Troan, O., Dec, W., Li, X., Bao, C., Matsushima, S., Murakami, T., and T. Taylor, "Mapping of Address and Port with Encapsulation (MAP)", draft-ietf-softwire-map-07 (work in progress), May 2013.

[I-D.savolainen-stateless-pd]

Savolainen, T. and J. Korhonen, "Stateless IPv6 Prefix Delegation for IPv6 enabled networks", draft-savolainen-stateless-pd-01 (work in progress), February 2010.

[I-D.wakikawa-mext-haha-interop2008]

Wakikawa, R., Shima, K., and N. Shigechika, "The Global HAHa Operation at the Interop Tokyo 2008", draft-wakikawa-mext-haha-interop2008-00 (work in progress), July 2008.

[I-D.yokota-dmm-scenario]

Yokota, H., Seite, P., Demaria, E., and Z. Cao, "Use case scenarios for Distributed Mobility Management", draft-yokota-dmm-scenario-00 (work in progress), October 2010.

[NFV-WHITEPAPER]

Network Operators, ., "Network Functions Virtualization, An Introduction, Benefits, Enablers, Challenges and Call for Action", SDN and OpenFlow SDN and OpenFlow World Congress, October 2012.

[RFC3963] Devarapalli, V., Wakikawa, R., Petrescu, A., and P. Thubert, "Network Mobility (NEMO) Basic Support Protocol", RFC 3963, January 2005.

[RFC5213] Gundavelli, S., Leung, K., Devarapalli, V., Chowdhury, K., and B. Patil, "Proxy Mobile IPv6", RFC 5213, August 2008.

[RFC5555] Soliman, H., "Mobile IPv6 Support for Dual Stack Hosts and Routers", RFC 5555, June 2009.

[RFC5844] Wakikawa, R. and S. Gundavelli, "IPv4 Support for Proxy Mobile IPv6", RFC 5844, May 2010.

- [RFC6275] Perkins, C., Johnson, D., and J. Arkko, "Mobility Support in IPv6", RFC 6275, July 2011.
- [RFC6301] Zhu, Z., Wakikawa, R., and L. Zhang, "A Survey of Mobility Support in the Internet", RFC 6301, July 2011.
- [RFC6333] Durand, A., Droms, R., Woodyatt, J., and Y. Lee, "Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion", RFC 6333, August 2011.
- [RFC6459] Korhonen, J., Soininen, J., Patil, B., Savolainen, T., Bajko, G., and K. Iisakkila, "IPv6 in 3rd Generation Partnership Project (3GPP) Evolved Packet System (EPS)", RFC 6459, January 2012.
- [RFC6877] Mawatari, M., Kawashima, M., and C. Byrne, "464XLAT: Combination of Stateful and Stateless Translation", RFC 6877, April 2013.

Authors' Addresses

Satoru Matsushima
Softbank Telecom
1-9-1,Higashi-Shimbashi,Minato-Ku
Tokyo 105-7322
Japan

Email: satoru.matsushima@g.softbank.co.jp

Ryuji Wakikawa
Softbank Mobile
1-9-1,Higashi-Shimbashi,Minato-Ku
Tokyo 105-7322
Japan

Email: ryuji.wakikawa@gmail.com

INTAREA Working Group
Internet Draft
Intended status: Proposed Standard
Expires: January 2014

Youval Nachum
Marvell
Linda Dunbar
Huawei
Ilan Yerushalmi
Tal Mizrahi
Marvell
July 15, 2013

Scaling the Address Resolution Protocol for Large Data Centers
(SARP)
draft-nachum-sarp-06.txt

Abstract

This document introduces SARP, an architecture that uses proxy gateways to scale large data center networks. SARP is based on fast proxies that significantly reduce switches' FDB (MAC table) sizes and ARP/ND impact on network elements in an environment where hosts within one subnet (or VLAN) can spread over various locations. SARP is targeted for massive data centers with a significant number of VMs that can move across various physical locations.

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/lid-abstracts.txt>.

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

This Internet-Draft will expire on January 15, 2014.

Copyright Notice

Copyright (c) 2013 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 Simplified BSD License.

