通用网络虚拟封装

Geneve: Generic Network Virtualization Encapsulation

译者：北京-小武

Sina 微博：北京-小武

技术 blog：http://blog.csdn.net/night_elf_1020

SDNAP 群成员：北京-小武（252480296）

(另请参考 VXLAN　NVGRE　STT 的相关标准)

Network Working Group
Internet-Draft
Intended status: Informational

J. Gross
T. Sridhar
VMware
Expires: August 18, 2014
P. Garg
Microsoft
C. Wright
Red Hat
I. Ganga
Intel
February 14,2014

通用网络虚拟封装（Geneve: Generic Network Virtualization Encapsulation）

摘要：

网络虚拟化的进展（Network Virtualization）涉及了包括软件或硬件的隧道终端、传输 Fabrics 和集中控制集群在内设备大量功能的协作。这些隧道试图集合不同元素的组成完整系统，导致了所有这些组件对隧道的需求产生了影响。如果一个隧道协议能跟上系统演进的步伐，那么其灵活性就是最为重要的方面。本草案描述了 Geneve 机制，一种被设计用于对这些功能和需求变化的确认与调整的协议。

备案录现状：

本互联网草案的提交严格遵守 BCP 78 和 BCP 79 的所有条款。

互联网草案是 IETF（Internet Engineering Task Force）的工作文件（working documents）。需要注意的是其他工作组也有可能对此草案有贡献。至今的互联网草案可参见 http://datatracker.ietf.org/drafts/current/。互联网草案文档有效期是 6 个月并且期间可能会被更新、替换或者其他文档吸纳。把还处于完善中的互联网草案作为参考材料或者引用是不合适的。本分草案将在 2014 年 8 月 18 过期。

目录

通用网络虚拟封装

1.介绍

网络长期以来就有包括隧道，TAG 等各种各样封装机制。然而网络虚拟化功能的兴起为其带来了一股新的吸引力，并随着新协议的引入封装机制的种类有了一定的增幅。在这个领域的大量协议涌现，包括从 VLANs [IEEE.802.1Q-2011]和 MPLS [RFC3031]到最近的 VXLAN [I-D.mahalingam-dutt-dcops-vxlan]、NVGRE[I-D.sridharan-virtualization-nvgre]和 STT [I-D.davie-stt]等，经常导致人们对新的封装格式需求的怀疑和引发其涌现的网络虚拟化是什么的疑问。

当大量的封装协议寻找隔离底层不同区域网络网桥的时候，网络虚拟化将传输网络作为集成系统中多个组件之间的的提供互联的功能。很多时候这个系统很类似与机架交换机在 IP Underlay 网络所起到的总线（backplane）和线卡起到的边缘网隧道联接的作用。从这个角度来看，隧道协议的需求从元素据必要性的数量（the quantity of metadata necessary）和传输节点来说有着显著的不同。现在的有些工作，比如 VL2 和 NVO3 工作组[I-D.ietf-nvo3-dataplane-requirements]已经描述了一些关于数据平面须支持网络虚拟化的特性。然而一种定义需求的附加条件是数据报文携带传输系统传统的需求。一些元素据在所有协议上的使用，对于虚拟化环境里使用至少 24 比特空间的标志来隔离不同的租户来说，必然不是不相干的相关概念而已。这一点经常被用来描述解决 VLAN 仅有 12bit 大小的限制问题，并且当在那个场景下或任何场景下作为一个租户的确切标记 1600 万的空间已经是一个非常大的范围。然而实际情况是元素据并不是唯一的一种用于标识组合和对他们从一群空间中开始其他信息快速编码的方式。事实上与机架交换机的线卡之间交换元素据的标记（TAGS）比起来，24 比特范围的标识符用起来也是相当少的。这类元数据（metadata）有数不清的使用，范围从存储简单安全策略的入端口到基于内容的内部插入的高级中间件（interposing advanced middleboxes）的服务。

现有的隧道协议从各方面尝试去满足新需求的各个方面，仅是为了通过改变控制平面的实现和发展来快速提取（rendered out of）相关数据。另外，软件及硬件的组件和控制器都有不同的优势和演化速度，这一点实际上应该被看做一个优点而不是阻碍或限制。本草案描述了 Geneve，是一种通过提供一个对于隧道封装但不是固定格式以避免上述这些整个系统框架问题的协议。

1.1 用语约定

关键词 "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT"，"SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" 在本文档中用法的解释如同 RFC 2119 中的描述。在本文档中，这些词语仅仅被全大写的时候才如此被解释。这些单词小写的情况下并不适用于仅仅在 RFC-2119 标准的描述。

1.2.术语

本文档有一下条目：

Checksum offload：很多硬件网卡在发送和接收报文时对于上层协议检验和计算和检验的一种优化实现。通常包括 IP 和 TCP/UDP 校验，否则的话需要在软件协议栈中进行计算。

Clos network： 一种将网络 Fabrics 组合成比单个交换机容量大很多且保持无阻塞带宽的一种多点互连技术。ECMP 被用于将数据流分载到多条由 fabric 组成交换机的多条链路上。可能是“leaf and spine”结构也可能是“fat tree”拓扑。

ECMP：Equal Cost Multipath， 一种通过对报文头部字段哈希计算后来从多条等价最优路由中选择其中一跳的一种路由机制，用来保证当避免一条流的重排序时对带宽的较大利用。

Geneve：Generic Network Virtualization Encapsulation，本文档描述的隧道协议草案。

LRO：Large Receive Offload，在接收端好 LSO 的功能一样，表示该处能将多个协议分段（主要是 TCP）合并成的最大数据单元。

NIC：Network Interface Card，网卡应该是隧道终端或传输设备的一部分，能够处理 Geneve 报文或者位于 Geneve 报文处理流程里面。

OAM：Operations, Administration, and Management，一系列用于监视网络并解决问题的工具。

Transit device：一种隧道路径的转发元素。一个传输设备可能有能力解析 Geneve 数据帧格式，但并不封装或终结 Geneve 报文。

LSO：Large Segmentation Offload，在很多商用网卡上提供了这种功能，允许传给网卡的数据单元比网卡 MTU 大一些的，这样可以提高性能，并且网卡要负责将数据单元分成多份并附带有正确协议头部。对应到 TCP/IP 具体的功能，这个特性就是众所周知的 TSO（TCP Segmentation Offload）。

Tunnel endpoint：将以太网帧封装到 Geneve 头部里的一个组件，反之也可。作为最终的隧道元素据的处理者，终端对隧道报文头部的解析能力有最高优先级的需求。隧道终端可能是软件实现，也可能是靠硬件实现，甚至是软硬件结合。

VM：Virtual Machine，虚拟机器。

2.设计要求

Geneve 被设计用语支持网络虚拟化的使用场景，其经典的做法是通过在位于 hypervisors 里的虚拟交换机、物理交换机、中间件或其他电子设备之间建立隧道作为支路干线（backplane）。任何一个 IP 网络均可被用于 CLOS 底层设施（underlay），其通常在所有连接点之间选择使用 ECMP 链路作为选择以提供持续的双向通路。

