

[19] 中华人民共和国国家知识产权局

[51] Int. Cl<sup>7</sup>

G06F 13/14



# [12] 发明专利说明书

G06F 13/376 H04L 12/28

H04L 12/407 H04L 12/66

H04L 12/56

[21] ZL 专利号 97192716.2

[43] 授权公告日 2003 年 4 月 2 日

[11] 授权公告号 CN 1104687C

[22] 申请日 1997.1.30 [21] 申请号 97192716.2

[30] 优先权

[32] 1996. 1. 31 [33] US [31] 08/597,520

[32] 1996. 11. 22 [33] US [31] 60/024,272

[86] 国际申请 PCT/US97/01595 1997. 1. 30

[87] 国际公布 WO97/28505 英 1997. 8. 7

[85] 进入国家阶段日期 1998. 8. 31

[71] 专利权人 伊普思龙网络公司

地址 美国加利福尼亚州

[72] 发明人 托马斯·里欧恩 彼得·纽曼

格里·敏夏尔 罗伯特·海登

峰青·廖 艾里克·霍夫曼

劳伦斯·B·胡思顿

威廉姆·A·罗伯森

审查员 朱世菡

[74] 专利代理机构 中国国际贸易促进委员会专利

商标事务所

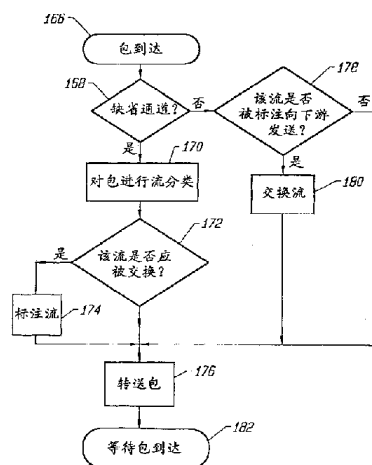
代理人 王以平

权利要求书 1 页 说明书 89 页 附图 50 页

[54] 发明名称 传输网络中在包路由选择和包交换之间动态转换的改进方法及设备

[57] 摘要

一种在交换及包路由选择之间高效地动态转换，以提供高通过量的方法及设备。本发明提供一种使用流分类及标记来重定向流，在网络中的上游节点及下游节点之间传输包的方法。该方法包括如下步骤：在上游节点和下游节点之间建立缺省虚拟通道(1602)，在下游节点接收包，在下游节点对该包进行流分类，以确定该包是否属于应在上游节点重定向的指定流(170)，在下游节点选择一个自由标记(1668)并通知上游节点属于该指定流的未来包应在已附加选定自由标记的情况下被发送(1676)。本发明的其它实施例包括在网络中传输包的系统中使用的基本交换单元(1、3)，交换网关单元(21)及交换代理器。



ISSN 1008-4274

1.一种在第一节点交换流的方法,所述第一节点具有指向第二节点的下  
游链路,及指向第三节点的上游链路,所述方法包括如下步骤:

在所述第一节点对一个第一包进行流分类,以确定所述第一包是否属  
于应在第三节点中被重定向的指定流;

在所述第一节点选择第一自由标记;

通知所述第三节点属于所述指定流的未来包应在附加所述选定的第一  
自由标记的情况下被发送;

在所述第二节点对一个第二包进行流分类,以确定所述第二包是否属  
于应在第三节点中被重定向的所述指定流;

在所述第二节点选择第二自由标记;

通知所述第一节点属于所述指定流的未来包应在附加所述选定的第二  
自由标记的情况下被发送;及

其中来自所述上游链路的所述指定流可在第2层由所述第一节点交换  
给所述下游链路。

2.按照权利要求1所述的方法,其中所述第二包是所述的第一包。

3.按照权利要求2所述的方法,其中所述第一、第二及第三节点使用  
ATM。

4.按照权利要求3所述的方法,其中所述第一及第二自由标记包括  
VPI/VCI。

5.按照权利要求2所述的方法,其中所述通知步骤由使所述节点之间的  
通信成为可能的IFMP软件执行。

## 传输网络中在包路由选择和包交换 之间动态转换的改进方法及设备

本发明涉及通信领域。更具体地说，在一个实施例中，本发明提供了一种用于在交换及包路由选择之间有效实现动态转换的方法及设备，从而在保持完整的网间协议（IP）路由选择功能的同时，提供高的包通过量。本发明把高速、高容量、多任务通信能力和简单性、可缩放性及坚固性结合在一起。

由于利用IP的英特网的普及及持续不断地发展，IP已经发展为当前使用的主要网络层协议。IP规定协议数据单元（PDU）格式及站-路由器及路由器-路由器的交互作用。IP为和英特网网络连接的站上的IP用户提供无连接数据传输服务。IP建立于其上的无连接模型提供了一个坚固且灵活的基础，在该基础上构造一个综合服务通信网络。所有主要的操作系统都包括IP工具，使IP及其伴随传输层（OSI参考模型的第4层）协议，和传输控制协议（TCP）实际上能够跨越所有的硬件平台得到应用。IP的主要优点之一是其惊人的可缩放性，它可在只具有几个用户的网络到企业级网络甚至全球性的英特网中成功运行。

随着英特网的快速发展，常规的IP路由器在英特网上处理通信的能力变得不够了。在今天的快速工作站，客户机-服务器计算及高带宽需求应用的情况下，网络日益受到通信拥塞问题的困扰。典型的问题包括网络响应时间变化无常，网络故障率高，及不支持延迟敏感型的应用。

局域网（LAN）交换器可提供快速、相对便宜的方法减轻共享介质LAN区段上的通信拥塞。交换技术是比共享介质集线器（hub）或简单的桥接器更为有效的在LAN中管理通信及分配带宽的方法。LAN交换器起数据链路层（OSI参考模型的第2层）包转送硬件引擎的作用，处理介质存取控制（MAC）地址，及执行简单的表查找功能。基于交换的网络能够提供更大的通过量，但是仍然存在广播淹没（broadcast flooding）及安全性底之类的问题。仍然需要在网络层（OSI参考模型的第3层）上运行

的路由器来解决这类问题。但是快速交换技术的能力超过了现有路由器的能力，从而造成路由器阻塞。英特网基于其上的常规 IP 包转送设备，IP 路由器的能力已不够了。路由器和正在形成的交换技术相比，昂贵、复杂，并且通过量受到限制。为了支持大的全企业网络及英特网增长的通信需求，IP 路由器需要运作更快，并且费用较低。

此外，为了支持对实时及多媒体应用（例如会议）日益增长的需求，需要服务质量（QOS）选择。目前 TCP/IP 不支持 QOS 选择。但是，随着更多类型的通信所要求的先进功能在 IP 中得以实现，常规的 IP 路由器已不足以作为包转送设备。

异步传输模式（ATM）是一种作为未来无路由器网络的基础的高速、可缩放的多业务技术。ATM 是一种通过量很高，可缩放，并且支持多种类型通信，包括音频、视频及数据的高效包转送技术。但是，ATM 是一种明显不同于当前诸如 IP 之类连网结构的连网技术，从而对 ATM 不存在明显的迁移途径。由于 ATM 的面向连接的结构，ATM 难以有效地支持现有的局域网通信，从而需要另外的一组非常复杂，并且未经测试的多层协议。这些协议带来的问题是交换虚拟电路（SVC）连接建立时间太长，难以接受。另外，使 TCP/IP 用户能够利用 SVC 发送或接收 ATM 通信需要采用更新的、未经证实的、非常复杂的协议。这些协议不能使在 TCP/IP 协议上运行的应用利用 ATM 的 QOS 特征，从而对网络管理者带来大量额外开销，而不能实现 ATM 的任一主要优点。此外，这些协议大多数重复了有良好基础的 TCP/IP 协议组的功能，必须解决网络中的问题的网络管理者需要学习这些复杂的协议，从而增加了拥有 ATM 设备的费用。鉴于经过时间考验并已调试的 IP 具有庞大的，且日益增长的安装用户基础（这一点由英特网的普及得到证实），因此转移到 ATM 是很困难的。

为了克服对上述问题的现有解决方案的不足，厂家已研究出了许多新的分布式路由选择结构。但是这些结构通常都较复杂、混乱，并重复了 IP 提供的功能。这些结构对网络管理者还会导致越来越复杂的问题。例如，功能的重复将导致网络管理功能的增加，使得查找网络问题非常困难。因此需要一种进行高速路由选择的系统，以避免阻塞，并避免增加网络管理复杂性。此外，需要提供一种和 IP 兼容，而没有不必要重复的连网结构。

本发明涉及网络通信，特别提供一种在交换及包路由选择之间有效实现动态转换的方法及设备，以便提供高的包通过量来解决上面讨论的问题。

根据一个实施例，本发明提供一种在网络中的上游节点和下游节点之间传输包的方法。下游节点位于上游节点的下游。该方法包括下述步骤：在上游节点和下游节点之间建立缺省虚拟通道，在下游节点接收一个包，并在下游节点对该包进行流分类，以确定该包是否属于应在上游节点重定向的指定流。该方法还包括在下游节点选择一个自由标记，并通知上游节点属于所述指定流的未来包应在已附加选定的自由标记的情况下被发送。

在另一个实施例中，本发明提供一种在第一节点交换流的方法，第一节点具有指向一个第二节点的下游链路，及指向一个第三节点的上游链路。该方法包括如下步骤：在第一节点对一个第一包进行流分类，以确定第一包是否属于应在第三节点中被重定向的指定流；在第一节点选择第一自由标记；通知第三节点属于所述指定流的未来包应在已附加选定的第一自由标记的情况下被发送。该方法还包括步骤：在第二节点对一个第二包进行流分类，以确定第二包是否属于应在第三节点中被重定向的所述指定流；在第二节点选择第二自由标记；通知第一节点属于指定流的未来包应在附加选定的第二自由标记的情况下被发送。该方法这样操作，可使来自上游链路的指定流可在第2层由第一节点交换给下游链路。

根据另一个实施例，本发明提供一种在网络中传输包的系统中的基本交换单元。该基本交换单元包括交换硬件，及和交换硬件联接的一个控制器。包括一个处理器及存储器的控制器控制交换硬件。基本交换单元还包括安装在可触知介质上的软件，该软件使基本交换单元能够在第3层IP包路由选择及第2层交换之间动态转换，从而优化包通信的通过量。

根据又一实施例，本发明提供一种在网络中传输包的系统中的交换网关单元。该系统包括一个通过通信链路和交换网关单元联接的基本交换单元。该交换网关单元包括一个网关控制器及软件。网关控制器包括一个处理器、存储器及多个NIC。安装在可触知介质上的软件使交换网关单元能够把包流重定向给基本交换单元，从而能在包路由选择及交换之间的进行动态转换，优化包通信的通过量。

根据又一实施例，本发明提供一种在网络中传输包的系统中的交换代理器（switching agent）。该系统包括通过通信链路和交换代理器联接的基本交换单元，其中基本交换单元包括一个控制器及一个交换引擎（switching engine）。交换代理器包括一个处理器、存储器及多个NIC，这些NIC中特定的一个提供所述通信链路，并且至少一个NIC能够和网络中的至少一个节点连接。交换代理器还包括安装在存储器的可触知计算机可读介质上的计算机可读程序代码。计算机可读程序代码使基本交换单元的控制器能够对流进行分类，并将来自第一节点的包重定向给网络中的第二节点，还使基本交换单元的控制器能够指挥交换代理器通过交换引擎把来自第一节点的流转送给第二节点。于是从基本交换单元的控制器卸下了包转送。

下面参考附图将更详细地说明本发明的这些及其它实施例，以及本发明的优点和特征。

图 1a 是根据本发明一个实施例的系统的的基本交换单元的简图；

图 1b 是根据本发明另一个实施例的系统的交换网关单元的简图；

图 1c 是根据本发明又一个实施例的系统的交换代理器的简图；

图 2a-2c 是根据本发明实施例的典型网络结构的简图；

图 3 是根据本发明实施例使用的典型计算机系统的简略系统方框图；

图 4 是根据本发明的一个实施例的典型 ATM 交换的简略系统方框图；

图 5a 是概括说明根据本发明的一个实施例，每个系统节点中的初始化过程的简图；

图 5b 是概括说明系统节点操作的简图；

图 5c 是概括说明初始化后，当包到达交换代理器的接口之一时，交换代理器上的过程的简图；

图 5d 是概括说明初始化后，当包通过缺省通道从交换代理器到达交换控制器的接口之一时，交换控制器（至少一个交换代理器通过一个通信链路，例如使用交换控制器的交换引擎，被连接到交换控制器上）的过程的简图；

图 6a 概括说明了在系统节点标注流的步骤；

图 6b 概括说明了在基本交换单元交换交换流的步骤;

图 6c 概括说明了在系统节点 (或交换节点) 转送包的步骤;

图 6d 概括说明了交换控制器中, 为三种情况下从源交换代理器接收的包标注流时执行的步骤;

图 6e 概括说明了为从连接的交换节点接收, 并预定给连接的交换代理器的一个接口的包标注流时, 交换控制器中执行的步骤;

图 7a-7b 说明了流类型 ( Flow Type ) 1 及类型 2 的流标识符的格式;

图 8a 根据本发明的一个实施例, 说明了类属 IFMP 相邻协议消息的结构;

图 8b 说明了一个带有长度可变数据字段的类属 IP 包 ( 现行版本 IPv4 ), 数据字段中可封装一个 IFMP 消息;

图 8c 是说明当接收带有输入 IFMP 相邻协议消息的包时, 系统节点操作的简图;

图 8d 是说明当输入的 IFMP 相邻协议消息不是 RSTACK 消息时, 发送者系统节点操作的状态图;

图 9a 根据本发明的一个实施例, 说明了类属 IFMP 重定向协议消息的结构;

图 9b 是说明当接收 IFMP 重定向协议消息时, 系统节点操作的简图;

图 9c-9g 说明了相应 IFMP 重定向协议消息的消息体 ( Message Body ) 394 中的 REDIRECT 消息元, RECLAIM 消息元, RECLAIM ACK 消息元, LABEL RANGE 消息元, 及 ERROR 消息元的结构;

图 10a 根据本发明的一个实施例, 说明了 ATM 数据链路路上的标志字段的格式;

图 10b-10e 根据本发明的实施例, 分别说明了缺省、流类型 0、流类型 1 及流类型 2 封装 IP 包;

图 11a 说明了封装 GSMP 包的格式;

图 11b 说明了 GSMP 相邻协议消息的格式;

图 11c 是说明当接收带有输入 GSMP 相邻协议消息的包时, 发送者的操作的简图;

图 11d 是说明当输入 IFMP 相邻协议消息不是 RSTACK 消息时, 发

送者的操作的状态图;

图 12 说明了类属 GSMP 连接管理 ( Connection Management ) 消息的格式;

图 13a-13e 是分别说明当接收 GSMP 连接管理加入支线 ( Add Branch )、删除支线 ( Delete Branch )、删除树 ( Delete Tree )、校验树 ( Verify Tree ) 及全部删除 ( Delete All ) 消息时, 接收器的操作的简图;

图 13f 说明了 GSMP 连接管理移动根 ( Move Root ) 消息的格式;

图 13g 是说明当接收带有输入 GSMP 连接管理移动根消息的包时, 发送者的操作的简图;

图 13h 说明了 GSMP 连接管理移动支线 ( Move Branch ) 消息的格式;

图 13i 是说明当接收带有输入 GSMP 连接管理移动支线消息的包时, 发送者的操作的简图;

图 14 说明了 GSMP 端口管理 ( Port Management ) 消息的格式;

图 15a 说明了封装 IFMP-C 包 1000 ;

图 15b 说明了可包含在图 15a 的封装 IFMP-C 包 1000 的 IFMP-C 消息字段 1006 中的典型 IFMP-C 消息 1012 的类属结构;

图 16a 说明了可包含在图 15a 的封装 IFMP-C 包 1000 的 IFMP-C 消息字段 1006 中的 IFMP-C 相邻协议消息 1040 的类属结构;

图 16b 是说明在 IFMP-C 相邻协议的三种可能状态下, 发送者 ( IFMP-C 控制器或 IFMP 代理器 ) 的操作的状态图;

图 17a 和 17b 分别说明了 IFMP-C 接口列表 ( Interface List ) 请求及响应消息的结构;

图 17c 和 17d 分别说明了 IFMP-C 接口查询 ( Interface Query ) 请求及响应消息的结构;

图 17e 说明了 IFMP-C 接口配置 ( Interface Configuration ) 请求消息 1170 的结构;

图 18a 说明了 IFMP-C 加入支线请求消息和 IFMP-C 删除支线请求消息的消息格式 1200 ;



图 18b 说明了用于图 18a 的 IFMP-C 加入支线请求消息和 IFMP-C 删除支线请求消息中的“截尾包”（“Truncate packet”）传输类型的数据传输（Data Transformation）字段 1240；

图 18c 说明了 IFMP-C 加入支线响应消息及 IFMP-C 删除支线响应消息的消息格式 1250；

图 18d 说明了 IFMP-C 删除树请求消息 1260 的结构；

图 18e 说明了 IFMP-C 移动支线请求消息 1300 的结构；

图 19a 说明了 IFMP-C 获得树统计表（Get Tree Statistics）请求消息 1400 的结构；

图 19b 说明了树数据（Tree Data）字段使用的树数据字段结构 1406；

图 20a 和 20b 分别说明了 IFMP-C 读取支线（Read Branch）请求消息 1420 及 IFMP-C 读取支线响应消息 1430 的结构；

图 21a 说明了 IFMP-C 节点信息（Node Information）请求消息 1440 的结构；

图 21b 和 21c 分别说明了 IFMP-C 接口统计（Interface Statistics）请求消息 1460 及 IFMP-C 接口统计响应消息 1470 的结构；

图 21d 说明了图 21c 的 IFMP-C 接口统计响应消息 1470 中的接口统计字段 1480 的结构；

图 21e 说明了图 21c 的 IFMP-C 接口统计响应消息 1470 的接口统计字段 1480 中的通用统计（General Statistics）字段 1494 的结构；

图 21f 说明了图 21c 的 IFMP-C 接口统计响应消息 1470 的接口统计字段 1480 中的专用统计（Specific Statistics）字段 1530（用于 ATM 接口）的结构；

图 21g 说明了图 21c 的 IFMP-C 接口统计响应消息 1470 的接口统计字段 1480 中的专用统计字段 1540（用于以太网接口）的结构。

## I. 概述

在此公开了一种在网络中传输包的改进方法及设备。该方法及设备特别可用于经局域网（LAN）、城域网（MAN）、广域网（WAN）、英特网、或类似网络高通过量传输能够携带音频、视频及数据信号的 IP 包，这里将举例说明，但是本发明并不局限于此。本发明还可用于需经网络传

输包的各种应用中。

这里描述的系统是一种动态交换及路由选择系统。该系统一般称为“交换系统”，但是应认识到该系统动态地提供数据链路层 2 的交换功能，及网络层 3 的路由选择及包转送功能。另外，该系统的“基本交换单元”也动态地提供数据链路层 2 的交换功能，及网络层 3 的路由选择和包转送功能。该系统的“交换网关单元”用作能够把现有局域网与基于环境和基本交换单元的网络相连的访问装置。类似于交换网关单元，“交换代理器”也用作能够把现有局域网与基于环境和至少一个基本交换单元相连的访问装置。交换网关单元和基本交换单元都具有独立的流重定向管理能力，都运行路由选择协议，并且在缺少任何流重定向的情况下，都可独立作出路由选择决定，对此下文将作进一步说明。于是交换网关单元和基本交换单元是同等物。相反，不具备独立流重定向管理能力的交换代理器根据来自基本交换单元的指令转送包。按照这种来自基本交换单元的指令进行操作，交换代理器可转送接收自基本交换单元的包，从而由基本交换单元转送的大部分包，现在可由交换代理器转送给代理器接口上的现有局域网及基于环境。这些环境可包括以太网、快速以太网、FDDI、千兆位以太网、或其它类型的局域网。由于这种包转送由交换代理器根据包转送指令进行，于是基本交换单元可具有更多时间来完成其它任务，例如运行路由选择协议，及减少转送包的等待时间。由交换代理器实现包转送减少了基本交换单元的交换控制器的工作量。于是，某些情况下，例如在某一节点不需要独立的流重定向管理能力，或者可更好地利用基本交换单元的能力的情况下，适于使用交换代理器。交换代理器也可用作交换网关单元的低成本替代物。该系统和网络协议（IP）的现行版本（IPv4）兼容，而且和网络协议的未版本（例如 IPv6）兼容。该系统在网络上提供包交换和包路由选择之间的动态转换，在避免路由器阻塞的同时，提供最佳的高速包通过量。

如图 1a 所示，根据本发明的一个实施例，交换系统的基本交换单元 1 包括交换引擎 3、交换控制器 5 及安装在交换控制器 5 上的系统软件 7。特别地，交换引擎 3 利用当前通用的常规异步传输模式（ATM）交换硬件。当然，根据应用的不同，其它交换技术，例如快速包交换、帧延迟、

千兆位以太网技术或其它技术也可用于提供本发明的交换引擎 3。本实施例中，交换引擎 3 是 ATM 交换器。通常和 ATM 交换器相关的任意软件被彻底除去，ATM 交换器在 ATM 适配层（Adaptation Layer）类型 5（AAL-5）之上。这样，信令发送、现有的任意路由选择协议、及任意局域网仿真服务器或地址分解服务器等被除去。交换控制器 5 是一个具有 ATM 网络适配器或网络接口卡（NIC）9 的计算机，ATM 网络适配器或网络接口卡（NIC）9 通过 ATM 链路 11 和交换引擎 3 连接。系统软件 7 安装在基本交换单元 1 中，更具体地说安装在作为交换控制器 5 的计算机中。

基本交换单元 1 的交换引擎 3 具有多个物理端口 13，端口 13 能够和多种设备，例如数据终端设备（DTE）、数据通信设备（DCE）、服务器、交换器、网关等连接。每个物理端口 13 可通过一个 ATM 链路和安装有 ATM 适配器或网络接口卡（NIC）的设备，或者另一基本交换单元的一个端口，或者交换网关单元的一个端口，或者交换代理器的一个端口相连。提供基本交换单元的交换引擎 3 的 ATM 交换硬件在数据链路层上操作（OSI 参考模型的第 2 层）。

当由系统软件 7 判定到基本交换单元被请求时，交换引擎 3 用来实现高速交换功能。交换系统的交换能力只受交换引擎 3 中使用的硬件的限制。于是本发明的本实施例能够利用 ATM 技术的高速、大容量、高带宽性能。当然，根据应用的不同，其它交换技术，例如快速包交换、帧延迟、千兆位以太网技术、或其它技术也可用于提供本发明的交换引擎 3。

在本发明的一个实施例中，交换控制器 5 是一个通过 ATM 链路 9 和 ATM 交换硬件 3 连接的计算机，系统软件安装在该计算机上。除了在第 3 层进行标准的无连接 IP 路由选择功能外，交换控制器 5 还根据本地情况为包做出流分类决定。

如图 1b 所示，根据本发明的另一实施例，交换系统的交换网关单元 21 包括网关交换控制器 23，及安装在网关交换控制器 23 上的系统软件 25。网关交换控制器 23 包括多个网络适配器或网络接口卡 27，及一个 ATM 网络接口卡 29。类似于基本交换单元 1 的交换控制器 5，网关交换控制器 23 也是一个装备有一个 ATM 网络接口卡 29 的计算机，ATM 网络接口卡 29

具有安装在该计算机上的系统软件 25。如前所述，交换网关单元用作能够把现有的局域网与基于环境和基本交换单元的一个网络相连的访问设备。于是，网络接口卡 27 的类型可以不同，例如 10BaseT 以太网网络接口卡、100BaseT 以太网网络接口卡、光纤分布数据接口（FDDI）网络接口卡、及其它网络接口卡，或者前述网络接口卡的任意组合。当然，网络接口卡 27 使用的具体类型取决于交换网关单元 21 访问的现有局域网和基于环境的类型。多个局域网可以连接到一个交换网关单元 21。ATM 网络接口卡 29 允许交换网关单元 21 通过 ATM 链路和基本交换单元 1 相连。当然，为了连接交换网关单元 21 和另一个交换网关单元，网络接口卡 27 也可以是 ATM 网络接口卡。

除了基本交换单元和交换网关单元外，本发明的系统还可包括经适当装备的主计算机、工作站或服务器。特别地，在装备有适当的 ATM 网络接口卡的主计算机、工作站或服务器上可安装一个系统软件子集，使得主机能够直接连接到基本交换单元上。

如图 1c 所示，根据本发明的又一个实施例的交换代理器 901 是一个计算机，它装有多用于连接现有局域网及基于环境的网络适配器或网络接口卡 903，一个用于连接基本交换单元 1 的 ATM 网络接口卡 905，及使交换代理器 901 能够根据基本交换单元 1 的指令转送包的适当系统软件 907。交换代理器用作能够把现有局域网与基于环境和至少一个基本交换单元相连的访问设备。于是，网络接口卡 903 可以是相同或不同的类型，例如 10BaseT 以太网网络接口卡、100BaseT 以太网网络接口卡、光纤分布数据接口（FDDI）网络接口卡、及其它网络接口卡，或者前述网络接口卡的任意组合。当然，网络接口卡 903 使用的具体类型取决于交换代理器 901 访问的现有局域网和基于环境的类型。多个局域网可以连接到交换代理器 901。ATM 网络接口卡 905 允许交换代理器 901 通过 ATM 链路和基本交换单元 1 相连。当然，根据基本交换单元 1 中采用的具体交换引擎技术（本实施例中为 ATM）来恰当选择网络接口卡 905。

基本交换单元、交换网关单元、交换代理器及系统软件允许用户以工作组、校园及城域网环境为目标建立灵活的 IP 网络，作为当前校园基于网拥挤问题的高性能、可缩放解决方案。使用本发明的系统，可实现各种网

络结构，以提供端对端无缝 IP 通信流，同时网络结构具有高带宽、高通过量及组件互操作性的特征。图 2a-2c 说明了根据本发明可能的多种网络结构中的少数几种。当然，图 2a-2c 只是示范性结构，许多替代结构也是可能的。

图 2a 表示了校园局域网结构的简图，在该网络中，基本交换单元 1 用作整个校园网络的集中 IP 包转送设备，校园网络具有几个和现有局域网连通的交换网关单元 21。基本交换单元 1 和包括三个服务器 31<sub>n</sub>（其中 n=1-3）的服务器场相连。每个服务器 31<sub>n</sub> 装备有一个系统软件子集，及一个 ATM 网络接口卡，以便能够通过对应的 ATM 链路 33<sub>n</sub>（其中 n=1-3）和基本交换单元 1 相连，ATM 链路 33<sub>n</sub> 是 OC-3（155Mbps）链路。使服务器直接通过高速 ATM 链路连接到基本交换单元 1 为频繁访问的服务器提高了包通过量。基本交换单元 1 也通过相应的 ATM 链路 33<sub>n</sub>（其中 n=4-6）和三个交换网关单元 21 相连，ATM 链路 33<sub>n</sub> 也是 OC-3 链路。通过链路 33<sub>4</sub> 和基本交换单元 1 相连的第一交换网关单元 21 还通过适当的链路 39<sub>1</sub> 和局域网 35<sub>1</sub> 相连，局域网 35<sub>1</sub> 可以是以太网或 FDDI 的某些类型。局域网基于 35<sub>1</sub> 通过适当的网络接口卡 43 和个人计算机、终端或工作站 41 相连。类似地，分别通过链路 33<sub>5</sub> 及 33<sub>6</sub> 和基本交换单元 1 连接的第二及第三交换网关单元 21 也分别通过以太网或 FDDI 链路 39<sub>2</sub> 及 39<sub>3</sub> 和局域网基于 35<sub>2</sub> 及 35<sub>3</sub> 连接。这样按照本发明，图 2 的结构使和不同局域网连接的用户能够使用无缝 IP 通信流通信，而不会产生阻塞。

作为另一个例子，图 2b 表示了工作组结构的简图。图 2b 说明了高性能工作组环境，其中几个主计算机 45 通过 ATM 链路 33<sub>m</sub> 和多个基本交换单元 1 连接，基本交换单元 1 和与带有用户设备 41 的局域网 35 相连的交换网关单元 21 连接。这种结构中，第一基本交换单元 1 通过 ATM 链路 33<sub>1</sub>（155Mbps）和第二基本交换单元 1 连接。多个主计算机 45 通过各自的 155Mbps ATM 链路 33<sub>x</sub>（其中 x=2-5）经各自的 ATM 网络接口卡 47 和第一基本交换单元 1 相连。另外，多个主计算机 45 通过各自的 25Mbps ATM 链路 33<sub>y</sub>（其中 y=8-10）经各自的 ATM 网络接口卡 49 和第二基本交换单元 1 相连。如前所述，装备有 ATM 网络接口卡的主计算机 45 上安装有一个系统软件子集，使得 TCP/IP 主机能够直接和基本交换单元相连。第一和第二基本交换单元 1 分别通过 ATM 链路 33<sub>6</sub>（155Mbps）和 33<sub>7</sub>

( 25Mbps ) 和交换网关单元 21 连接。第一及第二基本交换单元 1 通过以太网或 FDDI 链路 39 和交换网关单元 21 的连接使主计算机 45 的用户能够和连接到局域网 35 的用户设备 41 通信。用户设备 41 可以是具有适当的和任意以太网或 FDDI 局域网 35 连接的网络接口卡 43 的个人计算机、终端或工作站。这样主计算机的工作组就和校园网络的其余部分无缝集成在一起。

作为又一个例子, 图 2c 表示了使用一个基本交换单元 1; 几个交换代理器 911、913、915; 及一个系统节点 916 (例如, 另一个基本交换单元、交换网关单元、或主机) 的简单结构的简图。当然, 其它结构可以随意地包含更多的系统节点及其它组合。图 2c 举例说明了几个交换代理器 911、913 及 915, 每个代理器具有各自的用于不同以太网局域网  $917_n$  (本例中  $n=1-6$ ) 的接口, 每个以太网局域网  $917_n$  都具有连接的用户设备 (图中未表示出), 每个代理器通过 ATM 链路  $919_m$  (本例中  $m=1-3$ ) 和基本交换单元 1 相连, 基本交换单元 1 包括一个由 ATM 链路 923 与交换引擎 925 连接的交换控制器 921。当然, 局域网  $917_n$  可以是 FDDI、10BaseT 或 100BaseT 以太网、千兆位以太网、其它类型的网络, 或者这些类型网络的任意组合。和局域网  $917_n$  连接的用户设备可以是具有和局域网  $917_n$  连接的适当网络接口卡的个人计算机、终端、打印机、服务器、工作站等等。系统节点 916 通过 ATM 链路  $919_4$  连接到基本交换单元 1 的交换引擎 925 上。

通常, 图 2c 中的交换控制器 921 通过支配相应的接口 (用于包传输及包接收), 及通过指导交换代理器如何处理在特定流类型的特定流中接收的包来控制交换代理器。特定流类型及特定流可通过 IFMP-C 协议的操作由交换控制器 921 产生。如上所述, 交换控制器 921 连接到链路层交换器 (例如 ATM 交换器 925), 链路层交换器又连接到交换代理器 (例如 911、913、915) 和/或连接到另一系统节点 916 上。初始化中, 交换控制器 921 发送 IFMP-C 包给交换代理器, 允许交换控制器 921 了解第一个交换代理器的具体配置 (关于安装的网络接口等)。随后交换控制器 921 支配连接到交换代理器上的一个或多个网络接口  $917_n$  开始接收包。交换控制器 921 还在交换代理器中建立包处理, 以便把接收的某些包传输给交换控制器

921，而丢掉接收的其它包（例如，如果这些包是为未由交换控制器 921 处理的协议而接收）。如果交换控制器 921 检测到流可由交换代理器处理，而不用交换控制器 921 介入时，交换控制器 921 就使用 IFMP-C 指导交换代理器处理该包（例如丢掉包，使用一种或多种不同输出格式把该包转送出一个或多个接口，或者使用不同的服务类型在本地转送包）。转送包的同时对包进行变换（例如，减小包中的生存时间，更新 IP 标题检查和，对不同流类型格式的标题管理等等）。下面将对交换代理器、交换节点及交换控制器（例如图 2 所示结构中所表示的）的互操作进行详细说明。

根据本发明，系统通过使用系统软件在 ATM（或其它实施例中的替换技术）交换硬件上加入完备的 IP 路由选择功能，替代任意现有 ATM 交换控制软件来控制 ATM 交换器。于是，本发明的系统能够在网络层 IP 路由选择（需要时）和数据链路层交换（可能时）之间移动，以便以有效的方式产生高速度、高容量的包传输，而不会产生路由器阻塞的问题。