Table of Contents

1. Introduction	3
1.1. SARP Motivation.....	3
1.2. SARP Overview	6
1.3. SARP Deployment Options.....	8
2. Terms and Abbreviations Used in this Document	9
3. SARP Description	10
3.1. Control Plane: ARP/ND	10
3.1.1. ARP/NS Request for a Local VM	10
3.1.2. ARP/NS Request for a Remote VM	10
3.1.3. Gratuitous ARP and Unsolicited Neighbor Advertisement (UNA)	11
3.2. Data Plane: Packet Transmission	12
3.2.1. Local Packet Transmission	12
3.2.2. Packet Transmission Between Sites	12
3.3. VM Migration	13
3.3.1. VM Local Migration.....	13
3.3.2. VM Migration from One Site to Another	13
3.3.2.1. Impact to IP<->MAC Mapping Cache Table of VMs being moved	15
3.4. Multicast and Broadcast.....	15
3.5. Non IP packet	15
3.6. IP<->MAC caching on SARP Proxy	16
3.7. High availability and load balancing	17
3.8. SARP Interaction with Overlay networks	18
4. Conclusions	18
5. Security Considerations.....	18
6. IANA Considerations	19
7. References	19
7.1. Normative References.....	19

7.2. Informative References.....	20
8. Acknowledgments	21

1. Introduction

This document describes a proxy gateway technique, called Scalable Address Resolution Protocol (SARP), which reduces switches' Filtering Data Base (FDB) size and ARP/Neighbor Discovery impact on network elements in an environment where hosts within one subnet (or VLAN) can spread over various access domains in data centers.

The main idea of SARP is to represent all VMs (or hosts) under each access domain by their corresponding access (or aggregation) node's MAC address regardless whether the access (or aggregation) node is the VMs (hosts)' gateway or not. For example, when a host "a" under access domain "S" needs to communicate with peers on the same VLAN but connected to different access domains, SARP requires "a" to use remote access node's MAC address rather than peers' MAC addresses. By doing so, switches in each domain do not need to maintain a list of MAC addresses for all the VMs (hosts) in different access domains in their FDBs. Therefore, the switches' FDB size is limited regardless how VLAN is spread.

1.1. SARP Motivation

[RFC6820] has documented various impacts and scaling issues to data center networks when subnets span across multiple L2/L3 boundary routers.

Note: The L2/L3 boundary routers in this draft are capable of forwarding IEEE802.1 Ethernet frames (layer 2) without MAC header change. When subnets span across multiple ports of those routers, they are still under the category of a single link, or a multi-access link model recommended by [RFC4903]. They are different from the "multi-link" subnets described in [Multi-Link] and [RFC4903] which refer to a different physical media with the same prefix connected to a router and the layer 2 frames cannot be natively forwarded without header change.

Unfortunately, when the combined number of VMs (or hosts) in all those subnets is large, this can lead to switches' MAC table size

explosion and heavy impact on network elements. There are four major issues associated with subnets spanning across multiple L2/L3 boundary router ports:

1) Intermediate switches' MAC address table (FDB) explosion:

When hosts in a VLAN (or subnet) span across multiple access domains and each access domain has hosts belonging to different VLANs, each access switch has to enable multiple VLANs. Then, those access switches will be exposed to all MAC addresses among all the VLANs enabled.

For example, for an access switch with 40 physical servers attached, where each server has 100 VMs, there are 4000 hosts under the access switch. If indeed hosts/VMs can be moved anywhere, the worst case for the Access Switch is when all those 4000 VMs belong to different VLANs, i.e. the access switch has 4000 VLANs enabled. If each VLAN has 200 hosts, this access switch's MAC table potentially has $200 \times 4000 = 800,000$ entries.

It is important to note that the example above is relevant regardless of whether IPv4 or IPv6 are used.

The example illustrates a scenario that is worse than what today's L2/3 Gateway has to face. In today's environment where each subnet is limited to a few access switches, the number of MAC addresses the gateway has to learn is of a significantly smaller scale.

2) the ARP/ND processing load impact to the L2/L3 boundary routers;

All VMs periodically send NDs to their corresponding Gateway nodes to get gateway nodes' MAC addresses. When the combined number of VMs across all the VLANs is large, processing the responses to the ND requests from those VMs can easily exhaust the gateway's CPU utilization.

A L2/L3 boundary router could be hit with ARP/ND twice when the originating and destination stations are in different subnets attached to the same router and when those hosts do not communicate with external peers very frequently. The first hit is when the originating station in subnet-A initiates an ARP/ND request to the L2/L3 boundary router if the router's MAC is not in the host's cache; and the second hit is when the L2/L3

boundary router initiates an ARP/ND request to the target in subnet-B if the target is not in router's ARP/ND cache.

- 3) In IPv4, every end station in a subnet receives ARP broadcast messages from all other end stations in the subnet. IPv6 ND has eliminated this issue by using multicast. However, most devices support a limited number of multicast addresses, due to multicast filtering scaling. Once the number of multicast addresses exceeds the multicast filter limit, the multicast addresses have to be processed by devices' CPU (i.e. the slow path). It is less of an issue in DC without VM mobility because each port is only dedicated to one (or a few number of) VLANs. Thus, the number of multicast addresses hitting each port is significantly lower.
- 4) The ARP/ND messages are flooded to many physical link segments which can reduce the bandwidth utilization for user traffic; ARP/ND flooding is probably an insignificant issue in today's data center because the majority of data center servers are moving towards 1G or 10G ports. The bandwidth taken by ARP/ND, even when flooded to all physical links, becomes negligible compared to the link bandwidth. In addition, the IGMP/MLD snooping [RFC4541] can further reduce the ND multicast traffic to some physical link segments.

