

NV03 working group  
Internet Draft  
Category: Standards Track  
Expires: November 2017

L. Dunbar  
D. Eastlake  
Huawei  
Tom Herbert  
Google

July 8, 2016

## NVA Address Mapping Distribution (NAMD) Protocol

[draft-dunbar-nvo3-nva-mapping-distribution-03.txt](#)

### Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#). This document may not be modified, and derivative works of it may not be created, except to publish it as an RFC and to translate it into languages other than English.

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 April 8, 2015.

## Copyright Notice

Copyright (c) 2015 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.

## Abstract

This draft describes the mechanism for NVA to promptly and incrementally distribute the inner (TS) to outer (NVE) mapping and VN Context to relevant NVEs in a timely manner.

## Table of Contents

<a href="#">1.</a>	<a href="#">Introduction.....</a>	<a href="#">4</a>
<a href="#">2.</a>	<a href="#">Terminology.....</a>	<a href="#">4</a>
<a href="#">3.</a>	<a href="#">Overall Requirement for NVE&lt;-&gt;NVA Control Plane.....</a>	<a href="#">5</a>
<a href="#">4.</a>	<a href="#">Terminologies and Assumptions.....</a>	<a href="#">6</a>
<a href="#">5.</a>	<a href="#">Overview of NVA Address Mapping Distribution (NAMD) Protocol...</a>	<a href="#">7</a>
<a href="#">6.</a>	<a href="#">TLV for NVE reachable addresses.....</a>	<a href="#">7</a>
<a href="#">7.</a>	<a href="#">Push Mechanism.....</a>	<a href="#">8</a>
	<a href="#">7.1. Requesting Push Service.....</a>	<a href="#">9</a>
	<a href="#">7.2. Incremental Push Service.....</a>	<a href="#">12</a>
<a href="#">8.</a>	<a href="#">Pull Mechanism.....</a>	<a href="#">13</a>
	<a href="#">8.1. Pull Query Format.....</a>	<a href="#">14</a>
	<a href="#">8.2. Pull Response.....</a>	<a href="#">16</a>
	<a href="#">8.3. Cache Consistency.....</a>	<a href="#">19</a>
	<a href="#">8.4. Update Message Format.....</a>	<a href="#">20</a>
	<a href="#">8.5. Acknowledge Message Format.....</a>	<a href="#">21</a>
	<a href="#">8.6. Pull Request Errors.....</a>	<a href="#">21</a>
	<a href="#">8.7. Redundant Pull NVAs.....</a>	<a href="#">21</a>
<a href="#">9.</a>	<a href="#">Hybrid Mode.....</a>	<a href="#">21</a>
<a href="#">10.</a>	<a href="#">Redundancy.....</a>	<a href="#">22</a>
<a href="#">11.</a>	<a href="#">Inconsistency Processing.....</a>	<a href="#">22</a>
<a href="#">12.</a>	<a href="#">Protocols to consider to carry NAMD messages.....</a>	<a href="#">23</a>

Internet-Draft      NVA mapping distribution

[13.](#) Security Considerations.....[23](#)  
[14.](#) IANA Considerations.....[24](#)  
[15.](#) Acknowledgements.....[24](#)  
[16.](#) References.....[24](#)  
    [16.1.](#) Normative References.....[24](#)  
    [16.2.](#) Informative References.....[24](#)  
Authors' Addresses.....[25](#)

---

## Internet-Draft      NVA mapping distribution

### 1. Introduction

[Section 4.5](#) of [nvo3-problem-statement] describes the back-end Network Virtualization Authority (NVA) that is responsible for distributing the mapping information for entire overlay system. [\[nvo3-nve-nva-cp-req\]](#) defines the requirement for the control plane between NVA and NVE.

This draft describes a mechanism for NVA to promptly and incrementally distribute the inner (TS) to outer (NVE) mapping and VN Context to relevant NVEs in a timely manner.

For ease of description, the term "NAMD" is used to represent the NVA Address Mapping Distribution protocol.

### 2. Terminology

The following terms are used interchangeably in this document:

- The terms "Subnet" and "VLAN" because it is common to map one subnet to one VLAN.
- The term "Directory" and "Network Virtualization Authority (NVA)"
- The term "NVE" and "Edge"

Bridge: IEEE Std 802.1Q-2011 compliant device [[802.1Q](#)]. In this draft, Bridge is used interchangeably with Layer 2 switch.

NAMD Timeout: The time interval that an NVE can assume NVA is not reachable if the NVE hasn't received any updates from NVA during this time. NAMD Timeout is an unsigned

byte that gives the amount of time in seconds during which the NVA will send at least three update PDUs. An empty update is used as a keep alive. It defaults to 30 seconds.

DA: Destination Address

DC: Data Center

EoR: End of Row switches in data center. Also known as aggregation switches.

End Station: Guest OS running on a physical server or on a virtual machine. An end station in this document has at least one IP address and at least one MAC address, which could be in DA or SA field of a data frame.