利用下文将进一步说明的 Ipsilon 流管理协议（IFMP），系统软件使系统节点（例如基本交换单元、交换网关单元、或主计算机/服务器/工作站）能够把 IP 包分入基于某些共同特征的类似包的“流”中。流是从特定源发送给特定（单点传送或多点传送）目的地的包序列，这些包在它们的路由选择及它们所需的任意本地处理方法上相关。本发明能够根据流的类型对不同流进行不同处理。可通过把某些类型的流映到使用 ATM 交换引擎的单独 ATM 连接中对它们进行处理，以实现包的高速交换。某些类型的流，例如载有实时通信的流，具有服务要求等级的流，或者可能具有长保持时间的流可被构成，以便只要可能就被交换。其它类型的流，例如持续时间短的流或数据库查询由无连接 IP 路由选择处理。特定的包流和特定的 ATM 标记（即，ATM 虚拟路径标识符（VPI）和虚拟通道标识符（VCI））相联系。假定虚拟通道是单向的，于是每个链路输入方向的 ATM 标记由它所连接的输入端口所拥有。链路上传输的每个方向被分别处理。当然，每个方向上的流传输由系统分开处理，但是处理方式类似。

流分类是一个本地策略决定。当系统节点接收 IP 包时，该系统节点通过缺省通道传输该 IP 包。节点还把该 IP 包分属一个特定流，于是决定属于相同流的未来包是应在 ATM 交换引擎中优先直接交换，还是继续由节点

中的路由器软件逐段转送。如果决定交换包流，首先必须标注包流。为了标注包流，节点为该包流选择接收该包的输入端口的一个可用标记（VPI/VCI）。随后作出标注包流决定的节点存储该标记、流标识符及生存时间，并且逆流向包的前一来源节点发送 IFMP REDIRECT（重定向）消息。流标识符含有说明该流的标题字段集。生存时间规定重定向的有效时间长度。除非刷新流状态，否则生存时间期满，流和标记之间的联系就被删除。在刷新流状态之前，生存时间期满会导致更多属于该流的包将在相邻节点之间的缺省转送通道上传输。通过逆流发送一个重定向消息完成流状态刷新，该重定向消息具有和先前 REDIRECT 消息相同的标记及流标识符，及另一个生存时间。重定向消息请求上游节点通过标记指定的虚拟通道传输所有其它和流标识符标识的特征匹配的包。重定向决定也是一个由上游节点处理的本地决定，而流分类决定是由下游节点处理的本地决定。于是即使下游节点请求重定向特定包流，上游节点也可决定是接受还是忽略该重定向请求。另外，不应答重定向消息。相反，到达新的虚拟通道的第一个包指示重定向请求已被接受。

为通过 ATM 数据链路传输属于标识流的 IP 包，系统软件还使用了不同的封装，这取决于流的不同流类型。在本实施例中，使用了四种类型的封装。

除了 IFMP 之外，系统软件利用另一种协议——通用交换管理协议（GSMP），通过系统的基本交换单元的交换控制器和 ATM 硬件交换引擎之间的 ATM 链路建立通信，从而使第 2 层的交换（可能时）和第 3 层的 IP 路由选择及包转送（必须时）成为可能。特别地，GSMP 是一种控制 ATM 交换器的通用、非对称协议。即，交换控制器和 ATM 交换器之间是主从关系。GSMP 运行于初始化时跨越交换控制器和 ATM 交换器之间的 ATM 链路建立的虚拟通道上。单个交换控制器可在独立的虚拟通道上使用 GSMP 的多个实例来控制多个 ATM 交换器。同样包括在 GSMP 中的是 GSMP 相邻协议，该协议用来使交换控制器和 ATM 交换器之间的 ATM 链路两头的状态同步，以便发现链路另一端的实体身份，并检测该实体身份的变化。

GSMP 允许交换控制器跨越 ATM 交换器建立并释放连接，在点对多



点连接上加入及删除叶,管理交换端口,请求配置信息及请求统计。GSMP还允许ATM交换器通知交换控制器诸如链路正在下行(GOING DOWN)之类的事件。

假定交换器具有多个端口,其中每个端口是输入端口及输出端口的组合。ATM信元从外部通信链路通过输入虚拟通道在一个输入端口到达该ATM交换器,并从一个输出端口通过输出虚拟通道离开该ATM交换器去往外部通信链路。如前所述,端口或链路上的虚拟通道由它们的VPI/VCI标注。通过把一个或多个输入虚拟通道(或根)和一个或多个输出虚拟通道(或支线)相连接,形成一个跨越ATM交换器的虚拟通道连接。虚拟通道连接由它们到达的输入端口,及它们的输入虚拟通道的VPI/VCI标注。在交换器中,每个端口具有一个由输入ATM的VPI/VCI索引的硬件查找表,表中的条目由交换器中的本地控制处理器控制。

对于GSMP,当建立每个虚拟通道连接时,通过向虚拟通道连接分配一个优先级,可使建立的每个虚拟通道连接具有一定的服务质量(QOS)。对于共享同一个输出端口的虚拟通道连接,如果两个信元同时位于交换器中,那么等级较高连接上的ATM信元和等级较低连接上的ATM信元相比,更可能离开交换器。交换器的多个端口支持的优先级数目从端口配置消息中获得。不同交换器可以不同的方式支持多点传送,例如,交换器可能对多点传送连接的支线数目、支持的多点传送的数目,可分配给多点传送连接的输出支线的不同VPI/VCI的数目有限制,和/或在同一输出端口上支持特定多点传送连接的单个支线。于是可随意规定故障代码。

交换器分配32-位端口数来描述交换端口。端口数可被配置在关于交换器的物理结构(例如,架、槽、端口)的子字段中。每个交换端口还含有交换器分配的端口话路号。在端口不断上行的同时,端口的端口话路号保持。但是,如果在端口已经下行,或不可用之后,或在一个动力循环之后,端口返回向上状态,则端口的端口话路号将改变。使用随机数的某些形式来分配端口话路号,并且端口话路号允许交换控制器检测链路故障,并保持状态同步。

除了IFMP和GSMP之外,一些实施例中的系统软件还使用了另一协议,下文将进一步说明的客户机Ipsilon流管理协议(IFMP-C),通过

基本交换单元的交换控制器和交换代理器之间的链路来建立通信，从而当需要时，把第3层的包转送分配给交换代理器。特别地，IFMP-C是一种控制交换代理器的通用、非对称协议。即，交换控制器和交换代理器之间是主从关系。通过使用IFMP-C，交换代理器上的接口好象本地连接在交换控制器上的接口，从而交换控制器/交换代理器外表上看来好象一个系统节点。通常，IFMP-C运行于初始化时跨越交换控制器和交换代理器之间的链路建立的虚拟通道上。单个交换控制器可在独立的虚拟通道上使用IFMP-C的多个实例来控制多个交换代理器。系统起动时，交换控制器启动连接到交换控制器上的每个ATM接口上的一个IFMP-C受话器（受话器连接到ATM接口的缺省VCI上），交换代理器开始在缺省VCI上发送周期SYN消息。当交换控制器从交换代理器接收SYN消息时，交换控制器启动包括在IFMP-C协议中的IFMP-C相邻协议。通过由链路每一侧使用，IFMP-C相邻协议用于跨越交换控制器和交换代理器之间的链路使状态同步，以便发现链路另一端的实体的身份，及检测该实体身份的变化。当IFMP-C相邻协议已使链路两侧状态同步，则链路的每一侧都具有一个识别链路另一侧的实例号。

在完成同步之后，IFMP-C允许交换控制器确定交换代理器上的哪些端口或接口（及它们的属性）可用，并配置每个接口，从而使之可用于转送包。假定交换代理器具有多个端口或接口，其中每个接口或端口是一个输入端口和一个输出端口的组合物。一旦确定并配置了接口，则IFMP-C用于产生、修改及删除转送支线。每个转送支线由输入数据及输出数据组成。在交换代理器中，每个接口具有一个由输入包的输入数据/输出数据索引的硬件查找表，表中的通路由交换代理器中的本地控制处理器控制。输入数据包括几部分信息（根据特定实施例，例如有输入接口、优先级、输入标志、关键字数据及关键字屏蔽），每部分信息成为输入信息的一部分。如果输入数据的任一部分变化，则认为包具有不同的转送输入通路。输出数据包括说明和输入数据匹配的包应被如何发送的几部分信息（根据特定实施例，包括输出接口、删除长度、变换、变换数据、标题数据、服务质量类型及服务质量数据）。一个输入通路可能具有多个输出通路。当包到达交换代理器的接口时，交换代理器搜索和该输入接口相关的输入通路。

可按从最低优先级到最高优先级的顺序搜索这些通路。当找到一个匹配的输入通路时，输出支线的信息用于转送该包。

在 IFMP-C 中，链路级硬件的管理（例如，打开虚拟通道及在以太网中加入硬件地址过滤器）留给交换代理器完成。如果输入关键字屏蔽包括链路级地址位，则交换代理器应确保其将接受这些地址。如果屏蔽不包括链路级寻址信息，则交换代理器不应调整过滤器。这样交换代理器可用对其硬件最有效的方式控制链路级过滤，并且交换代理器必须在关键字中包括足够的链路级信息，以便恰当地过滤包。交换代理器为混杂及多点传送混杂模式管理交换代理器的状态，从而交换代理器不会超出要求的动作之外，试图对代码路径进行不恰当的优化。

下面将根据本发明的一个具体实施例，进一步说明 IFMP、GSMP 及 IFMP-C。

## II. 系统硬件

### A. 控制器硬件

图 3 是可用作基本交换单元 1（如图 1a 所示）中的交换控制器 5 来执行本发明系统软件的典型计算机系统 51 的系统方框图。图 3 还说明了可用作交换网关单元 21（如图 1b 所示）中的交换网关控制器 23 来执行本发明系统软件的计算机系统的一个例子，图 3 还说明了可用作装有系统软件子集的主计算机/服务器/工作站的典型计算机的一个例子。当然，为主机配备了其它部分，例如显示器、屏幕及键盘。如图 3 所示，计算机系统 51 包括中央处理器 69、系统存储器 71、I/O 控制器 73、固定磁盘 79、网络接口 81 及只读存储器（ROM）83 之类的子系统。当然，计算机系统 51 还可为主机选择地包括显示器 53、键盘 59、显示适配器 75 及活动磁盘 77。箭头，例如箭头 85 表示计算机系统 51 的系统总线结构。但是，这些箭头只是用来举例说明任何链接子系统的互连方案的。例如，可使用局部总线把中央处理器 69 和系统存储器 71 及只读存储器 83 相连接。其它适于和本发明一起使用的计算机系统可包括更多的或较少的子系统。例如，另一个计算机系统可包括多个处理器 69（即，多处理器系统）或一个超高速缓冲存储器。

在本发明的一个实施例中，用作交换控制器的计算机是标准的基于

Intel 中央处理器 ( CPU ) 的计算机, 它装备有标准的外围部件互连 ( PCI ) 总线, 还装备有 ATM 网络适配器或网络接口卡 ( NIC )。该计算机使用 ATM 网络接口卡通过 155Mbps ATM 链路和 ATM 交换器相连。本实施例中, 系统软件安装在固定磁盘 79 上, 固定磁盘是该计算机的硬盘。本领域有经验人员知道, 系统软件可存储在 CD-ROM、软磁盘、磁带或其它存储计算机可读代码的可触知介质上。

图 3 所示的计算机系统 51 只是适于和本发明一起使用 ( 作为基本交换单元的交换控制器, 作为交换网关单元的交换网关控制器, 或者作为主计算机/服务器/工作站 ) 的计算机系统的一个例子。此外, 图 3 说明了可用作交换代理器 901 ( 如图 1c 所示 ) 的安装有至少一个系统软件子集 ( 为 IFMP-C 可操作性而提供 ) 的计算机系统的一个例子。应认识到路由选择协议的系统软件不必安装在用作交换代理器 901 的计算机系统上, 从而该系统软件子集可在一个嵌入设备上运行。

于是, 可从用作交换代理器 901 的计算机系统中省去固定磁盘 79, 从而对于某些可使用交换代理器 901 代替交换网关单元的网络, 将导致设备费用降低。适于和本发明一起使用的子系统的其它配置对本领域有经验人员来说是显而易见的。另外, 交换网关单元可装备有多个其它网络接口卡, 以便能够连接各种类型的局域网。在交换网关单元中可使用用于不同类型局域网基干的其它网络接口卡或者另外的适配器。例如, 可使用 SMC 10M/100M 以太网网络接口卡或 FDDI 网络接口卡。

在不限本发明范围的情况下, 表 1 根据上述实施例提供了市场上买得到, 可用于控制器操作的部件列表。对本领域有经验人员来说, 表 1 所列的部件显然仅仅代表那些可以和本发明结合使用的部件, 并且表 1 所列部件是为装配根据本发明特定实施例的设备而提供的。本领域有经验人员可用其知晓的各种部件容易地进行替代, 或者进行功能的结合或分离。

**表 1**  
**控制器组件**

微处理器	Intel Pentium 133Mhz 处理器
系统存储器	16Mbyte RAM/256K 超高速缓冲存储器
主板	Intel Endeavor 主板
ATM 网络接口卡	Zeitnet PCI ATM 网络接口卡 ( 155Mbps )
固定磁盘或硬盘	500Mbyte IDE 磁盘
驱动器	标准软盘驱动器, CD-ROM 驱动器
电源	标准电源
机箱	标准机箱

### B. 交换硬件

如上所述, ATM 交换硬件提供基本交换单元的交换引擎。ATM 交换引擎使用厂家独立 ATM 交换硬件。但是, 根据本发明的 ATM 交换引擎并不依靠其通常的面向连接的 ATM 路由选择及信号发送软件 ( SSCOP、Q.2931、UNI3.0/3.1 及 P-NNI )。相反, 完全放弃了所有的 ATM 协议及软件, 并且基本交换单元依靠系统软件来控制 ATM 交换引擎。将在后面对系统软件进行详细说明。

独立可用的 ATM 部件可被装配到一个典型 ATM 交换器结构中。例如, 图 5 是 ATM 交换器 3 ( 图例为 16-端口的交换器 ) 的结构简略方框图, 根据本发明的一个实施例, ATM 交换器 3 可用作基本交换单元的交换硬件引擎。但是, 根据本发明的另一个实施例, 市场上买得到的 ATM 交换器也可用作基本交换单元的交换引擎。交换硬件 3 的主要功能部件包括一个开关磁芯, 一个微控制器集合体, 及一个收发机子部件。通常, 开关磁芯实现第 2 层交换, 微控制器集合体为 ATM 交换器提供系统控制, 收发机子部件为接口及来自物理层信号的基本传输和接收而提供。在本例中, 开关磁芯基于 MMC 网络 ATMS2000 ATM 交换芯片集 ( Switch Chip Set ), 该芯片集包括白 ( White ) 芯片 100、灰 ( Grey ) 芯片 102、Mbuf 芯片 104、端口接口装置 ( Port Interface Device ( PIF ) ) 芯片 106 及公用数据存储器 108。开关磁芯还可选择地包括 VC 活动检测元件

( Activity Detector ) 110 及早期包丢弃 ( Early Packet Discard ) 功能件 112。还可包括包计数器, 图中未表示出。白芯片 100 提供配置控制及配置状态。除了和白芯片 100 进行配置状态及控制通信外, 灰芯片 102 还负责利用交换表进行直接寻址及数据传输。MBUF 芯片 104 负责在 PIF 芯片 106 及公用数据存储单元 108 之间移动信元通信。公用数据存储单元 108 用作交换器中的信元缓冲。PIF 芯片 106 管理 MBUF 芯片和交换端口硬件之间来回的数据传输。包括一个存储器元件的 VC 活动检测元件 110 提供有关每个活动虚拟通道的信息。早期包丢弃功能件 112 当需要时能够丢弃某些 ATM 信元。包计数器使交换器能够计数通过所有输入及输出端口的所有包。总线 114、115、116、117 提供交换器的各个部件之间的连接。微控制器集合体包括一个中央处理器 ( CPU ) 130、动态随机存取存储器 ( DRAM ) 132、只读存储器 ( ROM ) 134、快擦除存储器 136、DRAM 控制器 138、对偶通用异步收发器 ( DUART ) 端口 140 和 142、以及外部定时器 144。CPU130 起微控制器的作用。ROM134 起本地引导 ROM 的作用, 包括所有的交换代码图像、基本的低级操作系统功能, 以及诊断信息。DRAM132 提供常规的随机存取存储器功能, DRAM 控制器 138 ( 它可由现场可编程门阵列 ( FPGA ) 装置或类似装置实现 ) 提供对 DRAM132 的刷新控制。快擦除存储器 136 可被微控制器访问, 以获得硬件修正控制、序列号标识、及用于可制造性和跟踪的各种控制代码。DUART 端口 140 和 142 作为和诊断、监视及其它目的的通信资源相连的接口。需要时, 外部定时器 144 中断 CPU130。收发机子部件包括位于 PIF 芯片 106 和物理收发机 ( 图中未表示出 ) 之间的物理接口装置 146。接口装置 146 完成对数据流的处理, 并实现 ATM 物理层。当然交换器的部件可布置在印刷电路板, 该电路板可驻留在机架上, 使其安装或放置在一个台式装置上, 这取决于所使用机箱的情况。

在不限制本发明范围的情况下, 表 2 根据上述实施例提供了市场上买得到, 可用于交换引擎的操作的部件列表。对本领域有经验人员来说, 表 2 所列部件显然仅仅代表那些可以和本发明结合使用的部件, 并且表 2 所列部件是为装配根据本发明特定实施例的设备而提供的。本领域有经验人员可用其知晓的各种部件或者可得到的交换器容易地进行替代, 或者进行功

能的结合或分离。当然，如前所述，使用除 ATM 之外的其它技术（例如帧延迟、快速包交换、或千兆位以太网）的交换引擎要使用恰当的部件。

表 2

## 交换器部件

开关磁芯	
芯片集	MMC 网络 ATMS2000 ATM 交换芯片集（白芯片、灰芯片、MBUF 芯片、PIF 芯片）
公用数据存储	标准存储器模块
包计数器	标准计数器
微控制器集合体	
CPU	Intel960CA/CF/HX
DRAM	标准 DRAM 模块
ROM	标准 ROM
快擦除存储器	标准快擦除存储器
DRAM 控制器	标准 FPGA, ASIC 等
DUART	16552 DUART
外部计时器	标准计时器
收发机子部件	
物理接口	PMC-Sierra PM5346

### III. 系统软件功能

和上文概括说明的一样，IFMP 是指示相邻节点把第 2 层“标记”加入指定包“流”的协议。流是从特定源发送给特定目的地的包序列，这些包在它们的路由选择及其所需的逻辑处理方法上相关。标记指定一个虚拟通道，允许高效地存取用于该流的超高速缓存路由选择信息。标记还允许更多属于该指定流的包在第 2 层被交换，而不是在第 3 层进行路由选择。即，如果在网络中的一个特定节点上，上游链路和下游链路都重定向一个流，那么该特定节点将在数据链路层交换该流，而不是在网络层进行路由选择并转送该流。

图 5a 概括说明了根据本发明的一个实施例，每个节点中的初始化过

程。一旦在步骤 160 启动系统，则在步骤 162，每个系统节点在所有端口上建立缺省虚拟通道。随后在步骤 164，系统节点等待包到达任一端口。

图 5b 概括说明了根据本发明的在第 3 层路由选择和第 2 层交换之间动态转换的系统节点的操作。初始化之后，在步骤 166，一个包到达系统节点的一个端口。如果该包在缺少虚拟通道上被接收（步骤 168），则在步骤 170，系统节点对该包进行流分类。流分类确定该包是否属于一种流。在步骤 172，系统节点确定该包所属的流是否应被优先交换。如果系统节点确定该流应被交换，则在步骤 174，系统节点标注该流，随后在步骤 176 转送该包。转送该包之后，在步骤 182，系统节点等待包到达。一旦一个包到达，系统节点返回步骤 166。如果在步骤 168，系统节点确定该包没有到达缺省虚拟通道，则系统节点不对该包进行步骤 170 的流分类。当包到达一个备选虚拟通道时，该包属于一个已被标注的流。于是，如果该流也被标注向下游发送（步骤 178），则在步骤 180 系统节点交换该流。交换该流涉及在交换器中，在上游链路标记和下游链路标记之间建立一个连接。在步骤 180 中交换该流之后，系统节点在步骤 176 向下游转送该包。如果该流未被标注向下游发送（步骤 178），则系统节点不交换该流，而是在步骤 176 向下游转送该包。当然，只有是基本交换单元的系统节点才执行步骤 180。其它系统节点（例如交换网关单元或主机）按图 5 所示操作，但是不执行步骤 180，因为步骤 178 的结果不用于交换网关单元或主机（因为这些类型的系统节点没有下游链路）。

图 5c 和 5d 分别概括说明了根据本发明的交换控制器，及通过通信链路连接到该交换控制器上的交换代理器的操作。要注意的是交换代理器通常遵循图 5a 说明的初始化过程。图 5c 概括说明了初始化完成后，当一个包到达交换代理器的一个接口（步骤 1600）时，交换代理器上的过程。如果该包未在缺省虚拟通道上被接收（在步骤 1602 中确定），那么交换代理器在步骤 1604 存取束缚在该指定通道上的树。当包不到达虚拟通道时，该包属于一个已被标注的流，并且该流已被交换。于是交换代理器转送该包（步骤 1606），并且随后等待另一个包到达（步骤 1608）。但是，如果在缺省虚拟通道上接收到该包（在步骤 1602 中确定），那么在步骤 1610，交换代理器搜索它的支线表以寻找一个匹配的输入支线。如果未找到匹配



的输入支线（步骤 1612），则交换代理器在步骤 1614 中把该包发送给交换控制器，并等待下一个包（步骤 1616）。如果找到一个匹配的输入支线（步骤 1612），则交换代理器在步骤 1618 中以指定包的形式转送该包。随后交换代理器检查是否为该包指定了“失败”模式。如后文说明的一样，失败模式指示交换代理器在该包被传输之后，应在匹配该输入支线条目的下一优先级继续在输入支线表中寻找匹配的输入支线。如果未指定失败模式（步骤 1620），则随后交换代理器仅仅等待下一个包到达（步骤 1622）。但是如果指定了失败模式（步骤 1620），则随后交换代理器在下一优先级继续在支线表中寻找匹配的输入支线（步骤 1624）。从步骤 1624，交换代理器确定是否找到下一优先级的匹配输入支线（步骤 1626）。如果未找到，则交换代理器等待下一个包的到达（步骤 1622）。但是，如果找到下一优先级的匹配输入支线，则交换代理器从步骤 1626 转而以指定包的形式转送该包（步骤 1618），过程从步骤 1620 继续进行。

图 5d 概括说明了初始化完成后，当一个包从交换代理器到达交换控制器的一个接口的虚拟通道时，交换控制器（至少一个交换代理器通过通信链路，例如，使用交换控制器的交换引擎连接到该交换控制器上）上的过程。即，图 5d 说明了当包被发送给交换控制器（图 5c 的步骤 1614）时，交换控制器上的过程。

在包从交换代理器到达（步骤 1650）之后，交换控制器在步骤 1652 对该包进行流分类。如前所述，流分类确定该包是否属于一种流。从步骤 1652，交换控制器在步骤 1654 确定该包所属的流是否应被交换。如果交换控制器在步骤 1654 确定该流不应被交换，则交换控制器不交换该流，而只是转送该包（步骤 1656），随后等待下一个包（步骤 1658）。如果交换控制器在步骤 1654 确定该流应优先被交换，则交换控制器在步骤 1660 标注该流，并转送该包（步骤 1656），等待下一个包（步骤 1658）。

图 6a 概括说明了与在系统节点（或交换节点）的上游链路中标注流有关的步骤，例如图 5b 的标注流步骤 174 所表示的一样。系统节点是交换网关单元或主机时，系统节点标注流如图 6a 的步骤 190、192、200 及 202 所示。当标注流步骤开始时（步骤 190），在步骤 192，基本交换单元在上游链路上选择一个自由标记  $x$ 。随后在步骤 200，系统节点在上游链路

上发送 IFMP REDIRECT 消息（如虚线 193 所示）。之后系统节点在步骤 202 转送该包。系统节点是基本交换单元时，标注流由步骤 194、196、198 说明。当标注流步骤开始时（步骤 190），在步骤 192，基本交换单元在上游链路上选择一个自由标记  $x$ 。随后在步骤 194，基本交换单元的交换控制器在交换控制器的控制端口上选择一个暂时标记  $x'$ 。在步骤 196，交换控制器随后向硬件交换引擎发送一个 GSMP 消息，把上游链路上的标记  $x$  映到控制端口的标记  $x'$ 。随后在步骤 198 中，交换控制器等待，直到从硬件交换引擎接收到一个指示映射成功的 GSMP 确认消息为止。一旦接收到回应，在步骤 200，基本交换单元在上游链路上发送一个 IFMP REDIRECT 消息。步骤 200 之后，系统节点返回如图 56 所示的步骤 176。

图 6b 概括说明了与在基本交换单元中交换流有关的步骤，例如图 5b 的交换流步骤 180 所表示的一样。如上所述，只有是基本交换单元的系统节点才能执行交换流步骤。当在步骤 210 开始交换流过程时，基本交换单元中的交换控制器在步骤 212 发送一个 GSMP 消息，把上游链路的标记  $x$  映到下游链路的标记  $y$ 。标记  $y$  是基本交换单元的下游节点分配给该流的标记。当然，该下游节点已经以图 5b 和 6a 规定的方式，用在 192 步骤中选择的自由标记  $y$  标注了该流。步骤 212 之后，基本交换单元中的交换控制器在步骤 214 等待来自基本交换单元中的硬件交换引擎的 GSMP 确认消息，以指示映射成功。从而流完全在基本交换单元的硬件交换引擎中，在第 2 层被交换。随后基本交换单元在步骤 176 中着手转送包。

图 6c 概括说明了与在系统节点中转送包有关的步骤，例如图 5b 的转送包步骤 176 所示的一样。系统节点在步骤 218 开始转送包过程。如果包所属的流在下游链路上未被标注（步骤 220），则在步骤 222，系统节点在缺省虚拟通道上通过下游链路发送该包，随后转到等待状态 224 等待包的到达。但是，如果包所属的流在指示系统节点先前接收一个 IFMP REDIRECT 消息，以标注该流一个生存时间的下游链路上被标注（？），则系统节点在步骤 226 检查该流的重定向生存时间是否已期满。如果生存时间未到期，则在步骤 228，系统节点在 IFMP REDIRECT 消息中标注的虚拟通道上发送该包，随后转到等待状态 224。如果生存时间期满，则系统节点在步骤 230 自动删除流重定向。随后系统节点着手在缺省通道上发

送该包（步骤 222），并返回图 5B 所示步骤 182 的等待状态。

如上所述，图 6a-6c 说明的是不包含交换代理器的情况下，系统节点（或交换节点）的互操作。图 6d-6e 说明的是当至少一个交换代理器连接到基本交换单元上时，交换节点的互操作，如下所述。

图 6d 概括说明了为从连接的源交换代理器接收的包标注流时，在交换控制器中执行的步骤，例如如图 5d 的标注流步骤 1660 所示的一样。图 6d 中说明了三种情况：当希望把包流发送给该源交换代理器的另一个接口时；当希望把包流发送给另一个连接的交换代理器，即目的地交换代理器的一个接口时；及当希望把包流发送给另一个连接的系统节点（或交换节点，例如另一个基本交换单元，交换网关单元，或主机）的一个接口时。

如图 6d 所示，如果希望把从源交换代理器接收的包流发送给同一个交换代理器的另一个接口（在步骤 1662 中确定），则交换控制器（在步骤 1664）使用 IFMP-C 支配源交换代理器在该交换代理器的目的接口上，把对于该流接收的带有适当标题及变换的未来包转送出。

如果不希望把从源交换代理器接收的包流发送给同一个交换代理器上的另一个接口（在步骤 1662 中确定），则在步骤 1666 中确定是否希望把从源交换代理器接收的包流发送给目的交换代理器上的一个接口。如果希望，则交换控制器（在步骤 1668）在交换控制器和源交换代理器之间的上游链路上选择一个自由标记  $x$ ，并在交换控制器和目的交换代理器之间的下游链路上选择一个自由标记  $y$ （在步骤 1670）。随后交换控制器在步骤 1672 使用 GSMP 把  $x$  映到  $y$ 。在步骤 1674，交换控制器使用 IFMP-C 支配目的交换代理器在目的接口上，把对于该流在标记  $y$  接收的带有适当标题及变换的未来包转送出。随后交换控制器（在步骤 1676）使用 IFMP-C 支配源交换代理器，把该流的带有适当标题及变换的未来包发给标记  $x$ 。

如果不希望把从源交换代理器接收的包流发送给目的交换代理器的一个接口（在步骤 1666 中决定），则希望把从源交换代理器接收的包流发送给另一个连接的系统节点（或“交换节点”，例如另一个基本交换单元，交换网关单元，或主机）的一个接口。随后交换控制器（在步骤 1680）在交换控制器和源交换代理器之间的上游链路上选择一个自由标记  $x$ 。在步骤 1682，交换控制器等待下游链路上的一个自由标记  $y$  被交换节点选择，

并通过 IFMP 通信。之后，交换控制器在步骤 1684 使用 GSMP 把  $x$  映到  $y$ 。在步骤 1686，交换控制器使用 IFMP-C 支配源交换代理器，把该流的带有适当标题及变换的未来包发给标记  $x$ 。

图 6e 概括说明了为包标注流（从步骤 1700 开始）时，在交换控制器中执行的步骤，包接收自连接的交换节点，并计划发给连接的交换代理器的一个接口。当希望把从源交换节点接收的包流发送给目的交换代理器的一个接口时，交换控制器（在步骤 1702）在交换控制器和源交换节点之间的上游链路上选择一个自由标记  $x$ ，并且（在步骤 1704）在交换控制器和目的交换代理器之间的下游链路上选择一个自由标记  $y$ 。使用交换控制器在步骤 1706 使用 GSMP 把  $x$  映到  $y$ 。在步骤 1708，交换控制器使用 IFMP-C 支配目的交换代理器的目的接口，把对于该流在标记  $y$  接收的带有适当标题及变换的未来包转送出。在步骤 1710，交换控制器使用 IFMP 请求上游交换节点向标记  $x$  传输该流的未来包。上述概括说明的更多细节将在下文说明。

## A. IFMP&ATM 数据链路上的流标注传输

### 1. IFMP

利用 Ipsilon 流管理协议（IFMP），系统软件使系统节点（例如基本交换单元、交换网关单元、或主计算机/服务器/工作站）能够把 IP 包分入基于某些共同特征的相似包的“流”中。流由“流标识符”指定。特定流的流标识符给出包标题中定义流的字段集的内容或值。在属于该特定流的所有包中，包标题的字段集的内容都是相同的。可以规定几种“流类型”。每种流类型的规定用于识别流的包标题的字段集。例如，一种流类型可规定包标题的字段集，该包标题标记该流为具有在运行于站上的应用之间携带数据的包，而另一种流类型可规定另一种包标题的字段集，该包标题标记该流为具有在站之间携带数据的包。

在本发明的一个实施例中，规定了三种流类型：流类型 0，流类型 1，流类型 2。当然，也可规定不同的或更多的流类型。流类型 0 用于把 IP 包从缺省封装改变为新的封装。流类型 0 的流标识符的格式为空，因此其长度为 0。流类型 1 规定包标题的字段集，该包标题标记该流为具有在运行于站上的应用之间携带数据的包。流类型 1 可用于具有 UDP 及 TCP 协议

的包的流，在这些包中，IP标题后的头四个八位位组规定用于指示应用的一个源端口数及一个目的端口数。流类型1的流标识符具有4个32-位的字。图7a中附图标记240指示流类型1的流标识符的格式，包括（按照从最高位（MSB）到最低位（LSB）的顺序）作为第一字的版本字段、网间标题长度字段（英特网 Header Length（IHL））、服务类型字段、生存时间字段及协议字段；作为第二字的源地址字段；及作为第三字的目的地址字段。流标识符中的这些字段来自流类型1的IP包的标题。流类型1的流标识符还包括作为第四字的源端口数字段及目的端口数字段（IP包中IP标题后的头四个八位位组）。流类型2规定另一种包标题的字段集，该包标题标记该流为具有在站之间携带数据的包，而没有规定在站上运行的应用。流类型2的流标识符具有3个32-位的字。图7b中附图标记250指示的流类型2的流标识符的格式，包括来自IP包标题的版本字段、网间标题长度字段（英特网 Header Length（IHL））、服务类型字段、生存时间字段及协议字段，源地址字段，目的地址字段。类型流2的流标识符的格式和流类型1的流标识符的格式相同，只是没有第四字。各种流类型的流标识符的分级特性使得可对IP包进行最具体的匹配操作，简化了流分类。

