



(12) 发明专利申请

(10) 申请公布号 CN 103595553 A

(43) 申请公布日 2014. 02. 19

(21) 申请号 201310314144. X

(22) 申请日 2013. 07. 24

(30) 优先权数据

13/557, 138 2012. 07. 24 US

(71) 申请人 埃克斯帝网络有限公司

地址 加拿大魁北克

(72) 发明人 安德斯·瓦尔曼

(74) 专利代理机构 北京信慧永光知识产权代理

有限责任公司 11290

代理人 曹正建 陈桂香

(51) Int. Cl.

H04L 12/24 (2006. 01)

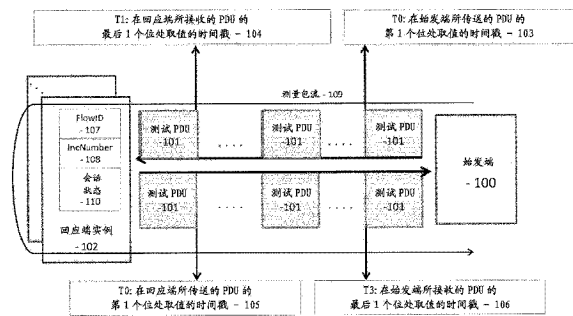
权利要求书2页 说明书10页 附图5页

(54) 发明名称

回应端实例的自动建立

(57) 摘要

在通信网络中的始发端和回应端之间建立服务操作管理维护 (OAM) 会话的方法包括通过所述回应端连续地监控由所述始发端传送的任何测试包。在由第一时间戳表示的时间处, 所述始发端向所述回应端传送测试包。在由第二时间戳表示的时间处, 所述回应端接收所述测试包, 并生成用于识别所述回应端的唯一流标识符。所述回应端将所述测试包传送给所述始发端。所述回应端向所述测试包分配序列号。在由第四时间戳表示的时间处, 所述始发端接收所述测试包。所述回应端至少在所述 OAM 会话的持续时间内存储所述唯一流标识符。



1. 一种在通信网络中的始发端和回应端之间建立服务操作管理维护 (OAM) 会话的方法,其包括:

通过所述回应端连续地监控由所述始发端传送的任何测试包;  
在由第一时间戳表示的时间处,通过所述始发端向所述回应端传送测试包;  
在由第二时间戳表示的时间处,通过所述回应端接收所述测试包;  
通过所述回应端生成用于识别所述回应端的唯一流标识符;  
通过所述回应端向由所述回应端传送的所述测试包分配序列号;  
在由第三时间戳表示的时间处,通过所述回应端将所述测试包传送回所述始发端;以

及

在由第四时间戳表示的时间处,通过所述始发端接收所述测试包,  
其中,所述回应端至少在所述 OAM 会话的持续时间内存储所述唯一流标识符。

2. 根据权利要求 1 所述的方法,其中通过所述回应端自动地定义所述流标识符;或者经由包括 TWAMP 控制面的信令交换或经由包括 CLI、基于网络的配置或 SML 的配置方法自动地定义所述流标识符。

3. 根据权利要求 1 所述的方法,其中通过包括 MAC 地址对各种方法、包括唯一 OAM 协议标识符的各种方法、使用附加 OAM 字段的各种方法、使用 IPv4 数据的各种方法或使用 IPv6 数据的各种方法来建立所述流标识符。

4. 根据权利要求 1 所述的方法,其中在所述始发端和所述回应端之间不存在建立互换、明确协商或初步配置步骤的情况下建立所述流标识符。

5. 根据权利要求 1 所述的方法,其还包括:

如果所述回应端在预定时间段内没有从所述始发端接收到后续测试包,通过所述回应端丢弃所述唯一流标识符。

6. 根据权利要求 1 所述的方法,其还包括:

如果所述回应端从所述始发端接收所述测试包时和接收后续包时之间的时延超过预定阈值,停止回应端实例,并重启用于保存先前所分配的唯一流标识符的新回应端实例。

7. 根据权利要求 1 所述的方法,其中所述回应端包括优先地与所述回应端相关联的标记号。

8. 根据权利要求 7 所述的方法,其还包括:

在重启所述回应端时使所述流标识符递增;  
针对所递增的流标识符,通过所述回应端分配第二标记号;  
通过所述回应端将所述第二标记号传送给所述始发端。

9. 根据权利要求 1 所述的方法,其中所述回应端包括在重启所述回应端之前与所述回应端相关联的第一标记号,且其中所述回应端还包括在重启所述回应端之后与所述回应端相关联的第二标记号,所述第一标记号不同于所述第二标记号。

10. 根据权利要求 1 所述的方法,其中所述始发端经配置用于检测所述回应端的重启。

11. 根据权利要求 10 所述的方法,其中所述始发端通过检查所述第一时间戳、所述第二时间戳、所述第三时间戳、所述第四时间戳和所述序列号来检测所述回应端的所述重启。

12. 根据权利要求 1 所述的方法,其还包括:

通过所述始发端测量所述第一时间戳和所述第二时间戳之间的间隔;以及

如果所述间隔超过预定阈值,通过所述始发端确定所述回应端已经被重启。

13. 一种在通信网络中的始发端和回应端之间建立服务操作管理维护(OAM)会话的方法,其包括:

通过所述回应端连续地监控由所述始发端传送的任何测试协议数据单元;  
通过所述回应端接收测试协议数据单元;  
通过所述回应端生成用于识别所述回应端的唯一标识符;  
通过所述回应端将所述测试协议数据单元传递到所述始发端;以及  
通过所述回应端向由所述回应端传递的所述测试协议数据单元分配序列号,  
其中,所述回应端至少在所述 OAM 会话的持续时间内存储所述唯一标识符。

14. 一种用于在通信网络中的始发端和回应端之间建立服务操作管理维护(OAM)会话的系统,其包括:

始发端,其经配置用于生成测试协议数据单元,并沿着测试路径将所述测试协议数据单元传送到回应端;以及

所述回应端,其经配置用于从所述始发端接收所述测试协议数据单元,并将所述测试协议数据单元传递给所述始发端,所述回应端经配置还用于生成用于识别所述回应端的唯一标识符,并至少在所述 OAM 会话的持续时间内存储所述唯一标识符,所述回应端经配置还用于向由所述回应端传送的所述测试协议数据单元分配序列号,

其中,所述始发端经配置用于从所述回应端接收所述测试协议数据单元。

## 回应端实例的自动建立

[0001] 相关申请的交叉参考

[0002] 2012 年 7 月 24 日提交的美国专利申请 No. 13/557, 102 的全部内容以引用的方式并入本文。

### 技术领域

[0003] 本发明涉及以太网 (Ethernet) 操作管理维护 (Operations Administration and Maintenance ;OAM) 构架的范畴下的测试会话的自动建立。

### 背景技术

[0004] 许多年来,以太网被用作 LAN(局域网)技术,且许多企业已经使用诸如简单网络管理协议 (SNMP)、ICMP 回应(或 IP 因特网包探索 (IPping))、IP 路由追踪 (IP Traceroute) 和思科单向链路检测协议 (Cisco Unidirectional Link Detection Protocol ;UDLD) 等因特网协议来管理这些网络。以太网操作管理维护 (Ethernet Operations Administration and Maintenance ;EOAM) 是用于建立、监视和检修城域网 (Metropolitan Area Network ;MAN) 以及广域网 (Wide Area network ;WAN) 的协议。由于目前存在较大且复杂的网络以及通过提供端到端服务 (end-to-end service) 而涉及不同操作者的广大用户群,因此以太网作为网络技术的使用就产生了对新的 OAM 协议组的需要。