如图 1 中是一个 hypervisor 的例子，TOR 交换机用 Geneve 隧道在物理服务器 WAN 上联口的一个简单 CLOS 网络上提供互联。这些隧道被用于封装和转发虚拟机或物理链路相关的数据帧。

为了满足网络虚拟化的需求，隧道协议应该有能力利用 underlay 和 overlay 网络中各种网络设备的功能的差异（和演进）关系。作为被关注的数据平面隧道协议，需要满足以下几点以：

■ 数据平面无论在通常情况下还是扩展的情况都是足够的以满足对现在和将来控制平面的支撑

■ 隧道组件必须能在软件和硬件的最低级普通特征上（the lowest common denominator）都没有约束性(restricting capabilities)的有效实现

■ 在已有的 IP Fabric 上有很高的性能

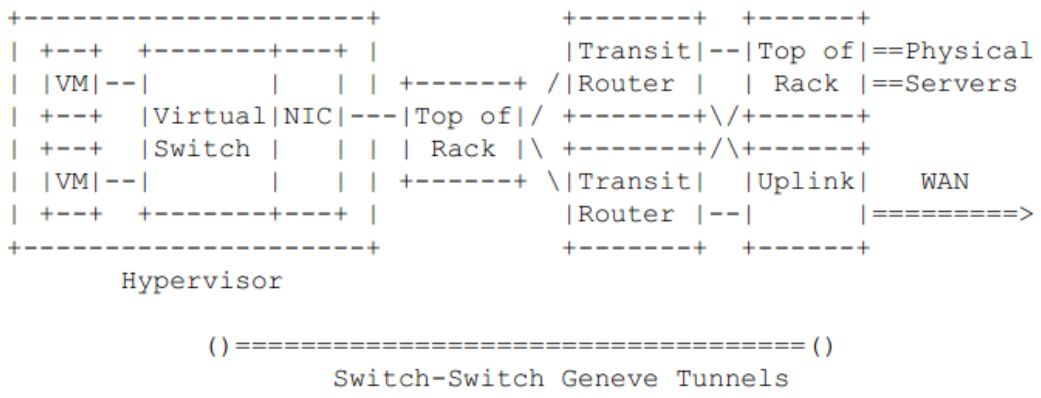


图 1: Geneve 部署样例

这些要求将在后续章节有详细描述。

2.1 控制平面独立性

尽管一些网络虚拟化的协议已经包含了一个控制平面作为隧道格式标准的一部分（最著名的例子是在 VXLAN 的原始标准中描述了一个基于组播学习的控制平面），这些标准仅仅是被认为大量的描述了数据格式。VXLAN 数据帧格式实际上已经被认为一种基于 VXLAN 的控制平面的多样性。数据格式的固定有一个非常明显的优势：大多数的协议仅仅有一些不太重要的区别和重复的工作也没有多大有点。

然而这些还不能说是控平面，它们在很多基本的方面都不同。标准化也在需求、目标和部署场景明显减少了多样性（is also less clear given the wide variety）。

实际上，Geneve 旨在提供一种比较单一的隧道格式标准，以满足实现多种控制平面不用指定任何一种隧道协议的能力。本文同时还提出了一种数据格式及增加将来控制平面增强其不过时的可能性。

为达到这种级别有效的灵活性，需要一个可选的设施以允许新的元数据类型能被定义、部署和终止或销毁。使用选项也允许不同的产品的差异性，通过鼓励每一个设备商的核心领域独立开发，从而导致整个领域的快速发展。到现在为止实现选项最通常的机制是 TLV 格式（Type-Length-Value）。需要注意的是，当选项能用于支持数据帧的非线速转发时，这和数据帧的隔离与直接路由一样的重要（对虚拟机来说，例子就是前面给出的基于如端口的安全策略和服务生效（service interposition）都需要在数据报文中添加 TAG）。因此，限制某些简化路径的控制帧的扩展性就有需要被限制，因为其不满足设计需求。

2.2 高效的实现

软件的灵活性和硬件的性能往往是冲突的并且很难被解决。对于一系列给定的功能，很明显的需要使性能最大化。然而这并不意味着新的特性也能在原来速率运行的情况下现今也不被允许。因此对已一个协议的高效实现来说是一些列常规功能可通过一种优雅的机制合理的通过交叉平台（across platforms）来合理处置，并且能在合适的场景下适用于高级特性。

可变头部长度的使用和选项在协议中在硬件中是否真的是一种高效的实现经常被提出质疑。为了解释这个问题在 Geneve 的本文中首先明白将硬件分为两种类型是非常重要的：隧道终端和传输设备。终端必须能够解析各种变长头部，包括各种选择和执行动作。因此这些设备在协议里积极参与（actively participating），他们受 Geneve 影响最大。而传输设备可以调整他们的输出到更容易接收。作为一种新的功能变得详尽的定义以被添加到终端里，支持的选项能够被设计用来约定限制（ordering restrictions）和其他技术能简单解析。传输设备或许能够解释选项和参与 Geneve 报文处理。然而作为非终结设备，他们不用封装或终结 Geneve 报文。传输设备对 Geneve 报文的处理时的参与（participation）是可选的项。

另外，无论隧道终端还是传输设备都可能用于网卡的 offload 功能比如校验 offload 内容以提高 Geneve 报文的处理性能。Geneve 可变长度头部的出现不应影响隧道终端和传输设备作为 offload 功能来的使用。

2.3. 使用标准的 IP Fabrics

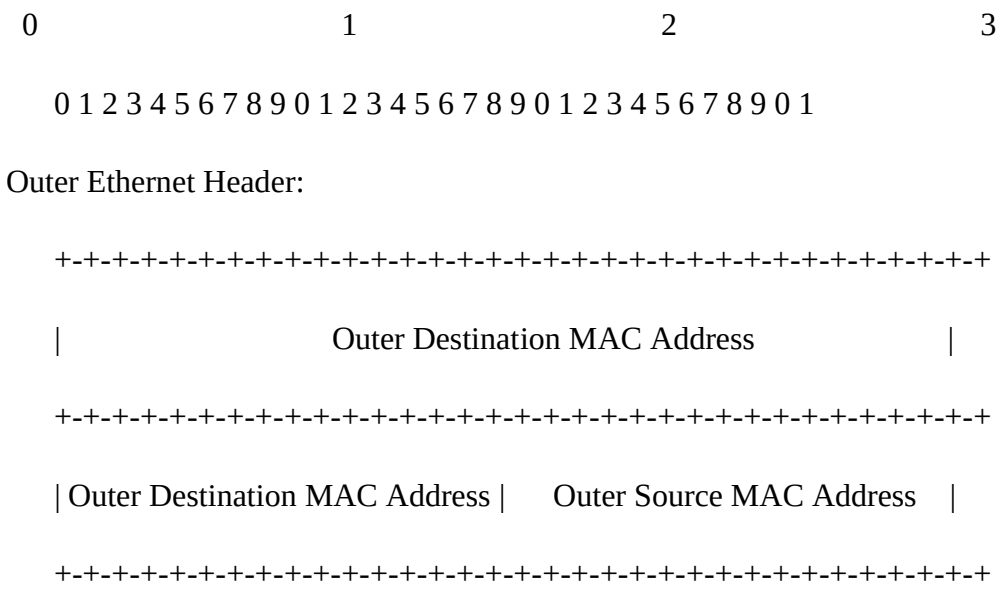
IP 技术已经很牢固的战局了传输机制领域的地位，并且随着多次的技术演进使其更强壮、有效和低廉。所以使用 IP fabrics 作为 Geneve 的传输网络是理所当然的。幸运的是 IP 封装和寻址的使用有足够的能力通过在网络的交换和路由技术实现传输报文到争取终端的主要目标。