本发明能够根据流的类型对不同类型的流进行不同处理。某些类型的流，例如载有实时通信的流，具有服务要求等级的流，或者可能具有长保持时间的流可被构成，以便只要可能就被交换。其它类型的流，例如持续时间短的流或数据库查询由无连接IP包转送处理。另外，每种流类型还规定一种封装，该封装将在流类型被重定向之后使用。可对于不同的数据链路技术，指定用于每种流类型的封装。在本实施例中，系统使用用于ATM数据链路的封装，对此将在下文进一步说明。

特定的包流和特定的ATM标记相联系。根据本实施例，标记是虚拟路径标识符和虚拟通道标识符（VPI/VCI）。用于一个特定端口的多个标记是可在该端口使用的标记集（VPIs/VCI s）。假定虚拟通道是单向的，于是每个链路输入方向的ATM标记由它所连接的输入端口所拥有。当然，对于使用其它交换技术，例如帧延迟的实施例，数据链路连接标识符可用作标记。对于使用快速包交换技术的实施例，数据链路通道多路复用标识

符可用作为标记。

如前所述，流分类是一个本地策略决定。当系统节点接收一个 IP 包时，该系统节点通过缺省通道传输该 IP 包。节点还把该 IP 包分入一个特定流，于是决定属于同一流的未来包是应在 ATM 交换引擎中直接被交换，还是继续由节点中的路由器软件逐段转送。如果决定交换包流，节点为该流选择接收该包的输入端口的一个可用标记（VPI/VCI）。作出交换包流决定的节点随后存储该标记、流标识符及生存时间，并且逆流向包的前一来源节点发送 IFMP REDIRECT 消息。如前所述，流标识符含有说明该流的标题字段集。生存时间规定重定向的有效时间长度。除非刷新流状态，否则一旦生存时间期满，流和标记之间的联系将被删除。在刷新流状态之前，生存时间期满会导致更多属于该流的包将在相邻节点之间的缺省转送通道上传输。

通过逆流发送一个重定向消息完成流状态刷新，该重定向消息具有和先前的 REDIRECT 消息相同的标记及流标识符，及另一个生存时间。重定向消息请求上游节点通过标记指定的虚拟通道，传输所有其它和流标识符标识的特征匹配的包。重定向决定也是一个由上游节点处理的本地决定，而流分类决定是由下游节点处理的本地决定。于是即使下游节点请求重定向特定包流，上游节点也可决定是接受还是忽略该重定向请求。另外，重定向消息不被应答。相反，到达新的虚拟通道的第一个包指示重定向请求已被接受。

本发明中，系统软件的 IFMP 包括 IFMP 相邻协议及 IFMP 重定向协议。IFMP 相邻协议使系统节点（主机、基本交换单元或交换网关单元）可以发现链路另一端的系统节点的身份。此外，IFMP 相邻协议被用来使链路两端的的状态同步，以便当链路另一端的系统节点变化时能够检测到，及交换分配给链路的 IP 地址列表。使用 IFMP 重定向协议，只是在系统已使用 IFMP 相邻协议识别了链路另一端的其它系统节点，并且链路两端实现状态同步之后，系统才可发送 REDIRECT 消息。通过当前未实现状态同步的链路接收的任何 REDIRECT 消息必须被丢弃。在下面对系统操作的详细说明之后，将对 IFMP 相邻协议及 IFMP 重定向协议进行详细说明。

利用如图 2a 所示的局域网结构，说明本发明系统的流分类及重定向的

一个特定例子可用于说明本发明系统具有的优点。特别地，该例子主要针对图 2a 的第一及第二网关交换单元 21 和基本交换单元 1 之间的相互作用。系统启动时，在基本交换单元 1 的控制器，及每个相邻节点（本例中，为第一及第二交换网关单元 21）的控制器上运行的系统软件之间建立一个缺省的转送 ATM 虚拟通道。当通过网络层链路 39<sub>1</sub> 从局域网基于 35<sub>2</sub> 传输一个 IP 包时，该 IP 包由第一交换网关单元 21 通过其一个合适的局域网网络接口卡接收。随后，第一交换网关单元 21 的系统软件检查该 IP 包，并且随后对该 IP 包内容进行缺省封装，以便通过链路 33<sub>4</sub>（建立于交换网关单元 21 的 ATM 网络接口卡和基本交换单元 1 中的 ATM 交换硬件的一个选定端口之间）传输给基本交换单元 1。ATM 交换硬件随后把 ATM 信元转送给交换控制器 5 中的 ATM 网络接口卡 9，ATM 网络接口卡随后重新装配该包，并将 IP 数据报转送给交换控制器中用于 IP 路由选择的系统软件。交换控制器以常规方式，通过启动时在基本交换单元 1 和第二交换网关单元 21 之间建立的缺省转送通道转送该包。另外，基本交换单元 1 中的交换控制器对该包进行流分类，以确定属于同一流的未来包是应在该 ATM 硬件中直接被交换，还是应继续由系统软件逐段寻径转送。如果交换控制器软件本地决定应交换该流，则它从接收该的输入端口（端口 i）的标记空间（标记空间仅仅是 VPI/VCI 标记的范围）选择一个自由标记（标记 x）。交换控制器也在它的控制端口（交换控制器通过其和 ATM 交换器连接的实际或虚拟端口）选择一个自由标记（标记 x'）。使用 GSMP，系统软件指导 ATM 交换器把输入端口 i 上的标记 x 映到控制端口 c 上的标记 x'。当交换器向交换控制器返回一个 GSMP 确认消息时，交换控制器逆流向包的前一个来源节点（本例中，为第一交换网关单元 21）发送一个 IFMP REDIRECT 消息。该 REDIRECT 消息是一个从基本交换单元 1 发给第一交换网关单元 21 的请求，它请求在 REDIRECT 消息的标记指定的 ATM 虚拟通道上，传输所有其它标题字段和重定向消息的流标识符中指定字段匹配的包。除非在 REDIRECT 消息的生存时间期满之前刷新流状态，否则流和重定向消息的标记之间的联系应被删除，这样就导致流中更多的包在第一交换网关单元 21 和基本交换单元 1 之间的缺省转送通道（启动时初始建立的）上被传输。

如果第一交换网关单元 21 接受了在基本交换单元 1 发送的 REDIRECT 消息中做出的请求, 则属于该流的包将到达交换控制器的带有 ATM VPI/VCI 标记  $x'$  的端口  $c$ 。包将继续由系统软件重新装配并进行路由选择, 但是由于对于该流的前一个路由选择决定被高速缓存, 并由系统软件中的标记  $x'$  索引, 因此对包的处理速度加快了。于是, 可以看出流可被标注, 但是并不必需被交换。

在下游节点(本例中为第二交换网关单元)也和相同流的重定向相关的情况下, 交换的重要优点之一变得明显。当基本交换单元 1 通过它和第二交换网关单元 21 之间的缺省转送通道, 把属于该流的初始包送到第二交换网关单元 21 时, 下游节点(本例中为第二交换网关单元 21)重新装配该包, 并以常规方式转送该包。对于在其端口  $j$  接收的包, 第二交换网关单元 21 也进行流分类, 并根据表中表述的本地策略决定是交换属于该流的未来包, 还是在控制器中继续包转送。如果第二交换网关单元 21 决定应交换该流的未来包, 则它逆流向基本交换单元 1 发送它自己拥有的 REDIRECT 消息(具有其端口  $j$  上的自由标记  $y$ , 流标识符及生存时间)。当然基本交换单元 1 可接受或忽略该重定向请求。当基本交换单元 1 决定交换该流时, 基本交换单元 1 的交换控制器中的系统软件把标记端口  $i$  上的标记  $x$  映到端口  $j$  上的标记  $y$ 。这样, 通信不再发送给交换控制器处理器, 而是直接交换给所要求的 ATM 交换硬件的输出端口。于是, 所有其它属于该流的通信可完全在基本交换单元 1 的 ATM 交换硬件内被交换。当包从基本交换单元 1 的 ATM 交换器的一个端口到达时, 第二交换网关单元 21 使用它的 ATM 网络接口卡通过 ATM 链路  $33_5$  接收该包。随后第二交换网关单元重新装配该包, 并通过它的一个网络接口卡经链路  $39_2$  将该包发送给局域网  $35_2$ 。包计划发往的用户设备 41 通过用户设备的网络接口卡 43 从局域网  $35_2$  接收该包。

当系统节点(本例中为基本交换单元 1)接受一个 REDIRECT 消息时, 它还改变该重定向流的封装。系统节点会根据流类型使用一种不同类型的封装, 而不是在虚拟转送通道上使用 IP 包的缺省封装。这样基本交换单元 1 封装属于该流的未来包, 并通过标记  $y$  中记录的指定虚拟通道传输这些未来包。某些类型的封装会从 IP 包中除去某些字段。当这些字段被除去时,



发布 REDIRECT 消息的系统节点存储该字段，并且把该字段和指定的 ATM 虚拟通道联系起来。在本例的情况下，如果基本交换单元 1 接受由第二交换网关单元 21 发送的 REDIRECT 消息，则基本交换单元 1 存储字段，并把字段和由标记 y 指定的 ATM 虚拟通道联系起来。类似地，如果第一交换网关单元 21 接受由第一交换单元 1 发送的 REDIRECT 消息，则第一交换网关单元 21 存储字段，并把该字段和由标记 x 指定的 ATM 虚拟通道联系起来。使用输入标记访问存储的字段可以重建一个完整的包。通过，例如防止用户建立一个去往防火墙后面的一个许可目的地或服务的交换流，随后改变 IP 包标题来获得对禁止目的地的访问，这种方法提供了一种安全措施。

每个系统节点保留一个后台刷新计时器。当后台刷新计时器期满时，检查每个流的状态。如果从前次刷新周期以来，流已接收了通信，则系统节点通过逆流发送一个具有和初始 REDIRECT 消息相同的标记及流标识符，及一个新的生存时间的 REDIRECT 消息来刷新该流的状态。如果该流从前一个刷新周期以来没有接收通信，则系统节点除去该流的高速缓存的状态。通过逆流发布一个 RECLAIM（回收）消息回收该标记以便再次使用，这样系统节点除去该流的状态。但是，在上游节点发送一个由发布 IFMP RECLAIM 消息的节点接收的 IFMP RECLAIM ACK 消息之前，不删除流状态，并且不重新使用该标记。IFMP RECLAIM ACK 消息确认所请求标记的释放。系统节点确定一个流是否以两种不同的方式接收了通信，这取决于该流是被交换还是不被交换。对于被标注，但是不被交换的流，控制器为系统节点检查其自身的状态，以了解该流是否已在前一刷新周期中接收了任一通信。对于被交换的流，控制器使用 GSMP 消息为系统节点查询 ATM 交换硬件，以了解特定通道最近是否一直是活动的。于是，在本例中，基本交换单元 1 监视一个流的通信，如果该特定流被从第一交换网关单元 21 映到基本交换单元 1 的控制端口，或者通过基本交换单元 1 中的 ATM 交换器被从第一交换网关单元 21 映到第二交换网关单元 21。如果在前一刷新周期中，该流没有新的通信，则基本交换单元将发送 IFMP RECLAIM 消息，并且当接收到 IFMP RECLAIM ACK 消息时除去流状态。同样，第二交换网关单元 21 监视一个流的通信，如果该特定流被从基

本交换单元 1 的控制端口映到第二交换网关单元 21。此外，装备有适当系统软件的主计算机/服务器/工作站也装备有一个后台刷新计时器。通过监视映到其上的任意流的通信，主机可发送一个 IFMP RECLAIM 消息，并且当接收到 IFMP RECLAIM ACK 消息时除去流状态。

如上所述，IFMP 相邻协议被用于建立状态同步，及识别相邻的系统节点及交换 IP 地址。对 IFMP 相邻协议来说，对特定链路系统节点具有三种可能状态：SYNSENT（同步消息发送），SYNRCVD（同步消息接收），ESTAB（同步建立）。在系统可使用 IFMP 重定向协议发送任意重定向消息之前，要求链路两端的同步（当一个系统节点达到链路的 ESTAB 状态时）。

图 8a 说明了类属 IFMP 相邻协议消息 300 的结构。所有的 IFMP 相邻协议消息都封装在一个 IP 包中。图 8b 说明了具有可变长度数据字段的类属 IP 包（为当前版本 IPv4），一个 IFMP 相邻协议消息可被封装在该可变长度数据字段中。作为 IP 包含有 IFMP 消息的指示，封装 IP 包的 IP 标题中的协议字段必须含有十进制值 101。封装 IFMP 消息的 IP 包的标题的生存时间字段被设定为 1。另外，利用 IP 标题的目的地址字段中的地址，所有 IFMP 相邻协议消息被发送给限定的广播 IP 目的地址（255.255.255.255）。如图 8a 所示，IFMP 相邻协议消息 300 包括（按照从 MSB 到 LSB 的顺序）下述字段：作为第一个 32 位字的 8 位版本（302），8 位 Op 代码（304），及 16 位检查和（306）；作为第二个 32 位字的发送者实例（308）；作为第三个 32 位字的对等物实例（310）；作为第四个 32 位字的对等物身份（312）；作为第五个 32 位字的对等物下一序列号（314）；地址列表（316），这是一个 32 位字的数目可变域。

在 IFMP 相邻协议消息中，版本字段 302 指定当前使用的 IFMP 协议的版本（当可发展出其它版本时）。Op 代码 304 规定该 IFMP 相邻协议消息的功能。在本实施例中，有四种可能的 Op 代码，即 IFMP 相邻协议消息的功能：SYN（同步消息，Op 代码=0）、SYNACK（同步确认消息，Op 代码=1）、RSTACK（重置确认消息，Op 代码=2）、及 ACK（确认消息，Op 代码=3）。在每个系统节点中，需要一个计时器来周期地产生 SYN、SYNACK 及 ACK 消息。在本实施例中，计时器的周期为 1

秒,但是也可规定其它周期。如果计时器期满,并且系统节点处于 SYNSENT 状态,则系统节点重置计时器,并且发送一个 SYN IFMP 相邻协议消息。如果计时器期满,并且系统节点处于 SYNRCVD 状态,则系统节点重置计时器,并发送一个 SYNACK IFMP 相邻协议消息。如果计时器期满,并且系统节点处于 ESTAB 状态,则系统节点重置计时器,并发送一个 ACK IFMP 相邻协议消息。

检查和 306 是来自封装 IFMP 相邻协议消息的 IP 包的源地址、目的地址及协议字段与 IFMP 相邻协议消息的总长度的二进制反码和的 16 位二进制反码。检查和 306 由系统用于错误控制目的。

在讨论 IFMP 时,“发送者”是发送 IFMP 消息的系统节点,“对等物”是发送者向其发送链路 IFMP 消息的系统节点。

在 SYN、SYNACK 及 ACK IFMP 相邻协议消息中,发送者实例 308 是该链路的发送者的“发送号”。实例号指示一个链路的特定实例,它是一个 32 位非零数,该数在近来是唯一的,并且当链路或系统节点在下行之后复原时,保证将发生改变。于是,每个链路具有其自己的唯一实例号。发送者实例用来检测链路在下行之后,何时复原,或者链路另一端对等物的身份何时发生变化。(以类似的方式把发送者实例 308 用于 TCP 中的初始序列号 (ISN))。对于 RSTACK IFMP 相邻协议消息,发送者实例 308 被设定为导致产生 RSTACK 消息的输入消息的对等物实例字段 310 的值。

在 SYN、SYNACK 及 ACK IFMP 相邻协议消息中,发送者认为对等物实例字段 310 是该链路对等物的当前实例号。如果发送者不知道该链路对等物的当前实例号,对等物实例字段 310 将被设定为 0。在 RSTACK IFMP 相邻协议消息中,对等物实例字段 310 被设定为导致产生 RSTACK 消息的输入消息的发送者实例字段的值。

对于 SYN、SYNACK 及 ACK IFMP 相邻协议消息,对等物身份字段 312 是消息发送者认为在链路另一端的对等物的 IP 地址。发送者获取封装发送者接收的 SYN 或 SYNACK 消息的 IP 标题的源地址字段中的该 IP 地址,并且使用发送者正发送的 IFMP 相邻协议消息的对等物身份字段中的 IP 地址。当发送者不知道链路另一端的对等物的 IP 地址时,对等物身份字段

312 被设定为 0。对于 RSTACK 消息，等对物身份 312 被设定为导致产生 RSTACK 消息的输入 IFMP 相邻协议消息的 IP 标题的源地址字段的 IP 地址值。

对等物下一序列号字段 314 给出在下一个 IFMP 重定向协议消息中发送者希望到达的对等物的序列号字段的值。如果输入 IFMP 相邻协议 ACK 消息中的对等物下一序列号 314 的值大于该序列号的值（来自从接收输入 IFMP 相邻协议 ACK 消息的端口传输出的前一 IFMP 重定向协议消息）加 1，则该链路应被重置。

地址列表字段 316 是一个或多个 IP 地址的列表，这些 IP 地址由 IFMP 相邻协议消息的发送者分配给链路。该列表必须具有至少一个条目，该条目和 IFMP 相邻协议消息的 IP 标题的源地址相同。该列表的内容不被 IFMP 使用，但是可由路由选择协议采用。

图 8c 说明了当接收具有输入 IFMP 相邻协议消息的包时，系统节点的操作。启动系统之后，系统节点接收一个具有输入 IFMP 相邻协议消息的包（步骤 320）。在步骤 322，系统节点确定输入 IFMP 相邻协议消息是否是 RSTACK 消息。如果输入 IFMP 相邻协议消息不是 RSTACK 消息（例如，是 SYN、SYNACK 或 ACK 消息），则系统节点以图 8d 的状态图中说明的方式操作。如果输入 IFMP 相邻协议消息是 RSTACK 消息，则系统节点在步骤 324 检查输入 RSTACK 消息中的发送者实例及源 IP 地址，是否和接收输入 RSTACK 消息的端口的更新对等物校验器（Update Peer Verifier）操作自前一消息存储的值匹配。对于 IFMP 相邻协议，更新对等物校验器操作被定义为存储来自 SYN 或 SYNACK 消息中的发送者实例和源 IP 地址，该 SYN 或 SYNACK 消息在特定端口上接收自对等物。如果在步骤 324 值匹配，则系统节点在步骤 326 确定输入 RSTACK 消息中的对等物实例及对等物身份，是否匹配当前用于所有从接收输入 RSTACK 消息的端口传输出的 SYN、SYNACK 及 ACK 消息的发送者实例及源 IP 地址的值。如果在步骤 326 值匹配，则系统节点在步骤 328 确定系统节点是否处于 SYNSENT 状态。如果系统节点不处于 SYNSENT 状态，则系统节点在步骤 330 重置链路。如果在步骤 324 值不匹配，或者在步骤 326 值不匹配，或者系统节点处于 SYNSENT 状态，则系统节点在步骤 332 丢弃输入

RSTACK 消息，并等待另一个包到达。于是当一个 RSTACK IFMP 相邻协议消息到达系统节点时，系统节点重置链路，如步骤 334、336、338、340 及 342 所示。在步骤 334，系统节点为链路产生一个新的实例号。随后系统节点在步骤 336 删除对等物校验器（即，把对等物的发送者实例及源 IP 地址的存储值设定为 0）。在步骤 338，系统节点把序列号及对等物下一序列号设定为 0。随后系统节点在步骤 340 发送 SYN IFMP 相邻协议消息，并在步骤 342 进入 SYNSENT 状态。随后系统节点接收供处理的另一个包。

图 8d 是说明当输入 IFMP 相邻协议消息不是 RSTACK 消息时，发送者系统节点的操作的状态图。对于图 8d 的下述说明，条件“%B”被定义为：输入消息中的发送者实例和源 IP 地址和，由对接收输入 IFMP 相邻协议消息的端口的更新对等物校验器操作存储的前一消息的值匹配。图 8d 中的条件“%C”被定义为：输入消息中的对等物实例及对等物身份，和当前用于所有从接收输入 IFMP 相邻协议消息的端口传输出的 SYN、SYNACK 及 ACK 消息的发送者实例及源 IP 地址的值匹配。在图 8d 中，条件“A”表示发送者系统节点接收一个输入 SYNACK IFMP 相邻协议消息，并且满足条件 %C；条件“B”表示发送者系统节点接收一个输入 SYNACK IFMP 相邻协议消息，并且不满足条件 %C；条件“C”表示发送者系统节点接收一个输入 ACK IFMP 相邻协议消息，并且条件 %B 及 %C 都被满足；条件“D”表示发送者系统节点接收一个输入 ACK IFMP 相邻协议消息，并且条件 %B 及 %C 都不被满足。

如果发送者处于 SYNSENT 状态 350，并且从链路另一端的对等物接收一个输入 SYN IFMP 相邻协议消息，则发送者执行更新对等物校验器操作，并向对等物发送一个 SYNACK IFMP 相邻协议消息（如步骤 352 所示）。随后发送者从 SYNSENT 状态 350 转到 SYNRCVD 状态 354。如果发送者在 SYNRCVD 状态 354 时接收一个输入 SYN IFMP 相邻协议消息，则发送者在步骤 352 执行更新对等物校验器操作，并向对等物发送一个 SYNACK IFMP 相邻协议消息，但是仍保持在 SYNRCVD 状态 354。如果发送者处于 SYNRCVD 状态 354，并且条件 B 或 D 被满足，则发送者向对等物发送一个 RSTACK IFMP 相邻协议消息（如步骤 356 所示），并保留

在 SYNRCVD 状态 354。如果发送者处于 SYNRCVD 状态 354，并且条件 C 被满足，则发送者向对等物发送一个 ACK IFMP 相邻协议消息（如步骤 358 所示），并转到 ESTAB 状态 360。如果发送者处于 SYNRCVD 状态 354，并且条件 A 被满足，则发送者执行更新对等物校验器操作，并向对等物发送一个 ACK IFMP 相邻协议消息（如步骤 362 所示），并转到 ESTAB 状态 360。如果发送者接收 SYN 或 SYNACK IFMP 相邻协议消息，或者如果条件 C 被满足，则发送者处于并保持 ESTAB 状态 360。如果条件 D 被满足，同时发送者处于 ESTAB 状态 360，则发送者保持在 ESTAB 状态 360，并发送一个 RSTACK IFMP 相邻协议消息（如步骤 356）所示。当处于 SYNSENT 状态 350 时，如果发送者接收一个 ACK IFMP 相邻协议消息，或者条件 B 被满足，则发送者保持在 SYNSENT 状态 350，并发送一个 RSTACK IFMP 相邻协议消息（步骤 356）。如果当发送者处于 SYNSENT 状态 350 时，条件 A 被满足，则发送者执行更新对等物校验器操作，发送一个 ACK IFMP 相邻协议消息（步骤 362）并进入 ESTAB 状态 360。

如上所述，在系统使用 IFMP 相邻协议识别链路另一端的其它系统节点，及链路两端实现状态同步之后，IFMP 重定向协议用来跨越链路发送重定向消息。通过当前没有实现状态同步的链路接收的任何 IFMP 重定向消息必须被丢弃。

图 9a 说明了类属 IFMP 重定向协议消息 380 的结构。和所有的 IFMP 相邻协议消息一样，所有的 IFMP 重定向协议消息也被封装在一个 IP 包中。图 8b 说明了具有可变长度数据字段的类属 IP 包（为当前版本 IPv4），一个 IFMP 重定向协议消息可被封装在该可变长度数据字段中。作为 IP 包含有 IFMP 消息的指示，封装 IP 包的 IP 标题中的协议字段必须含有十进制值 101。封装 IFMP 消息的 IP 包的标题中的生存时间字段被设定为 1。利用 IP 标题的目的地址字段中的 IP 地址，一个 IFMP 重定向协议消息被发送给链路另一端的对等物的 IP 地址。如图 9a 所示，IFMP 重定向协议消息 380 包括（按照从 MSB 到 LSB 的顺序）下述字段：作为第一个 32 位字的 8 位版本（382），8 位 Op 代码（384），及 16 位检查和（386）；作为第二个 32 位字的发送者实例（388）；作为第三个 32 位字的对等物

实例 ( 390 ) ; 作为第四个 32 位字的序列号 ( 392 ) ; 及消息体 ( 394 ) , 这是一个 32 位字的数目可变域。

在 IFMP 重定向协议消息中, 版本字段 382 指定当前使用的 IFMP 协议的版本 ( 当可发展出其它版本时 ) 。 Op 代码 384 规定该 IFMP 重定向协议消息的功能。在本实施例中, 有五种可能的 Op 代码, 即 IFMP 重定向协议消息的功能: REDIRECT ( 重定向流消息, Op 代码=4 ) 、 RECLAIM ( 回收标记消息, Op 代码=5 ) 、 RECLAIM ACK ( 回收标记确认消息, Op 代码=6 ) 、 LABEL RANGE ( 标记范围消息, Op 代码=7 ) 及 ERROR ( 错误消息, Op 代码=8 ) 。

检查和 386 是来自封装 IFMP 重定向协议消息的 IP 包的源地址、目的地址及协议字段与 IFMP 重定向协议消息的总长度的二进制反码和的 16 位二进制反码。检查和 386 由系统用于错误控制目的。

在 IFMP 重定向协议消息中, 如同从 IFMP 相邻协议中得到的一样, 发送者实例 388 是该链路的发送者的“发送号”。在 IFMP 重定向协议中, 发送者认为对等物实例字段 3910 是该链路对等物的当前实例号, 如同从 IFMP 相邻协议中得到的一样。

序列号字段 392 使接收 IFMP 重定向协议消息的系统节点可以顺序处理 IFMP 重定向协议消息。对跨越链路发送的每个 IFMP 重定向协议消息, 序列号 392 加 1, 其模数为  $2^{32}$ 。重置链路时, IFMP 相邻协议把序列号设为 0。

消息体字段 316 含有一个或多个 IFMP 重定向协议消息元的列表。由于 Op 代码字段 384 应用于整个 IFMP 重定向消息, 因此该列表中的所有消息元具有相同的消息类型。单个包中包括的消息元的数目绝对不能使该 IFMP 重定向协议消息的总的大小超过基础数据链路的最大传输单元 ( MTU ) 的大小。一个单独的消息元用于标记范围或错误 IFMP 重定向消息。

图 9b 说明了当接收一个 IFMP 重定向协议消息时, 系统节点的操作。启动后, 系统节点在步骤 400 接收一个封装 IFMP 重定向协议消息的包。在步骤 402, 系统节点检查 IFMP 相邻协议消息当前是否已实现了链路的状态同步。如果未实现状态同步, 则系统节点丢弃封装接收的 IFMP 重定

向协议消息的包（步骤 404）。如果达到状态同步，则系统节点在步骤 406 检查来自 IP 标题的源 IP 地址，IFMP 重定向协议消息包的发送者实例 388，及对等物实例 390。如果系统节点在步骤 408 确定发送者实例 388 和输入 IFMP 重定向协议消息的源 IP 地址字段，不和对于接收输入 IFMP 重定向协议消息的端口的更新对等物校验器操作所存储的值匹配，则系统节点丢弃输入 IFMP 重定向协议消息包（步骤 404）。如果在步骤 408 值匹配，则系统节点在步骤 410 确定对等物实例字段 390 是否和 IFMP 相邻协议的发送者实例的当前值匹配。如果在步骤 408 值不匹配，则系统节点丢弃该包（步骤 404）。但是，如果在步骤 408 中值匹配，则系统节点根据需继续处理接收的 IFMP 重定向协议消息（步骤 412）。

如上概述，一个 IFMP 重定向协议消息可是一个 REDIRECT 消息，该消息用于指导一个相邻节点把一个或多个标记加入属于一个或多个特定流的包中持续一特定时间段。从下游节点接收 REDIRECT 消息的系统节点决定是否接收该 REDIRECT 消息做出的重定向请求，并重定向一个流。不以正式方式确认该 REDIRECT 消息。相反，指定流的具有附加标记的包的实际重定向表明系统节点接受了 REDIRECT 消息做出的重定向请求。REDIRECT 消息的消息体 394 中的每个 REDIRECT 消息元的结构如图 9c 所示。按照从 MSB 到 LSB 的顺序，REDIRECT 消息元 420 包括作为第一个 32 位字的 8 位流类型字段 422，8 位流 ID 长度字段 424，及 16 位生存时间字段 426；作为第二个 32 位字的 32 位标记字段 428；及流标识符 430，这是一个 32 位字的整数倍域。流类型字段 422 指定包含在流标识符字段 430 中的流标识符的流类型，流 ID 长度字段 424 规定 32 位字的整数倍域中的流标识符 430 的长度。生存时间字段 426 规定重定向有效的时间长度（秒）。如上概述，在生存时间字段 426 中规定的时间周期期满之后，流标识符和标记之间的联系应被丢弃。标记字段 428 含有用于指定流的标记，标记的格式取决于物理链路的类型，IFMP 重定向协议消息越过该物理链路被发送。流标识符字段 430 规定标记字段 428 中规定的标记应与之相关的流。

在 IFMP 重定向协议消息元中，流类型 0 的流类型=0，流 ID 长度=0；流类型 1 的流类型=1，流 ID 长度=4；流类型 2 的流类型=2，流 ID 长度



=3.