[0005] 因特网工程工作小组 (Internet Engineering Task Force ;IETF) 开发并提升了网络标准。单向主动测量协议 [RFC4656] (One-way Active Measurement Protocol ;OWAMP) 提供了用于在网络装置之间测量单向量度的公共协议。OWAMP 可双向地用于在两个网络组件之间的两个方向上测量单向量度。然而,它并不适用往返或双向测量。

[0006] 双向主动测量协议 (Two-Way Active Measurement Protocol ;TWAMP) 提供了基于标准的用于测量两个装置之间的往返程 IP 性能 (丢包、时延和抖动) 的方法。TWAMP 使用单向主动测量协议 (One-way Active Measurement Protocol ;OWAMP) 的方法和结构来定义用于测量双向或往返量度的方式。

[0007] 在 TWAMP 中存在有四个逻辑实体:控制客户端 (Control-Client)、会话发起端 (Session-Sender)、服务器 (Server) 以及会话回应端 (Session-Reflector)。控制客户端和会话发起端通常实施在一个物理装置 (“客户端”) 中,而服务器和会话发起端实施在正在执行双向测量的第二个物理装置 (“服务器”) 中。

[0008] 控制客户端和服务器建立 TCP(传送控制协议)连接并在该连接上交换 TWAMP 控制消息。当控制客户端期望开始测试时,客户将测试参数传送到服务器。如果服务器同意进行所期望的测试,则在客户端发送开始会话消息时立刻开始测试。作为测试的一部分,会话发起端将基于用户数据协议 (User Datagram Protocol ;UDP) 的测试包的流发送到会话回应端,且会话回应端使用基于 UDP 的响应测试包来响应所接收的每个包。当会话发起端从会话回应端接收响应包时,将信息用于计算两个装置之间的双向时延、丢包和包时延变化。

[0009] 在 TWAMP 的简化架构 (TWAMP Light) 中,在一个被称为控制器的主机中实施控制

客户端、服务器和会话发起端的作用（部分 RFC5357），且在另一被称为应答器的主机中执行会话回应端的作用。

[0010] TWAMP Light 提供了应答器的简单架构，其中这些应答器的作用简单用作网络中的简单架构测试点。控制器通过非标准构件与服务器建立测试会话。在建立测试会话之后，控制器将测试包传送给应答器。除了下述动作之外，应答器都遵循 TWAMP 的会话回应端性能。

[0011] 在 TWAMP Light 的情况下，会话回应端不必了解会话状态。如果会话回应端不了解会话状态，则会话回应端必须将所接收包的序列号复制到回应包的序列号字段。控制器接收所回应的测试包并收集双向量度。该架构允许收集双向量度。

[0012] TWAMP Light 消除了对 TWAMP 控制协议的需要，且假设会话回应端受到配置且通过非标准构件在其与服务器通信之间传送它的配置。会话回应端简单地将进入的包回应到控制器，并同时复制所需信息以及生成序列码和时间戳值。TWAMP Light 引入了一些额外的安全考量。用于控制应答器并建立测试会话的非标准构件提供了下述特征。非标准应答器控制协议具有经认证的操作模式。应答器可配置成仅接收经认证的控制会话。非标准应答器控制协议具有经认证且加密的用于激活 TWAMP 测试协议的模式。当 TWAMP Light 测量会话在经认证或加密的模式下操作时，会话回应端必须具有一些用于生成密匙的机制（这是因为 TWAMP 控制协议在该过程中正常地发挥作用，但并不存在于此）。

## 发明内容

[0013] 根据本发明的一个方面，一种在通信网络中的始发端和回应端之间建立服务操作管理维护 (OAM) 会话的方法包括通过所述回应端连续监控由所述始发端传送的任何测试包。在由第一时间戳表示的时间处，所述始发端向所述回应端传送测试包。在由第二时间戳表示的时间处，通过所述回应端接收所述测试包，并生成用于识别所述回应端的唯一流标识符。所述回应端向由所述回应端传送的所述测试包分配序列号。在由第三时间戳表示的时间处，所述回应端将所述测试包传送给所述始发端。在由第四时间戳表示的时间处，所述始发端接收所述测试包。所述回应端至少在所述 OAM 会话的持续时间内存储所述唯一流标识符。

[0014] 根据本发明的另一方面，一种在通信网络中的始发端和回应端之间建立服务操作管理维护 (OAM) 会话的方法包括通过所述回应端连续监控由所述始发端传送的任何测试协议数据单元。所述回应端接收测试协议数据单元。所述回应端生成用于识别所述回应端的唯一流标识符。所述回应端将所述测试协议数据单元传递到所述始发端。所述回应端向由所述回应端传递的所述测试协议数据单元分配序列号。所述回应端至少在所述 OAM 会话的持续时间内存储所述唯一流标识符。

[0015] 根据本发明的又一方面，一种在通信网络中的始发端和回应端之间建立服务操作管理维护 (OAM) 会话的系统包括始发端和回应端。所述始发端经配置用于生成测试协议数据单元，并沿着测试路径将所述测试协议数据单元传送到回应端。所述回应端经配置用于从所述始发端接收所述测试协议数据单元，并将所述测试协议数据单元传递给所述始发端。所述回应端经配置还用于生成用于识别所述回应端的唯一流标识符，并至少在所述 OAM 会话的持续时间内存储所述唯一流标识符。所述回应端经配置还用于向由所述回应端传送

的所述测试协议数据单元分配序列号。所述始发端经配置用于从所述回应端接收所述测试协议数据单元。

#### 附图说明

[0016] 通过阅读下述详细说明并参考附图,本发明的上述和其它优点将变得更加明显。

[0017] 图 1 示出了在建立时为使因特网电路获取资格所需的测量;

[0018] 图 2 示出了以太网服务区域;

[0019] 图 3 示出了如何在用于支持以太网虚拟电路 (Ethernet Virtual Circuit;EVC) 的不同下层之间的边界处代表性地得到测量点 (Measurement Points;MP);

[0020] 图 4 示出了产生用于回应端实例 (reflector instance) 的双向环回测量 (2-way loopback measurement) 和组成的时间戳的位置;

[0021] 图 5 示出了流数据的格式;

[0022] 图 6 示出了由始发端 (originator) 生成的流数据的编码;

[0023] 图 7 示出了由回应端生成的流数据的格式和编码。

[0024] 虽然本发明可以做出各种修改或替代形式,但在附图中以示例的方式示出了具体实施例且在下文中详细说明了这些具体实施例。然而,应明白的是,本发明并不限于所公开的特定形式。反向地,本发明覆盖了所有落入由随附权利要求限定的发明范围和精神内的所有变化例、等同例和替代例。

#### 具体实施方式

[0025] 在下述说明中,为了说明而非限制的目的,提出了诸如具体实施例、过程、技术等的具体细节,以便提供对本发明的彻底理解。然而,对于本领域技术人员来说,很显然本发明还可应用于除这些具体细节之外的实施例中。