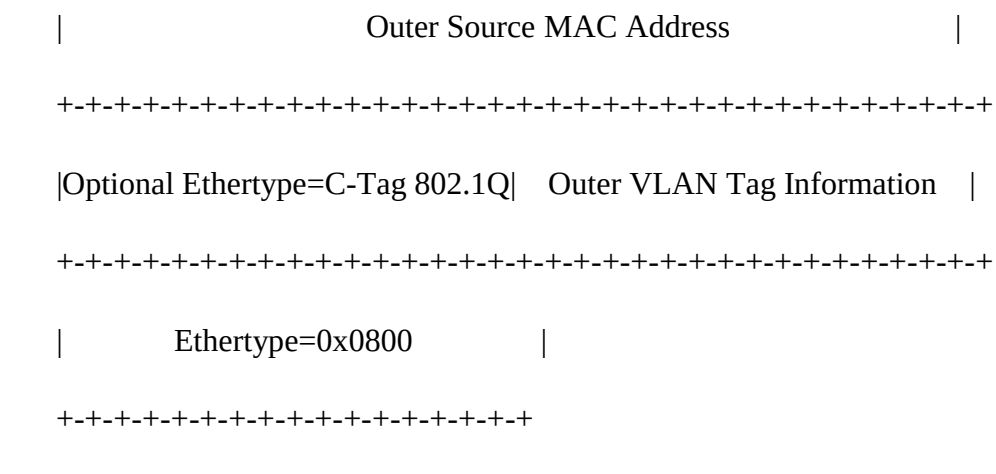
另外，最近所有的 underlay fabrics 被设计成利用数据流的并行模式，并没有没有引入重新排序的方式而是用多个链路中对每个流的传输负载进行分散处理。这些 ECMP（Equal cost multipathing）技术经典的做法是通过解析和哈希报文的地址及端口号来选择出口链接。然而如果没有额外的协议信息的话隧道经常导致 ECMP 性能的降低因为封装流从 fabric 中被隐藏并且仅仅终端地址能被用于哈希。因为 Geneve 在这些现有 fabrics 上工作良好的需求，从封装报文中提取隧道头信息是非常有必要的。最常用的技术就是使用 UDP 的源端口，这点将在 3.3 章节讨论。

3. Geneve 封装细节

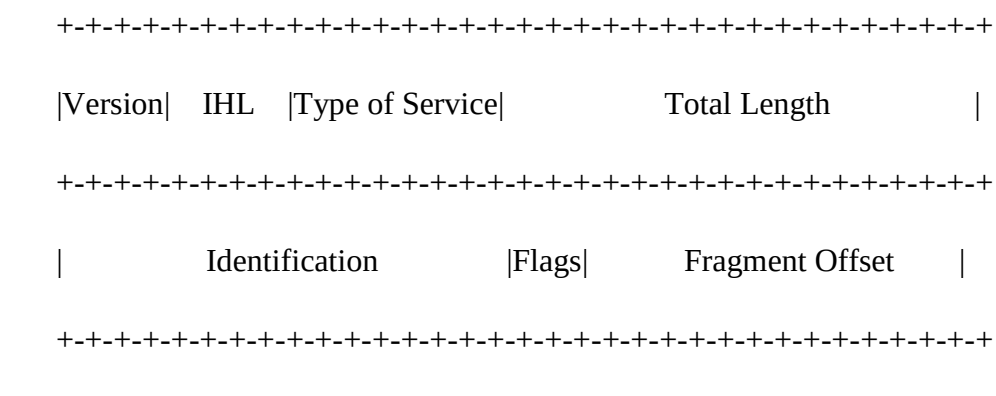
Geneve 帧格式是一个简化的封装在 ipv4 或 ipv6 的 UDP 里的隧道头部组成。一个固定的隧道头部提供了控制信息，以及一个关注简介方面的基础级别的功能和互操作性说明。这个头部经常后面跟随着一些列可变选项以便于将来创新。最后，内容包含了一个表征类型的协议报文单元。后续的讲解将提供以太网内容的 Geneve 帧在以太网上传输的示例。