上面详细说明了发送者及对等物进行的 REDIRECT 消息操作。另外，REDIRECT 消息元的其它特征包括标记管理及错误控制。如果 REDIRECT 消息元 420 的标记字段 428 中的标记在越过相应链路能够被处理的标记范围之外，则一个 LABEL RANGE 消息会返回给该 REDIRECT 消息元的发送者。该 LABEL RANGE 消息向发送者通知可越过该链路发送的标记范围。如果系统节点接收指定一个已被重定向流的 REDIRECT 消息元，则该系统节点对照为该重定向流存储的标记来检验接收的 REDIRECT 消息元中的标记字段。如果标记匹配，则系统节点把该重定向流的生存时间重置为接收的 REDIRECT 消息元的生存时间字段 426 中包含的生存时间。如果标记不匹配，则系统节点忽略接收的 REDIRECT 消息元，并且该流返回缺省状态。如果系统节点在 REDIRECT 消息元中任一字段中检测到错误，则丢弃该特定错误 REDIRECT 消息元。但是，相同 IFMP REDIRECT 消息体中的任何其它无错 REDIRECT 消息元不被丢弃或受到任何影响。如果系统节点不了解接收的 IFMP 消息中的 IFMP 协议的版本，则系统节点向发送该错误 REDIRECT 消息元的相邻节点返回一个 ERROR 消息。同样，如果系统节点不了解接收的 IFMP 消息中任一 REDIRECT 消息元中的流类型，则系统节点为每个不了解的流类型向发送每个特定 REDIRECT 消息元的相邻节点发送一个 ERROR 消息。

如上所述，IFMP 重定向协议消息可以是 RECLAIM 消息，该消息用于指导相邻节点使一个或多个流和当前与这些流结合的标记分离，并释放这些标记以便重新使用。于是从下游节点接收 RECLAIM 消息元的系统节点释放该标记，并向该下游节点发送一个 RECLAIM ACK 消息元作为对 RECLAIM 消息的正式确认。RECLAIM 消息的消息体 394 中的每个 RECLAIM 消息元的结构如图 9d 所示。按照从 MSB 到 LSB 的顺序，RECLAIM 消息元 432 包括第一个 32 位字中的 8 位流类型字段 434，8 位流 ID 长度字段 436，及 16 位保留字段 438；作为第二个 32 位字的 32 位标记字段 440；及流标识符 442，这是一个 32 位字的整数倍域。流类型字段 434 指定包含在流标识符字段 442 中的流标识符的流类型，流 ID 长度字段 436 规定 32 位字的整数倍域中的流标识符 442 的长度。本实施例中，保

留字段 438 未使用，由发送该 RECLAIM 消息元的系统节点置为零，并被接收该 RECLAIM 消息元的系统节点忽略。标记字段 440 含有要释放的标记。流标识符字段 442 识别标记字段 440 中的指定标记应与之分离的流。每个 RECLAIM 消息元施加于单个流及单个标记。在系统节点接收一个 RECLAIM 消息元之后，使一个流及一个标记分开，使该流返回缺省转送状态，并释放该标记，该系统节点必须发布一个 RECALIM ACK 消息元。RECLAIM ACK 消息元可被归到一个或多个 RECALIM ACK 消息中，并被返回给发送者作为完成回收操作的确认。

另外，RECLAIM 消息元的其它特征包括标记管理及错误控制。如果系统节点接收指定一个未知流的 RECLAIM 消息元，则该系统节点向该 RECLAIM 消息元的发送者返回一个带有相同标记 440 及流标识符 442 字段的 RECLAIM ACK 消息元。如果系统节点接收指示一个已知流的 RECLAIM 消息元，但是标记字段 440 中的标记当前未与该流捆绑，则该系统节点解开该流，并使该流返回缺省转送状态，并且向该 RECLAIM 消息元的发送者发布一个 RECLAIM ACK 消息元，该消息元含有先前与该流捆绑实际标记。如果系统节点在 RECLAIM 消息元中任一字段中检测到错误，则丢弃该特定错误 RECLAIM 消息元。但是，相同 IFMP RECLAIM 消息体中的任何其它无错 RECLAIM 消息元不被丢弃或受到任何影响。如果系统节点不了解接收的 IFMP 消息中的 IFMP 协议的版本，则系统节点向发送该错误 RECLAIM 消息元的相邻节点返回一个 ERROR 消息。同样，如果系统节点不了解接收的 IFMP 消息中任一 RECLAIM 消息元中的流类型，则系统节点为每个不了解的流类型向发送每个特定 RECALIM 消息元的相邻节点发送一个 ERROR 消息。

如上所述，IFMP 重定向协议消息可以是一个 RECLAIM ACK 消息，该消息用于确认成功释放一个或多个回收标记。在从下游节点接收 RECLAIM 消息元的系统节点释放标记之后，向发送该 RECALIM 消息元的节点发送一个 RECLAIM ACK 消息元。如果可能，应在所有在使用指定释放标记的链路上传输的数据队列已被发送后，才发送每个 RECLAIM ACK 消息元。RECLAIM ACK 消息的消息体 394 中的每个 RECLAIM ACK 消息元的结构如图 9e 所示。按照从 MSB 到 LSB 的顺序，RECLAIM

ACK消息元444包括第一个32位字中的8位流类型字段446，8位流ID长度字段448，及16位保留字段450；作为第二个32位字的32位标记字段452；及流标识符454，这是一个32位字的整数倍域。流类型字段446指定包含在流标识符字段454中的流标识符的流类型，流ID长度字段448规定32位字的整数倍域中的流标识符454的长度。本实施例中，保留字段450未使用，由发送该RECLAIM ACK消息元的系统节点置为零，并被接收该RECLAIM ACK消息元的系统节点忽略。标记字段452含有从流标识符454指定的流中释放的标记。流标识符字段454含有来自请求释放标记字段452中指定标记的RECALIM消息元的流标识符。

RECLAIM ACK消息元的其它特征包括标记管理及错误控制。如果系统节点接收指定一个流的RECLAIM ACK消息元，而对该流没有发布过RECLAIM消息元，则该RECLAIM ACK消息元被忽略。如果系统节点接收一个RECLAIM ACK消息元，而该消息元指定的标记不同于该流的RECLAIM消息中发送的标记，则系统节点处理接收的RECALIM ACK消息元，好象对RECLAIM消息中发送的标记的回收操作成功了一样。如果系统节点在RECLAIM ACK消息元中任一字段中检测到错误，则丢弃该特定错误RECLAIM ACK消息元。但是，相同IFMP RECLAIM ACK消息体中的任何其它无错RECLAIM消息元不被丢弃或受到任何影响。如果系统节点不了解接收的IFMP消息中的IFMP协议的版本，则系统节点向发送该错误RECLAIM ACK消息元的相邻节点返回一个ERROR消息。同样，如果系统节点不了解接收的IFMP消息中任一RECLAIM ACK消息元中的流类型，则系统节点为每个不了解的流类型向发送每个特定RECALIM ACK消息元的相邻节点发送一个ERROR消息。

如上所述，IFMP重定向协议消息可以是LABEL RANGE消息，如果在REDIRECT消息元的一个或多个中请求的标记在接收该REDIRECT消息的系统节点能够处理的标记范围之外，则该消息用于响应REDIRECT消息。LABEL RANGE消息向该REDIRECT消息的发送者通知在该链路上能被处理的标记范围。在LABEL RANGE消息中使用单个LABEL RANGE消息元。LABEL RANGE消息的消息体394中的LABEL RANGE消息元的结构如图9f所示。LABEL RANGE消息元456包括作

为第一个 32 位字的最小标记字段 458，及作为第二个 32 位字的最大标记字段 460。最小标记字段 458 及最大标记字段 460 分别是可在跨越特定链路的 IFMP 重定向协议消息中规定的标记的最小及最大值。只有在最小到最大标记（含）之间的标记值才可在跨越该链路的 IFMP 重定向协议消息中被指定。

如上所述，IFMP 重定向协议消息可以是 ERROR 消息，可发送该消息响应任意 IFMP 重定向协议消息。在 ERROR 消息中使用单个 ERROR 消息元。ERROR 消息的消息体 394 中的 ERROR 消息元的结构如图 9g 所示。按照从 MSB 到 LSB 的顺序，ERROR 消息元 462 包括 32 位字的 8 位错误代码字段 464 及 24 位参数字段 466。错误代码字段 464 规定发生了哪种类型的错误。每个 ERROR 消息可指定一个参数。如果系统节点在 IFMP 重定向协议消息元的任一字段中检测到错误，则丢弃该特定错误消息元，并发布一个 ERROR 消息。如果系统节点不能处理或不了解接收的 IFMP 消息中的 IFMP 协议的具体版本，则系统节点发送一个错误消息，该错误消息具有设定为值 1 的错误代码 464，及提供发送者能够了解或处理的 IFMP 协议最新版本的参数 466。同样，如果系统节点不了解接收的任一 IFMP 重定向协议消息元中导致该错误的流类型，则系统节点发送一个 ERROR 消息，该消息具有设定为值 2 的错误代码 464，及提供导致该错误的流类型的参数 466。

## 2. ATM 数据链路上的流标注传输

本发明使用 ATM 数据链路在系统节点之间传输 IP 包。对在 ATM 数据链路上传输的包进行流标注，并且如上所述根据该流的类型对包进行不同的封装。使用流分类，本发明可根据流的类型，对不同类型的流进行不同的处理（第 2 层路由选择或第 3 层交换）。另外，每个流类型还指定一个在该类型流重定向之后要使用的封装。在本实施例中，系统使用用于 ATM 数据链路的封装，在此将进行详细说明。当然，可为和本发明一起使用的不同硬件交换引擎的不同数据链路技术指定用于每种流类型的封装。

一个特定包流与一个特定 ATM 标记相关。图 10a 说明了本系统中 ATM 标记的 32 位标记字段的格式。如前所述，标记是虚拟路径标识符和虚拟通道标识符（VPI/VCI），假定虚拟通道单向。按照从 MSB 到 LSB

的顺序，图 10a 所示的 ATM 标记字段包括 4 位保留字段 472，12 位 VPI 字段 474 及 16 位 VCI 字段 476。在本实施例中，保留字段 472 由发送者系统节点设定为 0，并被接收该 ATM 标记的系统节点忽略。对于不支持全 12 位 VPI 的链路，VPI 字段 474 中的未使用位是字段 474 中的 MSBs（最高位），并被设定为 0。同样，对于不支持全 16 位 VCI 的链路，VCI 字段 476 中的未使用位是字段 476 中的 MSBs（最高位），并被设定为 0。

对于未被重定向流中的任意包，系统节点为 IP 包使用缺省封装。如果系统节点决定一种特定的流类型将被重定向，则系统节点使用每种流类型所特有的封装。一旦重定向一个流，则系统节点把用于重定向流的封装从通常使用的缺省封装改变为特定封装。系统节点根据被重定向的流类型使用一个不同类型的封装，而不是使用用于缺省转送链路上 IP 包的缺省封装。ATM 封装 IP 包可以是一个自身封装一个发往和/或来自运行系统软件子集的主计算机/服务器/工作站、基本交换单元或交换网关单元。

如上所述，在本发明的一个实施例中，规定了三种流类型：流类型 0、流类型 1 及流类型 2。流类型 0 用于把 IP 包从缺省封装改变为新的封装。流类型 1 用于在运行于站上的应用之间携带数据的包。流类型 2 用于在站之间携带数据的包，而没有识别在站上运行的应用。

本发明中，用于 ATM 数据链路上的 IP 包的缺省封装是图 10b 所示的逻辑链路控制/子网连接点（Logical Link Control/SubNetwork Attachment Point（LLC/SNAP））封装。图 10b 说明了缺省封装的 IP 包 480。缺省封装把一个 LLC/SNAP 标题加到封装在 ATM 适应层（Adaptation Layer）类型 5 公用部分收敛子层协议数据单元（Common Part Convergence Sublayer Protocol Data Unit（AAL-5 CPCS-PDU））的负载中的 IP 包的前头。按照从 MSB 到 LSB 的顺序，缺省封装 IP 包 480 包括 LLC/SNAP 标题（第一个 32 位字中的 24 位 LLC 字段 482，跟在其后的 SNAP 标题 484 的 8 位部分，及 SNAP 标题 484 的剩余 32 位字部分），IP 包 486（其长度为 32 位字的整数倍），填充字段 488 及 AAL-5 CPCS-PDU 尾部字段 490。填充字段 488 的范围为 0-47 个八位位组，尾部字段 490 为 8 个八位位组（四个 32 位字）。使用缺省封装的 IP 包 486 的 MOTU 为 1500 个八位位组。使用缺省封装的包被发送给预定的 VPI/VCI

( VPI=0, VCI=15, 这是根据特定实施例的缺省 VPI/VCI, 即包在缺省虚拟通道上被转送)。

流类型 0 封装用于把 IP 包从缺省封装改变为新的封装。图 10c 说明了流类型 0 封装 IP 包 492。使用流类型 0 的 IP 包被直接封装在 AAL-5 CPCS-PDU 的负载中, 没有前置 LLC/SANP 标题。按照从 MSB 到 LSB 的顺序, 流类型 0 封装 IP 包 492 包括 IP 包 494 (其长度为 32 位字的整数倍), 填充字段 496 及 AAL-5 CPCS-PDU 尾部字段 498。填充字段 496 的范围从 0-47 个八位位组, 尾部字段 498 为 8 个八位位组(四个 32 位字)。使用流类型 0 封装的 IP 包 494 的 MOTU 为 1500 个八位位组。属于从缺省虚拟通道重定向流的包使用流类型 0 封装, 并且被发送给 IP 包 494 中封装的 IFMP REDIRECT 消息元的标记字段中指定的 VPI/VCI (封装在 IP 包 494 中的 IFMP REDIRECT 消息元在流类型 0 封装中被发送)。

缺省及流类型 0 封装不涉及从封装的 IP 包中除去任何字段。但是, 流类型 1 及流类型 2 “封装”涉及从 IP 包中除去某些字段。当这些字段被除去时, 发布 REDIRECT 消息的系统节点存储除去的字段, 并把这些字段与 ATM 标记中指定的 ATM 虚拟通道联系起来。于是, 在目的地, 使用输入的 ATM 标记访问存储的字段, 就可以重构一个完整的 IP 包。

流类型 1 用于在站上运行的应用之间携带数据的包。图 10d 说明了一个流类型 1 封装的 IP 包。使用流类型 1 封装的 IP 包实质上被分解, 分解 IP 包的选定部分被直接封装在 AAL-5 CPCS-PDU 负载中, 没有前置 LLC/SNAP 标题。按照从 MSB 到 LSB 的顺序, 流类型 1 封装的 IP 包 500 包括作为第一个 32 位字, 来自分解的 IP 包的 IP 标题的 16 位总长度字段 502 及 16 位标识字段 504。总长度字段 502 的值不改变, 保留分解前的 IP 包总长度值。流类型 1 封装的 IP 包 500 还包括作为第二个 32 位字的 8 位标志字段 506, 12 位分段偏移字段 508, 及来自分解 IP 包的 IP 标题的 16 位检查和字段 510。如果 TTL 字段已设定为零, 则检查和字段 510 的发送值是对整个 IP 标题计算得到的检查和值。IP 标题中的版本、 IHL、 TOS、 TTL、 协议、 源地址及目的地址字段不作为流类型 1 封装的 IP 包 500 的一部分被传输。另外, 紧跟在 IP 标题(由 IHL 确定)后的前 4 个八位位组不作为流类型 1 封装的 IP 包 500 的一部分被传输。举例来说, 这前

4 个八位位组对应于 TCP 及 UDP 数据报的源端口及目的端口。源端口及目的端口字段识别站上运行的应用。此外，流类型 1 封装的 IP 包 500 包括数据 512。数据字段 512 后是填充字段 514 及 AAL-5 CPCS-PDU 尾部字段 516。填充字段 514 的范围从 0 到 47 个八位位组，尾部字段 516 为 8 个八位位组（四个 32 位字）。使用流类型 1 封装的 IP 包的 MOTU 为 1484 个八位位组。属于重定向流的，使用流类型 1 封装的包被发送给分解 IP 包中封装的相应流类型 1IFMP REDIRECT 消息元的标记字段中指定的 VPI/VCI（标记字段对应于 TCP 或 UDP 消息中的源和目的端口字段）。

流类型 2 用于在站之间携带数据的包，而不考虑什么样的应用在站上运行。图 10e 说明了流类型 2 封装的 IP 包。使用流类型 2 封装的 IP 包实质上被分解，分解 IP 包的选定部分被直接封装在 AAL-5 CPCS-PDU 负载中，不带前置 LLC/SNAP 标题。按照从 MSB 到 LSB 的顺序，流类型 2 封装的 IP 包 520 包括作为第一个 32 位字，来自分解的 IP 包的 IP 标题的 16 位总长度字段 522 及 16 位标识字段 524。总长度字段 522 的值不改变，保留分解前的 IP 包总长度值。流类型 2 封装的 IP 包 520 还包括作为第二个 32 位字的 8 位标志字段 526，12 位分段偏移字段 528，及来自分解 IP 包的 IP 标题的 16 位检查和字段 530。如果 TTL 字段已被设定为零，则检查和字段 530 的发送值是对整个 IP 标题计算得到的检查和值。IP 标题中的版本、IHL、TOS、TTL、协议、源地址及目的地址字段不作为流类型 2 封装的 IP 包 520 的一部分被传输。和流类型 1 封装不同另外，紧跟在 IP 标题（由 IHL 确定）后的前 4 个八位位组作为流类型 2 封装的 IP 包 520 的一部分被传输。此外，流类型 2 封装的 IP 包 520 包括数据 532。数据字段 532 后是填充字段 534 及 AAL-5 CPCS-PDU 尾部字段 536。填充字段 534 的范围从 0 到 47 个八位位组，尾部字段 536 为 8 个八位位组（四个 32 位字）。使用流类型 2 封装的 IP 包的 MOTU 为 1484 个八位位组。属于重定向流的，使用流类型 2 封装的包被发送给分解 IP 包中封装的相应流类型 2IFMP REDIRECT 消息元的标记字段中指定的 VPI/VCI。

如上所述，对于流类型 0、流类型 1 及流类型 2 封装，接受下游节点发送的 IFMP REDIRECT 消息的系统节点存储除去的字段，并且把这些字段与 ATM 标记指定的 ATM 虚拟通道联系起来，以便实现重定向包存取信

息的高速缓存。

## B. GSMP

系统软件还利用 GSM 协议通过 ATM 链路在交换控制器和系统的基本交换单元的 ATM 硬件交换引擎之间建立通信，从而能够实现第 2 层交换（可能时）及第 3 层 IP 路由选择（必需时）。特别地，GSMP（控制 ATM 交换器的通用非对称协议）运行于初始时跨越交换控制器和 ATM 交换器之间的 ATM 链路建立的虚拟通道上。单个交换控制器可在分开的虚拟通道上使用多个 GSMP 实例来控制多个 ATM 交换器。GSMP 也包括一个 GSMP 相邻协议。GSMP 相邻协议用来跨越交换控制器和 ATM 交换器之间的 ATM 链路使状态同步，以便发现链路另一端的实体的身份，并检测该实体身份的改变。

GSMP 允许交换控制器跨越 ATM 交换器建立并释放连接，在点对多点连接上加入及删除叶，管理交换端口，请求配置信息及请求统计。GSMP 还允许 ATM 交换器通知交换控制器诸如链路正在下行之类的事件。

如前所述，GSMP 是一种主-从协议。交换控制器向交换器发布请求消息。每个请求消息指示是否需要来自交换器的响应，并含有事务标识符，使得响应能够和特定请求相联系。交换器用指示成功或失败的响应消息应答。在本实施例中，GSMP 有五类消息：连接管理、端口管理、统计、配置及事件。除了事件消息类别外，其它四种类别都是请求-响应消息类别，每种具有一个请求消息格式及一个成功响应格式。除非另外指出，除指示失败本质的代码字段之外，失败响应消息和导致该失败的请求消息相同。除了这四种请求-响应消息类别外，GSMP 还包括一种事件消息类别，事件消息使得交换器能够产生异步事件消息，从而向交换控制器通知异步事件。当事件消息未被交换控制器确认时，事件消息具有单一的格式。本实施例中，具有多种不同的消息类型，即 GSMP 消息功能。五种 GSMP 消息类别的每一种，除了端口管理外，都具有许多不同的消息类型。

同样，GSMP 包括分配了特定消息类型的 GSMP 相邻协议消息。GSMP 相邻协议消息用于跨越 ATM 链路建立同步状态，并保持同步交换。除 GSMP 相邻协议消息之外，在 GSMP 相邻协议已经实现状态同步之前，其它 GSMP 消息不能越过 ATM 链路被发送。在当前没有实现状态同步的



ATM链路上接收的所有GSMP消息被丢弃。

本发明中，GSMP包的长度可变，并以和上面关于图10b说明的ATM数据链路上IP包的缺省封装相类似的方式，直接封装在具有前置LLC/SNAP标题的AAL-5 CPCS-PDU中。图11a说明了封装GSMP包540。缺省封装把一个LLC/SNAP标题加到封装在AAL-5 CPCS-PDU的负载中的GSMP包的前头。按照从MSB到LSB的顺序，缺省封装的GSMP包540包括LLC/SNAP标题（第一个32位字中的24位LLC字段542，跟在其后的SNAP标题544的8位部分，及SNAP标题544的剩余32位字部分），GSMP消息546（其长度为32位字的整数倍），填充字段548及AAL-5 CPCS-PDU尾部字段550。填充字段548的范围为0-47个八位位组，尾部字段550为8个八位位组（四个32位字）。使用缺省封装的GSMP消息546的MOTU为1500个八位位组。使用缺省封装的包被发送给缺省的VPI/VCI，即缺省虚拟通道。

图11b说明了可包含在图11a的封装GSMP包540的GSMP消息字段546中的GSMP相邻协议消息的结构。如图11b所示，GSMP相邻协议消息552包括（按照从MSB到LSB的顺序）下述字段：作为第一个32位字的8位版本字段554、8位消息类型字段556、8位结果字段558及8位代码字段560；作为第二个32位字的发送者实例字段562；作为第三个32位字的发送者端口字段564；下一个48位的发送者名字字段566；下一个48位的接收者名字字段568；下一个32位的接收者端口字段570；及下一个32位的接收者实例字段572。讨论GSMP消息时，“发送者”是发送GSMP消息的实体，“对等物”是发送者通过ATM链路向其发送GSMP消息的实体。所述实体可以是交换控制器或ATM交换器。

在GSMP相邻协议消息552中，版本字段554规定当前使用的GSMP协议的版本（当可发展出其它版本时）。消息类型字段556被设定为特定值（消息类型=96）以规定GSMP消息为GSMP相邻协议消息。结果字段556不用于GSMP相邻协议消息，由发送者实体设定为零，并被接收该GSMP相邻消息的实体忽略。

GSMP相邻协议消息的代码字段560规定消息的功能。在本实施例中，代码字段有四种可能值，即GSMP相邻协议消息的功能：SYN（同

步消息, 代码=0)、SYNACK(同步确认消息, 代码=1)、RSTACK(重置确认消息, 代码=2)、及ACK(确认消息, 代码=3)。在每个实体中, 为了周期地产生SYN、SYNACK及ACK消息, 需要一个计时器。对GSMP相邻协议, 对特定链路实体具有三种可能的状态: SYNSENT(同步消息发送)、SYNRCVD(同步消息接收)、ESTAB(同步建立)。在系统可发送非GSMP相邻协议消息的GSMP消息之前, 要求链路两端的同步(当一个实体达到链路的ESTAB状态时)。本实施例中, 计时器的周期为1秒, 但是也可规定其它周期。如果计时器期满, 并且发送者实体处于SYNSENT状态, 则发送者实体重置计时器, 并且发送一个SYN GSMP相邻协议消息。如果计时器期满, 并且发送者实体处于SYNRCVD状态, 则发送者实体重置计时器, 并发送一个SYNACK GSMP相邻协议消息。如果计时器期满, 并且发送者实体处于ESTAB状态, 则发送者实体重置计时器, 并发送一个ACK GSMP相邻协议消息。

在SYN、SYNACK及ACK GSMP相邻协议消息中, 发送者实例562是该链路的发送者实体的实例号。实例号指示链路的一个特定实例, 它是一个32位非零数, 该数在近来是唯一的, 并且当链路在下行之后复原时, 或链路另一端的实体身份变化时, 肯定会发生改变。于是, 每个链路具有其自己唯一的实例号。发送者实例562用来检测链路在下行之后, 何时复原, 或者ATM链路另一端的实体的身份何时改变。对于RSTACK GSMP相邻协议消息, 发送者实例562被设定为导致产生RSTACK消息的输入GSMP相邻协议消息的接收者实例字段572的值。

在SYN、SYNACK及ACK GSMP相邻协议消息中, 发送者端口字段是通过它发送GSMP消息的链路的本地端口号。如上所述, 端口号是本地分配的32位值。对于RSTACK GSMP相邻协议消息, 发送者端口字段564被设定为导致产生RSTACK消息的输入GSMP相邻协议消息的接收者端口字段570的值。

对于SYN、SYNACK及ACK GSMP相邻协议消息, 发送者名字字段是发送者实体的名字。48位发送者名字566在基本交换单元的操作语境中是唯一的。例如, IEEE 802 MAC地址可用作发送者名字字段。对于RSTACK GSMP相邻协议消息, 发送者名字字段566被设定为导致产生

RSTACK 消息的输入 GSMP 相邻协议消息的接收者名字字段 568 的值。

对于 SYN、SYNACK 及 ACK GSMP 相邻协议,接收者名字字段 568 是发送者实体认为其在 ATM 链路另一端的实体的名字。如果发送者不知道该实体的名字,则接收者名字字段 568 将被设定为 0。对于 RSTACK GSMP 相邻协议消息,接收者名字字段 568 被设定为导致产生 RSTACK 消息的输入 GSMP 相邻协议消息的发送者名字字段 566 的值。

在 SYN、SYNACK 及 ACK GSMP 相邻协议消息中,发送者认为接收者端口字段 570 是已由链路另一端的实体为链路分配的本地端口号。如果发送者实体不知道该实体的端口号,则接收者端口字段 570 被设定为 0。对于 RSTACK GSMP 相邻协议消息,接收者端口字段 570 被设定为导致产生该 RSTACK 消息的输入 GSMP 相邻协议消息的发送者端口字段 564 的值。

在 SYN、SYNACK 及 ACK GSMP 相邻协议消息中,发送者实体认为接收者实例 572 是已由链路另一端的实体为链路分配的当前实例号。如果发送者实体不知道链路另一端的当前实例号,则接收者实例字段被设定为 0。对于 RSTACK GSMP 相邻协议消息,接收者实例 572 被设定为导致产生该 RSTACK 消息的输入 GSMP 相邻协议消息的发送者实例字段 562 的值。

图 11c 是说明当接收输入 GSMP 相邻协议消息时,发送者实体操作的简图。启动后,发送者实体在步骤 582 接收 GSMP 相邻协议包。在步骤 584,发送者确定该输入 GSMP 相邻协议消息是否是 RSTACK 消息。如果输入 GSMP 相邻协议消息不是 RSTACK 消息(例如, SYN、SYNACK 或 ACK 消息),则发送者以图 11d 的状态图中说明的方式操作。如果输入 GSMP 相邻协议消息是 RSTACK 消息,则发送者实体在步骤 584 检查输入消息中发送者实例、发送者端口及发送者名字字段是否匹配由更新对等物校验器操作自前一消息存储的值。对于 GSMP 相邻协议,更新对等物校验器操作被定义为:存储从链路另一端的实体接收的 SYN 或 SYNACK 消息中的发送者实例、发送者端口及发送者名字字段。如果在步骤 584 值匹配,则发送者实体在步骤 586 确定输入 RSTACK 消息中的接收者实例、接收者端口及接收者名字字段,是否匹配当前在接收输入 RSTACK 消息端口

的输出 SYN、SYNACK、及 ACK 消息中发送的发送者实例、发送者端口及发送者名字字段的值。如果在步骤 586 值匹配，则系统节点在步骤 588 确定发送者实体是否处于 SYNSENT 状态。如果发送者实体不处于 SYNSENT 状态，则发送者实体在步骤 590 重置链路。如果在步骤 584 值不匹配，或者在步骤 586 值不匹配，或者发送者实体处于 SYNSENT 状态，则发送者实体在步骤 592 丢弃输入的 RSTACK 消息，并等待另一个包到达。于是当一个 RSTACK GSMP 相邻协议消息到达发送者实体时，发送者实体重置链路，如步骤 594、596、598 及 600 所示。在步骤 594，发送者实体为链路产生一个新的实例号。随后发送者实体在步骤 596 删除（即设为零）更新对等物校验器操作先前存储的发送者实例、发送者端口及发送者名字的存储值。随后在步骤 598，发送者实体发送 SYN GSMP 相邻协议消息，并在步骤 600 进入 SYNSENT 状态。随后发送者实体接收供处理的另一个包。

图 11d 是说明当输入 GSMP 相邻协议消息不是 RSTACK 消息时，发送者实体的操作的状态图。对于图 11d 的下述说明，条件“%B”被定义为：输入消息中的发送者实例、发送者端口及发送名字字段和更新对等物校验器操作自前一消息存储的值匹配。图 11d 中的条件“%C”被定义为：输入消息中的接收者实例、接收者端口及接收者名字和当前在输出 SYN、SYNACK 及 ACK 消息中发送的发送者实例、发送者端口及发送者名字的值匹配。在图 11d 中，条件“A”表示发送者实体接收一个输入 SYNACK GSMP 相邻协议消息，并且条件 %C 被满足；条件“B”表示发送者实体接收一个输入 SYNACK GSMP 相邻协议消息，并且条件 %C 不被满足；条件“C”表示发送者实体接收一个输入 ACK GSMP 相邻协议消息，并且条件 %B 及 %C 都被满足；条件“D”表示发送者实体接收一个输入 ACK GSMP 相邻协议消息，并且条件 %B 及 %C 都不满足。

如果发送者实体处于 SYNSENT 状态 602，并且从链路另一端的对等物接收一个输入 SYN GSMP 相邻协议消息，则发送者实体执行更新对等物校验器操作，并向对等物发送一个 SYNACK GSMP 相邻协议消息（如步骤 604 所示）。随后发送者从 SYNSENT 状态 602 转到 SYNRCVD 状态 606。如果发送者在 SYNRCVD 状态 606 时接收一个输入 SYN GSMP 相

邻协议消息，则发送者在步骤 604 执行更新对等物校验器操作，并向对等物发送 SYNACK GSMP 相邻协议消息，但是仍保持在 SYNRCVD 状态 606。如果发送者处于 SYNRCVD 状态 606，并且条件 B 或 D 被满足，则发送者向对等物发送 RSTACK GSMP 相邻协议消息（如步骤 608 所示），并保留在 SYNRCVD 状态 606。如果发送者处于 SYNRCVD 状态 606，并且条件 C 被满足，则发送者向对等物发送 ACK GSMP 相邻协议消息（如步骤 610 所示），并转到 ESTAB 状态 612。如果发送者处于 SYNRCVD 状态 606，并且条件 A 被满足，则发送者执行更新对等物校验器操作，并向对等物发送一个 ACK GSMP 相邻协议消息（如步骤 614 所示），并转到 ESTAB 状态 612。如果发送者接收 SYN 或 SYNACK GSMP 相邻协议消息，或者如果条件 C 被满足，则发送者处于并保持 ESTAB 状态 612。如果条件 D 被满足，同时发送者处于 ESTAB 状态 612，则发送者保持在 ESTAB 状态 612，并发送 RSTACK GSMP 相邻协议消息（如步骤 608）所示。当处于 SYNSENT 状态 602 时，如果发送者接收一个 ACK GSMP 相邻协议消息，或者条件 B 被满足，则发送者保持在 SYNSENT 状态 602，并发送一个 RSTACK GSMP 相邻协议消息（步骤 608）。当发送者处于 SYNSENT 状态 602 时，如果条件 A 被满足，则发送者执行更新对等物校验器操作，发送一个 ACK GSMP 相邻协议消息（步骤 614）并进入 ESTAB 状态 612。

除了 GSMP 相邻协议消息之外，其它类型的 GSMP 消息 546 包括 GSMP 连接管理（CM）消息，该消息是请求-响应消息。在基本交换单元中，交换控制器使用 GSMP CM 消息来跨越 ATM 交换器建立、删除、修改及校验虚拟通道连接。可以不考虑交换端口的状态即发布 GSMP CM 消息，当交换端口向上、向下、或者相反不可用时，可建立或删除连接。连接管理消息包括：加入支线，删除支线，删除树，校验树，删除全部，移动根及移动支线。如前面所述，虚拟通道连接是单向的，并包括一个输入虚拟通道及至少一个输出虚拟通道或支线。即，单点传送虚拟连接具有一个输出支线，多点传送虚拟连接具有两个或更多输出支线。

加入支线消息是用于建立一个虚拟通道连接，或者在现有虚拟通道连接中加入一个附加支线。本实施例中，单点传送和多点传送之间没有区别。

用于特定输入端口、输入 VPI 及输入 VCI 的第一加入支线消息建立单点传送连接。带有相同输入端口、输入 VPI 及输入 VCI 的第二加入支线消息建立单点传送连接，并通过加入另一个输出支线把单点传送转换为多点传送连接。利用更多的加入支线消息，可用同样的方式加入其它输出支线。另外，加入支线消息可用于检查 ATM 交换器中存储的连接状态。GSMP CM 消息中的删除支线消息用于删除虚拟通道连接的单个支线。例如，在具有两条支线的多点传送虚拟通道连接上使用删除支线消息删除一条支线，把多点传送连接转换为单点传送。删除支线消息也可用于通过删除虚拟通道连接中的最后支线来删除连接。另一种 GSMP CM 消息，删除树消息用于通过删除虚拟连接的所有剩余支线来删除整个虚拟连接。校验树消息是用于校验虚拟通道连接上的支线数的 GSMP CM 消息。删除全部消息是用于删除交换输入端口的所有连接的 GSMP CM 消息。移动根消息是用于把整个虚拟连接树从它的当前输入端口、输入 VPI 及输入 VCI 移动到新输入端口、输入 VPI 及输入 VCI 的 GSMP 消息。另一种 GSM CM 消息，移动支线消息用于把虚拟通道连接的单个输出支线从它的当前输出端口、输出 VPI 及输出 VCI 移动到同一虚拟通道连接上的新输出端口、输出 VPI 及输出 VCI。

图 12 说明了用作对加入支线、删除支线、删除树、校验树及删除全部消息的请求及响应的类属 GSMP CM 消息 620 的结构。类属 GSMP CM 消息可包含在图 11a 的封装 GSMP 包 540 的 GSMP 消息字段 546 中。如图 12 所示，类属 GSMP CM 消息 620 包括（按照从 MSB 到 LSB 的顺序）下述字段：8 位版本字段 622、8 位消息类型字段 624、8 位结果字段 626 及 8 位代码字段 628；32 位字事务标识符字段 630；32 位字端口话路号字段 632；32 位字输入端口字段 634；包括设定为零的 4 位字 636，12 位输入 VPI 字段 638 及 16 位输入 VCI 字段 640 的 32 位字输入标记字段；32 位字输出端口字段 642，包括设定为零的 8 位字 644，12 位输出 VPI 字段 646 及 16 位输出 VCI 字段 648 的 32 位字输出标记字段；16 位支线数目字段 650；8 位保留字段 652 及 8 位优先权字段 654。

除了 GSMP 相邻协议消息之外，所有的 GSMP 消息都包括以相同方式使用的版本字段 622、消息类型字段 624、结果字段 626、代码字段 628

及事务标识符字段 630。例如，GSMP 消息中的版本字段 622 规定当前使用的 GSMP 协议的版本（当可发展出其它版本时）。消息类型字段 624 被设定为指定该 GSMP 消息类型的特定值。例如，对 GSMP CM 加入支线消息指定一个特定值用于消息类型字段 624，对其它类型消息指定其它特定值。