[0026] 下面的讨论是对可实施本文所述的方法和装置的合适计算机处理环境的简要和概括说明。在一个非限制性示例中,将在诸如程序模块等可在分布式计算环境中执行的处理器可执行指令的一般范畴下说明该方法和装置,上述分布式计算环境中的任务是通过经由一个或多个网络链路的远程或本地处理设备来执行的。本领域技术人员可以理解,可使用任何数目的适当计算机系统配置来实施该方法,而不是仅限于所说明的配置。

[0027] 以太网服务 OAM 架构是由诸如 ITU-T Y. 1731 和 IEEE802.1ag 等标准定义的。这样允许通过双向测量功能 (即 2xOneWay) 对 2 层交换网络的时延和丢包进行有效测量,且能够测量和报告两个方向的所有所需的度量 (时延、抖动、丢包、重新排序等),从而确定并报告网络的性能。

[0028] 抖动是在忽视了任何丢包的情况下从流中选择的包之间的端到端单向时延的差异。时延被定义为从在源处正被传送的包的开始到在目的地正被接收的包的结束。例如,如果每隔 20ms 传送包且如果第二个包在第一个包被接收之后 30ms 被接收,则抖动为 -10ms。如果第二个包在第一个包被接收之后 10ms 被接收,则抖动为 +10ms。

[0029] 这就需要在直接在 2 层网络上进行操作的各种性能测量或者当在 3 层网络基础结构 (IPv4 和 / 或 IPv6) 中提供以太网服务时进行操作的各种性能测量。

[0030] 国际电信同盟 (International Telecommunication Union;ITU) 是联合国的信息

和通信技术 (ICTs) 的专门机构。ITU 标准 (被称为“建议”) 是 ICT 网络操作的基本原则。ITU-T Y. 1731 性能监视提供了基于标准的以太网性能监视, 该以太网性能监视包括对以太网帧时延、帧时延变化和帧丢包的测量及传送量。

[0031] 以太网服务 OAM 架构定义了数种用于连接性验证的功能以及性能监视。本发明的一个目的在于增强由 ITU Y. 1731 定义的基础以太网服务 OAM 消息。例如, 以太网环回 (Ethernet Loopback ;ETH-LB) 和双向帧时延测量 (two-way Frame Delay Measurement ; ETH-DM) 是双向的, 即, 始发端节点向响应端节点发送一个或多个测量帧, 响应端节点 (在处理和交换 MAC 地址之后等) 将测量帧发送回始发端节点。由于 ETH-DM 和 ETH-LB 并不提供基于 Y. 1731 的下行链路 (反向路径) 方向的专用序列号, 因此这两种测量功能都缺乏监视单向丢包、重新排序和复制包的能力。

[0032] 这就是为什么需要专有 ETH-VSP 测量功能的一个原因。通过 2xOneWay 方法获得的测量非常类似于通过 TWAMP 获得的测量, 其不同之处在于, 2xOneWay 响应端记录所接收包中的 IP-ToS (服务类型)。

[0033] 由于没有针对这些测试的序列号的明确追踪, 所以 ETH-DM 和 ETH-LB 不需要任何响应端实例以根据规格进行操作, 但 ETH-VSP 和 ETH-SLM 都需要该实例。ETH-SLM 是 Y. 1731 中的新测量功能, 其需要专用序列号。

[0034] 对 Y. 1731 中的测量功能的概述 :

[0035] ETH-LB : 测量并报告双向时延、抖动、丢包、重新排序等。为了改善 ETH-LB 消息中的可用测量, 将类型 - 长度 - 值 (Type-Length-Value ; TLV) 数据加入到 ETH-LBM 的有效载荷 (payload) 中, 以保持传送时间戳 ( $t_0$ )。由于不需要“主动”动作, 所以对明确的响应端实例没有形式要求。所有这一切所要求的是对管理响应端实例的网络组件进行配置以启用该功能。ETH-DM : 测量并报告双向时延、抖动、丢包、重新排序等。它在响应端使用两个可选时间戳 ( $t_1$  和  $t_2$ ) 的情况下也能够测量和报告单向时延和抖动。将类型 - 长度 - 值 (Type-Length-Value ; TLV) 数据加入到 ETH-DMM 的有效载荷中, 以保持“上行链路”的序列号。ETH-SLM : 测量并报告双向时延、抖动、丢包、重新排序等。它也测量和报告单向丢包、重新排序等。由于响应端需要追踪序列号, 所以对 ETH-SLM 消息的支持需要必须配置至少一个 MEP 的构造。ETH-VSP : 测量并报告双向和单向的所有量度。ETH-VSP 不依赖于 MEP 或 MIP, 但它可能需要启用以太网 OAM 以启用 ETH-VSP 响应端, 这取决于具体的实施。总之, 2xOneWay 和 TWAMP 的组合在其涉及所测量和报告的量度 (即时延、抖动、丢包、重新排序等) 时提供了与 ETH-VSP 相同的功能。

[0036] 下述图表示出了用于定义在建立时使以太网电路具有资格所需的时延和其它测量的要求。根据 ITU Y. 1563 制成的图 1 示出了以太网服务性能的分层特性。该网络包括连接导向链路或无连接链路, 这些链路连接到在网络内部处理以太网层的桥接器。在以太网帧每次通过以太网层时, 它都经过整合处理并通过下层连接发送到下一桥接器。下层是基于诸如 SDH、OTN、PDH、MPLS、ATM 和 ETY 等多种技术的层。所有以太网层和下层的性能都影响用于分发服务的网络的端到端性能。

[0037] 高层可用于启用端到端通信。上层可包括类似 IP、MPLS 和以太网等允许网络部署的较大可扩展性的协议。类似 TCP 等其他协议提供了在帧丢包的情况下重新传送帧的能力。不幸的是, TCP 中的两个缺点在于传送用户信息时增加了时延, 以及可能限制了最大通

知窗口尺寸、与带宽时延产品的交互以及与以太网服务的丢包和时延的流控制交互。本发明的一个目的在于与如下事实无关地执行所要求的测量：用于承载以太网虚拟电路 (EVC) 的链路 (或下层 -LL) 可以在 2 层或 3 层处操作。

[0038] 根据 ITU Y. 1564 制成的图 2 提供了以太网服务区域的简单示例, 并作为以太网服务激活测量进行参照。该测试的目的在于验证基于以太网的服务的配置和性能。该测试验证了包括承诺信息速率 (Committed Information Rate ;CIR)、过量信息速率 (Excess Information Rate ;EIR) 和其它属性的以太网服务属性。这示出了网络的用于支持以太网服务实例的不同部分。

[0039] 还示出了 UNI 参考点出现在接入链路的中部的情况, 或更具体地, UNI 是如下参考点, 该参考点的功能被拆分为客户 (UNI-C) 组件和网络 (UNI-N) 组件。从服务供应者的观点来看, 他们需要将服务从 UNI-N 分发到 UNI-C, 且基于该观点而创建了本发明所述的测试方法。