LISP: Locator/ID Separation Protocol

NVA: Network Virtualization Authority

NVE: Network Virtualization Edge

SA: Source Address

Station: A node, or a virtual node, with IP and/or MAC addresses, which could be in the DA or SA of a data frame.

ToR: Top of Rack Switch in data center. It is also known as access switches in some data centers.

TS: Tenant System

VM: Virtual Machines

VN: Virtual Network

VNID: Virtual Network Instance Identifier

### [3.](#) Overall Requirement for NVE<->NVA Control Plane

[Section 3.1](#) of [nvo3-cp-req] describes the basic requirement of inner address to outer address mapping for NV03. A NVE needs to know the mapping of the Tenant System destination (inner) address to the (outer) address (IP) on the Underlying Network of the egress NVE.

[Section 3.1](#) of [nvo3-cp-req] states that a protocol is needed to provide this inner to outer mapping and VN Context to each NVE that requires it and keep the mapping updated in a timely manner. Timely updates are important for maintaining connectivity between Tenant Systems.

### [4.](#) Terminologies and Assumptions

NVAs can be centralized or distributed with each NVA holding the mapping information for a subset of VNs. By saying that an NVA holds mapping information for a VN, it means that the NVA has mapping information for all the TSs in the VN.

Centralized NVA means that the NVA holds mapping information for all the VNs in the administrative domain. There could be multiple instances of centralized NVA for redundancy purpose.

A NVA could be instantiated on a server/VM attached to a NVE, very much like a TS attached to a NVE, or could be integrated within an NVE. When a NVA is a standalone server/VM attached to a NVE, it has to be reachable via the attached NVE by other NVEs. A NVA can also be instantiated on a NVE that doesn't have any TSs attached. The NVE-NVA control plane for NVA being attached to NVE (like a VM) will require additional functions on NVEs than NVA being embedded in a NVE.

NVA should have at least the following information for each TS:

- . Inner Address: TS (host) Address family (IPv4/IPv6, MAC,

virtual network Identifier MPLS/VLAN, etc)

- . Outer Address: The list of locally attached edges (NVEs); normally one TS is attached to one edge, TS could also be attached to 2 edges for redundancy (dual homing). One TS is rarely attached to more than 2 edges, though it could be possible;
- . VN Context (VN ID and/or VN Name)
- . Timer for NVEs to keep the entry when pushed down to or pulled from NVEs.
- . Optionally the list of interested remote edges (NVEs). This information is for NVA to promptly update relevant edges (NVEs) when there is any change to this TS' attachment to edges (NVEs). However, this information doesn't have to be kept per TS. It can be kept per VN.

By saying that a NVE is participating in a VN or the VN is active on the NVE, it means that the VN is enabled on the NVE and there is at least one TS of the VN being attached to the NVE.

## [5.](#) Overview of NVA Address Mapping Distribution (NAMD) Protocol

The inner-outer address mapping could change as TSs move from NVE to another. At any given period, probably only a small set of TSs would move, resulting in a small portion of changes on the inner-outer address mapping. Therefore, it is important to have a mechanism for NVA to send incremental updates to NVEs for the changes instead of entire database of the mapping entries. This document specifies the incremental update messages (TLVs) from NVAs to NVEs, to maintain data consistency between NVAs and NVEs.

The NAMD mechanism requires messages to distribute NVA content to all the NVEs, inform the incremental changes to the relevant NVEs, and maintain the database consistency between NVA and NVEs. This document specifies the structures (a.k.a. TLVs) of those messages, which are referred to as NAMD messages throughout this

document. The NAMD TLVs can be included in BGP or IGP protocol messages. How they are integrated with the BGP or IGP will be further specified in the corresponding working groups.

A NVA can offer services in a Push, Pull model, or the combination of the two.

In Push model, the NVE, upon restart or initialization, sends requests for all the interested VNs as a multicast to all the NVAs. NVAs with the requested VNs use NAMD messages to distribute the mapping entries to the requested NVEs. Whenever, there are changes in the mapping entries, NVA uses NAMD messages to only send the changed portion of the entries.

In the Pull model, an NVA periodically sends VN scoped broadcast messages to all NVEs. An NVE, upon receiving a unknown unicast or ARP/ND with unknown target NVE, sends the pull request to the NVA that supports the VN that the targets belongs to.

## 6. TLV for NVE reachable addresses

The Reachable Interface Addresses (IA) TLV is used to advertise a set of addresses within a VN being attached to (or reachable by) a specific NVE, and optionally the NVE Virtual Access Point.

These addresses can be in different address families. For example, it can be used to declare that a particular interface with specified IPv4, IPv6, and 48-bit MAC addresses in some particular VN is reachable from a particular NVE.