对于为请求消息的 GSMP 消息，结果字段 626 指示当输出成功时，对请求消息是否需要响应。请求消息中的结果字段 626 可含有 NoSuccessAck（指示如果输出成功，不需要响应）的值或 AckAll（指示如果输出成功，需要响应）的值。对于某些类型的 GSMP 请求消息，AckAll 是缺省的，结果字段 626 中的 NoSuccessAck 值被忽略。对于为响应消息的 GSMP 消息，结果字段 626 中含有成功（指示该请求成功）的值，或失败（指示该请求未成功）的值。在请求成功完成之前，不发送 GSMP 成功响应消息。GSMP 成功响应消息是返回的带有指示成功的结果字段 626 的相应 GSMP 请求消息的副本。对于不具有成功输出的 GSMP 请求消息，产生一个 GSMP 失败响应消息。GSMP 失败响应消息是返回的带有指示失败的结果字段 626 的相应 GSMP 请求消息的副本。响应 GSMP 请求消息的失败输出，发布 GSMP 失败响应消息的交换器不修改该交换器内的连接状态。

在 GSMP 响应消息中，代码字段 628 提供关于结果的更多信息。例如，GSMP 失败响应消息中的代码字段 628 可含有指定导致失败的错误类型。可确定多种不同的失败或其它类型代码，以便用于代码字段 628。可确定的失败代码的例子包括：特定于特殊消息类型的失败，其它失败代码未覆盖的指定原因，无效请求消息，未在该交换器中实现的指定请求消息，无效端口话路号，至少一个指定端口不存在，至少一个指定端口下行，至少一个指定 VPI/VCI 在至少一个指定端口上超出范围，指定连接不存在，指定输出支线不存在，已经在指定输出端口上为指定多点传送连接建立了指定输出支线，达到了交换器支持的多点传送连接的最大限度，到达了指定多点传送连接能够支持的支线的最大限度，或者和交换器支持的多点传送能力有关的一般性问题。当然也可提供其它代码。另外，代码字段 628 在成功响应消息或事件消息中可提供更多的信息。代码字段在 GSMP 请求响应中未被使用，并被设定为 0。

事务标识字段 630 用于使 GSMP 请求消息和它的 GSMP 响应消息联系起来。在 GSMP 请求消息中，交换控制器为字段 630 选择任一事务标识符值。在 GSMP 响应消息中，事务标识符字段 630 的值被设定为该 GSMP 响应消息响应的 GSMP 请求消息的事务标识符的值。由于 GSMP 事件消息不要求响应，因此事务标识符字段 630 被设定为 0。

上面对版本、消息类型、结果、代码及事件标识符字段的概括说明除 GSMP 相邻协议消息外，适用于所有的 GSMP 消息。适当时说明了与概括说明的差别。

对于 GSMP CM 消息，端口话路号字段 632 提供输入端口的话路号。特别地，端口话路号字段 632 的值给出了输入端口字段 634 中指示的交换输入端口的端口话路号。每个交换端口保留一个由交换器分配的端口话路号。当端口不断上升时，端口话路号保持不变。但是，在一个端口下降或不可用之后，该端口再上升后，将产生一个新的不同端口话路号。最好随机地选择新端口话路号。如果交换控制器发送一个端口话路号字段 632 中值无效的 GSMP CM 请求消息，则交换器通过发送一个带有指示导致该失败的无效端口话路号的代码字段 628 的 GSMP CM 失败响应消息，拒绝该 GSMP CM 请求消息。使用 GSMP 配置消息可获得当前端口话路号。

在 GSMP CM 消息中，输入端口字段 634 指示使用交换器分配的 32 位值的交换输入端口。输入 VPI 字段 638 识别到达输入端口字段 634 中指示的交换输入端口的 ATM 虚拟路径，输入 VCI 字段 640 识别到达输入 VPI 字段 638 中识别的虚拟路径的 ATM 虚拟通道。

在 GSMP CM 消息中，输出端口字段 642 指示使用交换器分配的 32 位值的交换输出端口。输出 VPI 字段 646 识别离开输出端口字段 642 中指示的交换输出端口的 ATM 虚拟路径，输出 VCI 字段 648 识别离开输出 VPI 字段 646 中识别的虚拟路径的 ATM 虚拟通道。

对于 GSMP CM 消息，支路数目字段 650 给出虚拟通道连接上的输出支线的数目。字段 650 用在 GSMP CM 校验树消息中。对于所有其它的 GSMP CM 消息，字段 650 由发送者实体设定为 0，并被接收者实体忽略。本实施例中，不用于 GSMP CM 消息的保留字段 652 由发送者实体设定为 0，并被接收者实体忽略。



GSMP CM 消息中的优先权字段 654 给出连接的优先权。最高优先权号为 0，最低优先权号为  $q-1$ ，其中  $q$ =交换输出端口支持的优先权数目。可从 GSMP 端口配置消息中获得每个交换输出端口的  $q$  值。当建立每个虚拟通道连接时，通过给它分配一个优先权，可建立具有一定服务质量 (QOS) 的虚拟通道连接。对于共享相同输出端口的虚拟通道连接，如果两个信元同时位于交换器中，则优先权较高的连接上的 ATM 信元和优先权较低的连接上的 ATM 信元相比，更可能离开交换器。优先权字段 654 用在 GSMP CM 加入支线及移动支线消息中。如果 GSMP CM 请求消息 (用于加入支线或移动支线) 具有交换器不支持的优先权字段 654 值，则交换器分配它能够支持的最接收的优先权。在其它 GSMP CM 消息中，优先权字段 654 由发送者实体设定为 0，并被接收者实体忽略。

加入支线消息是用于建立一个虚拟通道连接，或者向现有虚拟通道连接中加入一个附加支线的 GSMP CM 消息。连接由输入端口字段 634、输入 VPI 字段 638 及输入 VCI 字段 640 指定，输出支线由输出端口字段 642、输出 VPI 字段 646 及输出 VCI 字段 648 指定，它们具有由优先权字段 654 指定的连接优先权。另外，加入支线消息可用于检查 ATM 交换器中存储的连接状态。图 13a 是说明 ATM 交换器从交换控制器接收 GSMP 加入支线请求消息的操作的简图。在步骤 660，交换控制器发送一个由 ATM 交换器接收的 GSMP 加入支线请求消息。ATM 交换器在步骤 662 确定接收的加入支线请求消息的输入端口字段 634、输入 VPI 字段 638 及输入 VCI 字段 640 指定的虚拟通道连接是否在交换器中存在。如果交换器在步骤 662 确定该虚拟通道连接不存在，则 ATM 交换器在步骤 664 着手建立加入支线请求消息中指定的连接。如果交换器在步骤 662 确定虚拟通道连接确实存在，则交换器在步骤 666 确定接收的加入支线请求消息的输出端口字段 642、输出 VPI 字段 646 及输出 VCI 字段 648 指定的输出支线是否在该交换器中存在。如果交换器确定输出支线不存在，则该 ATM 交换器在步骤 668 加入一个如加入支线请求消息中指定的新输出支线。在步骤 664 或步骤 668 之后，交换器在步骤 670 确定操作是否已成功。如果操作不成功，则 ATM 交换器在步骤 672 向交换控制器发送一个加入支线响应消息，该响应消息是接收的加入支线请求消息的副本，它带有指示失败的结果字段 626。加入

支线响应消息还可用其代码字段 628 中的恰当的失败代码指定失败的类型。如果在步骤 670 确定成功地完成了操作，则 ATM 交换器在步骤 674 检查加入支线请求消息的结果字段 626 以确定当请求成功时是否需要响应。如果请求消息的结果字段指示 AckAll，则 ATM 交换器在步骤 676 向交换控制器发送一个成功响应。加入支线成功响应消息是接收的加入支线请求消息的副本，它带有指示成功的结果字段 626。如果交换器在步骤 666 确定加入支线请求消息中指定的输出支线已经存在，则交换器在步骤 680 检查请求消息的优先权字段 654 中指定的优先权是否不同于输出支线的当前优先权。如果交换器确定请求的优先权不同于当前优先权，则交换器在步骤 682 把输出支线的优先权改变为加入支线请求消息指定的优先权。如果优先权相同，则交换器不改变优先权（如步骤 684 所示）。

删除支线消息是用于删除虚拟通道连接的单个支线，或者在最后的支线情况下删除连接的 GSMP CM 消息。连接由输入端口字段 634、输入 VPI 字段 638 及输入 VCI 字段 640 指定，输出支线由输出端口字段 642、输出 VPI 字段 646 及输出 VCI 字段 648 指定。图 13b 是说明从交换控制器接收 GSMP 删除支线请求消息的 ATM 交换器的操作的简图。在步骤 690，交换控制器发送一个由 ATM 交换器接收的 GSMP 删除支线请求消息。ATM 交换器在步骤 692 确定接收的删除支线请求消息的输入端口字段 634、输入 VPI 字段 638 及输入 VCI 字段 640 指定的虚拟通道连接是否在交换器中存在。如果交换器在步骤 692 确定该虚拟通道连接确实存在，则交换器在步骤 694 确定接收的删除支线请求消息的输出端口字段 642、输出 VPI 字段 646 及输出 VCI 字段 648 指定的输出支线是否在该交换器中存在。如果交换器确定输出支线确实存在，则该 ATM 交换器在步骤 696 着手删除删除支线请求消息中指定的输出支线。在步骤 696 之后，交换器在步骤 698 确定删除操作是否已成功。如果确定成功完成了删除，则在步骤 700 交换器根据删除支线请求消息的结果字段 626 确定当请求成功时，是否需要响应。如果请求消息的结果字段指示 AckAll（需要成功响应），则交换器在步骤 702 向交换控制器发送一个删除支线成功响应消息。删除支线成功响应消息是接收的删除支线请求消息的副本，它带有指示成功的结果字段 626。如果在步骤 700 确定不需要成功响应，则交换器不提供响应（如步骤 704 所

示)。如果交换器在步骤 692 确定删除支线请求消息中指定的连接不存在, 或者如果交换器在步骤 694 确定删除支线请求响应 2 中指定的输出支线不存在, 或者如果交换器在步骤 698 确定删除操作不成功, 则交换器在步骤 706 向交换控制器发送一个带有恰当失败代码的删除支线失败响应消息。删除支线失败响应消息是接收的删除支线请求消息的副本, 它带有指示失败的结果字段 626, 及带有由它的代码字段 628 中的恰当失败代码指示的失败类型。

删除树消息用于通过删除连接的所有剩余支线来删除整个虚拟通道连接。连接由输入端口字段 634、输入 VPI 字段 638 及输入 VCI 字段 640 指定。删除树消息中的输出端口字段 642、输出 VPI 字段 646 及输出 VCI 字段 648 未被使用, 由交换控制器设定为零, 并被交换器忽略。图 13c 是说明从交换控制器接收 GSMP 删除树请求消息的 ATM 交换器的操作的简图。在步骤 710, 交换控制器发送一个由 ATM 交换器接收的 GSMP 删除树请求消息。ATM 交换器在步骤 712 确定接收的删除树请求消息的输入端口字段 634、输入 VPI 字段 638 及输入 VCI 字段 640 指定的虚拟通道连接是否在交换器中存在。如果交换器在步骤 712 确定该虚拟通道连接确实存在, 则交换器在步骤 714 着手删除删除树请求消息中指定的连接(从而删除整个树)。步骤 714 之后, 交换器在步骤 716 确定删除操作是否成功。如果确定成功完成了删除, 则在步骤 718 交换器根据删除树请求消息的结果字段 626 确定当请求成功时, 是否需要响应。如果请求消息的结果字段指示 AckAll(需要成功响应), 则交换器在步骤 720 向交换控制器发送一个删除树成功响应消息。删除树成功响应消息是接收的删除树请求消息的副本, 它带有指示成功的结果字段 626。如果在步骤 718 确定不需要成功响应, 则交换器没有任何响应(如步骤 722 所示)。如果交换器在步骤 712 确定删除树请求消息中指定的连接不存在, 或者如果交换器在步骤 716 确定删除操作不成功, 则交换器在步骤 724 向交换控制器发送一个带有恰当失败代码的删除树失败响应消息。删除树失败响应消息是接收的删除树请求消息的副本, 它带有指示失败的结果字段 626, 及带有由它的代码字段 628 中的恰当失败代码指示的失败类型。

校验树消息是用于校验虚拟通道连接上的支线数目的 GSMP CM 消

息。连接由输入端口字段 634、输入 VPI 字段 638 及输入 VCI 字段 640 指定。校验树消息中的输出端口字段 642、输出 VPI 字段 646 及输出 VCI 字段 648 未被使用，由交换控制器设定为零，并被交换器忽略。交换器认为指定虚拟通道连接应含有的支线的数目由校验树请求消息中的支线数目字段 650 给出。图 13d 是说明从交换控制器接收 GSMP 校验树请求消息的 ATM 交换器的操作的简图。在步骤 730，交换控制器发送一个由 ATM 交换器接收的 GSMP 校验树请求消息。ATM 交换器在步骤 732 确定交换器中是否存在由接收的校验树请求消息的输入端口字段 634、输入 VPI 字段 638 及输入 VCI 字段 640 指定的虚拟通道连接。如果交换器在步骤 732 确定该虚拟通道连接确实存在，则在步骤 734 交换器检查指定连接的实际支线数目，并将该实际数目和接收的校验树请求消息的支线数目字段 650 中的数目相比。如果交换器在步骤 736 确定支线数目匹配，则校验操作成功。如果确定成功地完成了校验，则交换器在步骤 738 根据校验树请求消息的结果字段 626 确定当请求成功时，是否需要响应。如果请求消息的结果字段指示 AckAll（需要成功响应），则交换器在步骤 740 向交换控制器发送一个校验树成功响应消息。校验树成功响应消息是接收的校验树请求消息的副本，它带有指示成功的结果字段 626。如果在步骤 738 确定不需要成功响应，则交换器没有任何响应（如步骤 742 所示）。如果交换器在步骤 732 确定校验树请求消息中指定的连接不存在，则交换器在步骤 744 向交换控制器发送一个带有恰当失败代码的校验树失败响应消息。校验树失败响应消息是接收的校验树请求消息的副本，它带有指示失败的结果字段 626，及带有由它的代码字段 628 中的恰当失败代码指示的失败类型。如果交换器在步骤 736 确定校验操作不成功，则交换器在步骤 746 把实际的支线数目设置到校验树失败响应消息的支线数目字段 650 中，并在代码字段设定为零的情况下将其发送给交换控制器。

删除全部消息是用于删除交换输入端口上的所有连接的 GSMP CM 消息。交换输入端口由输入端口字段 634 指定。在删除全部消息中，输入 VPI 字段 638、输入 VCI 字段 640、输出端口字段 642、输出 VPI 字段 646 及输出 VCI 字段 648 未被使用，由交换控制器设定为零，并被交换器忽略。图 13e 是说明从交换控制器接收 GSMP 删除全部请求消息的 ATM 交换器

的操作的简图。在步骤 750，交换控制器发送一个由 ATM 交换器接收的 GSMP 删除全部请求消息。在步骤 752，ATM 交换器确定在接收的删除全部请求消息的输入端口字段 634 中指定的交换器输入端口上是否存在任意连接。如果交换器在步骤 752 确定连接确实存在，则交换器在步骤 754 着手为接收的删除全部请求消息中指定的交换器输入端口删除所有的连接。随后交换器在步骤 756 确定完全删除操作是否成功。如果确定成功地完成了删除操作，则交换器在步骤 758 根据删除全部请求消息确定当请求成功时，是否需要响应。如果请求消息的结果字段指示 AckAll（需要成功响应），则交换器在步骤 760 向交换控制器发送一个删除全部成功响应消息。删除全部成功响应消息是接收的删除全部请求消息的副本，它带有指示成功的结果字段 626。如果在步骤 758 确定不需要成功响应，则交换器没有任何响应（如步骤 762 所示）。如果交换器在步骤 752 确定删除全部请求消息中指定的交换输入端口上不存在任何连接，则交换器在步骤 764 向交换控制器发送一个带有恰当失败代码的删除全部失败响应消息。删除全部失败响应消息是接收的删除全部请求消息的副本，它带有指示失败的结果字段 626，及带有由它的代码字段 628 中的恰当失败代码指示的失败类型。

移动根消息是用于把整个虚拟连接树从其当前输入端口、输入 VPI 及输入 VCI 移动到新输入端口、输入 VPI 及输入 VCI 的 GSMP CM 消息。图 13f 说明了用作请求及响应的 GSMP CM 移动根消息 770 的结构。GSMP CM 移动根消息 770 包括(按照从 MSB 到 LSB 的顺序)下述字段：8 位版本字段 622，8 位消息类型字段 624，8 位结果字段 626 及 8 位代码字段 628；32 位事务标识符字段 630；32 位端口话路号字段 632；32 位旧输入端口字段 772；设定为零的 4 位字 774，12 位旧输入 VPI 字段 776 及 16 位旧输入 VCI 字段 778；32 位新输入端口字段 780；设定为零的 8 位字 782，12 位新输入 VPI 字段 784 及 16 位新输入 VCI 字段 786；及 32 位保留字段 788。版本字段 622、消息类型字段 624、结果字段 626、代码字段 628、事务标识符字段 630 及端口话路号字段 632 的使用方法和前述用于其它 GSMP CM 消息的方法相同。保留字段 788 未被使用，由发送者设定为零，并被接收者忽略。在移动根消息中，当前虚拟通道连接由

旧输入端口字段 772、旧输入 VPI 字段 776 及旧输入 VCI 字段 778 指定，新的虚拟通道连接由新输入端口字段 780、新输入 VPI 字段 784 及新输入 VCI 字段 786 指定。图 13g 是说明从交换控制器接收 GSMP 移动根请求消息的 ATM 交换器的操作的简图。在步骤 790，交换控制器发送一个由交换器接收的 GSMP 移动根请求消息。交换器在步骤 792 确定在接收的移动根请求消息的旧输入端口字段 772、旧输入 VPI 字段 776 及旧输入 VCI 字段 778 中指定的虚拟通道连接是否在交换器中存在。如果交换器在步骤 792 确定该虚拟通道连接确实存在，则交换器在步骤 794 确定由接收的移动根请求消息的新输入端口字段 780、新输入 VPI 字段 784 及新输入 VCI 字段 786 指定的虚拟通道连接是否未被分配。如果在步骤 794 确定该虚拟通道连接未被分配，则交换器在步骤 796 着手移动现有虚拟通道连接的每个输出支线，以便建立如同移动根请求消息中指定的新的虚拟通道连接。步骤 796 之后，交换器在步骤 798 确定移动操作是否成功。如果确定成功地完成了移动操作，则交换器在步骤 800 根据移动根请求消息的结果字段，确定当请求成功时是否需要响应。如果请求消息的结果字段指示 AckAll（需要成功响应），则交换器在步骤 802 向交换控制器发送一个移动根成功响应消息。移动根成功响应消息是接收的移动根请求消息的副本，它带有指示成功的结果字段 626。如果在步骤 800 确定不需要成功响应，则交换器不提供任何响应（如步骤 804 所示）。如果交换器在步骤 792 确定在移动根请求消息中指定的旧的连接不存在，则交换器在步骤 806 向交换控制器发送一个带有恰当失败代码的移动根失败响应消息。移动根失败响应消息是接收的移动根请求消息的副本，它带有指示失败的结果字段 626，及带有由它的代码字段 628 中的恰当失败代码指示的失败类型。如果交换器在步骤 794 确定移动根请求消息中指定的新的虚拟通道连接被分配，则交换器不对现有连接进行任何修改，并在步骤 806 把移动根失败响应消息发送给交换控制器之前，把移动根失败响应消息中的代码字段设定为零（如步骤 808 所示）。

另一种 GSMP CM 消息，移动支线消息用于把虚拟通道连接的单个输出支线从它的当前输出端口、输出 VPI 及输出 VCI 移动到相同虚拟通道连接上的新输出端口、输出 VPI 及输出 VCI。图 13h 说明了用作请求及响应

的 GSMP CM 移动支线消息 820 的结构。GSMP CM 移动支线消息 820 包括（按照从 MSB 到 LSB 的顺序）下述字段：8 位版本字段 622，8 位消息类型字段 624，8 位结果字段 626 及 8 位代码字段 628；32 位字事务标识符字段 630；32 位字端口话路号字段 632；32 位输入端口字段 634；设定为零的 4 位字 636，12 位输入 VPI 字段 638 及 16 位输入 VCI 字段 640；32 位旧输出端口字段 822；设定为零的 8 位字 824，12 位旧输出 VPI 字段 826 及 16 位旧输出 VCI 字段 828；32 位新输出端口字段 830；设定为零的 8 位字 832，12 位新输出 VPI 字段 834 及 16 位新输出 VCI 字段 836；24 位保留字段 838；及优先权字段 840。版本字段 622、消息类型字段 624、结果字段 626、代码字段 628、事务标识符字段 630 及端口话路号字段 632 的使用方法和前述用于其它 GSMP CM 消息的方法相同。保留字段 838 未被使用，由发送者设定为零，并被接收者忽略。优先权字段 840 的使用方法和前述的 GSM CM 消息的优先权字段 654 的使用方法相同。在移动支线消息中，虚拟通道连接由输入端口字段 634、输入 VPI 字段 638 及输入 VCI 字段 640 指定。虚拟通道连接的旧支线由旧输出端口字段 822、旧输出 VPI 字段 826 及旧输出 VCI 字段 828 指定。虚拟通道连接的新支线由新输出端口字段 830、新输出 VPI 字段 834 及新输出 VCI 字段 836 指定。图 13i 是说明从交换控制器接收 GSMP 移动支线请求消息的 ATM 交换器的操作的简图。在步骤 842，交换控制器发送一个由交换器接收的 GSMP 移动支线请求消息。交换器在步骤 844 确定接收的移动支线请求消息的输入端口字段 634、输入 VPI 字段 638 及输入 VCI 字段 640 中指定的虚拟通道连接是否在交换器中存在。如果交换器在步骤 844 确定该虚拟通道连接确实存在，则交换器在步骤 846 确定在该虚拟通道连接上是否存在，由接收的移动支线请求消息的旧输出端口字段 822、旧输出 VPI 字段 826 及旧输出 VCI 字段 828 指定的旧输出支线。如果在步骤 846 确定旧输出支线存在，则交换器在步骤 848 着手加入由移动支线请求消息中的新输出端口字段 830、新输出 VPI 字段 834 及新输出 VCI 字段 836 指定的新输出支线，并删除移动支线请求消息中指定的旧输出支线。步骤 848 之后，交换器在步骤 850 确定移动操作是否已成功。如果确定成功地完成了移动操作，则交换器在步骤 852 根据移动支线请求消息的结果字段 626，确定

当请求成功时是否需要响应。如果请求消息的结果字段指示 AckAll（需要成功响应），则交换器在步骤 854 向交换控制器发送一个移动支线成功响应消息。移动支线成功响应消息是接收的移动支线请求消息的副本，它带有指示成功的结果字段 626。如果在步骤 852 确定不需要成功响应，则交换器不提供任何响应（如步骤 856 所示）。如果交换器在步骤 844 确定移动支线请求消息中指定的虚拟通道连接不存在，或者如果交换器在步骤 846 确定在该虚拟通道连接上不存在移动支线请求消息中指定的旧支线，或者如果交换器在步骤 850 确定移动支线操作不成功，则交换器在步骤 858 不修改任意连接状态，并在步骤 860 向交换控制器发送带有恰当失败代码的移动支线失败响应消息。移动支线失败响应消息是接收的移动支线请求消息的副本，它带有指示失败的结果字段 626，及带有由它的代码字段 628 中的恰当失败代码指示的失败类型。

通过提供交换端口管理，GSMP 端口管理（PM）消息可使端口开始工作、停止工作、被回转或被重置。图 14 说明了用作为请求及响应消息的 GSMP PM 消息 870 的结构。GSMP PM 消息 870 可包含在图 11a 中的封装 GSMP 包 540 的 GSMP 消息字段 546 中。如图 14 所示，GSMP PM 消息包括（按照从 MSB 到 LSB 的顺序）下述字段：8 位版本字段 622，8 位消息类型字段 624，8 位结果字段 626，8 位代码字段 628；32 位字事务标识符字段 630；32 位字端口字段 872；32 位字端口话路号字段 874；32 位事件序列号字段 876；8 位事件标志字段 878；8 位持续时间字段 880；及 16 位功能字段 882。版本字段 622、消息类型字段 624、结果字段 626、代码字段 628、事务标识符字段 630 及端口话路号字段 874 的使用方法和前述用于其它 GSMP CM 消息的方法相同。端口字段 872 给出 GSMP PM 消息施加给的端口的端口号。GSMP PM 消息具有特殊的消息类型字段，在功能字段 882 中可规定各种可能的功能。GSMP PM 消息的部分功能包括：提出功能、拆卸功能、内部回送功能、外部回送功能及重置事件标志功能。每个交换端口保持一个事件序列号及一组事件标志（一个事件标志用于一种类型的事件消息）。当端口被初始化时，事件序列号被设定为零，并且每次在该端口检测到可由事件消息报道的异步事件时，事件序列号增加，而不考虑事件消息是被发送，还是被接收。当交换端口



发送事件消息时，它在该端口设置相应的事件标志。在 GSMP PM 消息的重置事件标志功能重置相应的事件标志之前，不允许该端口发送同种类型的另一事件消息。事件标志的使用提供了简单的流控制，以便防止交换器用事件消息淹没交换控制器。在 GSMP PM 请求消息中，事件序列号字段 876 未被使用，由交换控制器设定为零，并被交换器忽略。在 GSMP PM 成功响应消息中，事件序列号字段 876 给出接收的 GSMP PM 请求消息中指定的交换端口的事件序列号的当前值。在具有指定重置事件标志的功能字段 882 的 GSMP PM 请求消息中，事件标志字段 878 中的特定位可用于重置端口字段 872 指定的交换端口中的相应事件标志。在具有指定重置事件标志的功能字段 882 的 GSMP PM 成功响应消息中，在请求消息中指定的事件标志已被重置后，事件标志字段 878 中的位被设定为指定端口的相应事件标志的当前值。通过在具有重置事件标志功能的 GSMP PM 消息中把事件标志字段设定为零，交换控制器能够在不改变事件标志的状态的情况下，获得事件标志的当前状态，及指定端口的当前事件序列号。在其它具有不同的指定的功能字段 882 的 GSMP PM 消息中，事件标志字段未被使用，由交换控制器设定为零，并被交换器忽略。持续时间字段 880 只在具有规定为内部回送、外部回送或双向回送的功能字段 882 的 GSMP PM 消息中使用。持续时间 880 提供任一回送状态保持有效的时间长度（秒）。当持续时间期满时，处于回送状态的端口自动返回工作状态。在具有不同的指定功能字段 882 的 GSMP PM 消息中，持续时间未被使用，由交换控制器设定为零，并被交换器忽略。在 GSMP PM 消息中，功能字段 882 规定要采取的动作（不考虑端口的当前状态，执行规定的动作）。提出功能使端口开始工作，而拆卸功能使端口停止工作。内部回送功能执行内部回送（从交换组织到达输出端口的 ATM 信元通过输入端口回送到交换组织）。外部回送功能执行外部回送（从外部通信链路到达输入端口的 ATM 信元在物理层被回送给通信链路，而不进入输入端口）。双向回送功能既执行内部回送，又执行外部回送。重置输入端口功能重置输入端口（删除到达指定输入端口的所有连接，重新初始化输入及输出端口硬件，以便在连接表中指定输入端口的所有 VPI/VCI 值为空）。重置事件标记功能重置如上所述的事件标志。

**GSMP 统计消息**允许交换控制器请求和交换输入及输出端口及虚拟通道相关的各种硬件计数器的值。提供了两类统计消息：VC 活动消息，及端口和 VC 统计消息。VC 活动消息用于确定一个或多个特定虚拟通道最近是否传输了通信。VC 活动消息含有一个或多个 VC 活动记录。每个 VC 活动记录用于请求并返回与单个指定虚拟连接相关的活动信息。如果交换器支持对每个虚拟连接进行通信量统计，则每个指定虚拟连接的通信量计数器中的当前值在 VC 活动记录的 VC 通信量计数字段中被返回。把通信量计数的当前值和指定虚拟连接的每一个的先前值进行比较，以确定在干涉期间每个虚拟连接是否是活动的。如果除了通信量统计之外，交换器还支持对每个虚拟连接进行通信量检测，则可使用 VC 活动记录中的标志字段为虚拟连接指示结果。端口和 VC 统计消息用于查询各种端口和 VC 特定通信及错误计数器。端口统计消息用于获取在消息的端口字段中指定的交换端口的统计值，VC 统计消息用于获取消息的端口字段中指定的交换输入端口上的虚拟通道（在消息的 VPI/VCI 字段中指定）的统计值。

**GSMP 配置消息**允许交换控制器确定基本交换单元中的 ATM 交换器的能力。定义了三种类型的 GSMP 配置消息：交换器配置、端口配置及全部端口配置。GSMP 配置消息对于请求消息及响应消息使用了不同的格式，因为在它们的字段中含有不同的信息。由交换控制器发送给 ATM 交换器，并由特殊消息类型字段指示的交换器配置请求消息向 ATM 交换器请求它的全局配置。随后交换器向交换控制器返回一个交换器配置响应消息，该消息包括用于 ATM 交换器的交换器类型及交换器名字的字段，及用于安装的交换控制固件的版本的字段。交换器类型由该交换器的制造商规定，以便识别该交换器产品，交换器名字可以是 48 位 IEEE802MAC 地址，或者在交换器操作语境中是唯一的其它数值。端口配置请求消息具有其自己的特殊消息类型字段，并由交换控制器发送给 ATM 交换器。端口配置请求消息向交换器请求在端口配置请求消息的端口字段中指定的单个交换端口的配置信息。交换器向交换控制器发送一个端口配置成功响应消息，该消息既包括指定端口输入方面的配置信息，又包括指定端口输出方面的配置信息。端口配置成功响应消息中的配置信息包括：端口的当前端口话路号，GSMP 可支持的输入端口上的连接表的 VPI 最小值，GSMP 可支持的输入

端口上的连接表的 VPI 最大值, GSMP 可支持的输入端口上的连接表的 VCI 最小值, GSMP 可支持的输入端口上的连接表的 VCI 最大值。配置信息还包括:端口的信元速率(每秒的 ATM 信元流量),端口的当前状态(即,下行、上升、不能利用、内部回送、外部回送或双向回送);端口类型(端口的物理传输接口的类型,例如,未知、155.52Mbps SONET STS-3c、44.736Mbps DS3、100Mbps 4B/5B 编码、155.52Mbps 8B/10B 编码、25Mbps ATM Forum 物理层、或 51Mbps ATM Forum 物理层);及输出端口可分配给虚拟通道连接的优先权的数目。提供的配置信息被称为端口的端口记录。交换控制器向 ATM 交换器发送一个全部端口配置请求消息,以请求所有交换端口的配置信息,该全部端口配置请求消息具有其自己的特殊消息类型字段。这样,全部端口配置请求消息并不指定一个特定的端口。交换器发送全部端口配置成功响应消息,该消息包括:响应消息包含的端口记录的数目,每个端口记录的字节长度,及每个端口的端口记录。每个端口的端口记录是和对于端口配置成功响应消息说明的同样的配置信息。当然,如果端口记录的数目超过为全部端口配置成功响应消息设置的规定最大数值,则可在多个均不超过规定最大数值的成功响应消息中发送端口记录。

GSMP 事件消息允许 ATM 交换器向交换控制器通知某些异步事件。如前所述,事件消息不被确认。根据异步事件的不同,事件消息可具有不同的消息类型。不同的事件消息包括端口上升事件消息、端口下行事件消息,无效 VPI/VCI 事件消息、新端口事件消息,及死亡端口事件消息。每个交换端口保持一个事件序列号及一组事件标志(一个事件标志用于一种类型的事件消息)。当交换端口发送事件消息时,它在该端口设置相应的事件标志。在 GSMP 端口管理消息的重置事件标志功能重置相应的事件标志之前,不允许该端口发送同种类型的另一事件消息。事件标志的使用提供了简单的流控制,以便防止交换器用事件消息淹没交换控制器。当端口被初始化时,事件序列号被设定为零,并且每次在该端口检测到可由事件消息报道的异步事件时,事件序列号增加,而不考虑事件消息是被发送,还是被接收。当前事件序列号包括在事件消息中,以便向交换控制器通知已在该端口发生,但由于简单流控制机制的行动还未由事件消息报告的异