[0040] 客户设备 (CE) 和操作者的网络通过 UNI 交换服务帧 (service frame)。服务帧是通过 UNI 向服务供应者传送的以太网帧 (被称为入口服务帧) 或通过 UNI 向 CE 传送的以太网帧 (被称为出口服务帧)。许多服务在双 UNI 上运行。它们由其属性来获取资格。这些属性包括但不限于: 连接类型; 业务参数; QoS (包括 VLAN 信息)、业务类型 (数据与管理的关系) 等; 带宽轮廓属性; 性能标准; FD、FDV、帧丢包率、可用性等。

[0041] 从而, 以太网虚拟电路可实际上能够以 2 层或 3 层跨越多个传送网络。这就对通过使用传统方法来有效地测量时延和丢包提出了挑战。

[0042] 根据 ITU-Ty. 1563 制成的图 3 示出了如何在用于支持 EVC 的不同下层之间的边界处代表性地获得测量点 (Measurement Point ;MP)。这些 MP (或回应端) 的位置对于获得充分测量来说非常重要。对于上述每个图, 双向时延测量通常是从源 (SRC) 到目的地 (DST) 并返回 SRC 来获得的, 且并不针对每个交换链路 (或下层) 提供详细的时延信息。

[0043] 回应端实例是非管理型或管理型回应端实例。非管理型回应端实例从回应端实例池中分配, 且在新的测量流到达时在运行期间被激活。产生唯一的流标识符 (FlowId) 以用于具体测量流, 且接着该标识符用于将测量流与所分配的回应端实例相关联。如果在超时时钟的持续时间内没有从相关联的测量流接收到消息, 则回应端实例被去激活, 停止与测量流的关联, 并被返还到回应端实例池中。

[0044] 管理型回应端实例通过控制接口被配置成与由唯一的流标识符 (FlowId) 识别的具体测量流相关联。在此情形中, 针对具体测量流预先分配回应端实例, 且在运行时不需要资源分配。通过控制接口移除回应端实例的配置, 从而使管理型回应端实例去激活并被返还到回应端实例池中。

[0045] 根据本发明的一个方面, 自动建立非管理型回应端, 因此消除 (或至少明显地降低) 了与各回应端的配置数据的定义和持续管理相关联的开销, 并消除了始发端和回应端之间对明确的 OAM 会话建立协议的需要。该自动建立不仅在以太网电路的起始激活期间是令人满意的, 而且在以太网电路被宣布是可操作的寿命期间的任何时间上也是令人满意的。

[0046] 结合服务数据单元 (service data unit ;SDU) 可以更好地理解协议数据单元 (protocol data unit ;PDU)。在不同的“层”中实施网络的特征或服务。例如, 通过物理层



在线路、光纤等上发送 1 或 0。通过数据链路层将 1 和 0 组织到数据块中并在线路上使它们安全地到达的正确位置处。通过网络层在多个相连的网络上传递数据块。通过传送层将数据分发到目的地处的正确软件应用程序。在各层之间（和应用程序与最顶层之间），这些层通过接口来传递服务数据单元。应用程序或高层了解 SDU 中的数据的结构，但接口处的下层并不了解；其将 SDU 中的数据作为有效负荷，并负责使有效负荷到达目的地处的相同接口。为实现该目的，协议层向 SDU 中增加它需要的某些数据以执行其功能。例如，高层可能增加端口号以识别应用程序，增加网络地址以帮助路由选择和 / 或增加代码以识别包中的数据类型以及检错信息。

[0047] 所有这些附加信息加上来自高层的原始服务数据单元构成了这一层处的协议数据。其意义在于，PDU 是结构化信息，其被沿着数据的行程传递到匹配协议层，其中数据的行程允许该层分发它所期望的功能或服务。匹配层或“同等级层 (peer)”对数据进行解码以抽取原始服务数据单元，并决定该数据是否没有错误以及接下来将该数据发送到哪个位置等。除非该数据已经达到最下层（物理层），否则通过使用协议“堆栈”中的下一下层的层将 PDU 传递到同等级层。当 PDU 通过接口从构建了 PDU 的层传递到仅传递 PDU（且因此不了解其内部结构）的层时，PDU 变成那一层的服务数据单元。在到达最下层之前，反复执行将寻址和控制信息增加到 SDU 以形成 PDU 以及将该 PDU 传递到下一下层的层以作为 SDU 的动作，且通过一些媒体传递数据以作为物理信号。

[0048] PDU 与 OSI 模型的前 4 个层如下地关联：

[0049] 1. 1 层（物理层）PDU 是位。

[0050] 2. 2 层（数据链路层）PDU 是帧。

[0051] 3. 3 层（网络层）PDU 是包。

[0052] 4. 4 层（传送层）PDU 是用于 TCP 连接的段或用于 UDP 连接的数据报。

[0053] 在建立和激活以太网电路时，需要能够获得精确的性能测量，以保证以太网虚拟连接 (Ethernet Virtual Connection ;EVC) 具有完全的与操作者的性能说明相一致的功能。

[0054] 另外非常期望的是，能够在 (1) 以太网电路的寿命期间（一天 24 小时，一周 7 天）执行类似的性能测量以确保正在进行的操作质量以及遵守任何服务水平协议 (Service Level Agreement ;SLA) 以及其他操作参数，以及 (2) 以太网电路处于服务中时执行类似的性能测量。

[0055] 根据本发明的一个方面，双向时延测量的起始端被称为始发端 100，回应端实例 102 响应于双向时延测量请求。

[0056] 图 4 是双向环回法测量的图。始发端 100 和回应端实例 102 之间的上述服务 OAM 会话的建立通常在进行测量测试之前要求广泛的配置和 / 或始发端 100 与回应端实例 102 之间的建立交换 (setup exchange) 以协商（每个会话的）唯一流标识符。本发明的一个方面涉及唯一 FlowID107 的自动建立，而不需要明确协商或初步配置步骤。

[0057] 通过始发端 100 向回应端实例 102 发送测试 PDU (Test PDU) 101，以开始双向测量测试。本发明的一个方面涉及能够在具有测量和报告两个方向的量度的能力的任何 2 层或 3 层网络上操作的双向方案。如果测量协议包含用于反向路径方向的专用序列号，则需要回应端实例 102。回应端实例 102 的主要目的在于向唯一的相关测试包流 109 提供专用序列

号计数器。该序列号计数器用于向“回应的”包分配序列号。通过相关的测量包流 109 的 FlowID107 来唯一地识别测量包流 109,且要求该 FlowID107 由用于服务特定测量包流 109 的响应端实例 102 来存储。

[0058] 无论何时在始发端 100 和响应端实例 102 之间建立服务 OAM 会话,都要求能够保存会话的相关状态信息的能力;FlowID107 就是用于此目的的。响应端实例 102 执行功能以提供用于服务测量包流 109 的状态实体或会话状态 (session state)110。该会话状态 110 针对与其相关的测量包流 109 保持唯一序列号计数器、终止计时器等。

[0059] 在接收到 Test PDU101 时,响应端实例 102 产生用于识别响应端实例 102 的唯一 FlowID107。FlowID107 可以具有任何大小,且被自动地定义成与服务 OAM 会话是否在 2 层或 3 层网络上操作无关。

[0060] 该 FlowID107 可通过不同方法进行设定,这些方法包括但不限于通过信令交换 (signaling exchange) (例如使用 TWAMP 控制面) 自动设定;通过配置方法 (CLI、基于 Web 的配置、XML 等) 或通过响应端 102 自动设定。