This document suggests using the Interface Addresses APPsub-TLV defined by [\[IA\]](#) except using NVE address subTLV in the fourth field shown below:

```
+---+---+---+---+---+---+---+---+---+---+
| Type = TBD                               | (2 bytes)
+---+---+---+---+---+---+---+---+---+---+
| Length                                   | (2 bytes)
+---+---+---+---+---+---+---+---+---+---+
```



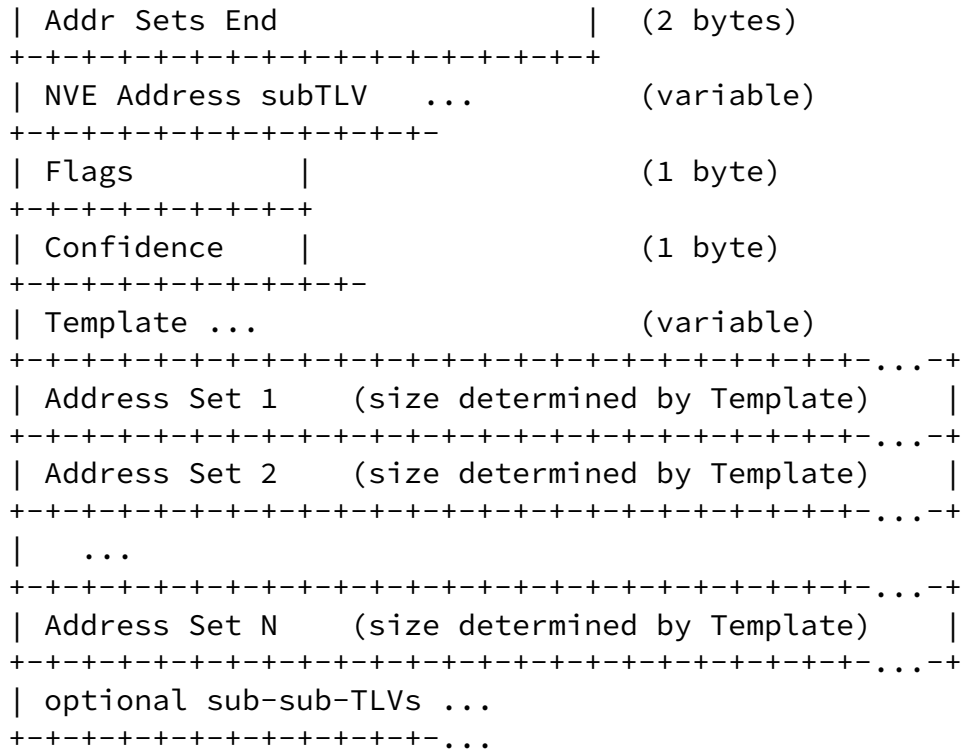


Figure 1. The Interface Addresses APPsub-TLV

Addr Sets End: The unsigned integer offset of the byte, within the IA APPsub-TLV [IA] value part, of the last byte of the last Address Set. This will be the byte just before the first sub-sub-TLV if any sub-sub-TLVs are present (see [Section 3](#)). If this is equal to Length, there are no sub-sub-TLVs. If this is greater than Length or points to before the end of the Template, the IA APPsub-TLV is corrupt and MUST be discarded. This field is always two bytes in size.

## 7. Push Mechanism

Under this mode, NVA pushes the inner-outer mapping for all the TSs of the VNs to relevant NVEs. This service is scoped by VN. A Push NVA also advertises whether or not it believes it has pushed complete mapping information for a VN. It might be pushing only a

subset of the mapping and/or reachability information for a VN.

The Push Model uses the NAMD messages as its distribution mechanism.

With the Push model, if the destination of a data frame arriving at the Ingress NVE can't be found in its inner-outer mapping database that are pushed down from the NVA, the Ingress edge could be configured with one or more of the following policies:

- simply drop the data frame,
- flood the data frames to other NVEs that have the VN enabled, or
- start the "pull" process to get information from Pull NVA.

When the NVE is waiting for reply from the Pull process, the NVE can either drop or queue the packet.

One drawback of the Push Mode is that it will push more mapping entries to an NVE than needed. Under the normal process of edge cache aging and unknown destination address flooding, rarely used entries would have been removed. It would be difficult for NVA to predict the communication patterns from/to TSs within one VN. Therefore, it is likely that the NVA will push down all the entries for all the VNs that are enabled on the NVE.

Another drawback with Push model: there really can't be any source-based policy. It's all or nothing.

#### [7.1](#). Requesting Push Service

When a NVE is initialized or re-started, it needs to send request to the relevant NVAs to push down the mapping information for the active VNs on the NVE. NVE could use Virtual Network scoped message to announce all the Virtual Networks in which it is participating to NVAs who have the mapping information for the VNs. A new subTLV (Enabled-VN TLV) is specified here for NVE to indicate all its interested VNs in the NAMD message. The new subTLV can be included in an IGP protocol message or BGP message.