步事件。端口上升事件消息向交换控制器通知指定端口已从下行状态改变为上升状态。当端口上升时，其输入端口上的所有连接被删除（输入端口的连接表为空），并且交换器分配一个新的端口话路号。端口下行事件消息向交换控制器通知指定端口已从上升状态改变为下行状态。如果交换器能够检测链路故障，则交换器向交换控制器发送一个端口下降事件消息，报告链路故障。当一个或更个 ATM 信元到达一个具有当前未被分配给虚拟通道连接的 VPI/VCI 的输入端口，则交换器向交换控制器发送一个无效 VPI/VCI 事件消息。无效 VPI/VCI 消息分别指定端口字段及 VPI/VCI 字段中的端口及 VPI/VCI。指定新端口数目的新端口事件消息通知交换控制器新端口已加入到交换器中。死亡端口事件消息通知交换控制器已从交换器除去一个端口。死亡端口事件消息在其端口字段及端口话路号字段中分别指定除去的端口数目，以及除去端口之前有效的端口话路号。

### C. IFMP-C

根据一些实施例，系统软件还可利用 IFMP-C 协议通过基本交换单元的交流控制器（在此称为 IFMP-C 控制器）和交换代理器（在此称为 IFMP-C 代理器）之间的链路建立通信，从而当需要时可分配第 3 层包转送。特别地，IFMP-C（一种控制 IFMP-C 代理器的通用目的非对称协议）在初始化时跨越 IFMP-C 控制器和 IFMP-C 代理器之间的链路建立的虚拟通道上运行。单个 IFMP-C 控制器可通过分离的虚拟通道使用 IFMP-C 的多个实例控制多个 IFMP-C 代理器。IFMP-C 还包括 IFMP-C 相邻协议，IFMP-C 协议用来使 IFMP-C 控制器及 IFMP-C 代理器之间的链路两端的同步，发现链路另一端的实体身份，及检测该实体的身份变化。

IFMP-C 允许 IFMP-C 控制器跨越 ATM 交换器建立并释放连接，在点对多点连接上加入及删除叶，管理 IFMP-C 接口，请求接口配置信息，及请求统计。IFMP-C 还允许 IFMP-C 代理器通知 IFMP-C 控制器诸如链路正在下行之类的事件。

如前所述，IFMP-C 是一种主-从协议。IFMP-C 控制器向 IFMP-C 代理器发布请求消息。每个请求消息指示是否需要来自 IFMP-C 代理器的响应，并含有事务标识符，使得响应能够和特定请求相联系。IFMP-C 代理器用指示成功或失败的响应消息应答。

IFMP-C 包括分配了特殊消息类型的 IFMP-C 相邻协议。GSMP 相邻协议用于跨越链路建立同步状态，并在 IFMP-C 控制器和 IFMP-C 代理器之间保持同步交换。除了 IFMP-C 相邻协议消息之外，在 IFMP-C 相邻协议已经实现状态同步之前，不能越过链路发送其它 IFMP-C 消息。在当前没有实现状态同步的链路上接收的所有其它 IFMP-C 消息被丢弃。本实施例中，IFMP-C 还具有四种其它的消息类别：接口、支线、控制及管理。这四种类别是请求-响应消息类别，每一种具有用于请求消息的格式及用于响应消息的格式。除非另外指出，除指示失败本质的代码字段之外，失败响应消息和导致该失败的请求消息相同。本实施例中，具有许多不同的消息类型，即 IFMP-C 消息的功能。在一个特定实施例中，每种 IFMP-C 消息类别具有多种不同的消息类型。

在本发明中，IFMP-C 包长度可变，并以和上面关于图 10b 说明的 ATM 数据链路上 IP 包的缺省封装相类似的方式，直接封装在具有前置 LLC/SNAP 标题的 AAL-5 CPCS-PDU 中。图 15a 说明了封装的 IFMP-C 包 1000。缺省封装把一个 LLC/SNAP 标题加到封装在 AAL-5 CPCS-PDU 的负载中的 IFMP-C 包的前头。按照从 MSB（首先被传输）到 LSB 的顺序，缺省封装的 IFMP-C 包 1000 包括 LLC/SNAP 标题（第一个 32 位字中的 24 位 LLC 字段 1002，跟在其后的 SNAP 标题 1004 的 8 位部分，及 SNAP 标题 1004 的剩余 32 位字部分），IFMP-C 消息 1006（其长度为 32 位字的整数倍），填充字段 1008 及 AAL-5 CPCS-PDU 尾部字段 1010。填充字段 1008 的范围为 0-47 个八位位组，尾部字段 1010 为 8 个八位位组（四个 32 位字）。使用缺省封装的 IFMP-C 消息 1006 的 MOTU 为 1500 个八位位组。在特定实施例中，使用缺省封装的包被发送给 VPI=0，VCI=15（缺省虚拟通道）。

图 15b 说明了可包含在图 15a 的封装 IFMP-C 包 1000 的 IFMP-C 消息字段 1006 中的典型 IFMP-C 消息 1012 的类属结构。如图 15b 所示，类属 IFMP-C 消息 1012 包括（按照从 MSB 到 LSB 的顺序）下述字段：作为第一个 32 位字的 8 位版本字段 1014、8 位消息类型字段 1016、8 位代码字段 1018 及 8 位标志字段 1020；作为第二个 32 位字的 16 位事务标识符 1022，及 16 位消息长度字段 1024；作为第三个 32 位字的发送者同步实例

字段 1026；作为第四个 32 位字的对等物同步实例字段 1028；及消息体字段 1030。讨论 IFMP-C 消息时，“发送者”是发送 IFMP-C 消息的实体，“对等物”是发送者通过链路向其发送 IFMP-C 消息的实体。所述实体可以是 IFMP-C 控制器或 IFMP-C 代理器。上面讨论的类属 IFMP-C 消息 1012 和 IFMP-C 相邻协议消息存在一些差别（将在下面讨论）。

当发送消息时，发送者把版本字段 1014 设定为对应于当前使用的 IFMP-C 协议的版本号（当可发展出其它版本时）。消息类型字段 1016 识别消息体字段 1030 的类型及格式。在特定实施例中，每种特殊消息具有唯一的消息类型。代码字段 1018 用于指示操作是成功还是失败，标志字段 1020 用于指示应如何处理包，以及需要哪种类型的响应。当发送者需要接收者通知操作已完成时，标志字段 1020 可被设定为确认-完成（`acknowledge-complete`）（`PLEASE_ACK`）标志。当发送者只是在操作失败时才要求通知时，发送者可把标志字段 1020 设置为确认-否定（`acknowledge-negative`）（`PLEASE_NACK`）。如果操作成功，并且请求消息在标志字段 1020 中具有 `PLEASE_ACK` 标志，则在响应消息中，标志字段 1020 应被设置为确认（`ACK`）标志，代码字段 1018 应被设置为预定值，例如在特定实施例中为 0。但是，如果操作失败，并且请求消息在标志字段 1020 中具有 `PLEASE_NACK` 标志，则响应消息的标志字段 1020 应被设置为否定-确认（`NACK`）标志，代码字段 1018 应指示该失败的具体原因。代码字段 1018 中使用的失败的完整列表将在下文详述。事务标识符 1022 唯一地标识每个消息。当消息要求响应时，响应消息具有事务标识符字段 1022 中的相同值，从而使得请求消息和响应消息彼此相关。根据特定实施例，消息长度字段 1024 以八位位组的形式给出消息的长度（包括 IFMP-C 标题，但是不包括任意 SNAP/LLC 封装）。发送者同步实例字段 1026 含有识别发送者的当前同步实例的值。如后所述，该值在 IFMP-C 相邻协议中被交换。当接收来自发送者的消息时，发送者同步实例字段 1026 中的值和来自 IFMP-C 相邻协议的接收者处的本地对等物实例的值比较。如果实例不匹配，则接收者忽略该包。对等物同步实例字段 1028 含有识别实例的值，发送者认为该实例为其对等物的当前同步实例。该值在 IFMP-C 相邻协议中被交换。当从发送者接收消息时，对等物同步实例字段 1028

中的值和来自 IFMP-C 相邻协议的接收者处的本地实例比较。如果实例不匹配，则接收者忽略该包。消息体字段 1030 含有任意的特定消息数据，如下面对多种消息类型的详细说明中所述。

图 16a 说明了可包含在图 15a 的封装 IFMP-C 包 1000 的 IFMP-C 消息字段 1006 中的 IFMP-C 相邻协议消息 1040 的类属结构。如图 16a 所示，IFMP-C 相邻协议消息 1040 包括(按照从 MSB 到 LSB 的顺序)下述字段：版本字段 1014、消息类型字段 1016、代码字段 1018；标志字段 1020；事务标识符字段 1022；消息长度字段 1024；32 位发送者实例字段 1042；32 位对等物实例字段 1044；16 位发送者类型字段 1046 及 16 位 ACK 间隔字段 1048；下一个 48 位的发送者名字字段 1050；下一个 48 位的对等物名字字段 1052。讨论 IFMP-C 相邻协议消息时，“发送者”是发送 IFMP-C 相邻协议消息的实体，“对等物”是发送者通过链路向其发送 IFMP-C 相邻协议消息的实体。所述实体可以是 IFMP-C 控制器或 IFMP-C 代理器。

在 IFMP-C 相邻协议消息 1040 中，版本字段 1014、消息类型字段 1016、标志字段 1020、事务标识符字段 1022 及消息长度字段 1024 的使用和上面对图 15b 的类属 IFMP-C 消息 1012 的说明相同。IFMP-C 相邻协议消息的消息类型字段 1016 被设定为特定值，以规定 IFMP-C 相邻协议消息的特定类型。在本实施例，在 IFMP-C 相邻协议消息中对于消息类型字段 1016，存在四种可能值：SYN（同步消息）、SYNACK（同步确认消息）、RSTACK（重置确认消息）、及 ACK（确认消息）。发送者实例字段 1042 被设定为发送者的链路当前实例号。实例号用于检测链路的一侧何时已重启动。链路的实例号在最近应是唯一的，并且当链路重新启动时应是唯一的。对等物实例字段 1044 被设定为发送者认为其是链路远端的当前实例号的值。在对等物实例字段 1044 中使用一个预定值（例如在特定实施例中为 0）来指示何时发送者不了解远程实例。为此该预定值可被保留，并且不用作供发送者实例字段 1042 里使用的有效实例号。发送者类型字段 1046 指示发送该消息的 IFMP-C 实体的类型（规定类型为 IFMP-C 控制器及 IFMP-C 代理器）。当一个实体接收任意 IFMP-C 相邻消息时，接收者把接收的消息中的发送者类型字段 1046 和接收者的类型比较，以便防止一个 IFMP-C 控制器和另一个 IFMP-C 控制器相邻，或者防止一个

IFMP-C 代理器和另一个 IFMP-C 代理器相邻。如果类型相同或者未确定，则该消息被忽略。如果类型是互补的（一个为 IFMP-C 控制器，另一个为 IFMP-C 代理器），则同步发生。ACK 间隔字段 1048 由 IFMP-C 控制器设定以指示最佳的确认（ACK）间隔。IFMP-C 代理器使用 ACK 间隔作为运行相邻协议的超时间隔。当发送 IFMP-C 相邻协议消息时，IFMP-C 代理器把 ACK 间隔字段 1048 设定为由 IFMP-C 控制器忽略的预定值（例如在一个特定实施例中为 0）。发送者名字字段 1050 是在该设备的操作语境中识别发送者的唯一值（如果可能，例如介质存取控制器（MAC）地址）。对等物名字字段 1052 由发送者设定为发送者认为是链路远端的对等物的名字的值。发送者把对等物名字字段 1052 设定为预定值（例如在一个特定实施例中为 0）来指示何时对等物名字未知。

如上所述，IFMP-C 相邻协议用于跨越连接 IFMP-C 控制器及 IFMP-C 代理器的链路建立状态同步，及识别链路状态的变化，及链路另一侧的重新起动。链路的每一侧都运行 IFMP-C 相邻协议。对 IFMP-C 相邻协议来说，对特定链路存在三种可能状态：SYNSENT（同步消息发送），SYNRCVD（同步消息接收），ESTAB（同步建立）。在 IFMP-C 控制器及 IFMP-C 代理器发送 IFMP-C 消息之前，要求链路两端的同步（当建立起邻近时，接口将处于 ESTAB 状态）。

在 IFMP-C 相邻协议中，可导致状态改变的事件有两类：定时驱动事件及包到达。图 16b 说明了这些状态改变，该图是说明在 IFMP-C 相邻协议的三种可能状态下，发送者实体（IFMP-C 控制器或 IFMP-C 代理器）的操作的状态图。IFMP-C 控制器通过把 ACK 间隔字段 1048 设定为特定值（例如在一个特定实施例中为 1 秒）在 IFMP-C 代理器上设置定时间隔。当然，在其它实施例中定时间隔也可使用其它值。如下所述，为了周期地产生 SYN、SYNACK 及 ACK IFMP-C 相邻协议消息，每个实体中需要一个定时器。

在此讨论 IFMP-C 相邻协议的定时驱动事件。如图 16b 所示，如果定时器期满（由  $t$  表示），并且发送者实体处于 SYNSENT 状态 1060，则发送者实体重置定时器，并发送 SYN IFMP-C 相邻协议消息（由 1062 表示）。根据一个特定实施例，这一动作（由虚线表示）只由 IFMP-C 代理器执行，



它具有通过发送 SYN 包启动同步的任务。如果定时器期满，并且发送者实体是 IFMP-C 控制器，则该实体仅重置定时器，而不发送 SYN 包。如果定时器期满，而发送者实体处于 SYNRCVD 状态 1064，则发送者实体重置定时器，并发送 SYNACK IFMP-C 相邻协议消息（由 1066 表示）。如果定时器期满，并且发送者实体处于 ESTAB 状态 1068，则发送者实体重置定时器，并发送 ACK IFMP-C 相邻协议消息（由 1070 表示）。如果链路两侧经过预定数目（例如：3）的超时期间，均未从链路另一侧接收 ACK 包，则为使 IFMP-C 相邻协议超时，链路两侧应重置该链路，并进入 SYNSENT 状态。录 IFMP-C 代理器超时，它应将 ACK 间隔恢复为缺省值。

除了上面讨论的定时驱动转变之外，IFMP-C 相邻包到达也会导致 IFMP-C 协议中的状态转变。当 IFMP-C 相邻消息到达一个实体时，根据该实体的当前状态、消息内容及消息类型采取一个动作。在 IFMP-C 相邻协议中下述操作被执行：更新对等物操作及重置链路操作。IFMP-C 相邻协议的每个实例保持确定对等物实例及对等物名字（即链路远端的对等物实体的实例及名字）的状态。更新对等物操作设置本地对等物状态使之和来自刚接收的 IFMP-C 相邻消息的发送者状态匹配。当重新启动同步时，重置链路操作为链路的局部侧面产生新的实例号，并通过把对等物实例及对等物名字设定为零，删除所有的对等物状态。

对图 16b 中的以下说明，条件“%X”被定义为：输入消息中的对等物实例及对等物名字字段与和链路相关的实例及名字的本地值匹配。条件“%Y”被定义为：输入消息中的发送者实例及发送者名字字段，与为对等物实例及对等物名字存储的发送者实例及发送者名字的值匹配。在图 16b 中，条件“A”表示发送者实体接收一个输入 SYNACK IFMP-C 相邻协议消息，并且条件%X 被满足；条件“B”表示发送者实体接收一个输入 SYNACK IFMP-C 相邻协议消息，并且条件%X 不被满足；条件“C”表示发送者实体接收一个输入 ACK IFMP-C 相邻协议消息，并且条件%X 及 %Y 均被满足；条件“D”表示发送者实体接收一个输入 ACK IFMP-C 相邻协议消息，并且条件%X 及 %Y 均不被满足；条件“E”表示发送者实体接收一个输入 RSTACK IFMP-C 相邻协议消息，并且条件%X 及 %Y

均被满足；条件“F”表示发送者实体接收一个输入 RSTACK IFMP-C 相邻协议消息，并且条件%X及%Y均不被满足；条件“G”表示发送者实体接收 SYN 或 SYNACK IFMP-C 相邻协议消息，并且条件%X被满足；条件“H”表示发送者实体接收 SYN 或 SYNACK IFMP-C 相邻协议消息，并且条件%X不被满足。

如同在此说明的一样，当实体处于 SYNSENT 状态 1060 时，可能发生多种转变。如图 16b 所示，如果发送者实体处于 SYNSENT 状态 1060，并且从链路另一端的对等物接收一个输入 SYN IFMP-C 相邻协议消息，则发送者实体执行更新对等物操作，并向对等物发送一个 SYNACK IFMP-C 相邻协议消息（如步骤 1072 所示）。随后发送者从 SYNSENT 状态 1060 转为 SYNRCVD 状态 1064。如果发送者处于 SYNSENT 状态 1060 的同时，接收一个输入 RSTACK IFMP-C 相邻协议消息，则发送者实体保留在 SYNSENT 状态 1060，但是丢弃该消息（步骤 1074 表示）。如果发送者或者接收输入 ACK IFMP-C 相邻协议消息，或者条件 B 被满足，则发送者实体保留在 SYNSENT 状态 1060，只是发送一个 RSTACK IFMP-C 相邻协议消息（由 1076 表示）。如果当发送者实体处于 SYNSENT 状态 1060 时，条件 A 被满足，则发送者实体更新对等物，发送 ACK IFMP-C 相邻协议消息（由 1078 表示），并转变为 ESTAB 状态 1068。

如同在此说明的一样，当实体处于 SYNRCVD 状态 1064 时，可能发生多种转变。如图 16b 所示，如果发送者实体处于 SYNRCVD 状态 1064，并且从链路另一端的对等物接收一个输入 SYN IFMP-C 相邻协议消息，则发送者更新对等物，并向对等物发送一个 SYNACK IFMP-C 相邻协议消息（如步骤 1072 所示），同时保留在 SYNRCVD 状态 1064。如果处于 SYNRCVD 状态的同时，条件 B 或 D 被满足，则发送者实体保留在 SYNRCVD 状态 1064，并发送一个 RSTACK IFMP-C 相邻协议消息（由 1078 表示）。如果实体处于 SYNRCVD 状态 1064 的同时，条件 F 被满足，则实体保留在 SYNRCVD 状态 1064，但是丢弃该消息（由 1080 表示）。如果处于 SYNRCVD 状态 1064 的同时，条件 E 被满足，则发送者实体执行重置链路操作，并发送一个 SYN IFMP-C 相邻协议消息（由 1082 表示），并从 SYNRCVD 状态 1064 转变到 SYNSENT 状态 1060。如果处于

SYNRCVD 状态 1064 的同时，条件 A 被满足，则发送者实体执行更新对等物操作，发送一个 ACK IFMP-C 相邻协议消息（由 1078 表示），并从 SYNRCVD 状态 1064 转变到 ESTAB 状态 1068。如果处于 SYNRCVD 状态 1064 的同时，条件 C 被满足，则发送者实体发送一个 ACK IFMP-C 相邻协议消息（由 1084 表示），并从 SYNRCVD 状态 1064 转变到 ESTAB 状态 1068。

如同在此说明的一样，当实体处于 ESTAB 状态 1068 时，可能发生多种转变。如图 16b 所示，如果处于 ESTAB 状态 1068 的同时，条件 D 或 H 被满足，则发送者实体保留在 ESTAB 状态 1068，并发送一个 RSTACK IFMP-C 相邻协议消息（由 1088 表示）。如果实体处于 ESTAB 状态 1068 的同时，条件 F 被满足，则实体保留在 ESTAB 状态 1068，只是丢弃该消息（由 1080 表示）。如果处于 ESTAB 状态 1068 的同时，条件 E 被满足，则发送者实体执行重置链路操作，发送一个 SYN IFMP-C 相邻协议消息（由 1086 表示），并从 ESTAB 状态 1068 转变到 SYNSENT 状态 1060。如果处于 ESTAB 状态 1068 的同时，条件 C 或 G 被满足，则发送者实体发送一个 ACK IFMP-C 相邻协议消息（由 1090 表示），并保留在 ESTAB 状态 1068。根据一个特定实施例，每个超时期间，链路一侧的每个实体不应发送多于一个的由包到达产生的 ACK IFMP-C 相邻协议消息。

IFMP-C 接口消息可包含在图 15a 的封装 IFMP-C 包 1000 的 IFMP-C 消息字段 1006 中。如前所述，在链路的同步状态建立之后，IFMP-C 接口消息用于发现并配置 IFMP-C 代理器上的接口。在一个特定实施例中，IFMP-C 接口消息包括接口列表请求及响应消息，接口列表错误消息，接口查询请求及响应消息，接口查询错误消息，接口配置请求及响应消息及接口配置错误消息。IFMP-C 接口列表消息及接口查询消息在消息类型字段 1016 中具有特定值。当然，在其它实施例中，也可有更多的消息，或者执行类似功能的消息。

接口列表请求及响应消息，及接口列表错误消息用于确定 IFMP-C 代理器上哪些接口可用。图 17a 及 17b 分别说明了接口列表请求及响应消息的结构。如图 17a 所示，接口列表请求消息 1100 具有如前关于图 15b 说明的类属格式，其消息体字段 1030 含有下一个 Cookie 字段 1112，这是一个

从前一接口列表响应返回的 32 位值。下一个 Cookie 字段 1112 用于支持跨越多于一个消息的接口列表。下一个 Cookie 字段中的值是由 IFMP-C 代理器返回的值，以便允许下一个消息请求在接口列表中前一个消息终止的地方继续。一个预定值（例如特定实施例中为 0）用于指示接口列表应从头开始。如果接口列表请求消息具有在其标志字段 1020 中设定的 PLEASE\_ACK 标志，并且操作成功，则 IFMP-C 代理器返回一个接口列表响应消息 1114（见图 17b），该消息的标志字段 1020 被设定为 ACK 标志，代码字段 1018 被设置为指示无错误的预定值（特定实施例中为 0）。如图 17b 所示，接口列表响应消息 1114 中的下一个 Cookie 字段 1112 是一个作为消息一部分被返回的 32 位值。如果下一个 Cookie 字段 1112 中的值为 0，则所有的接口都已被列举。如果该值不为 0，则该值用作下一个接口列表消息的变元以获取任何剩余的接口。IFMP-C 代理器为每个接口分配一个唯一的 32 位标识符，该标识符在其它 IFMP-C 消息中用于指示特定接口。接口列表响应消息 1114 在接口标识符 1 字段 1118，接口标识符 2 字段 1120 等等中，为可在响应消息中列举的 IFMP-C 代理器上的每个端口列举出所有的标识符。

如果接口列表请求失败，则 IFMP-C 代理器返回一个包括 IFMP-C 标题的接口列表错误消息。在接口列表错误消息中，应在标志字段 1020 中设置 NACK 标志，代码字段 1018 应指示失败的原因，事务标识符字段 1022 应和接口列表请求消息中的一样。失败的原因的一些例子包括：Cookie 无效，IFMP-C 代理器不能分配消息缓冲区以完成响应，妨碍请求完成的特定客户机错误，或其它一些原因，这些失败原因的每一个都具有供代码字段 1018 之用的特定值。

接口查询请求及响应消息及接口查询错误消息用于获取与 IFMP-C 代理器上的指定接口有关的属性。图 17c 和 17d 分别说明了接口查询请求及响应消息的结构。如图 17c 所示，接口查询请求消息 1130 具有如前关于图 15b 说明的类属格式，其消息体字段 1030 含有接口标识符字段 1132，这是一个唯一地标识属性正被查询的接口的 32 位标识符。如上所述，接口标识符由 IFMP-C 代理器分配，并通过接口列表消息获得。如果来自 IFMP-C 控制器的接口查询请求消息 1130 在其标志字段 1020 中具有设定的

PLEASE\_ACK 标志, 并且查询操作成功, 则 IFMP-C 代理器返回一个接口查询响应消息 1134 (见图 17d), 该消息的标志字段 1020 被设定为 ACK 标志, 代码字段 1018 被设置为指示无错误的预定值。如图 17d 所示, 接口查询响应消息的消息体字段 1030 还包括用于查询接口名字的 48 位接口名字字段 1136 (在特定实施例中, 可使用 MAC 地址), 8 位接口类型字段 1138 (为各种接口类型 (例如 ATM、以太网、快速以太网、千兆位以太网、FDDI 或其它局域网接口) 规定的值), 8 位介质类型字段 1140 (指示接口的物理介质, 例如多模光纤、类 (Category) 5 双绞线、单模光纤等)。此外, 接口查询响应消息还包括 32 位支持速度字段 1142 及 32 位当前速度字段 1144。支持速度字段 1142 指示查询接口支持的不同的传输速率 (例如, 10Mbps、25Mbps、100Mbps、155Mbps、622Mbps、1000Mbps, 以及其它速率)。对于支持多种传输速率的接口来说, 在字段 1142 中可设定多个标志。如果接口支持速度设置的自动协商, 则可在支持速度字段 1142 中设置自动协商标志。当前速度字段 1144 指示查询接口的当前传输速率。如果接口处在自动配置模式, 则在当前速度字段 1144 中表示出当前接口速度, 并且在支持速度字段 1142 中设置自动协商标志。IFMP-C 接口查询响应消息 1134 还包括 32 位的支持的双工字段 1146 (指示查询接口支持的双工速率, 例如半双工, 全双工, 或者双工设置的自动协商; 对于支持多个双工设置的接口可设置多个标志), 32 位的当前双工字段 1148 (指示接口的当前双工设置; 如果接口处于双工设置的自动协商模式, 则在字段 1146 将设置自动协商标志)。此外, 接口查询响应消息 1134 还包括 32 位的接口槽标识符字段 1150 (标识 IFMP-C 代理器上的接口占有的物理槽), 32 位的接口端口标识符字段 1152 (识别 IFMP-C 代理器上的接口占有的物理端口), 16 位接口标志字段 1154, 及 16 位接口状态字段 1156。接口标志字段 1154 规定查询接口上的配置选项的当前设置, 每个标志指示不同的状态 (例如接口正在上升, 接口处于混杂模式, 接口接受所有的多点传送包等)。接口状态字段 1156 指示关于链路的不能由 IFMP-C 控制器改变的当前状态信息 (例如 IFMP-C 控制通信正在通过该接口, 或者该接口上的链路处于上升)。另外, 接口查询响应消息 1134 包括 32 位的最小接收标记字段 1158、32 位的最大接收标记字段 1160, 32

位的最小传输标记字段 1162，及 32 位的最大传输标记字段 1164。如果接口是 ATM，则最小接收标记字段 1158 和最大接收标记字段 1160 分别指示在其上接口可接收的最小及最大 VCI。如果接口是 ATM，则最小传输标记字段 1162 及最大传输标记字段 1164 分别指示接口可传输的最小及最大 VCI。如果查询的接口不是 ATM，字段 1158、1160、1162 及 1164 被设定为零。

如果接口查询请求失败，则 IFMP-C 代理器发送一个接口查询错误消息，该消息包括 IFMP-C 标题，它具有设置在标志字段 1020 中的 NACK 标志，在其代码字段 1018 中指示的恰当的失败原因，及和接口查询请求消息 1130 中同样的事务标识符字段 1022。可在接口查询错误消息的代码字段 1018 中指示的失败原因的例子包括接口标识符无效，IFMP-C 代理器不能分配消息缓冲区完成该响应，妨碍请求完成的特定客户机错误，及其它原因。

接口配置请求及响应消息和接口配置错误消息用于使 IFMP-C 控制器能够改变 IFMP-C 代理器上的接口配置。图 17e 说明了接口配置请求消息 1170 的结构。如图 17e 所示，接口配置请求消息 1170 具有如前关于图 15b 说明的类属格式，其消息体字段 1030 含有接口标识符字段 1132（唯一地标识其配置要被改变的接口），16 位清除标志字段 1172，16 位设置标志字段 1174，32 位速度字段 1176，及 32 位双工字段 1178。清除字段 1172 指示在指定接口上，IFMP-C 代理器应消除哪些标志，以影响接口的操作。清除标志的例子包括在管理上拆卸接口，使接口脱离混杂模式，阻止接口接收所有的多点传送包等。设置标志字段 1174 指示 IFMP-C 代理器应在指定接口上设置哪些标志，以影响该接口的操作。设置标志的例子包括在管理上使接口上行，使接口处于混杂模式，允许接口接收所有的多点传送包等。速度字段 1176 使 IFMP-C 控制器能够改变指定接口的速度设置（在接口支持的速度之中，例如自动协商、10Mbps、25Mbps、100Mbps、155Mbps、622Mbps、1000Mbps 或其它）。双工字段 1178 使 IFMP-C 控制器能够改变指定接口的双工设置（在接口支持的双工设置中，例如自动协商的双工，半双工，或全双工）。速度字段 1176 或双工字段 1178 设置成表示速度或双工设置不应被改变的预定值（例如在特定实施例中为

0)。如果来自 IFMP-C 控制器的接口配置请求消息 1170 具有在其标志字段 1020 中设置的 PLEASE\_ACK 标志，并且操作成功，则 IFMP-C 代理器返回一个接口配置响应消息，该消息由 IFMP-C 标题字段（除消息体字段外，图 15b 所示的所有字段）构成，标志字段 1020 中设置为 ACK 标志，代码字段 1018 设置为指示无错误的预定值。

如果接口配置请求消息失败，则 IFMP-C 代理器返回一个由 IFMP-C 标题构成的接口配置错误消息。在接口列表错误消息中，NACK 标志应被设置在标志字段 1020 中，代码字段 1018 应指示失败的原因，并且事务标识符字段 1022 应和接口列表请求消息中的事务标识符字段一样。失败原因的例子包括接口标识符无效，使用了无效配置参数，妨碍请求完成的特定客户机错误，或其它原因，每个原因具有一个供代码字段 1018 使用的特定值。

除 IFMP-C 相邻和接口消息外，其它类型的 IFMP-C 消息 1012 包括 IFMP-C 支线请求、响应及错误消息。存在 6 种类型的 IFMP-C 支线消息以产生、修改及删除转送支线。具体地，IFMP-C 支线消息包括：加入支线、删除支线、删除树、移动支线、读取支线及获取树统计。给予每个转送支线一个优先权值。当产生多个匹配时，使用优先权值最低的支线。

如前所述，基本转送通过支线完成，其中每个支线由两个部分构成：输入数据及输出数据。输入数据提供足够的信息，使输入包能够和支线匹配，输出数据是转送和输入数据匹配的包所必需的信息。用于修改 IFMP-C 代理器的转送状态的基本操作是加入支线及删除支线。加入多个具有相同输入数据的支线将产生可用于把每个输入包转送给多个目的地的树。例如，具有一个特定输入数据及几个输出数据的第一加入支线消息建立起单点传送包转送。具有相同输入数据及其它输出数据的第二加入支线消息通过加入另一个与该相同输入数据相关的输出支线把单点传送转送转变为多点传送转送。借助于更多的加入支线消息，可用相同的方式加入其它输出支线。删除支线消息是用于删除单个支线的 IFMP-C 支线消息。对具有两个支线的多点传送连接使用删除支线消息可除去具有指定输出数据的支线，把多点传送转送转变为单点传送转送。另一种 IFMP-C 消息，移动支线消息用于改变现有支线上的现有输出信息。其它的 IFMP-C 支线消息包

括用于获取输入数据被指定的转送通路的统计值的获取树统计消息，及用于诊断及调试目的，从而使 IFMP-C 控制器能够检索 IFMP-C 代理器上的所有转送支线的读取支线消息。

加入支线消息及删除支线消息是均使用图 18a 所示消息格式 1200（只是具有不同的消息类型）的 IFMP-C 支线消息。如图 18a 所示，IFMP-C 加入/删除支线请求消息格式 1200 具有前面关于图 15b 说明的类属格式，其消息体字段 1030 包括：输入接口标识符字段 1201，16 位输入优先级字段 1202，16 位输入标志字段 1204，32 位输入接口标识符字段 1206，24 位保留字段 1208，8 位关键字长度字段 1210，8 位输出标题长度字段 1212，8 位删除长度字段 1214，8 位变换类型字段 1216，8 位变换数据长度字段 1218，32 位服务质量句柄字段 1222，预定长度输入关键字数据字段 1224，预定长度输入关键字屏蔽字段 1226，预定长度输出标题数据字段 1228 及变换数据字段 1230。