[0061] 为了消除配置过程的复杂度和服务 OAM 会话握手的复杂度,FlowID107 能够通过响应端实例 102 自动确定,而不需要与始发端 100 进行明确协商,如下:

[0062] a) 当始发端 100 和响应端 102 之间的通信出现在 2 层网络上时,使用一对 MAC 地址生成 / 创建唯一的 FlowID107,其中 MAC 源地址属于始发端 100:

[0063]  $\text{FlowID} = \text{Mac 源地址} + \text{StreamLabel} [+ \text{Mac 目的地地址}]$

[0064] b) 当 OAM 协议支持唯一标识符的概念时,其可用于得到唯一 FlowID107。例如,ITU Y. 1731 标准定义了包括 4 个八位组的测试标识符 (TestID) 的以太网综合丢包测量 (Ethernet Synthetic Loss Measurement;ETH-SLM)PDU。当存在此标识符时,FlowID107 如下所示:

[0065]  $\text{FlowID} = \text{Mac 源地址} + \text{TestID} + [\text{Mac 目的地地址} + \text{SrcMepId}]$

[0066] c) 或者,如果正在使用 VLAN 接口,FlowID107 可用于包括诸如 MAC 目的地地址和 VLAN[6] 标识符 (VlanID) 等附加字段,如下所述:

[0067]  $\text{FlowID} = \text{Mac 源地址} + \text{StreamLabel} + \text{Mac 目的地地址} + \text{VlanID}$

[0068] d) 在 3 层 IPv4 网络上操作时可如下地获得唯一 FlowID107:

[0069]  $\text{FlowID} = \text{IPv4 源地址} + \text{源 UDP 端口} + \text{StreamLabel}$  或

[0070]  $\text{FlowID} = \text{IPv4 源地址} + \text{源 UDP 端口} + [\text{IPv4 目的地地址} + \text{目的地 UDP 端口}] + \text{StreamLabel}$

[0071] e) 在 3 层 IPv6 网络上操作时可如下地获得唯一 FlowID107:

[0072]  $\text{FlowID} = \text{IPv6 源地址} + \text{源 UDP 端口} + \text{StreamLabel}$  或

[0073]  $\text{FlowID} = \text{IPv6 源地址} + \text{源 UDP 端口} + [\text{IPv6 目的地地址} + \text{目的地 UDP 端口}] + \text{StreamLabel}$

[0074] 上述 FlowID107 具有可变长度。一种实施可以在 FlowID107 的末端处增加位填充 (bit-padding),以获得具体数量的位 (例如总共 64 或 128 位),而与使用哪些组件来建立该实施无关。

[0075] 始发端 100 需要持续追踪针对特定接口分配的流标签 (如果使用的话)。

[0076] ITU Y. 1731 协议可经由被称为 (ETH-VSP) 的厂商特定 OAM-PDU 的概念而扩展。将

类型 - 长度 - 值 (Type-Length-Value ;TLV) 流数据添加到 ETH-VSP 编码中,以在所示实施例中支持双向测量测试。时间戳 (T0-T3) 使用 IEEE1588-2004 定义的时间戳格式。ETH-VSP 的 TLV (Type-Length-Value ;TLV) 流数据被分配了 MType = 209。

[0077] 当始发端 100 传送 Test PDU101 时,时间戳 T0 在 Test PDU101 的开始处取值。当回应端实例 102 接收 Test PDU101 时,时间戳 T1 在 TestPDU101 的末端处取值。时间戳 T2 由回应端实例 102 在被传送回始发端 100 的 Test PDU101 的开始处取值。当始发端 100 接收回应端实例 102 传送的 Test PDU101 的末端时,时间戳 T 由自始发端 100 取值。

[0078] 图 5 示出了始发端 100 和回应端实例 102 所使用的类型 - 长度 - 值 (Type-Length-Value ;TLV) 流数据的格式。

[0079] 图 6 示出了始发端所生成的类型 - 长度 - 值 (Type-Length-Value ;TLV) 流数据编码。

[0080] 图 7 示出了回应端实例 102 在响应于从始发端 100 接收的 Test PDU101 时使用的类型 - 长度 - 值 (Type-Length-Value ;TLV) 流数据编码。由于自动建立回应端实例 102,所以在预定时间段 (通常大约为 10 秒) 内从始发端 100 (由唯一 FlowID107 识别) 没有接收 Test PDU 的任何时候,都实施超时机构以拆下回应端实例 102 和不相关源。如果在空闲计时器终止后应当从相同始发端 100 接收新的 Test PDU101,回应端实例 102 自动生成新的唯一 FlowID107。因此,需要互补标签向始发端 100 表明已经自动建立新的回应端实例 102。这是通过由回应端实例 102 返回到始发端 100 的类型 - 长度 - 值 (Type-Length-Value ;TLV) 流数据中的标记号 (Incarnation Number ;IncNum) 400 而实现的。

[0081] 根据本发明的一个方面,能够检测回应端实例 102 是否已经被重启,且因此能够检测服务 OAM 会话有关的重要状态信息是否可能已经被丢包。

[0082] 一种用于检测此重启条件的方法是向每个回应端实例 102 分配唯一的标记号。该标记号 (IncNum400) 是从全局标记计数器生成的无符号整数值,该全局标记计数器在每次分配一个新标记号时增加 1:

[0083]  $IncNum = IncNumCnt;$

[0084]  $IncNumCount = IncNumCnt+1。$

[0085] 上述方案被实施为自动操作以确保在将新的 IncNum400 分配给回应端实例 102 的同时增加标记号计数器。

[0086] 全局标记号计数器在系统启动处立即被初始化为任意值,以使在系统重启后再次使用回应端实例 102 的同一 IncNum400 的可能性最小化。全局标记计数器的环绕率 (wrap-around rate) 取决于新测量流的到达率和已经运行的测量流的重新激活率。

[0087] 根据本发明的另一个方面,回应端实例 102 将新生成的 IncNum400 与旧的或之前的 IncNum400 进行比较,如果值相同,则回应端实例 102 生成新 IncNum400 以避免向用于服务一个和相同测量包流 109 的回应端实例 102 分配相同的 IncNum400。

[0088] 分配 IncNum400 的另一方法依赖于始发端 100,以通过分析由用于 OAM 会话的回应端实例 102 返回的下一 Test PDU101 (包括但不限于本质上包括相同信息的 ETH-VSP、EYH-SLM、TWAMP 和其它 PDU) 来检测回应端实例 102 的重启。OAM PDU 是包含用于监视器、测试和检修链路的 OAM 控制和状态信息的 IEEE802.3 慢速协议帧。始发端 100 检测回应端实例 102 的重启的方式是查看所接收包 (ETH-VSP) 的时间戳 (T0-T3) 和序列号 (401)。可

在出现较长的丢包时段时且下一 ETH-VSP 应答的“下行链路”序列号 401 远低于预期序列号时,能够确定回应端实例 102 的重启,且由包数据率划分的(“上行链路”序列号 300 中的)间隔足够大,以拒绝简单的包重新排序条件。