Statistics done by Merit Network [ARMD-Statistics] has shown that the major impact of a large number of mobile VMs in Data Centers is to the L2/L3 boundary routers, i.e., issue 2 above. A L2/L3 boundary router could be hit with ARP/ND twice when the originating and destination stations are in different subnets attached to the same router and those hosts do not communicate with external peers often enough. The first hit is when the originating station in subnet-A initiates an ARP/ND request to the L2/L3 boundary router if the router's MAC is not in the host's cache; and the second hit is when the L2/L3 boundary router initiates ARP/ND requests to the target in subnet-B if the target is not in router's ARP/ND cache.

Overlay approaches, e.g. [NVo3-PROBLEM], can hide hosts (VMs) addresses in the core but does not prevent the MAC table explosion problem (Issue 1) unless the NVE is on a server.

The scaling practices documented in [ARP-ND-PRACTICE] can only reduce some ARP impact to L2/L3 boundary routers in some scenarios, but not all.

In order to protect router CPUs from being overburdened by target resolution requests, some routers rate limit the target MAC resolution requests to CPU. When the rate limit is exceeded, the incoming data frames are dropped.

In traditional Data Centers, it is less of an issue because the number of hosts attached to one L2/L3 boundary router is limited by the number of physical ports of the switches/routers. When Servers are virtualized to support 30 plus VMs, the number of hosts under one router can grow 30 plus times. In addition, the traditional data center has each subnet nicely placed in a limited number of server racks, i.e., switches under router only need to deal with MAC addresses of those limited subnets. With subnets being spread across many server racks, the switches are exposed to VLAN/MAC of many subnets, greatly increasing the size of the FDB.

The solution proposed in this draft can eliminate or reduce the likelihood of inter-subnet data frames being dropped and reduce the host MAC addresses exposed to FDB on intermediate switches.

1.2. SARP Overview

SARP is a proxy gateway technique to reduce switches' FDB (MAC table) sizes and ARP/ND impact on network elements in an environment where hosts within one subnet (or VLAN) can spread over various access domains in data centers.

Note: The Guidelines to proxy developers [RFC4389] have been carefully considered for the SARP protocols. Section 3.3 has demonstrated how SARP works when VMs are moved from one segment to another.

In order to enable VMs to be moved across greater number of servers while maintaining their MAC/IP addresses unchanged, the layer-2 network (e.g. VLAN) which interconnect those VMs may

spread across different server racks, different rows of server racks, or even different data centers.

For ease of description, let's break the entire network which interconnects all those VMs into two segments: interconnecting segment and "access" segments. While the "Access" network is mostly likely Layer 2, the "interconnecting" segment might be not.

The SARP proxies are located at the boundaries where the "Access" segment connects to its "Interconnecting" segment. The boundary node could be a Hypervisor virtual switch, a Top of Rack switch, an Aggregation switch (or end of row switch), or a data center core switch. Figure 1 depicts an example of two remote data centers that are managed as a single flat Layer 2 domain. SARP proxies are implemented at the edge devices connecting the data center to the transport network. SARP significantly reduces the ARP/ND transmissions over the "interconnection" network. The ARP/ND broadcast/multicast messages are bounded by the SARP proxies.