3.1. IPV4 Geneve 帧格式

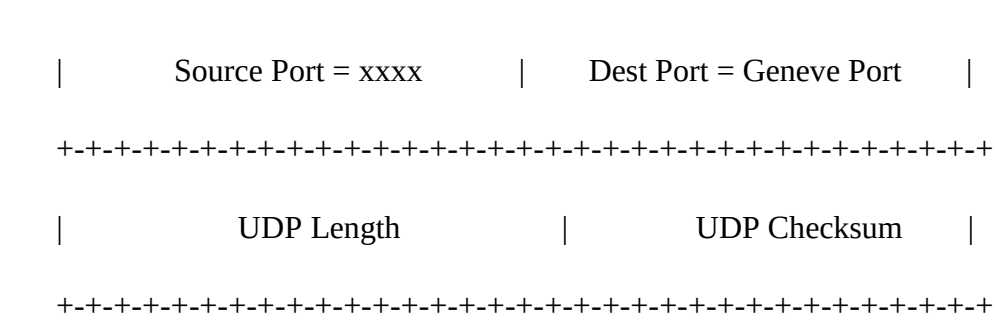




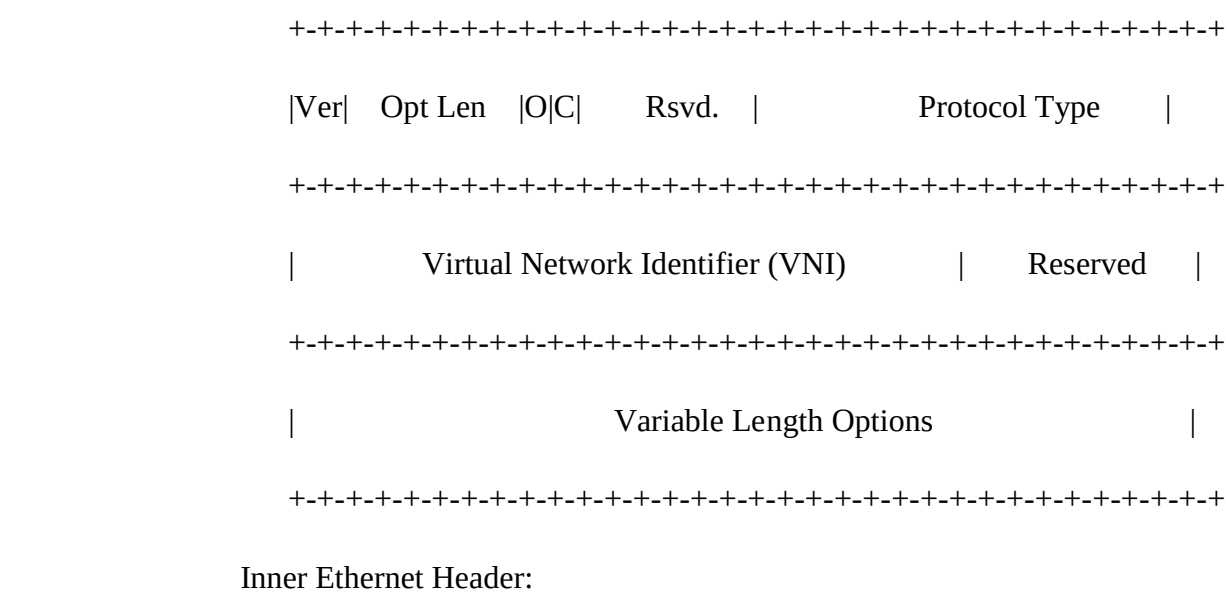
Outer IPv4 Header:



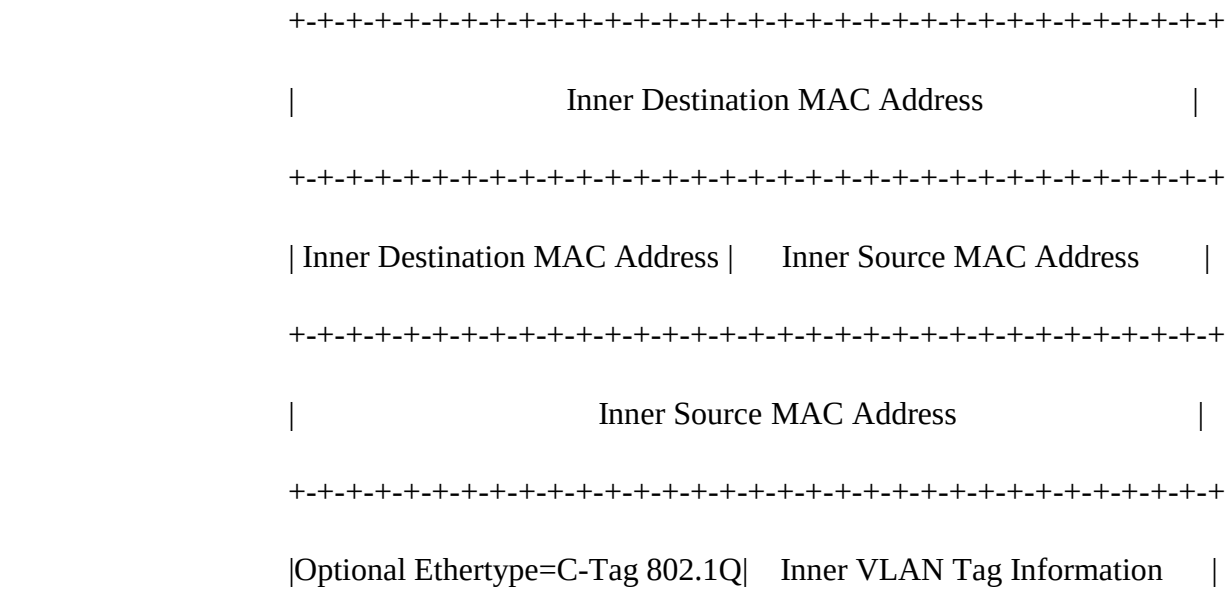
Outer UDP Header:



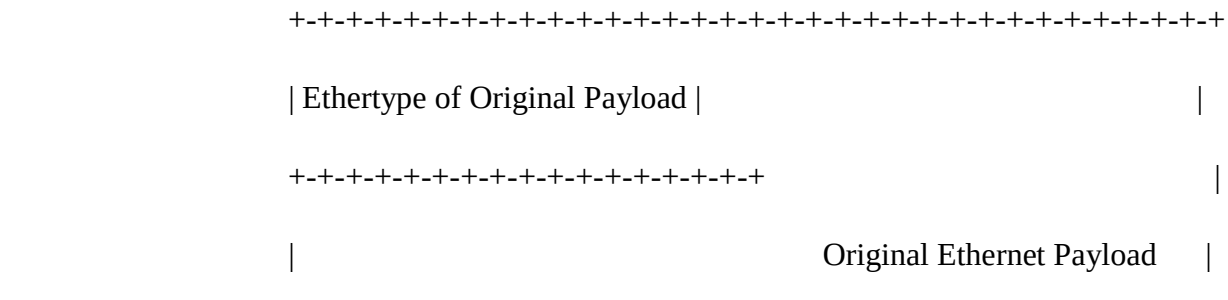
Geneve Header:



Inner Ethernet Header:



Payload:



```
|                                                               |
| (Note that the original Ethernet Frame's FCS is not included) |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Frame Check Sequence:

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     New FCS (Frame Check Sequence) for Outer Ethernet Frame     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

3.2. IPv6 Geneve 帧格式

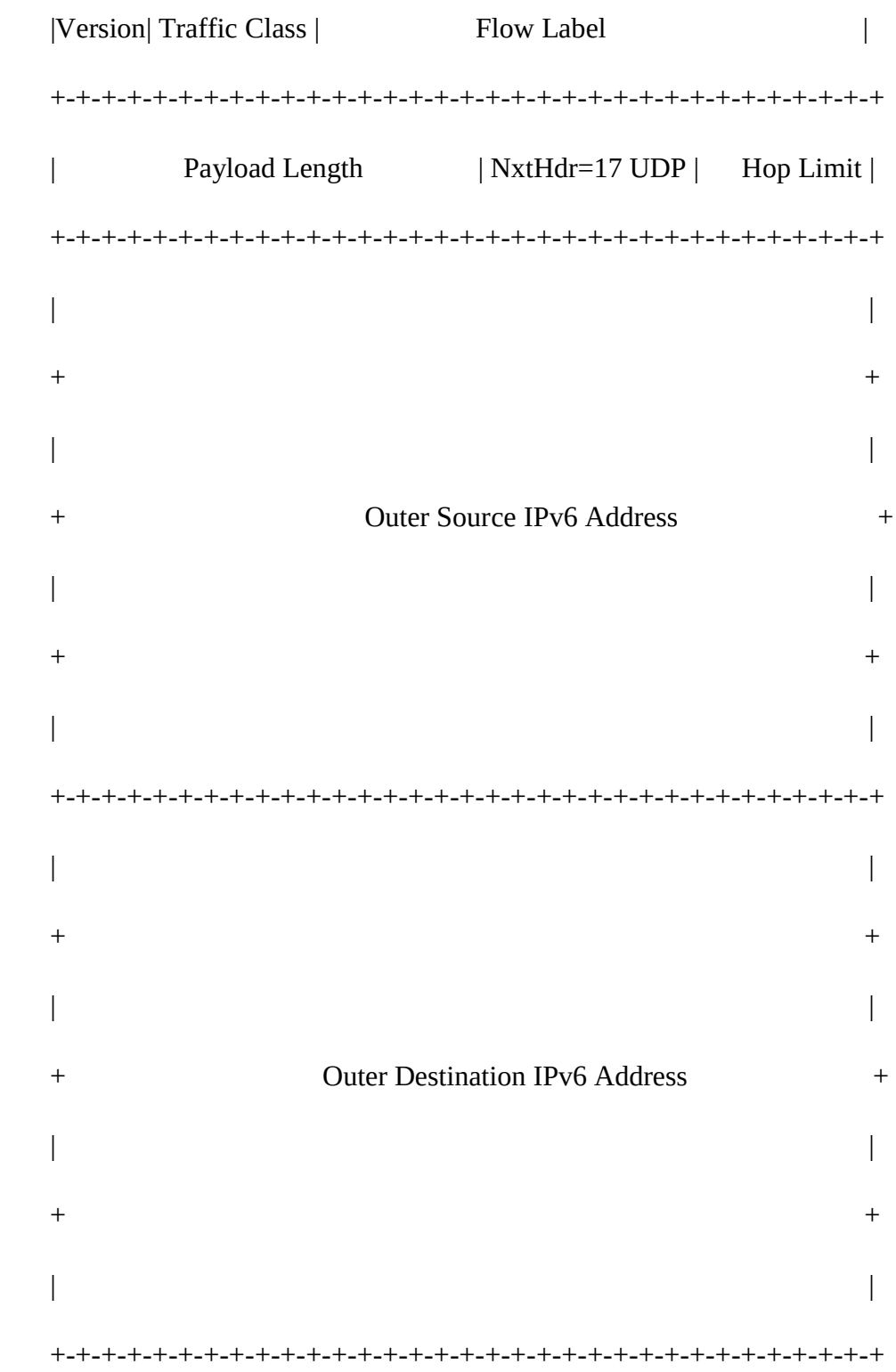
```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
```

Outer Ethernet Header:

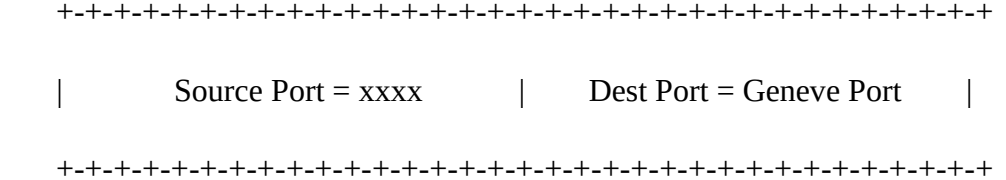
```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   Outer Destination MAC Address               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Outer Destination MAC Address |     Outer Source MAC Address    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   Outer Source MAC Address                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Optional Ethertype=C-Tag 802.1Q|   Outer VLAN Tag Information    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Ethertype=0x86DD                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

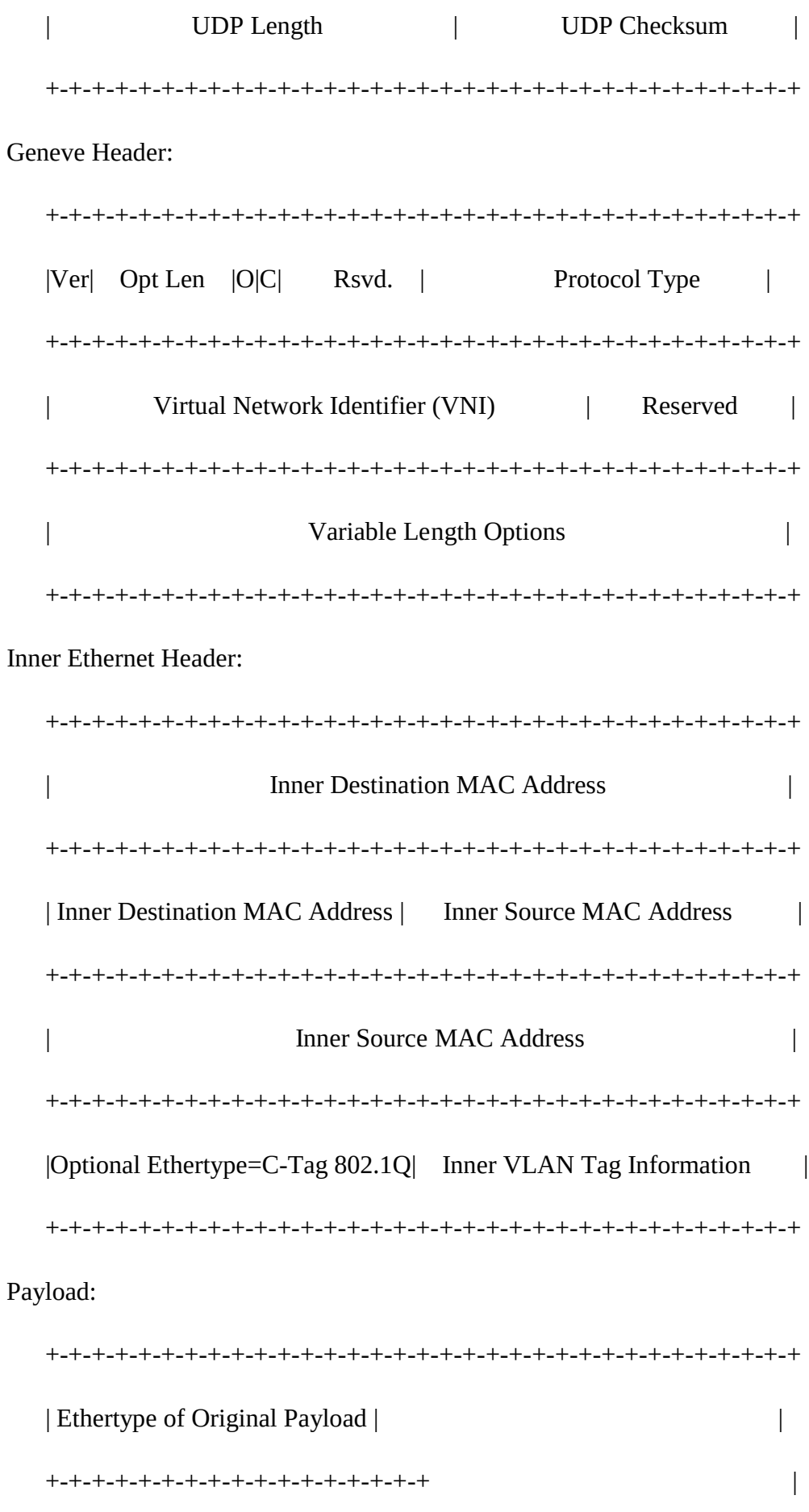
Outer IPv6 Header:

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```



Outer UDP Header:




| UDP Length | UDP Checksum |
Geneve Header:
|Ver| Opt Len |O|C| Rsvd. | Protocol Type |
| Virtual Network Identifier (VNI) | Reserved |
| Variable Length Options |
Inner Ethernet Header:
| Inner Destination MAC Address |
| Inner Destination MAC Address | Inner Source MAC Address |
| Inner Source MAC Address |
|Optional Ethertype=C-Tag 802.1Q| Inner VLAN Tag Information |
Payload:
| Ethertype of Original Payload | |

```
|                                                 Original Ethernet Payload    |
|                                                                              |
| (Note that the original Ethernet Frame's FCS is not included)                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Frame Check Sequence:

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     New FCS (Frame Check Sequence) for Outer Ethernet Frame                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

3.3. UDP 头部

使用 UDP（RFC0768）来基于以太网和 iP 网络无连接的封装主要原因是为路由器的沿用 ECMP 提供更多信息。因此头部字段主要有一下几个：

源端口: 一个隧道终端入方向选择定的端口号。此源端口号应该对某个封装流的所有报文是一样的，当用来防止当被重排序后导致使用不同的转发路径的时候。倡导通过多条链路将流进行均匀分布，这样的源端口号就可能被通过封装报文头部被用于比如 5-tuple 等 HASH 运算。因为一个端口号代表了一个流标识而不是一耳光真正的 UDP 连接，因此整个 16 比特范围都可能被用于来扩大信息熵值

目的端口: 固定的知名目的端口号都是被 IANA 来分配。这个端口号必须被双向流都能使用。因为这个端口号还没有被具体分配下来，所以实现中推荐的做法是可以配置的。

UDP 长度：整个 UDP 报文包括 UDP 头部的长度。

UDP 检验和：无论是 IPV4 还是 Ipv6（RFC6935，IPv6 and UDP Checksums for Tunneled Packets）传输封装报文的检验和可能是全零。当一个 UDP 检验和全零的 UDP 报文到达时，其必须被接收和封装。如果入方向隧道终端选择用非零值来封装报文校验和，其必须是正确计算得到的 UDP 的检验和。这样一旦接收到这样的报文出方向终端也会检验其有效性。如果接受者实施了校验并且校验和是不正确的那么报文必须被丢弃。反之，报文必须被接收并解封装。在网路可靠性不高或者报文没有被其他校验或循环冗余校验包含的情况下，建议开启 UDP 检验和功能来包含 Geneve 头部与可选项的完整性。

3.4.隧道头部字段

版本号(Ver ，2 bits): 现在版本号是 0. 终端对于接收到的未知版本号的报文必须丢弃。无隧道终结 Geneve 报文处理流程的设备对于未知版本号的报文必须将其作为带有未知 payload 的 UDP 报文进行处理。

可选项长度(Opt Len，6 bits):可选字段长度，以四字节倍数计量，不包括 8 个自己的隧道固定头部的长度。这个长度对于 Geneve 头部的支持最小 8 个字节，最大 260 个字节。Payload 头部的开始可以通过使用 Geneve 头部终端作为基址进行偏倚获得。

O (1 bit): 代表该报文是 OAM 帧，其包含了一些控制信息而不是数据内容。终端一定不能转发这些内容并且传输节点一定不允许试图解析并处理它。因为这些控制信息帧是非常稀少的，所以建议终端用一个高优先级队列来传输这些报文（例如，将 ASIC 转发的这些报文有意的定性到特定的 CPU 或者在网卡（NIC）上将这些报文隔离到到特定的流量控制）。传输设备也必须不改变这些基于这些报文比特级的转发路径，比如 ECMP 的链路选择等。

C (1 bit): 关键选项标志。一个活多个选项可以设置关键项比特。如果这个比特被设置了那么隧道终端必须解析这个选项的列别来解析各个关键选项。如果没有任何一个选项类型可以支持那么终端需要默然丢弃这个带有 C 比特设置的帧（包括无效的组合比如 C 比特职位并且选项长度是零或没有任何选项的相应 C 位被置上）。如果这个比特没有被设置对到终端可能通过选项长度提取所有选项并且转发这个解封装的帧。传输设备不允许丢弃或修改这些报文的任何比特。

保留位（Rsvd，6 bits）： 保留字段必须在传输过程中全零并且接收时不关心。

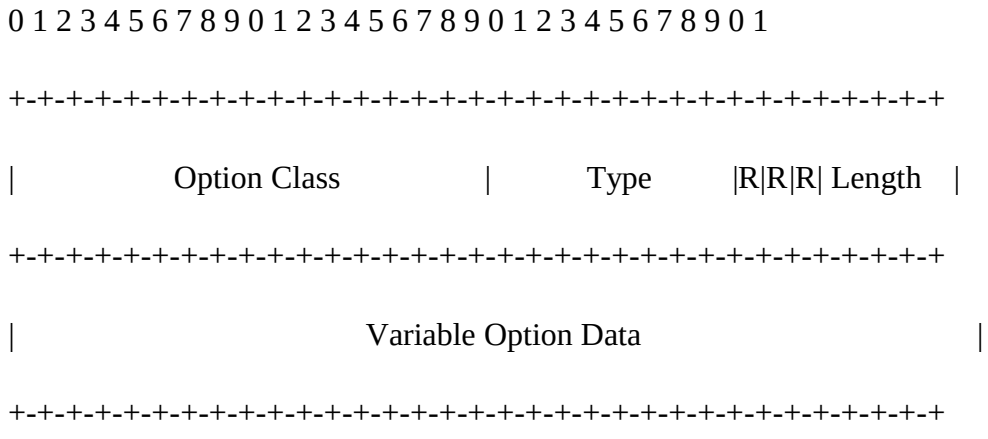
协议类型（Protocol Type ，16 bits）：协议字段单元的类型出现在 Geneve 头部后面。这个遵循了 EtherType 在以太网中的做法，其典型代表值是 0x6558。

虚拟网络标识（VNI ： Virtual Network Identifier，24 bits）：一个特定虚拟网络的标识在很多情况下可能代表了 L2 网络的分段，然而控制平面定义的是解封装报文的转发逻辑。VNI 可能被用于当做 ECMP 转发决策的一部分，或者当通过多 CPU 进行负载均衡时作为一种对封装报文里重叠地址空间的区分机制。

保留字段（Reserved ，8 bits）： 保留字段必须在传输过程中全零并且接收时不关心。传输设备必须维持转发行为的一致性且无需考虑选项长度（Opt Len）的值，包括 ECMP 链路选择。这些设备应能够避免被重排序到低优先级路径的方式转发包含选项的报文。

3.5.隧道选项字段

0 1 2 3



Geneve Option

Geneve 头部基址紧跟的是零个或多个以 TLV（Type-Length-Value）格式的选项字段。每一个选项包含了四个字节的可选头部长度和基于类型解析得到的一个不定长度的可选数据。

可选类型（Option Class ，16 bits）： 类型（Type）字段的命名空间。IANA 将被请求来创建一个“Geneve 选项类”来为那些对创建可选类型感兴趣的团体、技术专家和提供商分配标识。每一个团体可能分配相互独立的类型来进行试验和快速创新。随着时间的推移，某几种特定的可选项被众人所知并且给定的实现也是一些源当中的可选类型。另外，IANA 将被请求为标准和试验等选项保留一个具体的特定范围。

类型（Type，8 bits）：类型表明了在这个选项中包含的数据格式。选项主要被设计用于未来的扩展和创新，因此这些选项的标准化形势将被在不同的文档中定义。选项类型的最高顺序比特置位表明了这是个关键选项。如果一个接收终端没有识别出这个选项并且这个比特被职位那么这个数据帧必须被丢弃。如果任何选项的这个关键比特被置位那么在 Geneve 报文头部的基址的 C 位也必须被职位。传输设备不能丢弃所有这些基于此比特的（on the basis of this bit）报文。