For 24-bits VN ID, there could be 16 million VNs. Multiple ways can be used to express the interested VNs:

- Starting VN & End VNs & bit map for the VNs in between.
- Starting VN & End VN (for the VNs that are contiguous)

- Individual VN listing (for a small number of VNs that are not contiguous)

Therefore 3 different types of subTLV are specified:

```

+---+---+---+---+---+
| INT-VN-TYPE-1 |           (1 byte)
+---+---+---+---+---+
|   Length      |           (1 byte)
+---+---+---+---+---+---+---+---+---+---+---+---+
|           Start VN ID           | (4 bytes)
+---+---+---+---+---+---+---+---+---+---+---+---+
| VNID bit-map....
+---+---+---+---+---+---+---+---+---+---+---+---+

```

Figure 2. Enabled-VN TLV using bit map

```

+---+---+---+---+---+
| INT-VN-TYPE-2 |           (1 byte)
+---+---+---+---+---+
|   Length      |           (1 byte)
+---+---+---+---+---+---+---+---+---+---+---+---+
|           Start VN ID           | (4 bytes)
+---+---+---+---+---+---+---+---+---+---+---+---+
|           End VN ID             | (4 byptes)
+---+---+---+---+---+---+---+---+---+---+---+---+

```

Figure 3. Enabled-VN TLV using Range

```

+---+---+---+---+---+
| INT-VN-TYPE-3 |      (1 byte)
+---+---+---+---+---+
|   Length      |      (1 byte)
+---+---+---+---+---+
|                VN ID          | (4 bytes)
+---+---+---+---+---+
|                VN ID          | (4 bytes)
+---+---+---+---+---+
|                VN ID          | (4 bytes)
+---+---+---+---+---+
|      .      .      .
+---+---+---+---+---+

```

Figure 4. Enabled-VN TLV using list

- Type: indicating different ways to express the VNs that NVE is participating: INT-VN-TYPE-1 is for using bit map to express the interested VNs; INT-VN-TYPE-2 is for using range to express the interested VNs (if the interested VNs are contiguous); IT-VN-TYPE-3 is for using individual VN list to express the interested VNs.

- Length: Variable.

- RESV: 4 reserved bits that MUST be sent as zero and ignored on receipt.

- Start VN ID: The 24-bit VN-ID that is represented by the high-order bit of the first byte of the VN-ID bit-map.

VN-ID bit-map: The highest-order bit indicates the VN equal to the start VN ID, the next highest bit indicates the VN equal to start VN ID + 1, continuing to the end of the VN bit-map field.

If this sub-TLV occurs more than once in a Hello, the set of enabled VNs is the union of the sets of VNs indicated by each of the Enabled-VLAN sub-TLVs in the Hello.

When NVA is distributed, there could be multiple NVAs with each hosting mapping information for a subset of VNs.

Each NVA advertises its availability to push mapping information for a particular virtual network to all NVEs who participate in the VN. NVEs subscribe the relevant NVAs.

## Internet-Draft      NVA mapping distribution

The subscription is VN scoped, so that a NVA doesn't need to push down the entire set of mapping entries. Each Push NVA also has a priority. For robustness, the one or two NVAs with the highest priority are considered as Active in pushing information for the VN to all NVEs who have subscribed for that VN.

## 7.2. Incremental Push Service

Whenever there is any change in TS' association to an NVE, which can be triggered by TS being added, removed, or de-commissioned, an incremental update has to be sent to the NVEs that are impacted by the change. Therefore, proper sequence numbers have to be maintained by NVA and edges NVEs. NAMD incremental message is used to update and maintain the database consistency between NVAs and NVEs. We assume that NVA gets notification from an authoritative source, such as VM management system when TS-NVE attachment changes occur.

A new TLV is needed for to carry NAMD timeout value and a flag for NVA to indicate it has completed all updates.

If the Push NVA is configured to believe it has complete mapping information for VN X then, after it has actually transmitted all of its messages for VN X it sets the Complete Push (CP) bit to one. It then maintains the CP bit as one as long as it is Active.

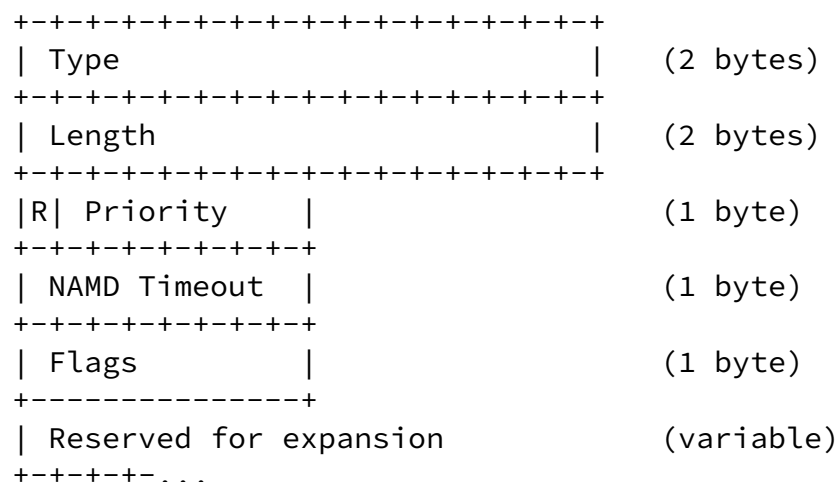


Figure 3. NAMD Complete TLV

Flags: A byte of flags defined as follows:

---

Internet-Draft      NVA mapping distribution

0	1	2	3	4	5	6	7
+---+---+---+---+---+---+---+---+							
UN	CP	RESV					
+---+---+---+---+---+---+---+---+							

The UN flag indicates that the NVA will accept and properly process NVA- PDUs sent by unicast

The CP flag is to indicate that NVA has completed its update.