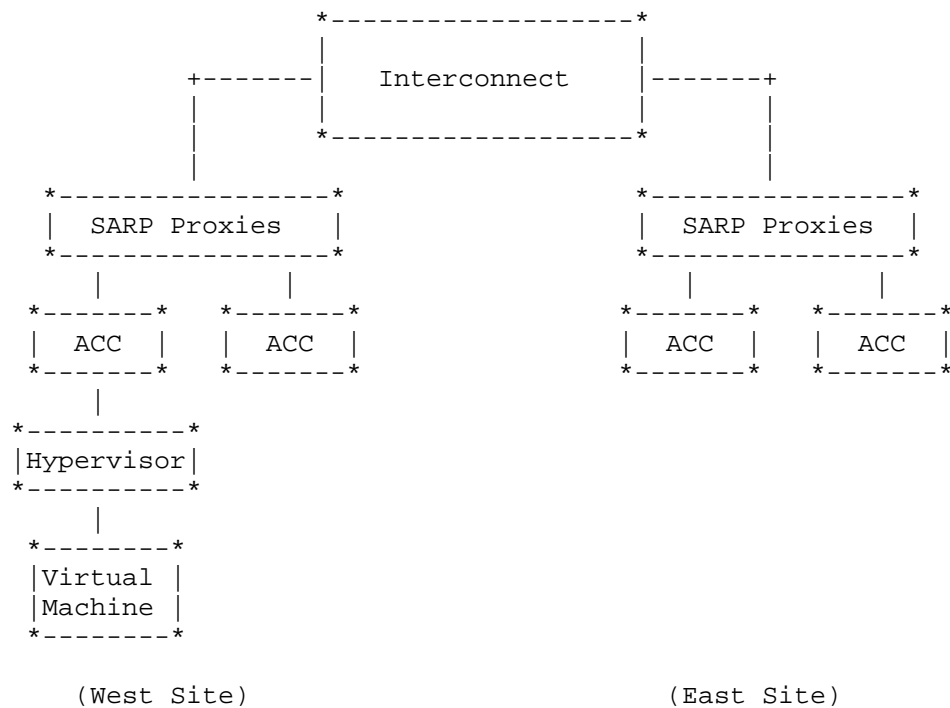


Figure 1 SARP Networking Architecture Example.

1.3. SARP Deployment Options

SARP deployment is tightly coupled with the data center architecture. SARP proxies are located at the point where the Layer 2 infrastructure connects to its Layer 2 cloud using overlay networks. SARP proxies can be located at the data center edge (as Figure 1 depicts), data center core, or data center aggregation. SARP can also be implemented by the hypervisor (as Figure 2 depicts).

To simplify the description, we will focus on data centers that are managed as a single flat Layer 2 network, where SARP proxies are located at the boundary where the data center connects to the transport network (as Figure 1 depicts).

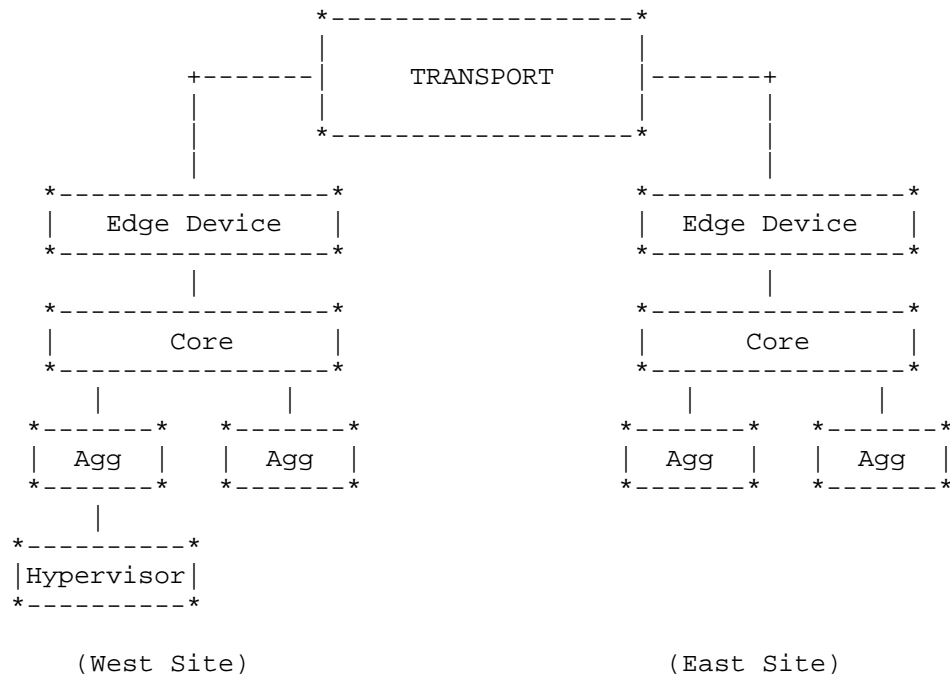


Figure 2 SARP deployment options.

2. Terms and Abbreviations Used in this Document

ARP: Address Resolution Protocol

FDB: Filtering Data Base, which is used for Layer-2 switches (IEEE802.1Q). Layer 2 switches flood data frames when DA is not in FDB, whereas routers drop data frames when the DA is not in the Forwarding Information Base (FIB). That is why Filtering Data Base (FDB) is used for Layer 2 switches.

FIB: Forwarding Information Base

IP-D: IP address of the destination virtual machine

IP-S: IP address of the source virtual machine

MAC-D: MAC address of the destination virtual machine

MAC-E: MAC address of the East Proxy SARP Device

MAC-S: MAC address of the source virtual machine

NA: IPv6 ND's Neighbor Advertisement

ND: IPv6 Neighbor Discovery Protocol. In this document, ND also refers to Neighbor Solicitation, Neighbor Advertisement, Unsolicited Neighbor Advertisement messages defined by RFC4861

NS: IPv6 ND's Neighbor Solicitation

SARP Proxy: The components that participates in the SARP protocol.

UNA: IPv6 ND's Unsolicited Neighbor Advertisement

VM: Virtual Machine

3. SARP Description

3.1. Control Plane: ARP/ND

This section describes the ARP/ND procedure scenarios. In the first scenario, VMs share the same Access Segment. In the second scenario, the source VM is local Access Segment and the destination VM is located at the remote Access Segment.

