Network Virtualisation: The Killer App for IPv6?

Mark Smith
markzzzsmith@yahoo.com.au
Ausnog 2014
Network Virtualisation
Network virtualisation, primarily in the data centre
The networks being virtualised are links and corresponding IP subnets.
They are being virtualised over the top of a physical network
Virtual Network #1, 192.0.2.0/24
Virtual Network #2, 172.18.254.0/23
Virtual Network #3, 2001:db8:0:1234::/64
Physical Network
Virtualisation is achieved by tunnelling unicast, multicast and broadcast traffic over the top of a 'physical' IP network.
Encapsulation Methods
A number of encapsulation methods for network virtualisation have or are being developed and deployed.
VXLAN

“VXLAN: A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks”
IETF RFC7348

with authors from:
Cumulus Networks, Arista, Broadcom, Cisco, VMware, Citrix, Red Hat
NVGRE

“NVGRE: Network Virtualization using Generic Routing Encapsulation”

IETF ID: draft-sridharan-virtualization-nvgre

with authors from:

Microsoft, Arista, Intel, Google, Hewlett-Packard, Broadcom, Emulex
“A Stateless Transport Tunneling Protocol for Network Virtualization”

IETF ID: draft-davie-stt

with authors from:

Nicira Networks
What's Common?

Header/header field used to **carry virtual network identifier**

Header/header field used to **better facilitate traffic load balancing**
Anything Else?
“Use of VXLAN with IPv6 transport is detailed below.”

“Figure 2 VXLAN Frame Format with IPv6 Outer Header”

RFC7348
“The outer IP header: Both IPv4 and IPv6 can be used as the delivery protocol for GRE. The IPv4 header is shown for illustrative purposes.”

draft-sridharan-virtualization-nvgre
“The TCP segments shown in Figure 2 are of course further encapsulated as IP datagrams, and may be sent as either IPv4 or IPv6.”

draft-davie-stt
IPv6 == IPv4?
IPv6 != IPv4!
Change may create Opportunities
I think there are at least 8 IPv6 VN encapsulation opportunities.
A Terminology Detour
IETF NVO3 Working Group

WG focussed on "Network Virtualization Over Layer 3"

Defining frameworks, terminology and methods
Tenant System

Physical or logical device sending or receiving packets over a virtual network e.g., hosts or routers
NVO3 Terminology

Tenant Packets

Frames or packets sent over a virtual network by Tenant Systems
NVO3 Terminology

(IPv6) Underlay Network

Physical IPv6 network across which Tenant Packets are tunnelled
Virtual Network Context ID

Identifier used to distinguish Tenant Packet virtual network membership when tunnelled across the IPv6 Underlay Network
Network Virtualization Edge (NVE)

Device or function encapsulating or decapsulating virtual network traffic at edge of IPv6 Underlay Network

a.k.a. Tunnel End-Point
IPv6 VN Encapsulation Opportunities
Flow Labels for VNs
IPv6 Flow Label

20 bits in size

Can hold $2^{20}$ or 1 Million values
IPv6 Flow Label

Intended to be used as one of the inputs into traffic load balancing e.g. ECMP, LAG
IPv6 Flow Label

Alternative to UDP or TCP header as load balancing input

They may be hidden by IPsec or behind a number of IPv6 Extension Headers
Opportunity #1
Opportunity

Copy whole or part of the Virtual Network Context ID into the Flow Label field
The Virtual Network Context ID value will now be used as an input into IPv6 Underlay Network load balancing.
Benefit

Don't need to add UDP header for load balancing purposes as VXLAN does, avoiding increasing tunnelling overhead.
Benefit

Don't need to use part of GRE key field to encode entropy for load balancing as NVGRE does, avoiding special case field definition.
<table>
<thead>
<tr>
<th>IPv6 Header</th>
<th>Virtual Network Header</th>
<th>Tenant Packet</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow Label</td>
<td>Virtual Network Context ID</td>
<td></td>
</tr>
</tbody>
</table>
Opportunity #2
Opportunity

Use **Flow Label** field to **carry Virtual Network Context ID**
Benefit

This will reduce tunnelling overhead
Flow Label is a *best effort* field, to allow the network to modify it.