## [8.](#) Pull Mechanism

Under this mode, an NVE pulls the mapping entries from the NVA when its cache doesn't have the mapping entries.

The main advantage of Pull Mode is that the mapping is stored only where it needs to be stored and only when it is required. In addition, in the Pull Mode, NVEs can age out mapping entries if they haven't been used for a certain period of time. Therefore, each NVE will only keep the entries that are frequently used, so its mapping table size will be smaller than a complete table pushed down from NVA.

The drawback of Pull Mode is that it might take some time for NVEs to pull the needed mapping from NVA. Before NVE gets the response from NVA, the NVE has to buffer the subsequent data frames with destination address to the same target. The buffer could overflow before the NVE gets the response from NVA. However, this scenario should not happen very often in data center environment because most likely the TSs are end systems which have to wait for (TCP) acknowledgement before sending subsequent data frames. Another option is forward, not flood,

subsequent frames to a default location, if the NVE is configured with a default node that has the ability to forward data frames when the NVE doesn't have the mapping information. This node can be the gateway, or a re-encapsulating NVE in NAMD context.

It worth noting that the practice of an edge waiting and dropping packets upon receiving an unknown DA is not new. Most deployed routers today drop packets while waiting for target addresses to be resolved. It is too expensive to queue subsequent packets while resolving target address. The routers send ARP/ND requests to the target upon receiving a packet with DA not in its ARP/ND cache and wait for an ARP or ND responses. This practice minimizes flooding when targets don't exist in the subnet. When the target doesn't exist in the subnet, routers generally re-send

an ARP/ND request a few more times before dropping the packets. The holding time by routers to wait for an ARP/ND response when the target doesn't exist in the subnet can be longer than the time taken by the Pull Mode to get mapping from NVA.

### [8.1.](#) Pull Query Format

Here are some events that can trigger the pulling process:

- o An NVE receives a data frame from the attached TSs with a destination whose attached NVE is unknown, or
- o The NVE receives an ingress ARP/ND request for a target whose link address (MAC) or attached NVE is unknown.

Each Pull request can have queries for multiple inner-outer mapping entries. The message format is defined below:

```

0           1           2           3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| Ver  | Type  | Flags | Count |      Err      |      SubErr      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                                     Sequence Number                                     |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| QUERY 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+...

```

```

| QUERY 2
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+...
| ...
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+...
| QUERY K
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+...

```

Figure 4. Pull Query TLV

Type: 1 for Query. Queries received by an NVE that is not a Pull NVA result in an error response unless inhibited by rate limiting.

Flags, Err, and SubErr: MUST be sent as zero and ignored on receipt.

Count: Number of QUERY Records present. A Query message Count of zero is explicitly allowed, for the purpose of pinging a Pull NVA server to see if it is responding. On receipt of such

an empty Query message, a Response message that also has a Count of zero is sent unless inhibited by rate limiting.