In all scenarios, the VMs (source and destination) share the same L2 broadcast domain.

3.1.1. ARP/NS Request for a Local VM

When source and destination VMs are located at the same Access Segment, the Address Resolution process is as described in [ARP] and [ND]. When the VM sends an ARP request or IPv6's Neighbor Solicitation (NS) to learn the IP to MAC mapping of another local VM, it receives a reply from the other local VM with the IP-D to MAC-D mapping.

3.1.2. ARP/NS Request for a Remote VM

When the source and destination VMs are located at different Access Segments, the Address Resolution process is as follows.

In our example, the source VM is located at the west Access Segment and the destination VM is located at the east Access Segment.

When the source VM sends an ARP/NS request to find out the IP to MAC mapping of a remote VM, if the local SARP proxy doesn't have the ARP cache for the target IP address or the cache entry has expired, the ARP/NS request is propagated to all Access Segments which might have VMs in the same virtual network as the originating VM, including the east Access Segment.

The destination VM responds to the ARP/NS request and transmits an ARP reply (IPv4) or Neighbor Advertisement (IPv6) having the IP-D to MAC-D mapping.

The east SARP proxy functions as the proxy ARP of its Local VMs. The east SARP proxy modifies the ARP reply or NA message's source MAC-D to MAC-E and forwards the modified ARP reply or NA message to all the SARP proxies.

The West SARP Proxy forwards the modified ARP reply message to the source VM.

The west SARP proxy can also function as an IP<->MAC cache of the Remote VMs. By doing so, it significantly reduces the volume of the ARP/ND transmission over the network.

When the west SARP proxy caches the IP<-> MAC mapping entries for remote VMs, the timers for the entries to expire should be set relatively small to prevent stale entries due to remote VMs being moved or deleted. For environment where VMs move more frequently, it is not recommended for SARP Proxy to cache the IP<-> MAC mapping entries of remote VMs.

3.1.3. Gratuitous ARP and Unsolicited Neighbor Advertisement (UNA)

Hosts (or VMs) send out Gratuitous ARP (IPv4) and Unsolicited Neighbor Advertisement - UNA (IPv6) for other nodes to refresh IP<->MAC entries in their cache.

The local SARP processes the Gratuitous ARP or UNA in the same way as the ARP reply or IPv6 NA, i.e. replace the source MAC with its own MAC.

3.2. Data Plane: Packet Transmission

3.2.1. Local Packet Transmission

When a VM transmits packets to a destination VM that is located at the same site, there is no change in the data plane. The packets are sent from (IP-S, MAC-S) to (IP-D, MAC-D).

3.2.2. Packet Transmission Between Sites

Packets that are sent between sites traverse the SARP proxy of both sites. In our example, all packets sent from the VM located at the west site to the destination VM located at the east site traverse the west SARP proxy and the east SARP proxy.

The source VM follows its ARP table and sends packets to (IP-D, MAC-E) destination addresses and with (IP-s, MAC-S) as the source addresses.

The west SARP proxy can either 1) simply forward the data frame to MAC-E, or 2) replace the packet source address to its own source address (MAC-W), keeps the destination address to be (MAC-E), and forwards the packet to the east proxy SARP.

It is recommended for west SARP proxy to replace Source Address with its own if the "interconnecting segment" has address learning enabled. Otherwise nodes in the "interconnecting segment" can't learn the address of the switch on which west SARP proxy is running unless the switch sends out frames periodically.

When the east proxy SARP receives the packet, it replaces the destination MAC address to be (MAC-D) based on the packet destination IP (i.e., IP-D), but it does not change the source MAC addresses. When the destination VM receives the packet, the Source Address field would be the MAC address of the VM on the west side or the MAC address of the west side SARP proxy,

Noted: it is common for data center network to have security policies to enforce some VMs can communicate with each other, and some VMs can't. Most likely, those policies are configured by VM's IP addresses. Even though the originating VM's MAC address might be lost when the packet arrives at the destination VM, the originating VM's IP address is still present in the data packets for security policy to be enforced.

Noted: for the option which doesn't need west SARP to change source MAC of the data frames, the originating VM's MAC will be

present when the data frames arrive at the destination VMs. Therefore, this option is valuable when hosts/VMs need to validate source VMs MAC addresses to comply any policies imposed.

Noted: Most hosts/VMs refresh its IP<->MAC mapping cache, with the Source MAC and Source IP of a received data frame. For the option which west SARP changes data frame's source MAC with its own MAC address, the destination VM's IP<->MAC cache can be refreshed with the valid mapping of the Source-VM-IP <->West-SARP-MAC. For the option of West SARP not changing source MAC, the destination VM has to turn off the learning of IP<->MAC mapping from the received data frames.