[0089] 根据本发明的另一方面,始发端 100 通过检查返回的 ETH-VSP 响应中的时间戳 T0 和 T1 来检测回应端实例 102 的重启。即使在考虑对测试包的可能重新排序以及这些包的传送速率的情况下,时间戳 T0 和 T1 之间的非常大的间隔可表示回应端实例 102 的重启或回应端实例 102 的时钟的调整/重置。

[0090] 另一方法是在处于有效无序 ETH-VSP 响应包的范围外的长丢包时段之后(通过始发端 100)接收具有无序下行链路序列号 400 的 ETH-VSP 响应。

[0091] 在回应端实例 102 在开始服务 OAM 会话时重启的情况下(其中重启发生足够快以在触发长丢包超时之前出现),始发端 100 通过使用比刚被接收的 ETH-VSP 早的(旧)时间戳 T0 检测到 ETH-VSP 响应中的下行链路序列号 400 已经在之前被使用,由此检测重启条件。该方法要求始发端 100 持续追踪先前针对最后“n”个测试包发布的测试包的时间戳 T0 和下行链路序列号 400,该最后“n”个测试包适配在测试包的重新排序的可接受范围内。重新排序的可接受范围的范围“n”可被定义为 SLA 轮廓的部分或通过其他配置组件来定义。

[0092] 上述实施例可应用于基于软件的、基于 HW 的和可插入(FPGA)的始发端和回应端。

[0093] 本发明包括具有如下处理器的系统,这些处理器用于提供处理信息和基于输入确定结果的各种功能。通常,可通过硬件和软件组件的组合来实现该处理。硬件方面可包括操作地耦合的硬件部件的组合,这些硬件部件包括微处理器、逻辑电路、通信/网络端口、数据滤波器、存储器或逻辑电路。这些处理器可适于执行由计算机可执行代码规定的操作,该计算机可执行代码可存储在计算机可读媒体中。

[0094] 本文中的上述方法的步骤可通过适当的可编程处理装置来实现,可编程处理装置例如为用于执行软件或存储指令的外部通用计算机或板上现场可编程门阵列(FPGA)或数字信号处理器(DSP)等。通常,如计算机和软件领域的技术人员所了解,本发明实施例所使用的用于处理或估计的物理处理器和/或机器可包括一个或多个根据本发明示例性实施例的启示而进行编程的基于网络或不基于网络的通用计算机系统、微处理器、板上现场可编程门阵列(FPGA)、数字信号处理器(DSP)、微控制器等。如软件领域的技术人员所了解的,编程人员可基于示例性实施例的启示容易地准备适当的软件。另外,如电子领域的技术人员所了解的,可通过准备特定应用集成电路或通过互连常规组件电路的适当网络来实施示例性实施例的装置和子系统。因此,示例性实施例并不限于硬件电路和/或软件的任何特定组合。

[0095] 本发明的存储在计算机可读媒体中的任何一个或其组合中的示例性实施例可包括用于控制示例性实施例的装置和子系统、用于驱动示例性实施例的装置和子系统、用于处理数据和信号、用于启动示例性实施例的装置和子系统以及与人类用户交互等的软件。此软件可包括但不限于装置驱动器、固件、操作系统、开发工具、应用软件等。此计算机可读媒体还可包括本发明实施例的用于执行在实施方法中所执行的全部或部分处理的计算机编程产品。本发明示例性实施例中的计算机代码装置可包括任何适当的可翻译或可执行的代码机制,包括但不限于脚本、可翻译程序、动态链路库(DLL)、Java 类和程序类型、完整的可执行程序等。另外,本发明实施例中的部分处理有助于更好的性能、可靠性、成本等。

[0096] 计算机可读媒体的通常形式例如包括软盘、软磁盘、硬盘、磁带、任何其他适当磁性媒体、CD-ROM、CDRW、DVD、任何适当的光学媒体、穿孔卡片、纸带、光学标记表、任何其他具有孔状或其他可光学识别适当标记的物理媒体、RAM、PROM、EPROM、FLASH-EPROM、任何其他适当的存储芯片或载体、载波或计算机可读的任何其他适当媒体。

[0097] 虽然图示和说明了本发明的特定实施和应用,但应当理解本发明不限于本文披露的精确构造和组成,且在不偏离所附权利要求所限定的发明范围的情况下可以从前面的说明中得到各种变形、修改和变化。

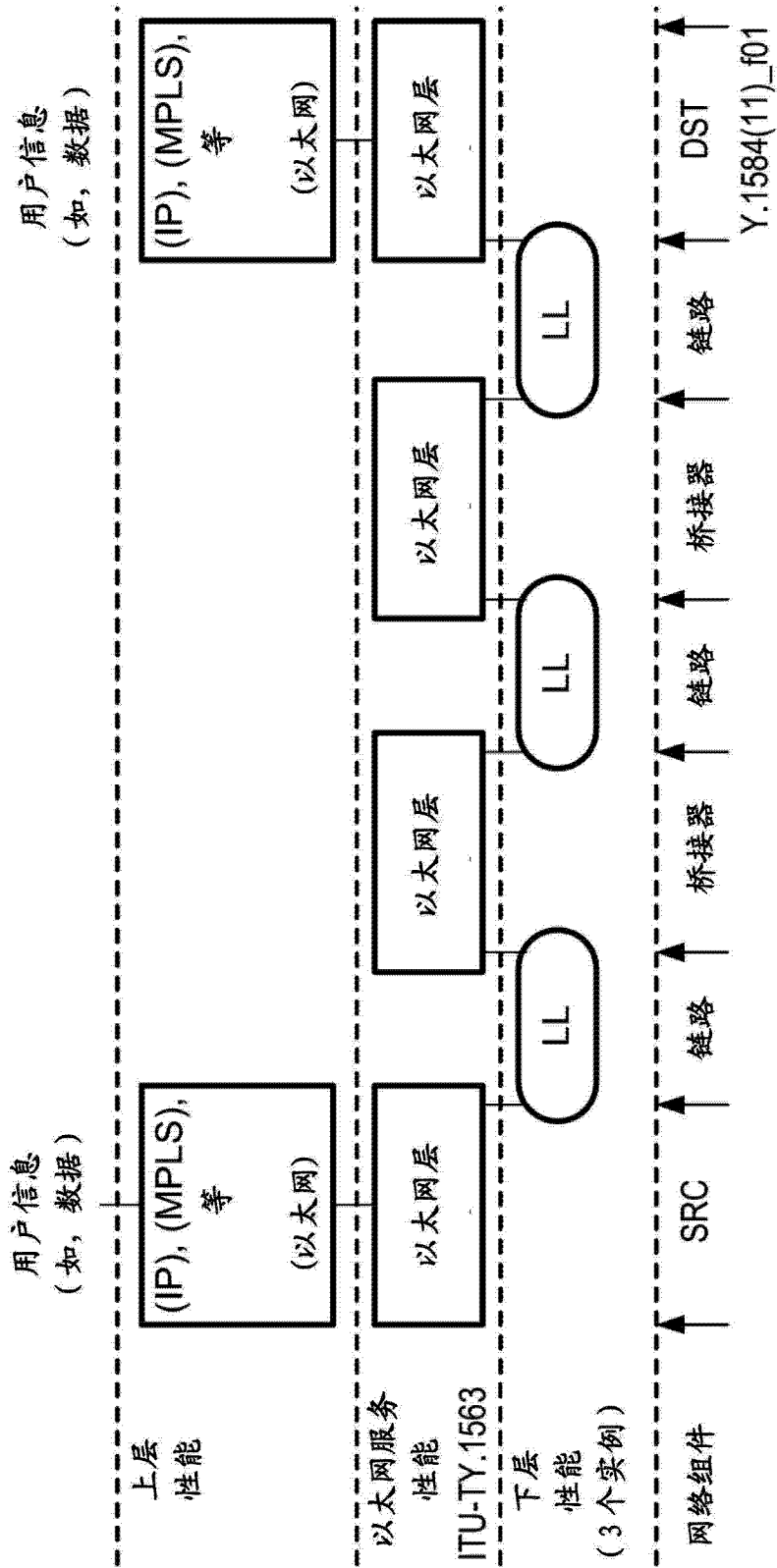


图1(现有技术)