QUERY: Each QUERY Record within a Pull Directory Query message is formatted as follows:

```

      0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|           SIZE           |   RESV   |   QTYPE   |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
      If QTYPE = 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                                   AFN                                   |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   Query address ...
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
      If QTYPE = 2, 3, 4, or 5
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   Query frame ...
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```



SIZE: Size of the QUERY record in bytes as an unsigned integer starting after the SIZE field and following byte. Thus the minimum legal value is 2. A value of SIZE less than 2 indicates a malformed QUERY record. The QUERY record with the illegal SIZE value and any subsequent QUERY records MUST be ignored and the entire Query message MAY be ignored.

RESV: A block of reserved bits. MUST be sent as zero and ignored on receipt.

QTYPE: There are several types of QUERY Records currently defined in two classes as follows: (1) a QUERY Record that provides an explicit address and asks for all addresses for the interface specified by the query address and (2) a QUERY Record that includes a frame. The fields of each are specified below. Values of QTYPE are as follows:

QTYPE	Description
-----	-----
0	reserved
1	address query
2	ARP query frame
3	ND query frame
4	RARP query frame
5	Unknown unicast MAC query frame
6-14	assignable by IETF Review

AFN: Address Family Number of the query address.

Address Query: The query is asking for any other addresses, and the address of NVE from which they are reachable, that correspond to the same interface, within the VN of the query. Typically that would be either (1) a MAC address with the querying NVE primarily interested in the NVE by which that MAC address is reachable, or (2) an IP address with the querying NVE interested in the corresponding MAC address and the NVE by which that MAC address is reachable. But it could be some other address type.

Query Frame: Where a QUERY Record is the result of an ARP, ND,

RARP, or unknown unicast MAC destination address, the ingress NVE MAY send the frame to a Pull NVA if the frame is small enough that the resulting Query message not exceeding the MTU.

If no response is received to a Pull Directory Query message within a timeout configurable in milliseconds that defaults to 200, the Query message should be re-transmitted with the same Sequence Number up to a configurable number of times that defaults to three. If there are multiple QUERY Records in a Query message, responses can be received to various subsets of these QUERY Records before the timeout. In that case, the remaining unanswered QUERY Records should be re-sent in a new Query message with a new sequence number. If an NVE is not capable of handling partial responses to queries with multiple QUERY Records, it MUST NOT send a Request message with more than one QUERY Record in it.

## [8.2.](#) Pull Response

There are several possibilities of the Pull Response:

1. Valid inner-outer address mapping, coupled with the valid timer indicating how long the entry can be cached by the NVE.  
The timer for cache should be short in an environment where VMs move frequently. The cache timer can also be configured.

2. The target being queried is not available. The response should include the policy if requester should forward data frame in legacy way, or drop the data frame.
3. The requestor is administratively prohibited from getting an informative response.

Pull NVA Response messages are sent as unicast to the requesting NVE. Responses are sent with the same VN. The specific data format is as follows:

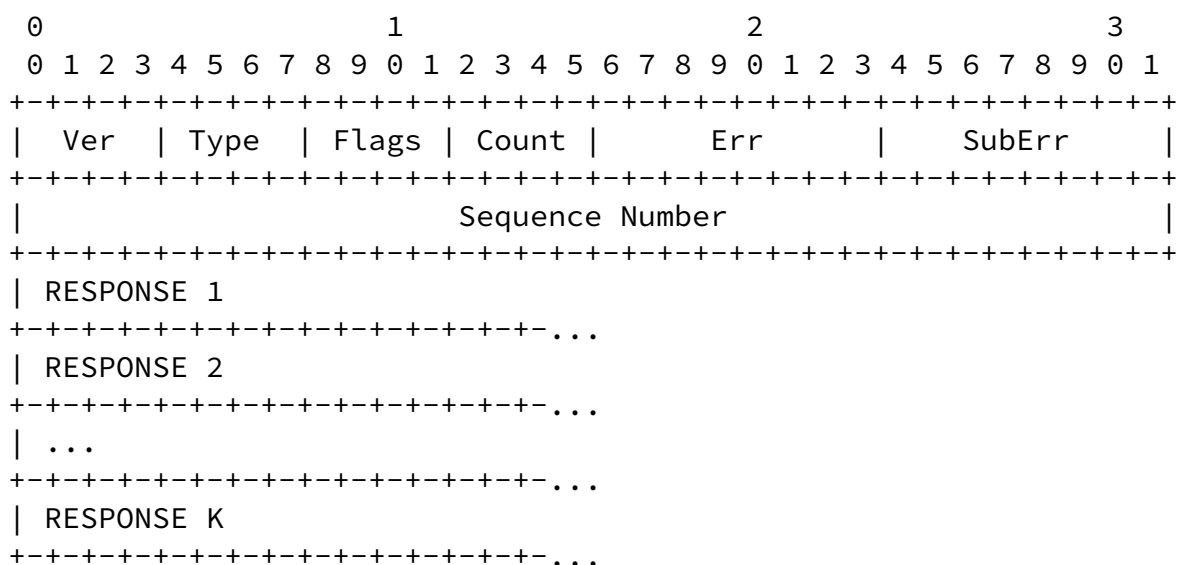


Figure 5. Pull Response TLV

Type: 2 = Response.

Flags: MUST be sent as zero and ignored on receipt.

Count: Count is the number of RESPONSE Records present in the Response message.

Sequence Number: There are many Pull Queries from NVEs; each Pull Query has a different sequence number. The Sequence Number in the Pull Response reflects the sequence number for the query.

Err, SubErr: A two part error code. Zero unless there was an error in the Query message, for which case see [Section 3.5](#).

RESPONSE: Each RESPONSE record within a Pull NVA Response message is formatted as follows:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
---	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----

```

+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|           SIZE           |OV| RESV  |   Index   |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                               Lifetime                               |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                               Response Data ...
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

SIZE: Size of the RESPONSE Record in bytes starting after the SIZE field and following byte. Thus the minimum value of SIZE is 2. If SIZE is less than 2, that RESPONSE Record and all subsequent RESPONSE Records in the Response message MUST be ignored and the entire Response message MAY be ignored.

OV: The overflow flag. Indicates, as described below, that there was too much Response Data to include in one Response message.

RESV: Four reserved bits that MUST be sent as zero and ignored on receipt.