3.3. VM Migration

3.3.1. VM Local Migration

When a VM migrates locally within its Access segment, the SARP protocol is not required to perform any action. VM migration is resolved entirely by the Layer 2 mechanisms.

3.3.2. VM Migration from One Site to Another

In our example, the VM migrates from the west site to the east site while maintaining its MAC and IP addresses.

VM migration might affect networking elements based on their respective location:

- Origin site (west site)
- Destination site (east site)
- Other sites

Origin site:

The Origin site is the site where the VM is before migration. It is the west site in our example.

Before the VM (IP=IP-D, MAC=MAC-D) is moved, all VMs at the west site that have an ARP entry of IP-D in their ARP table have the (IP-D to MAC-D) mapping. VMs on any other "Access Segments" will have ARP entry of (IP-D to MAC-W) mapping where MAC-W is the MAC address of the SARP proxy on the West Access Segment.

After the VM (IP-D) in the West Site moves to East Site, if there is gratuitous ARP (IPv4) or Unsolicited Neighbor Advertisement (IPv6) sent out by the destination hypervisor for the VM (IP-D), then the IP<->MAC mapping cache of VMs on all Access Segments will be updated by (IP-D to MAC-E) where MAC-E is the MAC address of the SARP proxy on the East Site. If there isn't any gratuitous ARP or Unsolicited Neighbor Advertisement sent out by the destination hypervisor, the IP<->MAC cache on the VMs in west site (and other sites) will eventually aged out.

Until IP<->MAC mapping cache tables are updated, the source VMs from the west site continue sending packets to MAC-D. Switches at the west site are still configured with the old location of MAC-D. This can be resolved by VM manager sending out a fake gratuitous ARP or Unsolicited Neighbor Advertisement on behalf of destination Hypervisor, shorter aging timer configured for IP<->MAC cache table, or by redirecting the packets to the proxy SARP of the west site.

Destination Site:

The destination site is the site to which the VM migrated, the east site in our example.

Before any gratuitous ARP or Unsolicited Neighbor Advertisement messages are sent out by the destination hypervisor, all VMs at the east site (and all other sites) might have (IP-D to MAC-W) mapping in their IP<->MAC mapping cache. IP<->MAC mapping cache is updated by aging or by a gratuitous ARP or UNA message sent by the destination hypervisor. Until IP<->MAC mapping caches are updated, the source VMs from the east site continue to send packets to MAC-W. This can be resolved by VM manager sending out a fake gratuitous ARP/UNA immediately after the VM migration, or redirecting the packets from the SARP proxy of the east site to the migrated VM by updating the destination MAC of the packets to MAC-D.

Other Sites:

All VMs at the other sites that have an ARP entry of IP-D in their ARP table have the (IP-D to MAC-W) mapping. ARP mapping is updated by aging or by a gratuitous ARP message sent by the destination hypervisor of the migrated VM and modified by the SARP proxy of the east site (IP-D to MAC-E) mapping. Until ARP tables are updated, the source VMs from the west site continue sending packets to MAC-W. This can be resolved by redirecting the packets from the SARP proxy of the west site to the SARP proxy of

the east site by updating the destination MAC of the packets to MAC-E.

3.3.2.1. Impact to IP<->MAC Mapping Cache Table of VMs being moved

When a VM (IP-D) is moved from one site to another site, its IP<->MAC mapping entries for VMs located at the other sites (i.e. neither east site nor west site) are still valid, even though most Guest OSs (or VMs) will refresh their IP<->MAC cache after migration.

The VM (IP-D)'s IP<->MAC mapping entries for VMs located at east site, if not refreshed after migration, can be kept with no change until the ARP aging time since they are mapped to MAC-E. All traffic originated from the VM (IP-D) in its new location to VMs located at the east site traverses the SARP proxy of the east Site. The ARP/UNA sent by the SARP proxy of the east site or by the VMs on east side can always refresh the corresponding entries in the VM (IP-D)'s IP<->MAC cache .

The VM (IP-D)'s ARP entries (i.e. IP to MAC mapping) for VMs located at west sites will not be changed either until the ARP entries age out or new data frames are received from the remote sites. Since all MAC addresses of the VMs located at the west site are unknown at the east site. All unknown traffic from the VM is intercepted by the SARP proxy of the east site and forwarded to the SARP proxy of the west site (just for ARP aging time). This can be resolved by the east SARP proxy having mapping entries for VMs in the west side. Upon receiving unknown packets, it can update the migrating VM with the new IP to MAC mapping by sending a modified gratuitous ARP with (IP-D to MAC-W) mapping.

Note that overlay networks providing the Layer 2 network virtualization services configure their Edge Device MAC aging timers to be greater than the ARP request interval.