下面说明图 18a 所示的字段（而不是图 15b 中所示的类属 IFMP-C 标题字段）。由 IFMP-C 代理器分配，并通过 IFMP-C 接口列表消息获得的输入接口标识符字段 1201 唯一地识别将输入支线应用于其的特定输入接口。输入优先级字段 1202（在特定实施例中为无符号整数）表示分配给该支线的优先级。当把输入包和输入关键字匹配时，首先匹配优先级最低的关键字。如果多个通路具有相同的优先级，则 IFMP-C 代理器会选择任一匹配的支线来转送包。输入标志字段 1204 应用于输入支线，该字段中指示的标志用于表示如果包和该转送通路匹配时，应采取的特定行为。可被标志的这些行为的例子包括：“失败”——在包被传输之后，在下一个优先级上搜索，而不是终止；或“放弃”——放弃所有与该输入通路匹配的包。输出接口标识符字段 1206 唯一地识别用于传输该包的接口。如果备用的保留字段 1208 未被使用，则在特定实施例中可由发送者实体设置为 0，并被接收者实体忽略。关键字长度字段 1210（在特定实施例中为 8 位无符号整数）以八位位组的形式给出输入关键字数据字段 1224 及输入关键字屏蔽字段的长度。输出标题长度字段 1212（在特定实施例中为 8 位无符号整数）以八位位组的形式给出输出标题数据字段 1228 的长度。8 位无符号整数的删除长度字段 1214 规定在施加变换数据字段 1230 中指示的变换之前，从包的



开始处删除的 8 位组数。

变换类型字段 1216 规定传输包之前，要对该包进行的修改的类型（例如，无修改、IP 包变换为 IFMP 流类型 1、IP 包变换为 IFMP 流类型 2、IFMP 流类型 1 变换为 IP 包、IFMP 流类型 2 变换为 IP 包、标准 IP 转送、包截尾或其它修改）。某些类型的修改需要不是该包组成部分的数据，这样要求的数据在变换数据字段 1230 中提供。具体地说，“IP 包变换为 IFMP 流类型 1（或 2）”变换类型把包从 IPv4（或者使用中的其它 IP 包版本）转换为前面讨论过的 IFMP 流类型 1（或 2）的封装。“IFMP 流类型 1（或 2）变换为 IPv4 包”变换类型通过使用变换数据字段 1230 重构删除的字段，及更新 TTL 及 IP 标题检查和，把以 IFMP 流类型 1（或 2）的形式到达的包转换为 IPv4 包。“标准 IP 转送”变换类型减小 IP 标题中的 TTL，并更新 IP 标题检查和。“截尾包”变换类型把包的大小限制为转送数据规定的大小（在变换数据字段 1230 中）。

如图 18a 所示，变换数据长度字段 1218 是规定包括在加入/删除支线消息格式 1200 中的变换数据字段 1230 的长度的 8 位无符号整数。如果变换在变换数据字段 1230 中不需要额外的数据，则变换数据长度字段 1218 被设置为 0。服务质量句柄字段 1222 指示 IFMP-C 代理器应如何处理与支线匹配的包，其长度由关键字长度字段 1210 规定。输入关键字数据字段 1224 含有和输入包比较，以了解其是否和转送通路匹配的数据。关键字数据字段 1224 中的数据拥有 MAC 地址之类的链路级信息，IP 地址之类的第 3 层信息等等，其长度由关键字长度字段 1210 规定。关键字屏蔽字段 1226 用于规定当把输入关键字数据字段 1224 中的信息和输入包比较时，关键字数据的相关位，其长度由输出标题长度字段 1212 规定。输出标题数据字段 1228 含有在传输包之前，应将其预挂给该包的标题（一般为链路级信息，例如以太网接口上的 MAC 标题，或 ATM 接口上的 VPI/VCI）。如上所述，变换数据字段 1230 含有执行指定变换所需的数据。对于“IFMP 流类型 1（或 2）变换为 IPv4”变换，变换数据字段 1230 中的数据和图 7a（或者流类型 2 的图 7b，除了在某些实施例中服务类型字段及协议字段可被保留外）所示的流标识符相似。当指定“截尾包”变换时，包的部分副本被发送给特定目的地，变换数据字段 1230 中的数据 1240 包括可由发送者设

置为 0，并被接收实体忽略的 16 位保留字段 1242，及 16 位截断长度字段 1244，如图 18b 所示。截断长度字段 1244 表示该包的多少个字节应从接口传输。如果到达的包的长度大于该值，则该包被截断到这个长度。截断长度是已删除任意字节之后的字节数，并且不包括可被加入的任意输出标题。

如果 IFMP-C 加入（或删除）支线请求消息具有在其标志字段 1020 中设置的 PLEASE\_ACK 标志，并且操作成功，则 IFMP-C 代理器返回一个具有图 18c 所示格式的 IFMP-C 加入（或删除）支线响应消息 1250，其标志字段 1020 中设置为 ACK 标志，代码字段 1018 设置为指示无错误的预定值。如图 18c 所示，响应消息具有类属 IFMP-C 的 IFMP-C 标题字段，其消息体含有 16 位保留字段 1252 及 16 位输出计数字段 1254。保留字段 1252 备用，可由发送者设置为 0，并被接收者忽略。无符号整数的输出计数字段 1254 含有当前操作施加后，共享相同输入信息的输出支线的数目（如果支线计数为 1，则其是单点传送支线；如果支线计数大于 1，则其是多点传送支线）。字段 1254 由 IFMP-C 控制器使用，以校验每次操作之后，支线状态的连贯性。

如果 IFMP-C 加入支线（或删除支线）请求失败，则 IFMP-C 代理器发送一个 IFMP-C 加入支线（或删除支线）错误消息。该错误消息由 IFMP-C 标题构成，其事务标识符字段 1022 和请求消息的事务标识符字段相同。其标志字段 1020 中设置为 NACK 标志，代码字段 1018 设置为指示失败原因的值。失败原因的例子包括：接口标识符之一无效；输入关键字长度大于 IFMP-C 代理器支持的最大长度；IFMP-C 代理器不支持或不识别输出变换；IFMP-C 代理器的资源不足以完成该请求；存在具有相同输入关键字，但是标志不同的另一支线；服务质量参数无效或不被 IFMP-C 代理器所支持；IFMP-C 代理器不支持输入关键字或屏蔽；试图加入 IFMP-C 代理器上已存在的指定支线；妨碍请求完成的特定客户机错误；或其它原因。

如图 18d 所示，IFMP-C 删除树请求消息 1260 具有含有对于图 18a 说明的多数字段的格式。删除树请求消息格式 1260 具有前面关于图 15b 说明的类属格式，其消息体字段 1030 包括：输入接口标识符字段 1201，输入优先级字段 1202，输入标志字段 1204，56 位保留字段 1262，关键

字长度字段 1210，预定长度的输入关键字数据字段 1224 及预定长度的输入关键字屏蔽字段 1226。

输入接口标识符字段 1201 唯一地识别应将输入支线（对该输入支线，树将被删除）应用于其的输入接口。输入关键字数据字段 1224 提供和输入包比较，以了解其是否和将被删除的转送通路匹配的数据，输入关键字屏蔽字段 1226 规定当将输入关键字数据和输入包比较时，关键字数据的相关位。保留字段 1262 备用，可由发送者设置为 0，并被接收者忽略。

如果 IFMP-C 删除树请求消息 1260 具有在其标志字段 1020 中设置的 PLEASE\_ACK 标志，并且操作成功时，则 IFMP-C 代理器返回一个为 IFMP-C 标题字段（除没有消息体字段之外，如图 15b 所示）的 IFMP-C 删除树响应消息，其标志字段 1020 中设置为 ACK 标志，代码字段 1018 设置为指示无错误的预定值。如果 IFMP-C 删除树请求失败，则 IFMP-C 代理器返回一个为 IFMP-C 标题字段（除没有消息体字段之外，如图 15b 所示）的 IFMP-C 删除树错误消息，其标志字段 1020 中设置为 NACK 标志，代码字段 1018 设置为识别失败原因（例如接口标识符无效，具有请求中指定的输入关键字的支线不存在，存在另一个具有指定输入关键字，但是标志不同的支线，或者其它原因）的值。

图 18e 说明了 IFMP-C 移动支线请求消息 1300 的结构，它含有对于图 18a 说明的多数字段。移动支线请求消息格式 1300 具有前面关于图 15b 说明的类属格式，其消息体字段 1030 包括（按照从 MSB 到 LSB 的顺序）：输入接口标识符字段 1201，输入优先级字段 1202，输入标志字段 1204，32 位旧输出接口标识符字段 1302，24 位保留字段 1304，8 位关键字长度字段 1210，8 位旧输出标题长度字段 1308，8 位旧删除长度字段 1310，8 位旧变换类型字段 1312，8 位旧变换数据长度字段 1314，32 位旧服务质量句柄字段 1318，32 位新输出接口标识符字段 1320，32 位保留字段 1322，8 位新输出标题长度字段 1324，8 位新删除长度字段 1326，8 位新变换类型字段 1328，8 位新变换数据长度字段 1330，32 位新服务质量句柄字段 1334，预定长度的输入关键字数据字段 1224，预定长度的输入关键字屏蔽字段 1226，预定长度的旧输出标题数据字段 1340，预定长度的旧变换数据字段 1342，预定长度的新输出标题数据字段 1344，预定长

度的新变换数据字段 1346。上面列举的大多数字段在前面对于图 18a 的描述中说明了，只要注意到确定旧支线的输出数据的字段的值，被确定占据旧支线位置的新支线的输出数据的字段中的值所代替，就容易理解移动支线请求消息 1300 中的其它各个字段。

如果 IFMP-C 移动支线请求消息 1300 在其标志字段 1020 中设置有 PLEASE\_ACK 标志，并且操作成功，则 IFMP-C 代理器返回一个 IFMP-C 移动支线响应消息，它具有和 IFMP-C 加入/删除支线响应消息相同的格式（如图 18c 所示），其标志字段 1020 中设置为 ACK 标志，代码字段 1018 设定为指示无错误的预定值。如果 IFMP-C 移动支线请求失败，则 IFMP-C 代理器返回一个为 IFMP-C 标题字段（除没有消息体字段之外，如图 15b 所示）的 IFMP-C 移动支线错误消息，其标志字段 1020 中设置为 NACK 标志，代码字段 1018 设置为识别失败具体原因（例如接口标识符无效，输入关键字长度大于支持的最大长度，IFMP-C 代理器资源不足，输出变换不被支持或不识别，不存在具有和请求中为初始支线指定的相同输入关键字的支线，存在另一个具有指定输入关键字，但是标志不同的支线，服务质量参数无效或不被支持，妨碍请求完成的特定客户机错误，或者其它原因）的值。

作为另一种 IFMP-C 支线消息类型，IFMP-C 获取树统计消息由 IFMP-C 控制器使用，以确定何时通路不再被使用，从而可以收回这些通路。每次通路用于转送包时，IFMP-C 代理器使计数器值增加。当树上的每个支线将被使用相同次数时，为每个树保持该通路。如图 19a 所示，IFMP-C 获取树统计请求消息 1400 具有前面关于图 15b 说明的类属格式，其消息体字段 1030 包括一系列树数据信息：在特定实施例中为树数据 1 字段 1402 和树数据 2 字段 1404。其它树数据字段也可包括在以消息体字段 1030 的形式存在的列表中。图 19b 说明了所有树数据字段使用的树数据字段结构 1406。每个树数据字段结构 1406 包括（从 MSB 到 LSB）：输入接口标识符字段 1201，输入优先级字段 1202，输入标志字段 1204，40 位保留字段 1408，关键字长度字段 1210，18 位记录大小字段 1410，64 位使用计数字段 1412，预定长度的输入关键字数据字段 1224 及预定长度的输入关键字屏蔽字段 1226。输入接口标识符字段 1201 唯一地识别应将输入

支线(将为该输入支线获取树统计值)应用于其的输入接口。保留字段 1408 备用,可由发送者设置为 0,并被接收者忽略。指示特殊树记录(例如,树数据 1 字段 1402 中的树记录)的大小的记录大小字段 1410 用于查找下一个树记录(例如,树数据 2 字段 1404 中的树记录)的开始处。使用计数字段 1412 是 64 位无符号整数,每次 IFMP-C 代理器使用指定树转送包时,使用计数字段 1412 的值增加。在请求消息中,使用计数字段由发送者设置为 0,并被接收者忽略。由于关于图 18a 已对剩余的字段进行了说明,因此在此不再描述这些字段。

如果 IFMP-C 获取树统计请求消息 1400 具有在其标志字段 1020 中设置的 PLEASE\_ACK 标志,并且操作成功,则 IFMP-C 代理器返回一个 IFMP-C 获取树统计响应消息,它具有和请求消息 1400(图 19a)相同的格式,其标志字段 1020 中设置为 ACK 标志,其代码字段 1018 设置为指示无错误的预定值。获取树统计响应消息还在树数据字段的使用计数字段中返回恰当的计数器值。

如果 IFMP-C 获取树统计响应失败,则 IFMP-C 代理器返回一个 IFMP-C 获取树统计错误消息,该错误消息为请求消息的 IFMP-C 标题字段(除了没有消息体字段外,如图 15b 所示),其标志字段 1020 中设置为 NACK 标志,代码字段 1018 设置为识别失败具体原因(例如,接口标识符无效,指定的输出树之一在 IFMP-C 代理器上不存在,存在具有指定的相同输入关键字,但是标志不同的另一个树,或其它原因)的值。

另一种 IFMP-C 支线消息类型是 IFMP-C 读取支线消息,它用于诊断及调试目的,从而允许 IFMP-C 控制器检索 IFMP-C 代理器上的所有转送支线。为了列举 IFMP-C 代理器上的所有支线,读取支线消息使用“获取下一个”操作。每次通路作好准备时,IFMP-C 代理器返回支线信息,及用作为下一次读取操作的变元的 cookie。该 cookie 对于 IFMP-C 控制器是含糊的,并由 IFMP-C 代理器使用以记住上次读取时停止的地方。IFMP-C 控制器可保留并使用一个预定值(例如在特定实施例中为 0)来获取 IFMP-C 代理器上的初始通路。直到列举了所有的表,此时 IFMP-C 代理器返回一个表示已到达列表末尾的指示时,获取转送通路请求的顺序串将获得成功。

图 20a 和 20b 分别说明了 IFMP-C 读取支线请求消息 1420 及 IFMP-C 读取支线响应消息 1430 的结构。如图 20a 所示，IFMP-C 读取支线请求消息 1420 具有前面关于图 15b 说明的类属格式，其消息体字段 1030 含有输入接口标识符字段 1201 及 32 位的下一 Cookie 字段 1422。输入接口标识符字段 1201 唯一地识别在其上支线应被读取的输入接口。下一 Cookie 字段 1422 是作为先前的读取支线响应消息的一部分返回的不透明 32 位值。下一 Cookie 字段 1422 中的值是由 IFMP-C 代理器保持，以便记录最后的读取支线请求的位置。一个预定值（特定实施例中为 0）用于指示应从列表的超始处开始。

如果 IFMP-C 读取支线请求消息具有在其标志字段 1020 中设置的 PLEASE\_ACK 标志，并且操作成功，则 IFMP-C 代理器返回一个 IFMP-C 读取支线响应消息 1430（见图 20b），其标志字段 1020 中设置为 ACK 标志，代码字段 1018 设置为指示无错误的预定值（特定实施例中为 0）。如图 20b 所示，读取支线响应消息 1430 具有前面关于图 15b 说明的类属格式，其消息体字段 1030 包括：输入接口标识符字段 1201，输入优先级字段 1202，输入标志字段 1204，输出接口标识符字段 1206，24 位保留字段 1432，关键字长度字段 1210，输出标题长度字段 1212，删除长度字段 1214，变换类型字段 1216，变换数据长度字段 1218，服务质量句柄字段 1222，下一 Cookie 字段 1422，预定长度的输入关键字数据字段 1224，预定长度的输入关键字屏蔽字段 1226，预定长度的输出标题数据字段 1228 及变换数据字段 1230。保留字段 1432 备用，由发送者设置为 0，并被接收者忽略。下一 Cookie 字段 1422 是由 IFMP-C 代理器保持，以便记录最后的请求的位置，并被用作为下一读取支线消息的输入的状态。由于其它字段已在前面关于图 18a 进行了说明，在此不再描述它们。

如果读取支线请求失败，则 IFMP-C 代理器返回一个由 IFMP-C 标题构成的读取支线错误消息。除了在标志字段 1020 中设置的 NACK 标志，及代码字段 1018 设定为指示失败原因的值之外，读取支线错误消息应和读取支线请求消息的 IFMP-C 标题相同。失败原因的例子包括消息中的接口标识符之一无效，cookie 无效，IFMP-C 代理器不能分配消息缓冲区以完成响应，妨碍请求完成的特定客户机错误，或其它原因。

除了 IFMP-C 相邻、接口及支线消息之外，IFMP-C 消息还包括 IFMP-C 重置消息之类的 IFMP-C 控制消息。IFMP-C 重置消息用于当 IFMP-C 控制器失去和 IFMP-C 代理器的联系，或者如果 IFMP-C 控制器认为 IFMP-C 代理器不可靠时，重新初始化 IFMP-C 代理器的状态。IFMP-C 重置请求消息指示 IFMP-C 代理器将其所有状态重置为初始条件。一旦接收到重置请求响应，则 IFMP-C 代理器除去所有的转送支线，并初始化接口，而没有重置 IFMP-C 相邻协议状态。IFMP-C 重置消息由图 15b 所示的 IFMP-C 标题（无消息体字段）组成，它具有设置为指示该消息为重置消息的消息类型字段 1016。如果重置请求消息具有在其标志字段 1020 中设置的 PLEASE\_ACK 标志，并且重置操作成功，则 IFMP-C 代理器返回一个由除了其标志字段 1020 设置为 ACK 标志，其代码字段设置为指示无错误的预定值之外，和重置请求消息相同的 IFMP-C 标题组成的重置响应消息。如果重置请求失败，则 IFMP-C 代理器返回一个由除了其标志字段 1020 设置为 NACK 标志，其代码字段设置为指示错误的特定原因（例如妨碍请求完成的特定客户机错误）的值之外，和重置请求消息相同的 IFMP-C 标题组成的重置错误消息。

除了 IFMP-C 相邻、接口、支线及控制消息之外，IFMP-C 消息还包括 IFMP-C 管理消息，IFMP-C 管理消息用于获得网络管理及诊断目的所需的信息。IFMP-C 管理消息包括多种消息类型：IFMP-C 节点信息消息及 IFMP-C 接口统计消息。

IFMP-C 节点信息消息获取关于运行 IFMP-C 的节点的信息（例如软件版本号等）。IFMP-C 节点信息请求消息具有如图 15b 所示的 IFMP-C 标题的类属格式（无消息体字段 1030），其消息类型字段 1016 标记该消息为 IFMP-C 节点信息消息。

如果 IFMP-C 节点信息请求消息具有在其标志字段 1020 中设置的 PLEASE\_ACK 标志，并且该操作成功，则 IFMP-C 代理器返回一个 IFMP-C 节点信息响应消息 1440（见图 21a），其标志字段 1020 设置为 ACK 标志，代码字段 1018 设置为指示无错误的预定值（特定实施例中为 0）。如图 21a 所示，IFMP-C 节点信息响应消息 1440 具有如前面关于图 15b 说明的类属格式，其消息体字段 1030 含有：48 位节点 ID 字段 1442，

48 位父 ID 字段 1444，16 位节点类型字段 1446，15 位保留字段 1448，32 位父层字段 1450，32 位父架字段 1452，16 位固件次版本字段 1454，及 16 位固件主版本字段 1456。节点 ID 字段 1442 是唯一地识别该节点的值（例如节点的 MAC 地址）。对于节点是诸如机箱中的插件板之类大系统的一部分的情况，父 ID 字段 1444 被设置为唯一地识别该节点的包容器（或父代）的值。如果该节点不是诸如机箱中的插件板之类大系统的一部分，则父 ID 字段 1444 被设置为指示无父节点的预定值（特定实施例中为 0）。节点类型字段 1446 是指示 IFMP-C 节点的类型 16 位无符号整数。节点类型字段 1446 中的值可由厂家分配，从而对于每种类型 IFMP-C 代理器，节点 ID 的组织独特标识符（OUI，其为 MAC 地址的高位 24 位）部分和节点类型的组合是唯一的。保留字段 1448 备用，由发送者设置为 0，并被接收者忽略。如果节点是诸如机箱中的插件板之类大系统的一部分，则父层字段 1450 被设置为对应于该节点在父代包容器中占据的槽。如果 IFMP-C 代理器不能确定槽信息，或者不是大的包容器的一部分，则父层字段 1450 可被设置为 0。如果节点是诸如机箱中的插件板之类大系统的一部分，则父架字段 1452 被设置为对应于该节点在父代包容器中占据的槽。如果 IFMP-C 代理器不能确定架信息，或者不是大的包容器的一部分，则父架字段 1452 可被设置为 0。固件次版本字段 1454 指示当前运行的 IFMP-C 代理器固件的次版本，固件主版本字段 1456 识别当前运行的 IFMP-C 代理器固件的主版本。

如果节点信息请求失败，则 IFMP-C 代理器返回一个由 IFMP-C 标题构成的节点信息错误消息。除了标志字段 1020 中设置的 NACK 标志，及代码字段 1018 设置为指示失败原因的值之外，节点信息错误消息应和节点信息请求消息的 IFMP-C 标题相同。失败原因的例子包括 IFMP-C 代理器不能分配消息缓冲区以完成该响应，妨碍请求完成的特定客户机错误，或其它原因。

另一种 IFMP-C 网络管理消息类型，IFMP-C 接口统计消息被用于获取关于 IFMP-C 代理器上的每个接口的信息。一个 IFMP-C 接口统计请求消息允许 IFMP-C 控制器在单个请求消息中请求对多个接口的统计。如图 21b 所示，IFMP-C 接口统计请求消息 1460 具有如图 15b 所示的类属格



式，其消息体字段 1030 包括：16 位保留字段 1462，16 位接口数目字段 1464，其后是多个接口标识符字段 1466，1468 及其它字段。保留字段 1462 备用，可由发送者设置为 0，并被接收者忽略。接口数目字段 1464 指示请求消息中的接口标识符字段数目。接口统计请求消息 1460 包括多个接口标识符字段（例如 1466 及 1468），从而列举出 IFMP-C 控制器关心的每个接口的标识符。在特定实施例中，应答必须装在单个消息中，从而 IFMP-C 代理器为同样数量的接口返回装在单个响应消息中的多个响应。

如果 IFMP-C 接口统计请求消息具有在其标志字段 1020 中设置的 PLEASE\_ACK 标志，并且该操作成功，则 IFMP-C 代理器返回一个 IFMP-C 接口统计响应消息 1470（见图 21c），其标志字段 1020 设置为 ACK 标志，代码字段 1018 设置为指示无错误的预定值（特定实施例中为 0）。如图 21c 所示，IFMP-C 接口统计响应消息 1470 具有如前面关于图 15b 说明的类属格式，其消息体字段 1030 含有：提供关于接口的通用统计（对不同接口类型通用的统计），及专用统计（只适用于特定接口类型的统计）的接口统计字段（例如 1472、1474）。如图 21d 所示的典型的接口统计字段结构 1480 包括：8 位保留字段 1482，8 位接口类型字段 1484，16 位记录长度字段 1486，32 位接口标识符字段 1488，16 位通用统计长度字段 1490，专用统计长度字段 1492，通用统计字段 1494，及专用统计字段 1496。保留字段 1482 备用，并可由发送者设置为 0，并被接收者忽略。接口类型字段 1484 描述被查询接口的类型（例如 ATM、以太网或其它局域网接口）。接口标识符字段 1488 唯一地指示统计数据描述的是哪一个接口。通用统计长度字段 1490 以八位位组的形式规定通用统计字段 1494 的长度，专用统计长度字段 1492 以八位位组的形式规定专用统计字段 1496 的长度。通用统计字段 1494 含有与所有接口相关的通用统计，并具有图 21c 所示的结构。专用统计字段 1496 含有接口类型专用统计，具有图 21f 及 21g 所示的示范结构（用于 ATM 及以太网接口）。其它结构可用于其它局域网络接口。

如图 21e 所示，通用统计字段 1494 包括：64 位接收八位位组字段 1500，它指示在指定接口上接收的八位位组的数目；64 位接收多点传送包字段 1502，它指示在指定接口上接收的，作为多点传送包寻址的包的数目；

64 位接收广播包字段 1504，它指示在指定接口上接收的，作为广播包寻址的包的数目； 64 位接收单点传送包字段 1506，它指示在指定接口上接收的，作为单点传送包寻址的包的数目； 64 位接收废弃字段 1508，它指示对于指定接口在输入时抛弃的包的数目； 64 位接收错误字段 1510，它指示指定接口上的接收错误的数目。 64 位接收未知字段 1512，它指示在指定接口上接收的具有未识别出协议的包的数目； 64 位传输八位位组字段 1514，它指示在指定接口上传输的八位位组的数目； 64 位传输多点传送包字段 1516，它指示在指定接口上传输的，作为多点传送包寻址的包的数目； 64 位传输广播包字段 1518，它指示在指定接口上传输的，作为广播包寻址的包的数目； 64 位传输单点传送包字段 1520，它指示在指定接口上传输的，既不是作为多点传送包，也不是作为广播包寻址的包的数目； 64 位传输废弃字段 1522，它指示当在指定接口上传输时，抛弃的包的数目；及 64 位传输错误字段 1524，它指示在指定接口上已产生的传输错误的数目。

图 21f 及 21g 分别表示了 ATM 及以太网接口的专用统计的专用统计字段 1496。如图 21f 所示，ATM 接口的专用统计字段结构 1530 包括：64 位接收信元字段 1532，它指示在指定接口上接收的 ATM 信元的数目；64 位传输信元字段 1534，它指示在指定接口上传输的 ATM 信元的数目；64 位 AAL5 CRC 错误字段 1535，它指示在指定接口上接收的带有错误 AAL5 循环冗余检查和（CRC）的包的数目；及 64 位物理错误字段 1536，它指示在指定 ATM 接口上已产生的物理错误的数目。用于以太网接口的专用统计字段 1496 的另一个例子包括：64 位接收 CRC 错误字段 1542，它指示在指定接口上已接收的带有错误 CRC 的包的数目；及 64 位传输冲突字段 1544，它指示当试图在指定接口上发送包时，产生的冲突的数目。

如果 IFMP-C 接口统计请求失败，则 IFMP-C 代理器返回一个由 IFMP-C 标题组成的 IFMP-C 接口统计错误消息。除了设置在标志字段 1020 中的 NACK 标志，及代码字段设置为指示失败原因的值之外，接口统计错误消息应和接口统计请求消息的 IFMP-C 标题相同。失败原因的例子包括接口标识符无效，列举的接口的数目和消息中的接口数目不一致，IFMP-C 代理器不能分配消息缓冲区以完成响应，妨碍请求完成的特定客户机错误，或者其它原因。

#### IV. 结论

在此要求的发明提供了一种在网络中传输包的改进方法及设备。当然上述说明只是说明性的，而非对本发明的限制。通过研究上述说明，对本领域有经验人员来说，许多实施例是显而易见的。通过例子，主要在能够携带音频、视频、图象、传真及数据信号的 IP 包（但是并限于此）的传输方面在此对本发明进行了说明。通过其它例子，对本发明和特殊的组件及操作速度一起进行了说明，但是本发明并不限于此。此外，尽管对具体的实施例说明了消息类型、错误等的具体例子，但是这些仅是例子，而其它实施例可以使用不同的、更多的、较少的特征，或者说明特征的组合。于是本发明的范围不应参考上面的说明来确定，而应参考权利要求，以及本领域普通技术人员做出的这些权利要求的等效物的全部范围来确定本发明的范围。