Would need to protect its value using some other VN header checksum.
/64s for NVEs
IPv6 Networks should have lots of /64s

RFC6177 minimum for a site is global /56 or 256 x /64s

RFC4193 Unique Local Address space is /48 or 65536 x /64s
Opportunity #3
Opportunity

Use /64s to identify NVEs in the IPv6 Underlay Network, instead of individual unicast 128 bit addresses
Benefit

Now have **more conventional /64 subnet routing** in the IPv6 Underlay Network.
Benefit

May get better IPv6 underlay network forwarding performance, as apparently some routers are better at forwarding for prefix lengths $\leq /64$.
Lower 64 bits?

The complete unicast IPv6 addresses in the IPv6 Underlay Network packets would be the 'Subnet-Router' anycast address.
Subnet-Router Anycast Address

The **Subnet-Router** anycast address is the /64 prefix combined with an 8 octet **Interface Identifier (IID) of all-zeros**.
/64 per NVE
IIDs for Tenants
Opportunity #4
Opportunity

Copy whole or part Tenant Packet addresses into the 8 octet IID portions of IPv6 Underlay Network packet unicast addresses
Benefit

Tenant System addresses will now be used as inputs into IPv6 Underlay Network load balancing.
Load balancing methods for IPv6 traffic are expected to use at least

source and destination addresses

flow label

as inputs
Benefit

Also exposes Tenant System address information to IPv6 Underlay Network operator's analysis and troubleshooting tools such as Netflow/IPFIX
Opportunity #5
Opportunity

Carry Tenant Packet addresses in the IID portions of IPv6 Underlay Network packet unicast addresses, and remove them from the Tenant Packets while being tunnelled.

Restore them in the Tenant Packets at destination NVE.
Benefit

This will reduce tunnelling overhead
Opportunity #6
Opportunity

Copy or carry other Tenant Packet field values into IID's in remaining octets
Benefit

Reduce tunnelling overhead even further
Benefit

Contribute more Tenant Packet field values to IPv6 Underlay Network load balancing
Copy and remove 8 octets of both Ethernet Source Address and Type/Length field into outer IPv6 Source Address 8 octet IID
Easy IPv6 Underlay Multicast
IPv6 Multicast Addresses

An IPv6 multicast address is an identifier for a group of interfaces (typically on different nodes). An interface may belong to any number of multicast groups. Multicast addresses have the following format:

```
+---------+---------+---------+----------------+
|  8      |  4      |  4      |  112 bits     |
+---------+---------+---------+----------------+
|11111111|flags|scope|            |
```

- **8 bits**: IPv6 version (always 11111110 for multicast)
- **4 bits**: Flags (reserved)
- **4 bits**: Scope (reserved)
- **112 bits**: Group ID
IPv6 Multicast Group ID

An IPv6 multicast address is an identifier for a group of interfaces (typically on different nodes). An interface may belong to any number of multicast groups. Multicast addresses have the following format:

```
+----------+----------+----------+
| 8        | 4        | 4        |
+----------+----------+----------+
| 11111111 | flgs      | scop     |
+----------+----------+----------+
        | 112 bits  | group ID |
```
RFC3307, “Allocation Guidelines for IPv6 Multicast Addresses”, narrows it down to 32 bits for better mapping to link-layer multicast addresses
RFC3307 32 bit Group ID Structure

Permanent IPv6 Multicast Address
::0000:00001 - ::3fff:ffff (1 billion)

Permanent IPv6 Multicast Group Identifier
::4000:0000 - ::7fff:ffff (1 billion)

Dynamic IPv6 Multicast Addresses
::8000:0000 - ::fff:ffff (2 billion)
IPv6 Multicast Address Scope

An IPv6 multicast address is an identifier for a group of interfaces (typically on different nodes). An interface may belong to any number of multicast groups. Multicast addresses have the following format:

```
<table>
<thead>
<tr>
<th>8</th>
<th>4</th>
<th>4</th>
<th>112 bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>11111111</td>
<td>flgs</td>
<td>scop</td>
<td>group ID</td>
</tr>
</tbody>
</table>
```
IPv6 Multicast Address Scope

Specifies a multicast traffic forwarding domain, independent of the IPv6 Hop Count

e.g., link-local, admin-local, site-local, organization-local
Opportunity

IANA could allocate a permanent multicast group ID to Virtual Network Encapsulation over IPv6, as per RFC3307 guidelines.