3.4. Multicast and Broadcast

To be added in a future version of this document

3.5. Non IP packet

To be added in a future version of this document

3.6. IP<->MAC caching on SARP Proxy

ARP/NS Requests for a VM located at a remote site require flooding messages over the interconnecting network to all sites which have enabled the virtual network on which the VM belongs to. This scenario is described in details at 3.1.2. In such cases, SARP caching can reduce the number of ARP/ND transmissions over interconnecting networks.

In the example presented at section 3.1.2. the source VM is located at the west site and the destination VM is located at the east site. When the source VM sends an ARP or Neighbor Solicitation request to discover the IP to MAC mapping of the remote VM, the request can be intercepted by the west SARP proxy.

The west SARP proxy learns or refreshes the source IP to source MAC mapping and looks up the IP to MAC translation of the destination IP. If the destination IP entry is found and is valid, the west SARP proxy replies with an ARP reply or Neighbor Advertisement without propagating the packet to other sites. Otherwise, the packet is propagated to all sites which have the virtual network enabled including the east site.

The propagated ARP/NS request is intercepted again by the east SARP proxy. It learns or refreshes the source IP to source MAC mapping and looks up the destination IP to MAC translation. If the destination IP entry is found and is valid the SARP proxy replies with an ARP reply or NA without propagating the ARP request to the east site. Otherwise, the ARP/NS request is broadcasted within the east site.

The destination VM responds to the ARP/NS request and transmits an ARP reply or NA having the IP-D to MAC-D mapping.

The east side SARP proxy intercepts the ARP reply or NA and learns or refreshes the Destination IP to Destination MAC mapping, replace the source MAC with its own MAC before sending the ARP reply or NA to the west SARP proxy (so that requesting VM can learn the IP-D to MAC-E mapping).

The West SARP Proxy intercepts the ARP reply or NA and learns or refreshes the Destination IP to Destination MAC mapping and propagates the ARP reply to the source VM.

The SARP proxies maintain an ARP caching table of IP to MAC mapping for all recent ARP/NS requests and replies. This table allows the SARP proxy to respond with low latency to the ARP/NS requests sent locally and avoid the broadcast transmissions of such requests over the transport network and all over the broadcast domains at the remote sites.

3.7. High availability and load balancing

The SARP proxy is located at the boundary where the local Layer 2 infrastructure connects to the interconnecting network. All traffic from the local site to the remote sites traverses the SARP proxy. The SARP proxy is subject to high availability and bandwidth requirements.

The SARP architecture supports multiple SARP proxies connecting a single site to the transport network. In SARP architecture all proxies can be active and can backup one another. The SARP architecture is robust and allows the network administrator to allocate proxies according to the bandwidth and high availability requirements.

Traffic is segregated between SARP proxies by using VLANs. An SARP proxy is the Master-SARP proxy of a set of VLANs and the Backup-SARP proxy of another set of VLANs.

For example the SARP proxies of the west site (as Figure 1 depicts) are SARP proxy-1 and SARP proxy-2. The west site supports VLAN-1 and VLAN-2 while SARP proxy-1 is the Master SARP proxy of VLAN-1 and the Backup proxy of VLAN-2 and SARP proxy-2 is the Master SARP proxy of VLAN-2 and the Backup SARP proxy of VLAN-1. Both proxies are members of VLAN-1 and VLAN-2.

The Master SARP proxy updates its Backup proxy with all the ARP reply messages. The Backup SARP proxy maintains a backup database to all the VLANs that it is the Backup SARP proxy.

The Master and the Backup SARP proxies maintain a keepalive mechanism. In case of a failure the Backup proxy becomes the Master SARP proxy. The failure decision is per VLAN. When the Master and the Backup proxies switchover, the backup SARP proxy can use the MAC address of the Master SARP proxy. The backup SARP proxy sends locally a gratuitous ARP message with the MAC address of the Master SARP proxy to update the forwarding tables on the

local switches. The backup SARP proxy also updates the remote SARP proxies on the change.

3.8. SARP Interaction with Overlay networks

SARP interaction with overlay networks providing L2 network virtualization (such as IP, VPLS, Trill, OTV, NVGRE and VxLAN) is efficient and scalable.

The mapping of SARP to overlay networks is straightforward. The VM does the destination IP to SARP proxy MAC mapping. The mapping of the proxy MAC to its correct tunnel is done by the overlay networks. SARP significantly scales down the complexity of the overlay networks and transport networks by reducing the mapping tables to the number of SARP proxies.

4. Conclusions

SARP distributes the Layer 2 Forwarding Information Base (FIB) from the edge devices (functioning as SARP proxies) to the VMs. By doing so, it significantly reduces table sizes on the edge devices. The source VM maintains the mapping of its destination VMs to the destination site/cloud in the ARP table. The destination VM IP is translated to the destination MAC address of the SARP proxy at the destination site. The SARP proxies only maintain Layer 2 FIB of local VMs and remote edge devices.