对于丢弃所有未知关键选项的需求在于整个隧道终端系统而不是某个特定的实现组件。例如，在一个由转发 ASIC 和通用 CPU 组成的系统里，这并不意味着整个报文必须在 ASIC 里丢弃。一种可能是实现是将报文通过线速控制通道送上 CPU 来进行慢路径异常处理。

保留位（R ，1 bit）：选项控制标记保留备将来使用。必须被全部置零并且接收时忽略。

长度（Length，5 bits）：选项的长度指不包括选项头部的以 4 字节的整数倍。每一个选项的全部长度在 4 和 128 字节之间。报文当中的所有选项的长度和不等于报文基础头部的选项长度（Opt Len）的时候是无效的并且必须被接收到的终端无任何动作的丢弃。

可变选项数据（Variable Option Data）：选项数据通过类型来解析。

3.5.1.选项处理

Geneve 选项的的用途是在隧道终端的发出和处理。选项可能被传输设备随着隧道路径被处理。本文档仅仅是明细了隧道终端对选项的处理。文档将来版本将提供传输设备对选项的处理细节。不处理 Geneve 报文选项字段的传输设备应该像其他 UDP 数据帧那样处理 Geneve 数据帧并且维持转发行为的持续性。在隧道终端，对选项字段的产生和解析最终在控制平面，这点超出了本文档的讨论范围。

然而 为了确保各种设备的互操作性终端设备上有两点需要确保：

1） 接收终端必须丢弃包含选项类型中 C 比特被置位的未知选项的报文

2） 发送终端不能假设选项被接受者按照传输的顺序处理。

4.实现和部署方面的考虑

4.1. Geneve 在 IP 中的封装

作为一种基于 IP 的隧道协议，Gevene 沿用了现有协议的很多属性和技术。对这些的应用将在后续详细描述，当然大多数常规能在 IP 层或 IP 隧道可以用的概念都能在 Geneve 协议里起作用。

4.1.1.IP 分片

为了防止分片并发挥最大性能，最好的方法是确保物理网络的 MTU 至少大于等于加上隧道头封装后的报文 MTU。手工或者更上层的配置可能被用于确保分片行为绝不会发生，然后在某些场合下这些配置可能不生效。Geneve 协议的报文在 IPV4 环境中传输推荐使用通过设置 IP 头部的 DF 位的路径 MTU 发现的功能（这点在 IPV6 中是默认配置）。在传输网络上使用路径 MTU 发现将提供封装端关于链路的软件状态，这可能被用于来防止或减少在虚拟网络中的分片。如有必要，需要注意分配可能在封装 payload 时进行。在虚拟网络中封装终端正好是 L3 节点也是有可能的，这种情况下终端可能使用得到的传输 MTU 和隧道头部长度来实现路径 MTU 发现或者把内部报文分成正确的大小。很多情况下，比如实现了一个完整的透明 L2 网桥，隧道终端和 payload 打交道是不太可能或没有必要的。在这些场景下，传输 IP 头部的分配可能起的作用是确保连接。如果一个报文被分片，终端应该使用传输链路的路径 MTU 以确保大小正是两端终端所需求的选择大小。注意一些实现可能不能支持分片或者其他一些普通的 IP 头部功能，比如选项或扩展头部等。

4.1.2.DSCP 和 ECN

当使用 Geneve 封装 IP 报文（包括基于以太网的方式）时，有多个选项来讲 DSCP 和 ECN 从报文内部映射到传输的隧道中并且在接收端进行反方向处理。RFC2983 列出了把 IP 头部内部和外部的 DSCP 映射的一些考虑。网络虚拟化可以被描述为是一种非常典型的管道模型，隧道报文头部的 DSCP 值基于一个策略来设置（也可能是一个固定值，一宗是基于内部流分类，另外就是一些其他分组流的机制）。统一模型（其将内部和外部的 DSCP 看成一个值，通过对内外报文头部的复制）的各方面可能也有使用，比如讲基于传输隧道出方向的报文头部内部的 DSCP 标记等。然而这种统一模型可能和挽留过虚拟化在概念上不一致，其主要目的在于提供一种针对封装流和物理网络之间的强隔离功能。文档 RFC6040 描述了这种基于 IP 隧道来传递 ECN 的机制并且将拥塞标记扩散到报文内部。这种行为应当在 Geneve 封装的 IP 报文内部亦被采用。

4.1.3.广播和组播

两个终端的 Geneve 隧道可能是点对点的单播也可能是用广播或组播地址。但并不需要内部和外部的地址都匹配上这个域。举例来说，在不支持组播的物理网络中，封装组播数据流可能要多个单播隧道或者基于策略的单播转发到本地来替代（可能这里需要被替代）。而对于支持组播的物理网络可能使用其一年复制报文的功能优势来实现对报文的封装。这种情况下，组播地址就可能在物理网络被分配给相应的租户，封装组播组或其他因素。这些组的分配是控制平面的一个组件，并且超出本文档的讨论范围。当一个物理环境的组播被使用时，Geneve 头部的 C 位可能被一组有复杂功能的设备所使用，因为每一个设备可能仅仅解释关注那些不是关键项，但对其却具有重大含义的选项。

4.2.网卡 Offloads 功能

现代的网卡现在都能提供各样的 offloads 功能以便能提高报文的处理效率。很多种 offloads 的实现，仅需要能够提供简易方式来对封装报文进行解析（例如校验和的 offloads 等）。然而，优化项中比如 LSO（**TCP Segmentation Offload**）和 LRO（**Large Receive Offload** ）等导致了一些选项自己的处理因为他们必须在多个报文之间进行复制或合并。这些情况下，就要求对 offloads 逻辑不需要改变一处理新引进的选项。为了实现这个功能，下面针对这些选项的定义做了些限制：

■ 当使用 LSO 功能时，网卡必须复制整个 Geneve 报文头部和整个选项，包括这些设备未知部分在每一个相关部分的内容。然而，支持设备的一个给定的选项定义可能覆盖这条规则并且导致不同的行为。相反的，当开启 LRO 功能时，网卡可能假定对这些选项的二进制比较足以确保相等结果的正确性并且把相同 Geneve 头部的报文可能进行合并。

■ 选项的排序是无意义的，并且不同选项顺序但拥有相同选项的报文可能会被有相似的处理流程；

■ 网卡使能 offloads 功能时决不能丢弃未知选项的报文，包括那些被标记的关键的报文。

对 Geneve 使用 offloads 功能上面所列举的例子在现有的实现中还没有任何需求。然而这些 offloads 机制现在正广泛地被部署到商用网卡里，这里所描述的规则目的也在于能够有效的处理现在和将来流过各种设备的选项。

4.3.内部 vlan 处理

Geneve 机制可以封装大量范围的各种协议并且一个给定的实现中可能仅对一个范围的常用协议进行支持。然而因为以太网被广泛部署，那么对于在封装内部的以太网帧中增加对 VLAN 行为的描述就非常有用。

像任何一个其他协议一样，支持内部 VLAN 头是一个可选项。在很多情况下，封装 vlan 的使用基于安全和实现的考虑可能被不允许。然而在其他情况下让 VLAN 的 trunk 功能在 Geneve 隧道中实现透传功能时，则非常有用。导致最终对 VLAN tag 的处理无论是隧道终端的入方向还是出方向都是基于终端或(和)控制平面的配置，并且是一类没有明确定义的数据格式部分。

5.互操作性问题

仅仅从数据平面来看，Geneve 机制没有引入任何互操作性问题，因为大多数设备上其就是一些 UDP 数据帧。然而，现在现在的虚拟网络环境里已经有一定数目的隧道协议被部署，因此要对这些协议的传输和兼容上有一些考虑。

自从 Geneve 机制是网络虚拟化中最常用三种协议(VXLAN、NVGRE 和 STT)功能的一个超集，因此其应该对现存控制平面明确接口(straightforward to port an)以最小的改进使其照常运行。无论新旧协议的数据帧格式只要支持心痛的功能集，就没有需求来进行费力的转换——终端将直接与其他每一个使用常规协议者进行通信，即使在一个单一的整体系统里二者仍然可能不同。作为传输设备主要功能是将数据帧基于 IP 头部进行转发，所有的协议动作类似并且这些设备不会引入额外的互操作性问题。