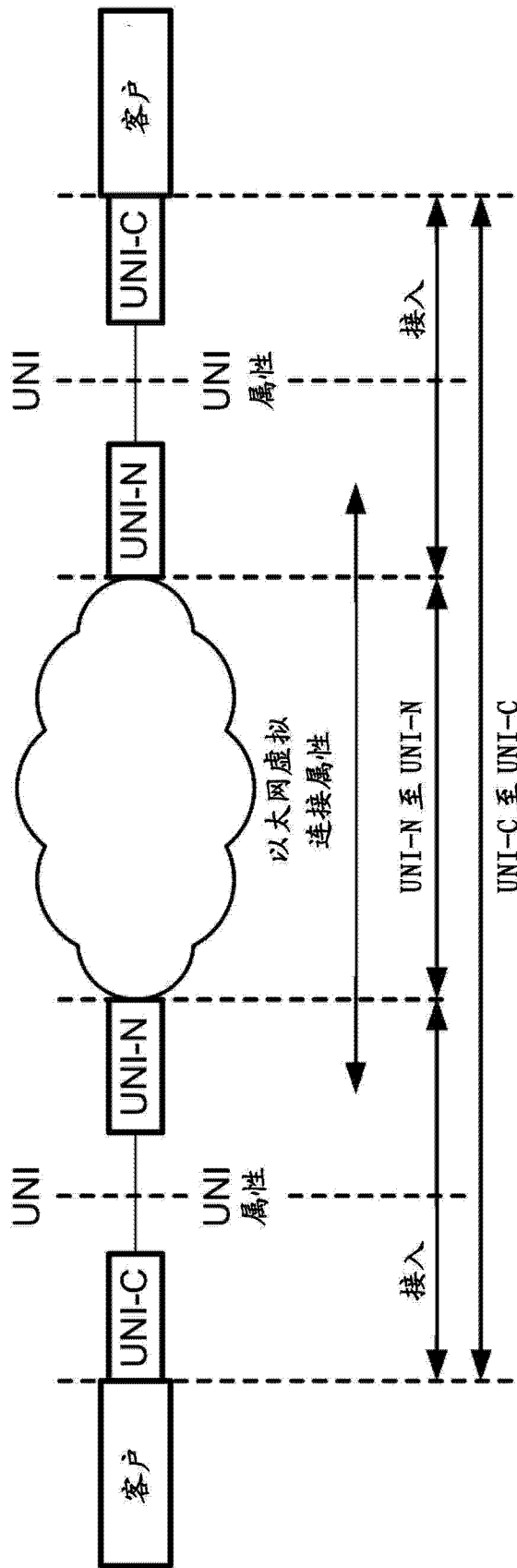


图 2(现有技术)

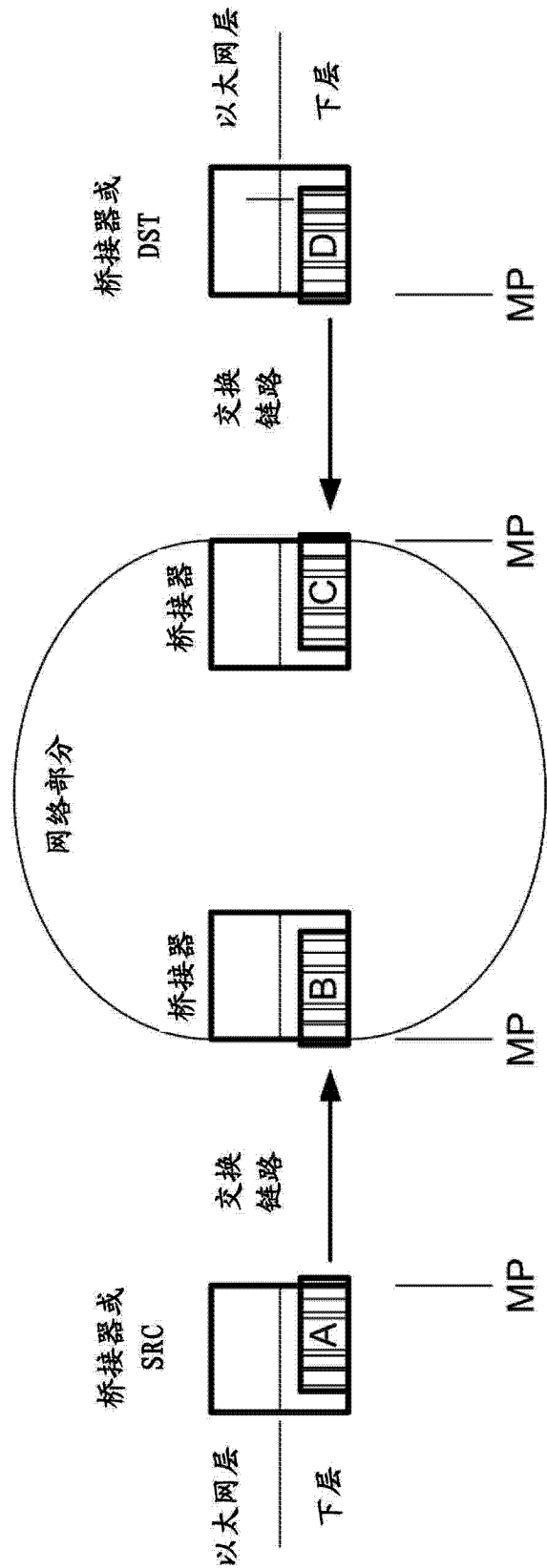


图 3(现有技术)