SARP proxies can support FAST VM migration and provide minimum transition phase. When SARP proxy indicates or is informed of VM migration, it can update all its peers and trigger a fast update.

SARP seamlessly supports Layer 2 network virtualization services over the overlay network and significantly reduces their complexity in terms of table size and performance. The overlay networks are only required to map MAC addresses of the SARP proxies to the correct tunnel.

5. Security Considerations

The SARP proxies are located at the boundaries where the local Layer 2 infrastructure connects to its Layer 2 cloud. The SARP proxies interoperate with overlay network protocols that extend the Layer-2 subnet across data centers or between different systems within a data center.

SARP control plane and data plane are traversed by the overlay network hence SARP does not expose the network to additional security threats.

SARP proxies may be exposed to Denial of Service (DoS) attacks by means of ARP/ND message flooding. Thus, the SARP proxies must have sufficient resources to support the SARP control plane without making the network more vulnerable to DoS than without SARP proxies.

SARP adds security to the data plane by hiding all the local layer 2 MAC addresses from potential attacker located at the remote clouds. The only MAC addresses that are exposed at remote sites are the MAC addresses of the SARP proxies.

6. IANA Considerations

There are no IANA actions required by this document.

RFC Editor: please delete this section before publication.

7. References

7.1. Normative References

- [ARP] Plummer, D., "An Ethernet Address Resolution Protocol", RFC 826, November 1982.
- [ND] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.
- [GratuitousARP] S. Cheshire, "IPv4 Address Conflict Detection", RFC 5227, July 2008.
- [IGMP-MLD-tracking] H. Aseda, and N. Leymann, "IGMP/MLD-Based Explicit Membership Tracking Function for Multicast Routers" (<http://tools.ietf.org/html/draft-ietf-pim-explicit-tracking-02>), Oct, 2012.
- [RFC826] D.C. Plummer, "An Ethernet address resolution protocol." RFC826, Nov 1982.
- [RFC1027] Mitchell, et al, "Using ARP to Implement Transparent Subnet Gateways" (<http://datatracker.ietf.org/doc/rfc1027/>)

- [RFC4389] Thaler, et al, "Neighbor Discovery Proxies (ND Proxy)", RFC4389, April 2006.
- [RFC4541] Christensen, et al, "Considerations for Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) Snooping Switches", RFC 4541, May 2006
- [RFC4861] Narten, et al, "Neighbor Discovery for IP version 6 (IPv6)", RFC4861, Sept 2007
- [RFC4903] Thaler, "Multilink Subnet Issues", RFC4903, July 2007.
- [RFC6820] Narten, et al, "Address Resolution Problems in Large Data Center Networks", RFC6820, Jan 2013.

7.2. Informative References

- [Impatient-NUD] E. Nordmark, I. Gashinsky, "draft-ietf-6man-impatient-nud"
- [ARMD-Statistics] M. Karir, J. Rees, "Address Resolution Statistics", draft-karir-armd-statistics-01.txt (expired), July 2011.
<https://datatracker.ietf.org/doc/draft-karir-armd-statistics/>
- [ARP_Reduction] Shah, et al, "ARP Broadcast Reduction for Large Data Centers", draft-shah-armd-arp-reduction-02.txt (expired), Oct 2011.
<https://datatracker.ietf.org/doc/draft-shah-armd-arp-reduction/>
- [ARP-ND-PRACTICE] Dunbar, Kumari, Gashinsky, "Practices for scaling ARP and ND for large data centers", draft-dunbar-armd-arp-nd-scaling-practices-06, Feb 2013
- [NVo3-PROBLEM] Narten, T., Gray, E., Black, D., Fang, L., Kreeger, L., Napierala, M., "Problem Statement: Overlays for Network Virtualization", draft-ietf-nvo3-overlay-problem-statement, work in progress, May 2013.

[Multi-Link] Thaler, et al, "Multi-link Subnet Support in IPv6",
draft-ietf-ipv6-multi-link-subnets-00.txt (expired),
Dec 2002. [https://datatracker.ietf.org/doc/draft-ietf-
ipv6-multilink-subnets/](https://datatracker.ietf.org/doc/draft-ietf-ipv6-multilink-subnets/)

8. Acknowledgments

We want to thank Ted Lemon in providing many rounds of valuable comments and suggestions to the draft.

This document was prepared using 2-Word-v2.0.template.dot.

Authors' Addresses

Youval Nachum
Email: youval.nachum@gmail.com

Linda Dunbar
Huawei Technologies
5430 Legacy Drive, Suite #175
Plano, TX 75024, USA
Phone: (469) 277 5840
Email: ldunbar@huawei.com

Ilan Yerushalmi
Marvell
6 Hamada St.
Yokneam, 20692 Israel
Email: yilan@marvell.com

Tal Mizrahi
Marvell
6 Hamada St.
Yokneam, 20692 Israel
Email: talmi@marvell.com