为了对传输功能有所帮助，这里强烈建议实现里支持 Geneve 机制和现存隧道协议的同步操作，因为其被期待与其他节点混合在一起做完一个常规的单一节点。最终，以前的协议可能因长久不被使用而被逐步淘汰。

6.安全考虑

作为一个 UDP/IP 报文，Geneve 报文没有继承任何安全机制。最终当一个攻击者访问了传输 IP 报文的 underlay 网络，其已经有能力窃取和欺骗报文。合法但恶意的终端也可能被其他租户用来伪造隧道头部的标记来获取对自己网络的访问权。

在一个特定的安全领域，比如被单一供应商操作的一个数据中心，最普通和最高性能的安全机制是隔离信任组件。隧道流量只能在一个单独的 vlan 里传播并且在任何不被信任的边界处被过滤掉。另外，隧道终端应该在环境里只能被服务供应者操作，比如 hypervisor 自身而不是一个客户 VM。

当穿越多个不被信任的链路时，比如公共因特网时，IPsec（RFC4301）可能被用于提供认证和（或）数据报文的加密处理。如果远端隧道终端不能被完全信任，例如其取决于客户的约定，那么净化所有的隧道元素据来防止租户基于跳的攻击也是有必要的。

7. IANA 考虑

一个 UDP 的目的端口号将在用户范围（1024-49151）被从 INAN 申请。另外，IANA 还被请求来串讲“Geneve 选项分类”的注册以用来分配选项类型。这个应该是一个已登记的可描述的 16 进制数字符串。在标准选项中标识符 0x0-0xFF 被 IETF 工作组文档中分配时保留，并且 0XFFFF 用于实验用途。另外，标识符将被分给任何对创建 Geneve 选项感兴趣的机构，并且基于先来先得的原则。

8.致谢

作者想感谢 Martin Casado, Bruce Davie and Dave Thaler 这些人的奉献、反馈和有用的建议。

9.参考文献

9.1.标准化参考资料

[RFC0768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, August 1980.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

9.2.引用化参考资料

[ETYPES] The IEEE Registration Authority, "IEEE 802 Numbers", 2013, <http://www.iana.org/assignments/ieee-802-numbers/ieee-802-numbers.xml>.

[I-D.davie-stt]

Davie, B. and J. Gross, "A Stateless Transport Tunneling Protocol for Network Virtualization (STT)", draft-davie-stt-04 (work in progress), September 2013.

[I-D.ietf-nvo3-dataplane-requirements]

Bitar, N., Lasserre, M., Balus, F., Morin, T., Jin, L., and B. Khasnabish, "NVO3 Data Plane Requirements", draft-ietf-nvo3-dataplane-requirements-02 (work in progress), November 2013.

[I-D.mahalingam-dutt-dcops-vxlan]

Mahalingam, M., Dutt, D., Duda, K., Agarwal, P., Kreeger, L., Sridhar, T., Bursell, M., and C. Wright, "VXLAN: A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks", draft-mahalingam-dutt-dcops-vxlan-08 (work in progress), February 2014.

[I-D.sridharan-virtualization-nvgre]

Sridharan, M., Greenberg, A., Wang, Y., Garg, P., Venkataramiah, N., Duda, K., Ganga, I., Lin, G., Pearson, M., Thaler, P., and C. Tumuluri, "NVGRE: Network Virtualization using Generic Routing Encapsulation", draft-sridharan-virtualization-nvgre-04 (work in

progress), February 2014.

[IEEE.802.1Q-2011]

IEEE, "IEEE Standard for Local and metropolitan area networks -- Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks", IEEE Std 802.1Q, 2011.

[RFC2983] Black, D., "Differentiated Services and Tunnels", RFC 2983, October 2000.

[RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, January 2001.

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.

[RFC6040] Briscoe, B., "Tunnelling of Explicit Congestion Notification", RFC 6040, November 2010.

[RFC6935] Eubanks, M., Chimento, P., and M. Westerlund, "IPv6 and UDP Checksums for Tunneled Packets", RFC 6935, April 2013.

[VL2] Greenberg et al, , "VL2: A Scalable and Flexible Data Center Network", 2009.

Proc. ACM SIGCOMM 2009

作者信息

Jesse Gross

VMware, Inc.

3401 Hillview Ave.

Palo Alto, CA 94304

USA

Email: jgross@vmware.com

T. Sridhar

VMware, Inc.

3401 Hillview Ave.

Palo Alto, CA 94304

USA

Email: tsridhar@vmware.com

Pankaj Garg

Microsoft Corporation

1 Microsoft Way

Redmond, WA 98052

USA

Email: pankajg@microsoft.com

Chris Wright

Red Hat Inc.

1801 Varsity Drive

Raleigh, NC 27606

USA

Email: chrisw@redhat.com

Ilango Ganga

Intel Corporation

2200 Mission College Blvd.

Santa Clara, CA 95054

USA

Email: ilango.s.ganga@intel.com