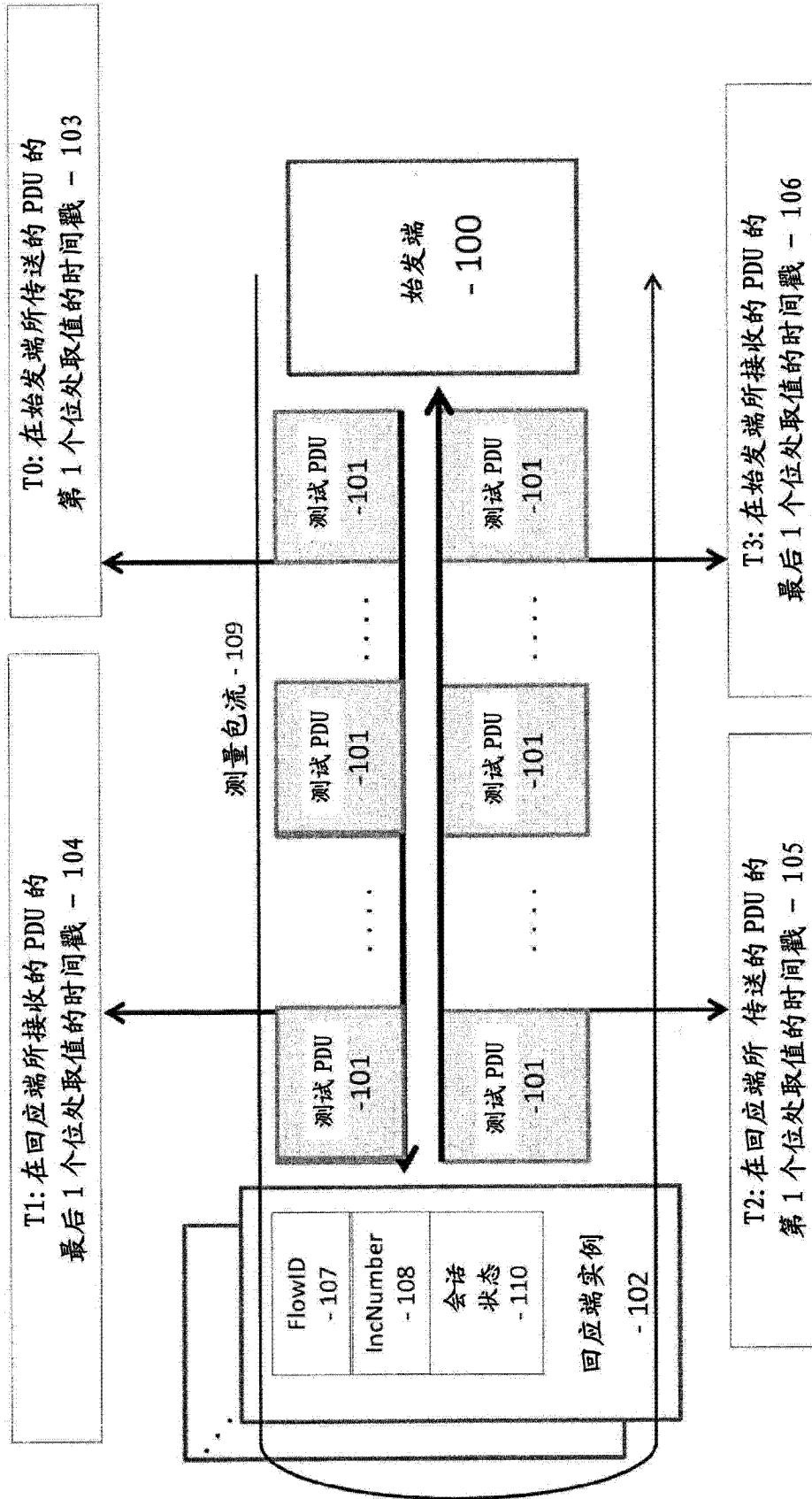


图 4



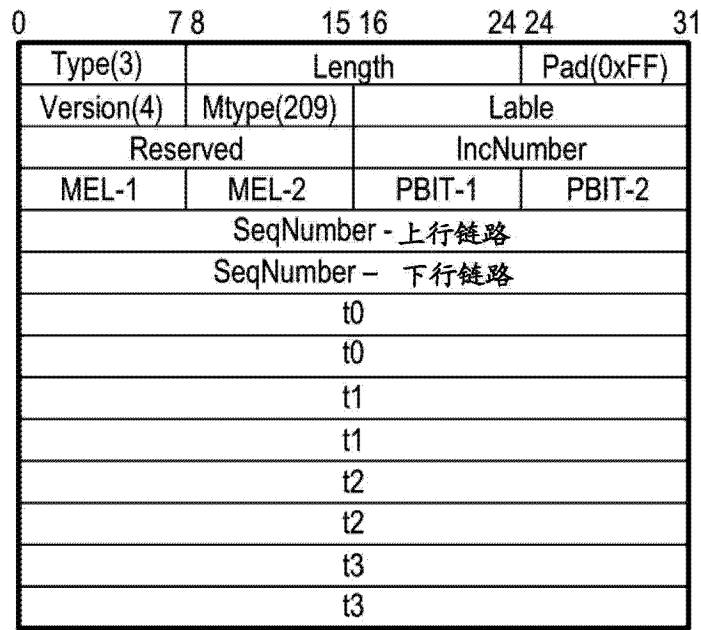


图 5

| 字段       | 大小 (bit) | 说明  |
|----------|----------|---|
| Type     | 8        | 总被设定为 3, 即 TLV 的类型 (3=数据)   |
| Length   | 16       | 数据 TLV 数据中的数据字段的长度  |
| Pad      | 8        | 总被设定为 ff16。用于对齐流的标头   |
| Version  | 8        | 流的版本, 当前版本是 4 (流的标头的第一字节)   |
| Mtype    | 8        | 流信息类型, 被设定为 209   |
| Lable    | 16       | MAC 地址对 (src/dst) 之间的唯一标签   |
| Reserved | 16       | 保留以将来使用。必须被设定为 0  |
| SeqNumUL | 32       | 用于上行链路/前向路径(从始发端 100 至 回应端 102)的序列号码                                  |
| t0       | 64       | 始发端 100 传送时间戳。<br>在传送时由始发端 100 设定                                     |
| t3       | 64       | 始发端 100 接收时间戳。始发端 100 在传送时将该时间戳设定为 0。在从回应端 102 接收到响应时始发端 100 再次设定该时间戳 |

图 6

| 字段       | 大小<br>(bit) | 说明   |
|----------|-------------|--|
| IncNum   | 16          | 前身编号。在建立回应端实例时被分配给回应端 102，并在回应时被发送回始发端 100。始发端 100 在传送时须将 IncNum 设定为 0。                        |
| SeqNumDL | 32          | 用于下行链路/反向路径方向(回应端 102 至始发端 100)的序列编号。始发端 100 在传送时须将 IncNum 设定为 0。                              |
| MEL-1    | 8           | 回应端 102 接收的 MEG 等级(MEL)。须由回应端 102 设定。始发端 100 在传送时须将 MEL 1 设定为 ff16。                            |
| MEL-2    | 8           | 回应端 102 传送的 MEG 等级(MEL)。须由回应端 102 设定。始发端 100 在传送时须将 MEL 2 设定为 ff16。                            |
| PBIT-1   | 8           | 回应端 102 接收的 VLAN PBIT 设定。如果接收接口是 VLAN 接口，仅是有效的且由回应端 102 设定。始发端 100 在传送时须将 PBIT 1 设定为 ff16。     |
| PBIT-2   | 8           | 回应端 102 传送的 VLAN PBIT 设定。如果到达(接收)接口是 VLAN 接口，仅是有效的且由回应端 102 设定。始发端 100 在传送时须将 PBIT 2 设定为 ff16。 |
| t1       | 64          | 回应端 102 接收时间戳。始发端 100 在传送时将该时间戳设定为 0。格式：秒数:ns  |
| t2       | 64          | 回应端 102 传送时间戳。始发端 100 在传送时将该时间戳设定为 0。格式：秒数:ns  |

图 7