NVEs would automatically subscribe to the Interface-Local, Link-Local and Site-Local scope multicast addresses for this permanent group ID.
Benefit

This would **simplify** and **automate** NVE multicast configuration.
IANA VN Multicast Addresses

IANA Assigned Permanent ID: 4fee:feee (e.g.)

Interface-Local Scope - ff01::db8:4fee:feee

Link-Local Scope - ff02::db8:4fee:feee

Site-Local Scope - ff05::db8:4fee:feee
Efficient IPv6 Underlay Multicast
Use a unicast prefix, up to /64, to generate a locally administered multicast address space

Global and ULA unicast prefixes should generate a globally unique unique multicast address space
This document introduces a new format that incorporates unicast prefix information in the multicast address. The following illustrates the new format:

```
|  8 |  4 |  4 |  8 |  8 |  64 |  32 |
+----+----+----+----+----+----+----+
|11111111| flgs| scop| reserved | plen | network prefix | group ID |
+--------+--------+-----------------+--------+-------------------+-----------------+
Opportunity #8
Opportunity

Use a single unicast prefix, known by all NVEs, and dynamic multicast Group IDs, to create per-VN multicast addresses.
Opportunity

NVEs would only subscribe to the multicast groups for the local VN segments they're attached to.
Opportunity

Tenant System multicasts and broadcasts would now only be sent to NVEs where the Tenant's Virtual Network is present
Benefit

This would reduce unnecessary multicast traffic on the IPv6 Underlay Network
Common NVE unicast prefix

2001:db8:dead:beef::/64

Per-VN Multicast Addresses


X = MC scope, xxx:xxxx = VN Context ID
Many VN multicast groups might exceed IPv6 Underlay Network multicast group capacity.

Mapping a number of VNs to each VN multicast group would be an alternative.
Opportunity #9
Opportunity

Use NVE individual /64 prefixes to generate NVE specific per-VN multicast addresses
Benefit

(I had an idea, but it didn't work out ... ask me afterwards if interested!)
So there's at least 8 opportunities to enhance virtual network encapsulation using IPv6
“Enhancing Virtual Network Encapsulation with IPv6”

IETF ID: draft-smith-enhance-vne-with-ipv6

draft-smith-enhance-vne-with-ipv6

More details of what I've presented here

Recently submitted to IETF NVO3 WG for consideration and comments
Thanks to Fred Baker and Brian Carpenter for their encouragement, review and comments

I'm interested in your thoughts and comments too!
The Bigger Picture
What problem are we really trying to solve with host and network virtualisation?
I think Virtual Hosts/Machines are really being used to create “(Network) Service Containers”
A Network Service is created using one or more applications, commonly and intentionally residing on the same host.
A Virtual Host/Machine is a way to bundle/bind together

application data

application configurations

application binaries and shared libraries

CPU, RAM and storage resource specifications
A single IP address and MAC address are also bound to the Virtual Host/Machine.
1 IP Address == 1 VM == 1 Network Service

1 MAC Address == 1 VM == 1 Network Service
1 IP Address == 1 Network Service

1 MAC Address == 1 Network Service
IP Addresses and MAC addresses aren't really host identifiers any more

They're Service Identifiers
The problem network virtualisation is solving is to allow the “Service Identifiers” to continue to be bound to the “Service Container”
Even when the 'Service Container' is moved to a different physical machine and physical location within in the physical network
So why is IPv6 better for this?
Two Reasons
73 billion billion times more “Service IDs” available than what IPv4 could provide

\( (2^{128} - 2^{32} \text{ or } 2^{96}) \)
With no need for NAT, there is no need for hard state in the IPv6 network.
Moving 'Service Containers' around the network doesn't require trying to shift around hard network state with them.
Finally
What I hope you've got out of this ...
Learned a few new things about IPv6
Seen how IPv6's differences from IPv4 can be taken advantage of.
Perhaps seen how we might do Virtual Networking over IPv6 in the future
Questions?
Thanks for listening!

CC image courtesy of Kiwthing
http://www.flickr.com/photos/kiwisaotome/8261132558/sizes/c/in/photostream/