Index: The relative index of the QUERY Record in the Query message to which this RESPONSE Record corresponds. The index will always be one for Query messages containing a single QUERY Record. If the Index is larger than the Count that was in the corresponding Query, that RESPONSE Record MUST be ignored and subsequent RESPONSE Records or the entire Response message MAY be ignored.

Lifetime: The length of time for which the response should be considered valid in units of 200 milliseconds except that the values zero and  $2^{16}-1$  are special. If zero, the response can only be used for the particular query from which it resulted and MUST NOT be cached. If  $2^{16}-1$ , the response MAY be kept indefinitely but not after the Pull NVA goes down or becomes unreachable. The maximum definite time that can be expressed is a little over 3.6 hours.

Response Data: There are various types of RESPONSE Records.

- If the Err field is non-zero, then the Response Data is a copy of the corresponding QUERY Record data, that is, either an AFN followed by an address or a query frame.

- If the Err field is zero and the corresponding QUERY Record was an address query, then the Response Data is the contents of an Interface Addresses APPsub-TLV [IA]. The maximum size of such contents is 253 bytes in the case when SIZE is 255.
- If the Err field is zero and the corresponding QUERY Record was a frame query, then the Response data consists of the response frame for ARP, ND, or RARP and a copy of the frame for unknown unicast destination MAC.

Multiple RESPONSE Records can appear in a Response message with the same index if the answer to a QUERY Record consists of multiple Interface Address APPsub-TLV contents. This would be necessary if, for example, a MAC address within a Data Label appears to be reachable by multiple NVEs. However, all RESPONSE Records to any particular QUERY Record MUST occur in the same Response message. If a Pull NVA holds more mappings for a queried address than will fit into one Response message, it selects which to include by some method outside the scope of this document and sets the overflow flag (OV) in all of the RESPONSE Records responding to that query address.

If no response is received from a Pull request within a configurable timeout, the request should be re-transmitted with the same Sequence Number up to a configurable number of times that defaults to three.

### 8.3. Cache Consistency

It is important that the cached information be kept consistent with the actual placement of VMs. Therefore, it is highly desirable to have a mechanism to prevent NVEs from using the staled mapping entries.

When there is any change in a Pull NVA, such as an entry being deleted or new entry added, and there may be unexpired stale information at some NVEs, the Pull NVA MUST send an unsolicited Update message to the relevant NVEs.

To achieve this goal, a Pull NVA server MUST maintain one of the following, in order of increasing specificity.

1. An overall record per VN of when the last returned query data will expire at a requestor and when the last query record specific negative response will expire.

## Internet-Draft NVA mapping distribution

2. For each unit of data (IA APPsub-TLV Address Set) held by the NVA and each address about which a negative response was sent, when the last expected response with that unit or negative response will expire at a requester.

Note: It is much more important to cache negative reply, because there are many invalid address queries. Study has shown that for each valid ND query, there are 100's of invalid address queries.

3. For each unit of data held by the NVA and each address about which a negative response was sent, a list of NVEs that were sent that unit as the response or sent a negative response to the address, with the expected time to expiration at each of them.

#### [8.4.](#) Update Message Format

An Update message is formatted as a Response message except that the Type field in the message header is a different value.

Update messages are initiated by a Pull NVA. The Sequence number space used is controlled by the originating Pull NVA and different from Sequence number space used in a Query and the corresponding Response that are controlled by the querying NVE.

The Flags field of the message header for an Update message is as follows:

```
+---+---+---+---+
| F | P | N | R |
+---+---+---+---+
```

F: The Flood bit. If zero, the response is to be unicast. If F=1, it is multicast to relevant NVEs.

P, N: Flags used to indicate positive or negative Update messages. P=1 indicates positive. N=1 indicates negative. Both

may be 1 for a flooded all addresses Update.

R: Reserved. MUST be sent as zero and ignored on receipt

Internet-Draft      NVA mapping distribution

### [8.5.](#) Acknowledge Message Format

An Acknowledge message is sent in response to an Update to confirm receipt or indicate an error unless response is inhibited by rate limiting. It is also formatted as a Response message.

If there are no errors in the processing of an Update message, the message is essentially echoed back with the Type changed to Acknowledge.

If there was an overall or header error in an Update message, it is echoed back as an Acknowledge message with the Err and SubErr fields set appropriately.

If there is a RESPONSE Record level error in an Update message, one or more Acknowledge messages may be returned.

### [8.6.](#) Pull Request Errors

If errors occur at the query level, they MUST be reported in a response message separate from the results of any successful queries. If multiple queries in a request have different errors, they MUST be reported in separate response messages. If multiple queries in a request have the same error, this error response MAY be reported in one response message.

### [8.7.](#) Redundant Pull NVAs

There could be multiple NVAs holding mapping information for a particular VN for reliability or scalability purposes. Pull NVAs advertise themselves by having the Pull Directory flag on in their Interested VNs sub-TLV [rfc6326bis].

A pull request can be sent to any of them that is reachable but it is RECOMMENDED that pull requests be sent to a NVA that is least cost from the requesting NVE.

## [9.](#) Hybrid Mode

For some edge nodes that have great number of VNs enabled and combined number of TSs under all those VNs are large, managing the inner-outer address mapping for TSs under all those VNs can be a challenge. This is especially true for Data Center

gateway nodes, which need to communicate with a majority of VNs if not all.

For those NVE nodes, a hybrid mode should be considered. That is the Push Mode being used for some VNs, and the Pull Mode being used for other VNs. It is the network operator's decision by configuration as to which VNs' mapping entries are pushed down from NVA and which VNs' mapping entries are pulled.

In addition, NVA can inform the NVE to use legacy way to forward if it doesn't have the mapping information, or the NVE is administratively prohibited from forwarding data frame to the requested target.

## [10.](#) Redundancy

For redundancy purpose, there should be multiple NVAs that hold mapping information for each VN. At any given time, only one or a small number of push NVAs is considered as active for a particular VN. All NVAs should announce its capability and priority to all the edges.

## [11.](#) Inconsistency Processing

If an NVE notices that a Push NVA is no longer reachable, it MUST ignore any mapping entries from that NVA because it is no longer



being updated and may be stale.

There may be transient conflicts between mapping information from different Push NVAs or conflicts between locally learned information and information received from a Push NVA. NVA may have a confidence level with address table information so, in case of such conflicts, information with a higher confidence value is preferred over information with a lower confidence. In case of equal confidence, Push NVA information is preferred to locally learned information and if information from Push NVAs conflicts, the information from the higher priority Push NVA is preferred.

## 12. Protocols to consider to carry NAMD messages

NAMD messages can be carried by IGP, BGP, or even OVSDb. NV03 WG only focuses on specifying the NAMD message structure. How NAMD TLVs are integrated with BGP or IGP messages will be discussed in the corresponding WGs, e.g. BESS WG.

OVSDb (Open vSwitch Database Management protocol - [RFC7047](#) by individual submission), is to bootstrap a vSwitch with the needed configuration (e.g. number of flow tables, the pipeline among those flow tables, path/link cost, Timer for Spanning Tree, Hello Timer, enabling Multicast snooping, etc). After OVSDb bootstrap a vSwitch, OpenFlow is used to dynamically pass down the flow entries.

Theoretically, some components of OVSDb can be potentially adopted (with update) to achieve the control plane between NVA and NVE. For example, changes to OVSDb are needed to address:

- How Edge nodes request for Push?
- How Edge nodes express the participated VNs?

- How NVA express the supported VNs ranges/list/?
- How Edge nodes feedback newly discovered attached TSs to NVA
- How Edge nodes exchange mapping among themselves.

### 13. Security Considerations

Incorrect information in NVA can result in a variety of security threats including the following:

Incorrect directory mappings can result in data being delivered to the wrong hosts/VMs, or set of hosts in the case of multi-destination packets, violating security policy.

Missing or incorrect data in NVA can result in denial of service due to sending data packets to black holes or discarding data on ingress due to incorrect information that their destinations are not reachable.

Push NVA data messages can be authenticated by including an Authentication TLV. See [[RFC5304](#)] and [[RFC5310](#)].

### 14. IANA Considerations

This section gives IANA allocation and registry considerations.

### 15. Acknowledgements

Special thanks to David Black, Dino Farinacci, Mingui Zhang, XiaoHu Xu for valuable suggestions and comments to this draft.

### 16. References

## 16.1. Normative References

- [RFC4971] J. Vasseur et al, "Intermediate System to Intermediate System (IS-IS) Extensions for Advertising Router Information", July 2007.
- [nvo3-nve-nva-cp-req] [draft-ietf-nvo3-nve-nva-cp-req-00](#), "Network Virtualization NVE to NVA Control Protocol Requirements", Kreeger, et al. July 31, 3013.
- [IA] - Eastlake, D., L. Yizhou, R. Perlman, "TRILL: Interface Addresses APPsub-TLV", [draft-ietf-trill-ia-appsubtlv](#), work in progress.

## 16.2. Informative References

- [802.1Q] IEEE Std 802.1Q-2011, "IEEE Standard for Local and metropolitan area networks - Virtual Bridged Local Area Networks", May 2011.
- [802.1Qbg] IEEE Std 802.1Qbg-2012, "Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks-Edge Virtual Bridging", July 2012.
- [RFC826] Plummer, D., "An Ethernet Address Resolution Protocol", [RFC 826](#), November 1982.
- [RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", [RFC 4861](#), September 2007.

### Authors' Addresses

Linda Dunbar  
Huawei Technologies  
5430 Legacy Drive, Suite #175  
Plano, TX 75024, USA

Phone: (469) 277 5840  
Email: linda.dunbar@huawei.com

Donald Eastlake  
Huawei Technologies  
155 Beaver Street  
Milford, MA 01757 USA  
Phone: 1-508-333-2270  
Email: d3e3e3@gmail.com

Tom Herbert  
Google  
Email: therbert@google.com