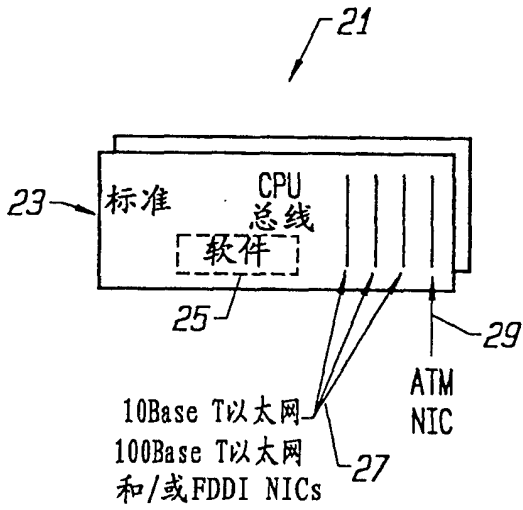


图 1b

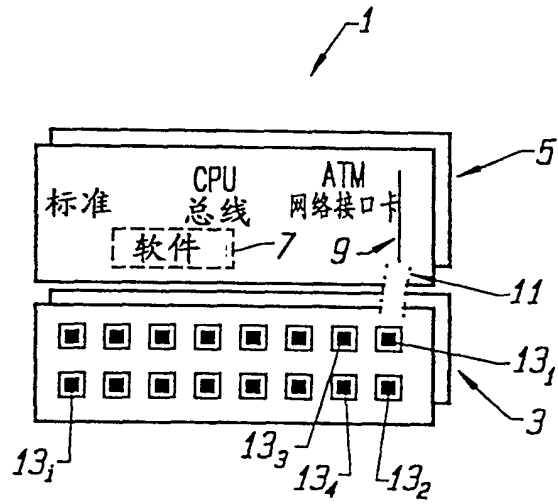


图 1a

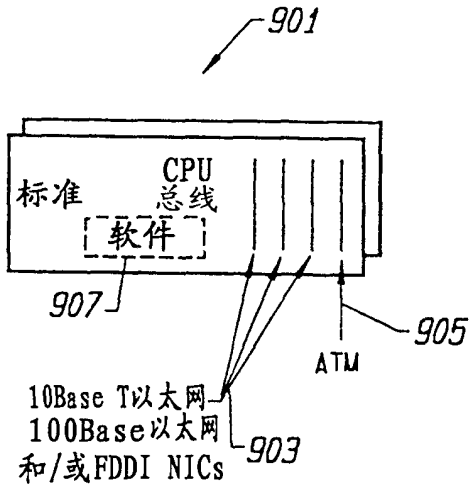


图 1c

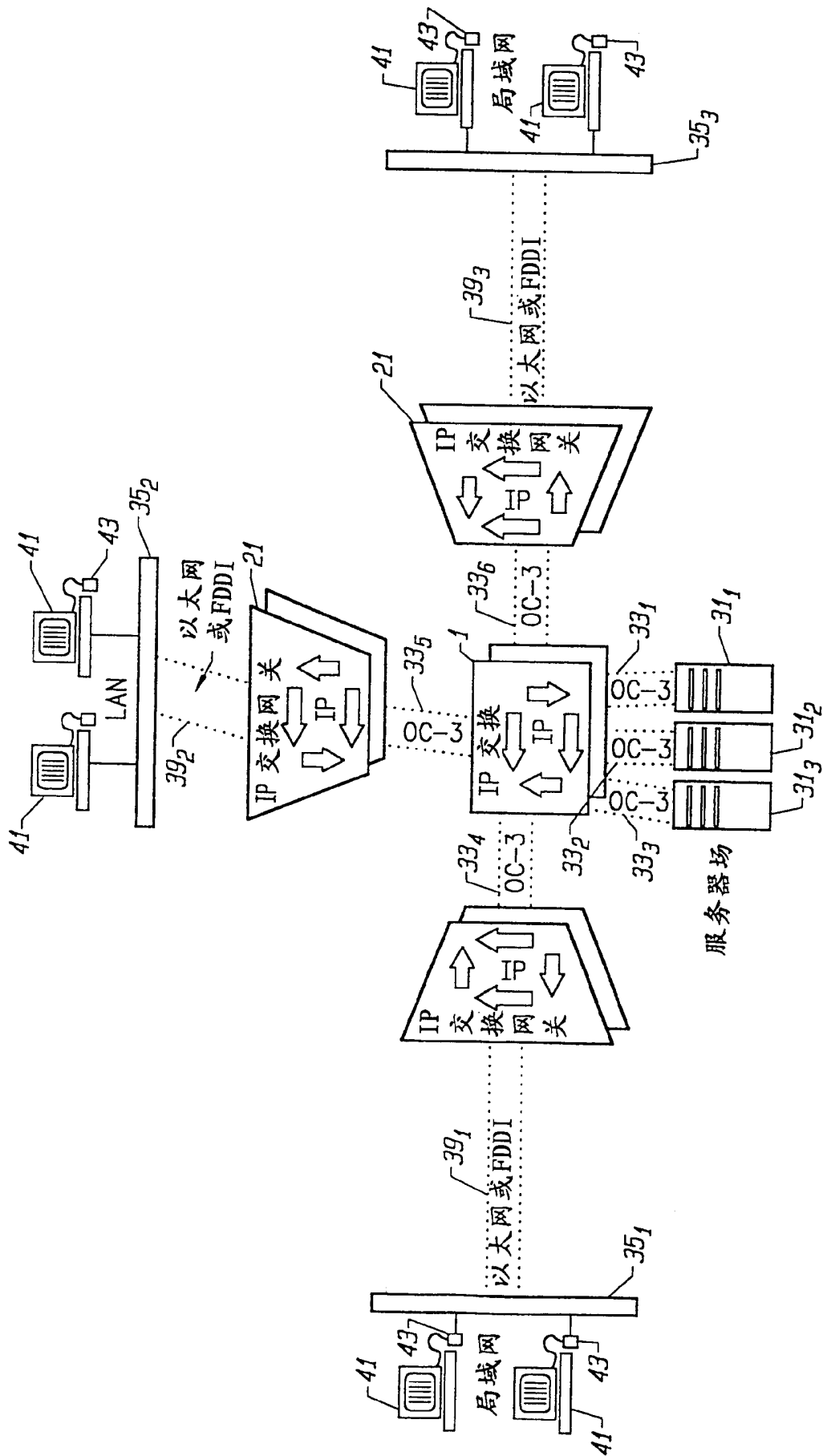


图 2a

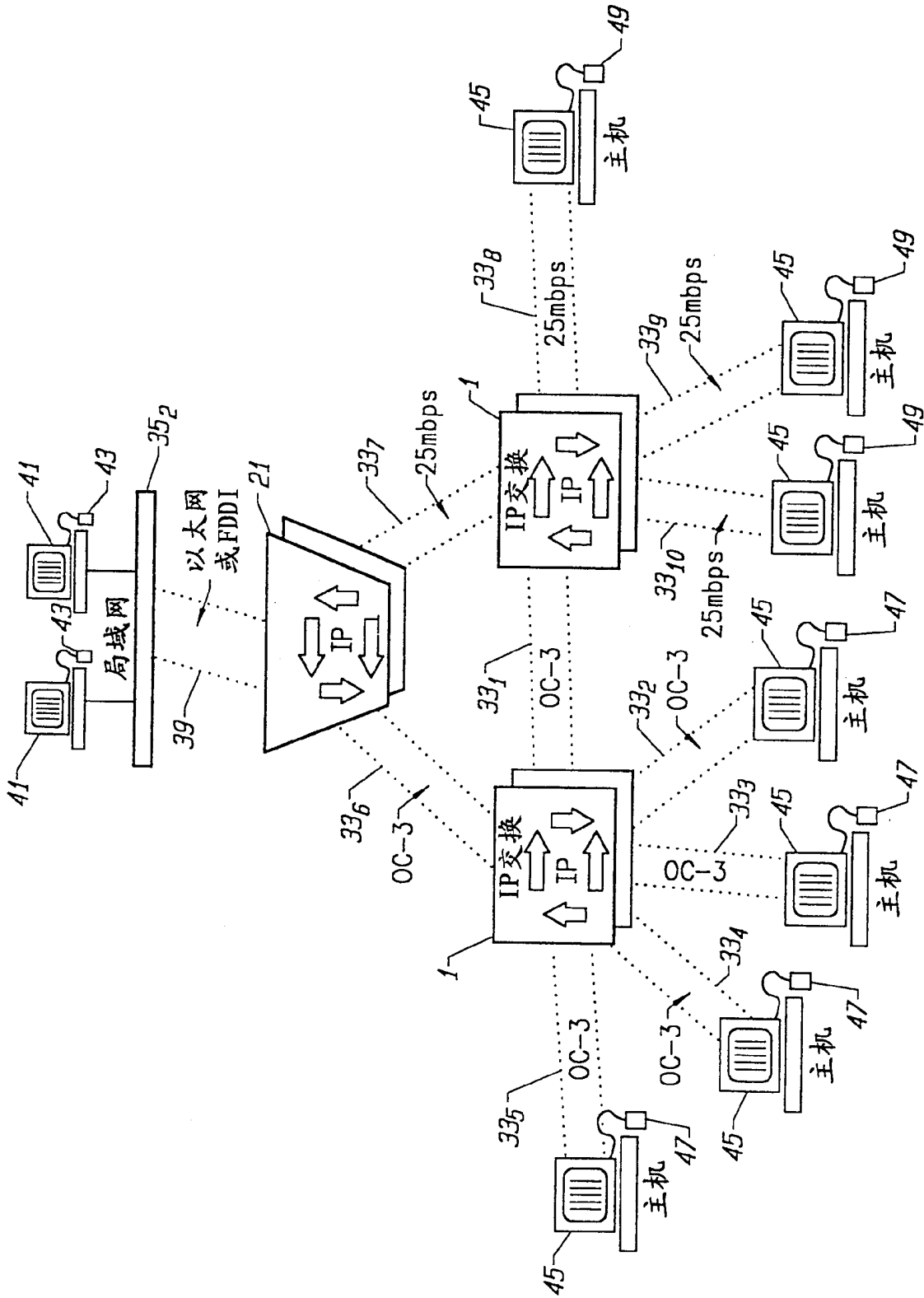


图 2b

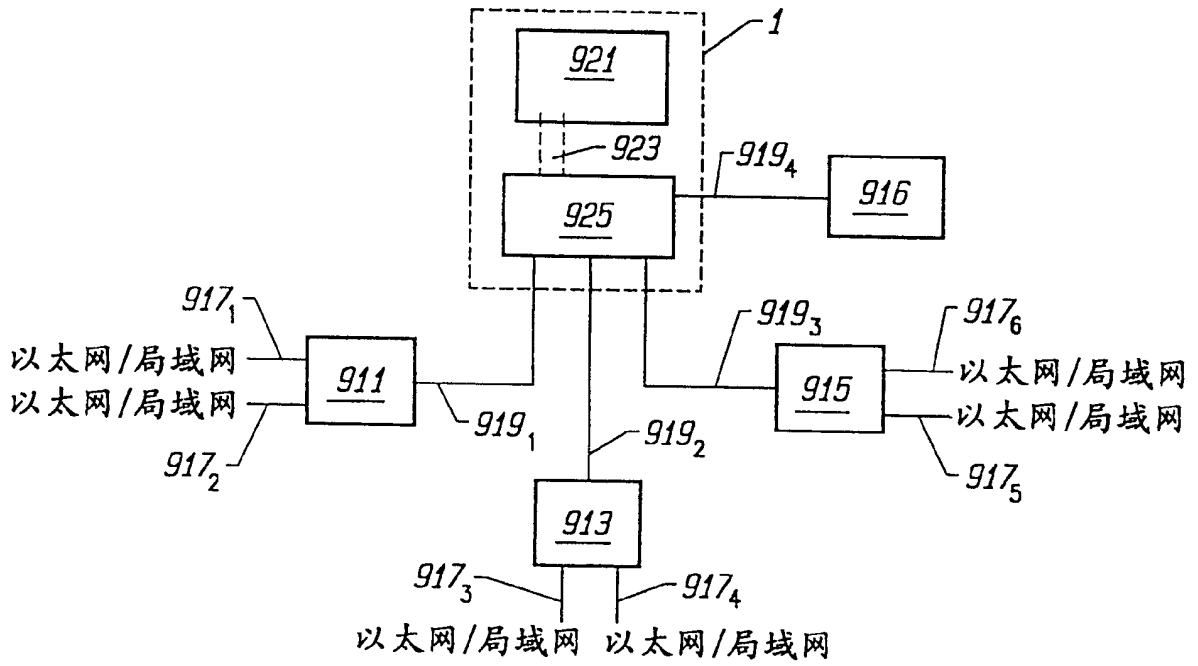


图 2c

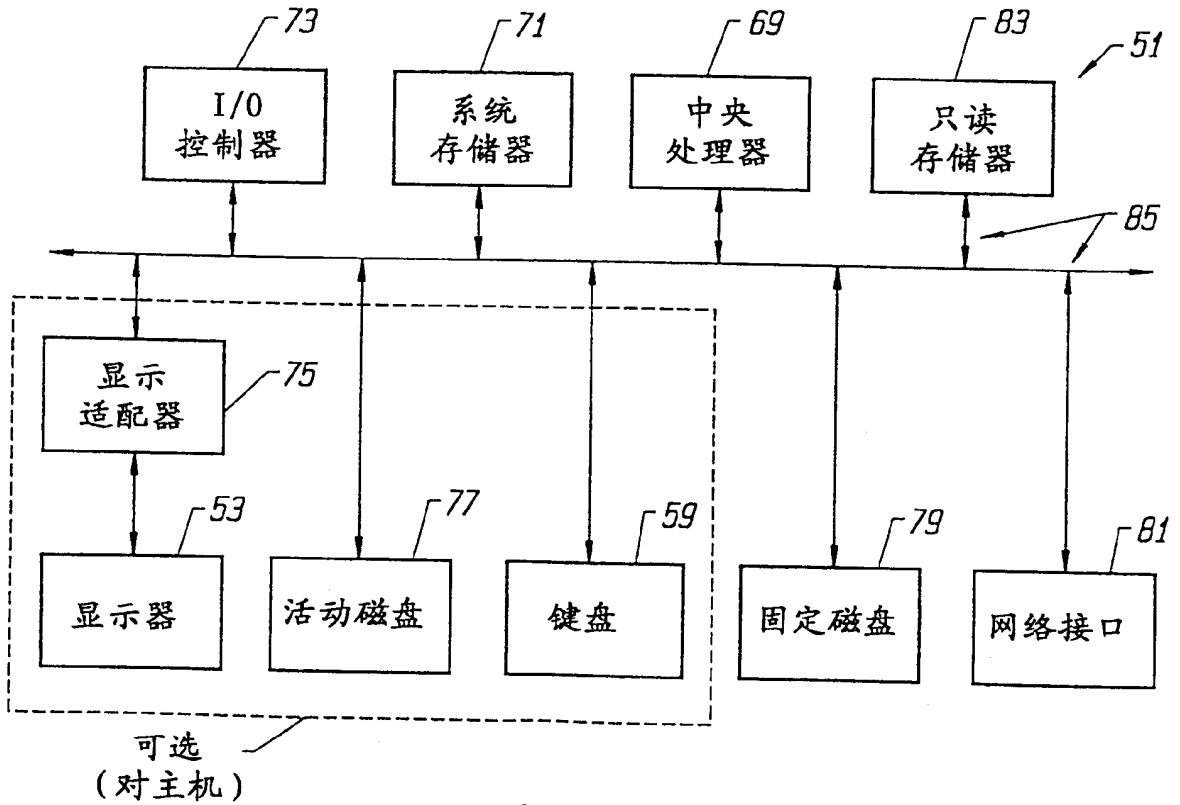


图 3

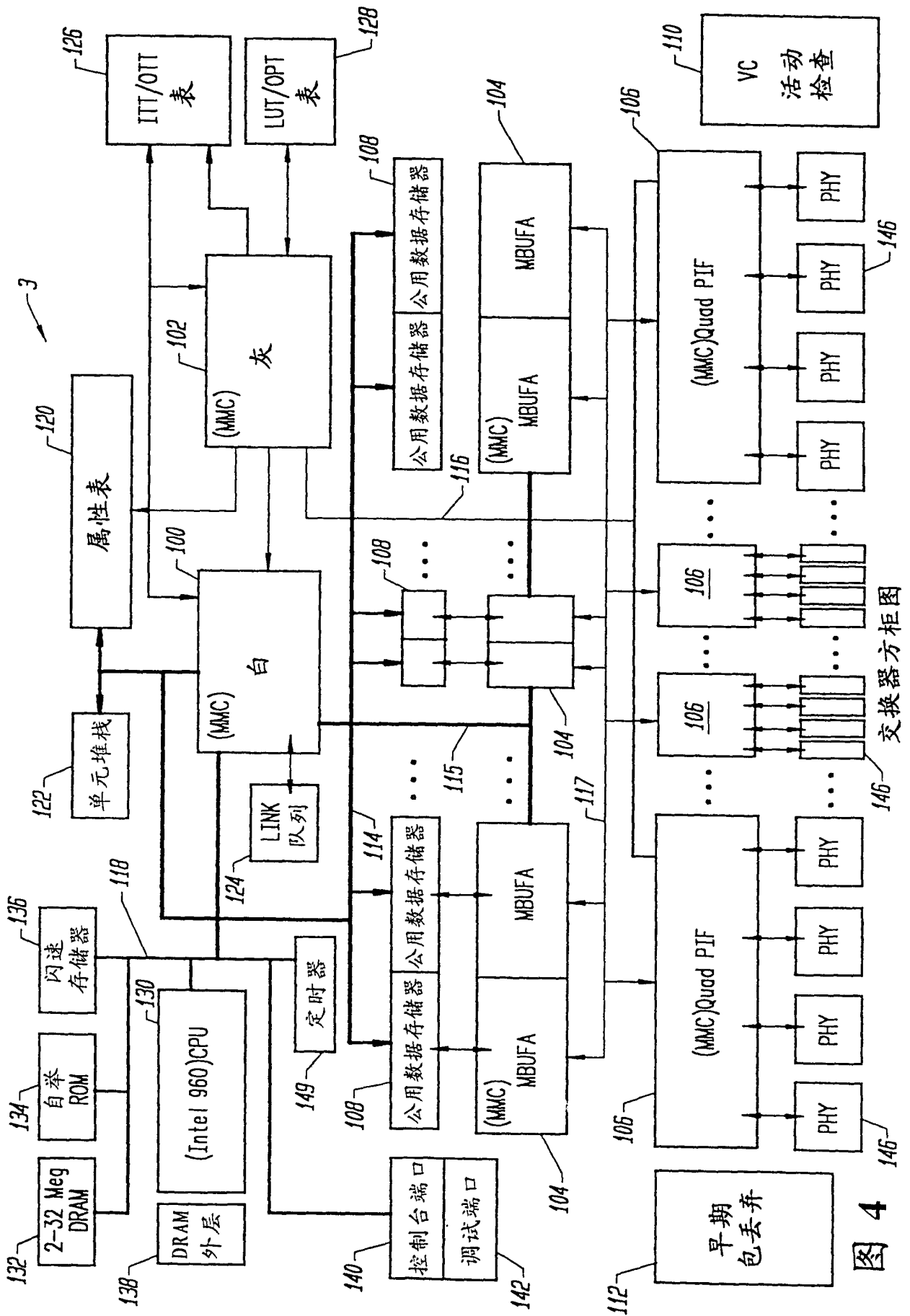


图 4



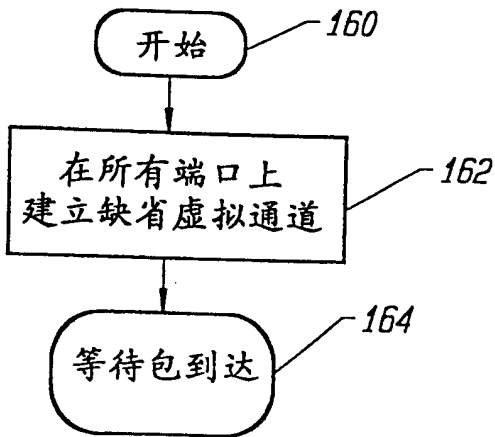


图 5a

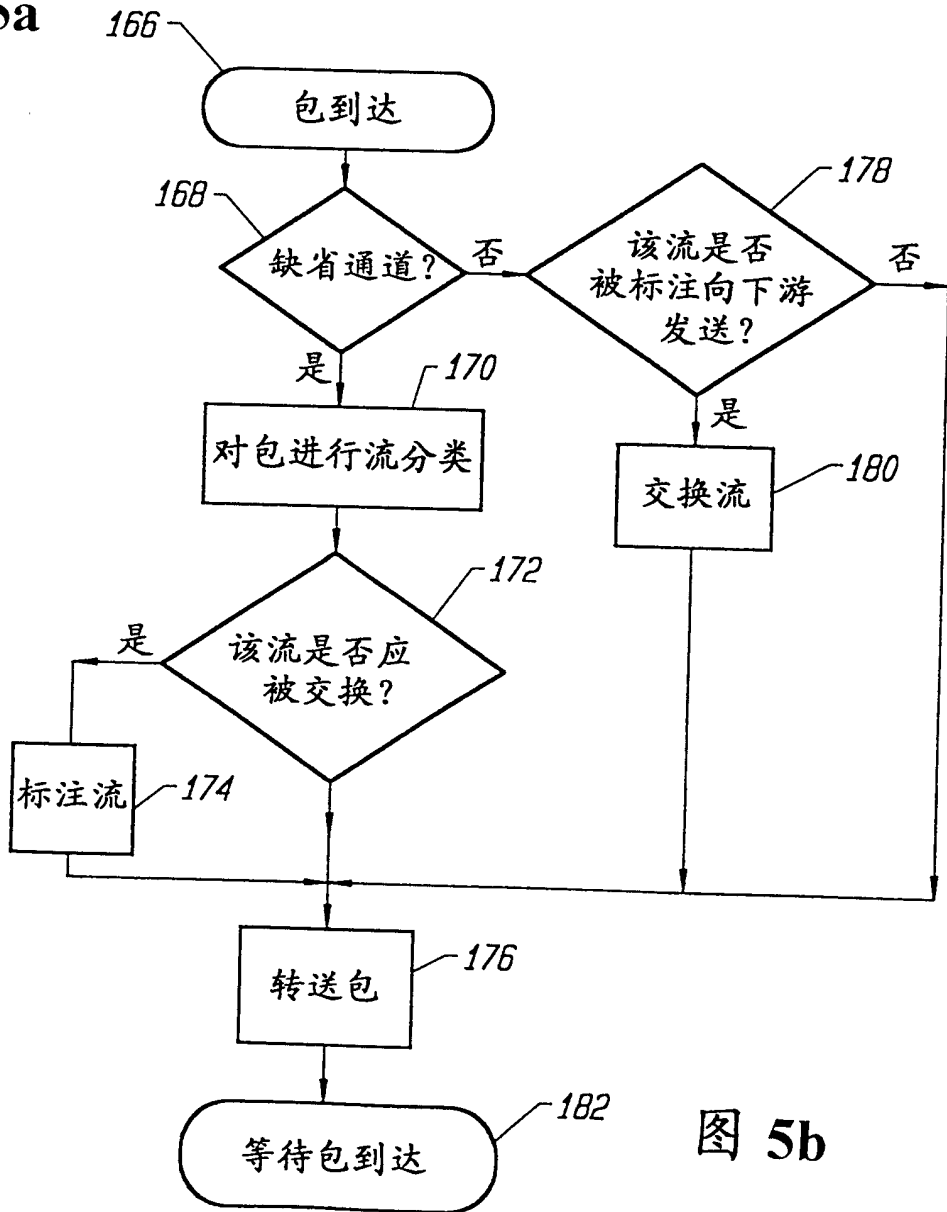


图 5b

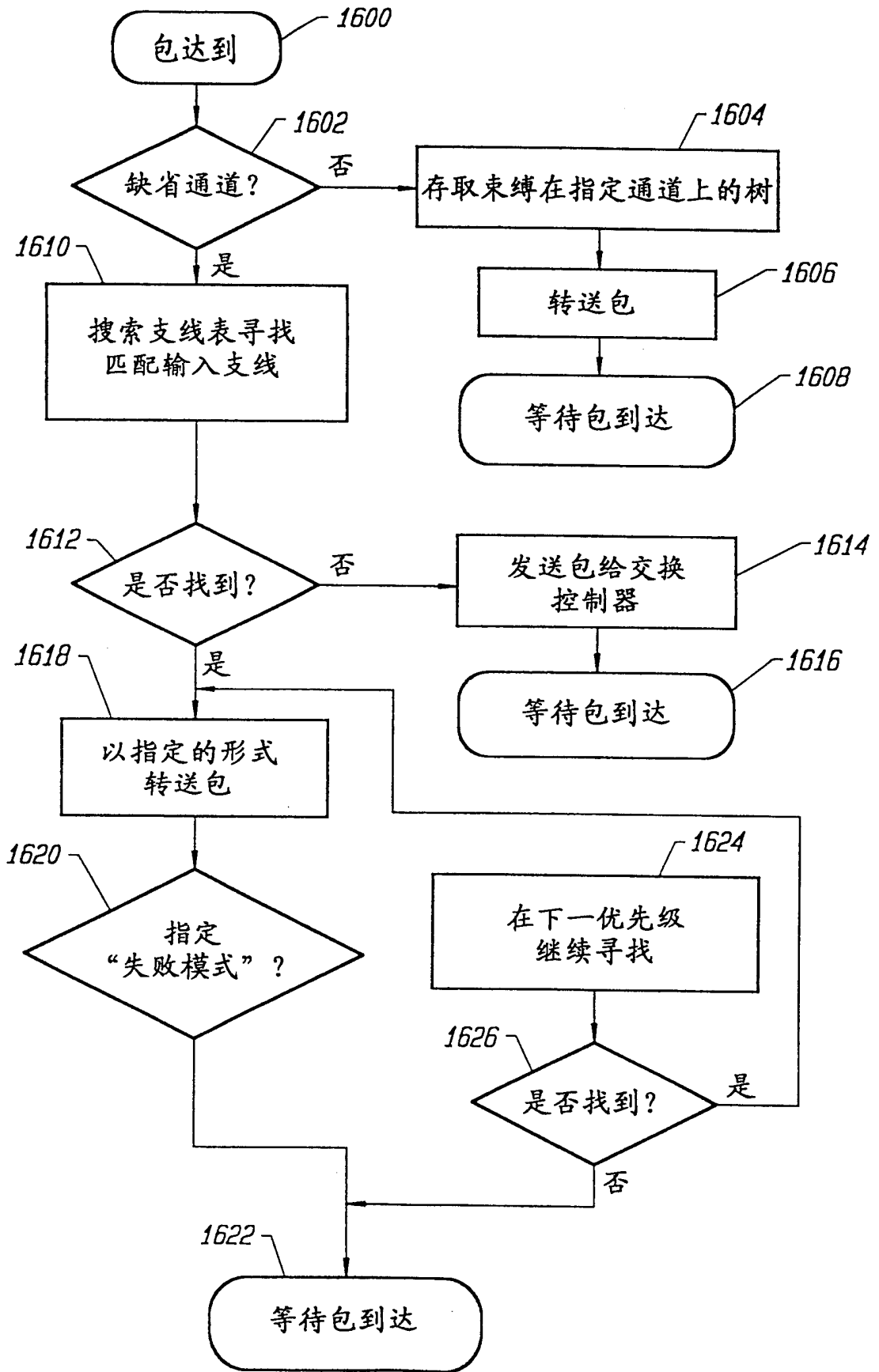


图 5c

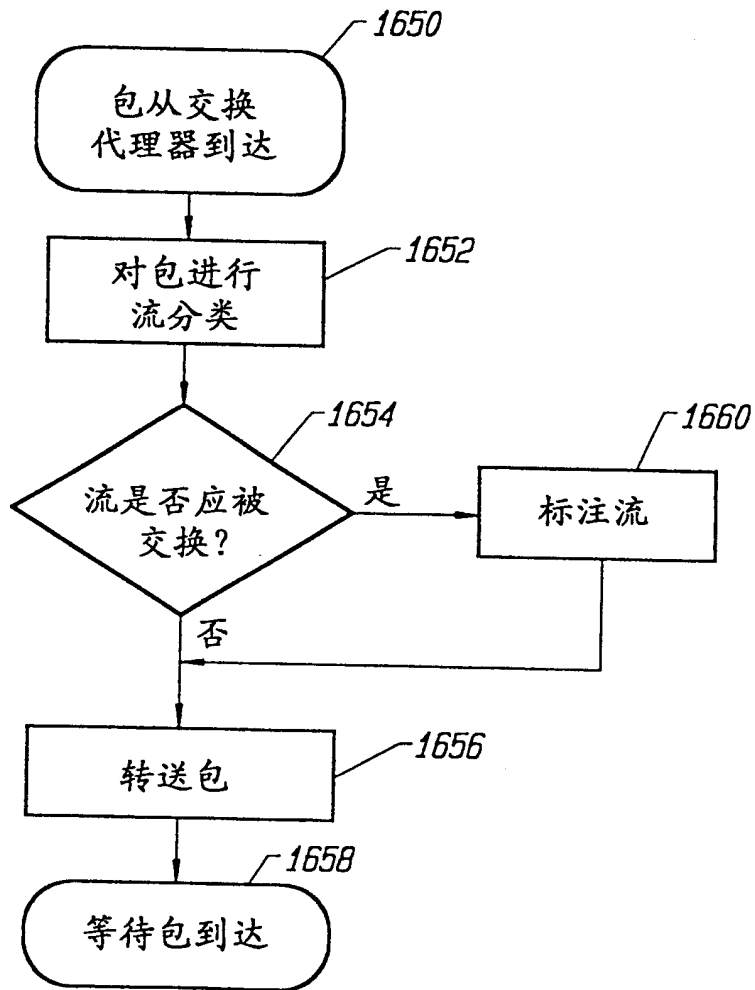


图 5d

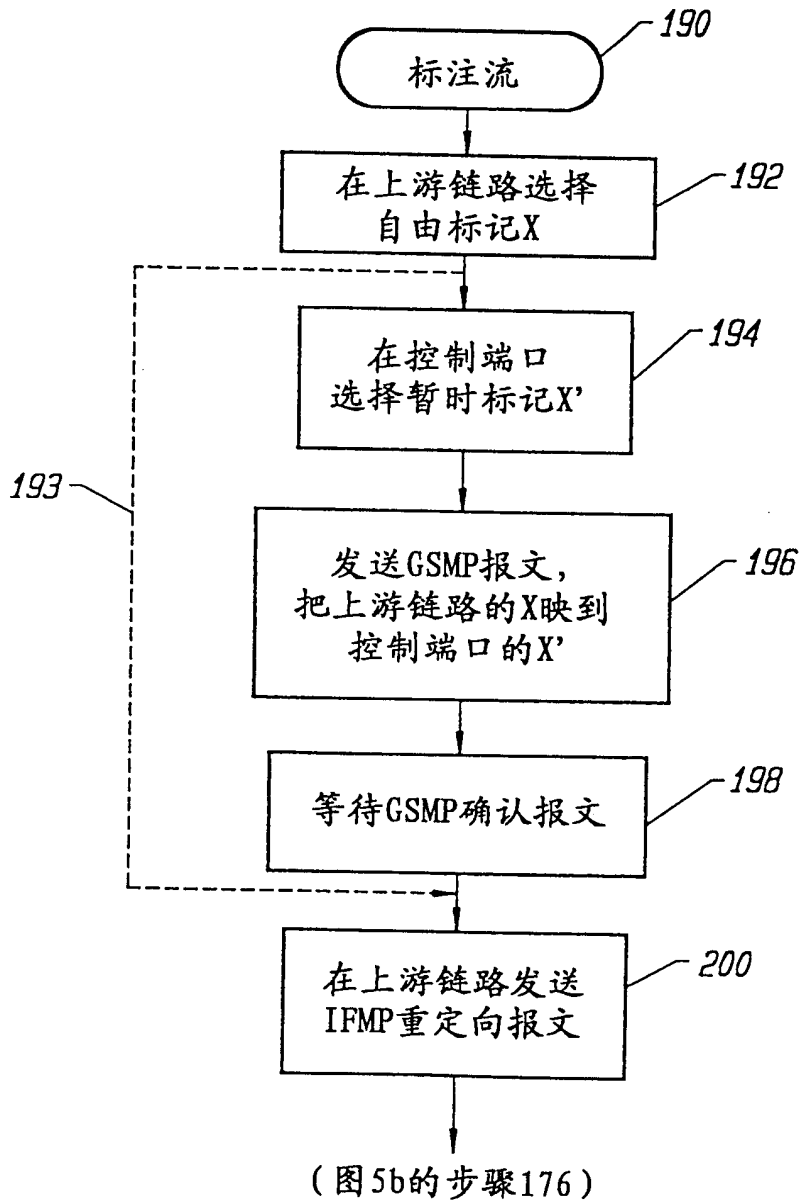
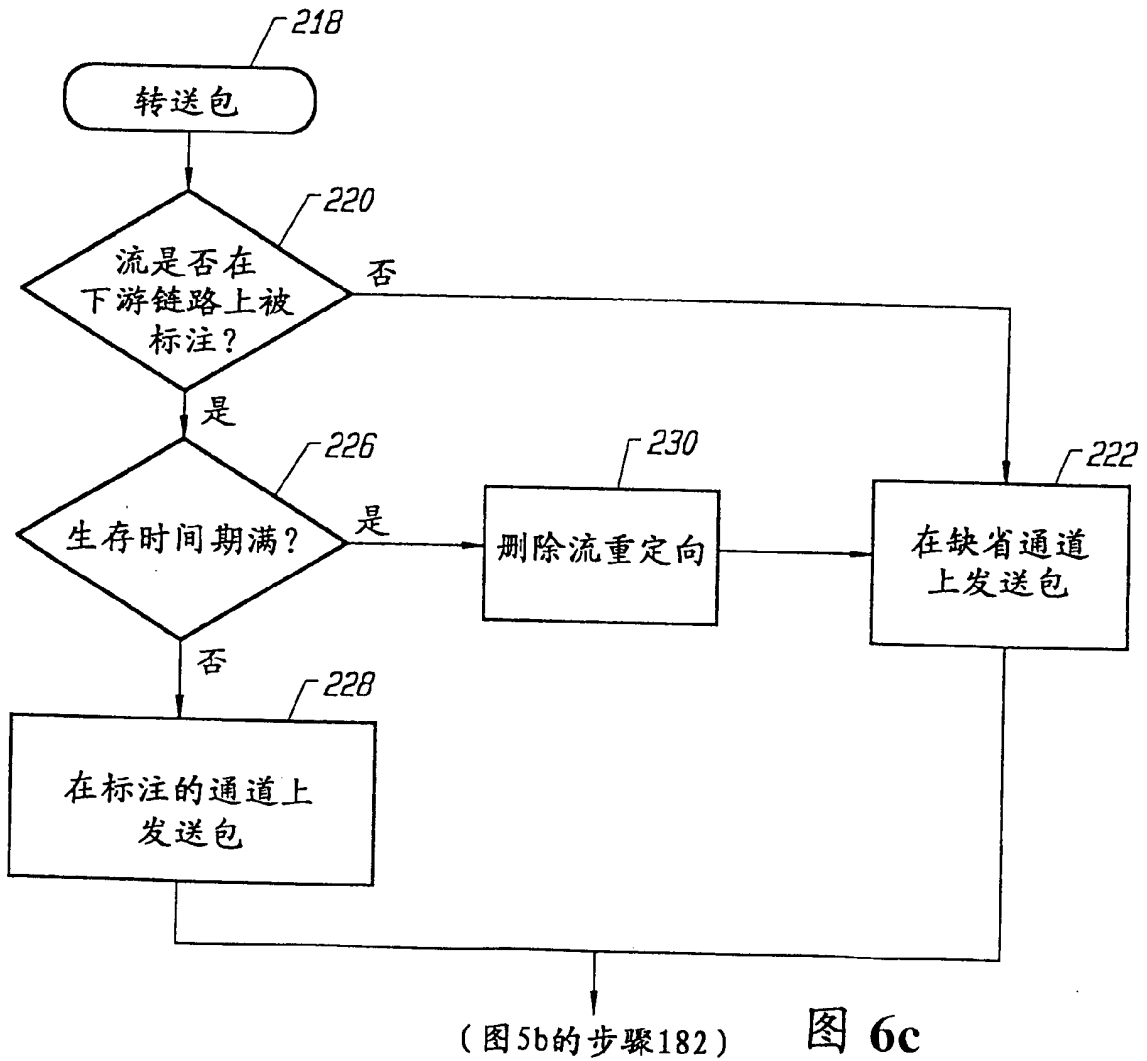
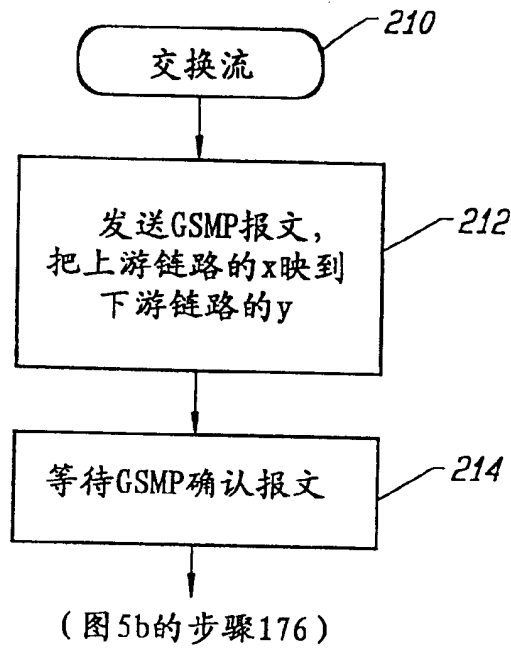


图 6a

图 6b



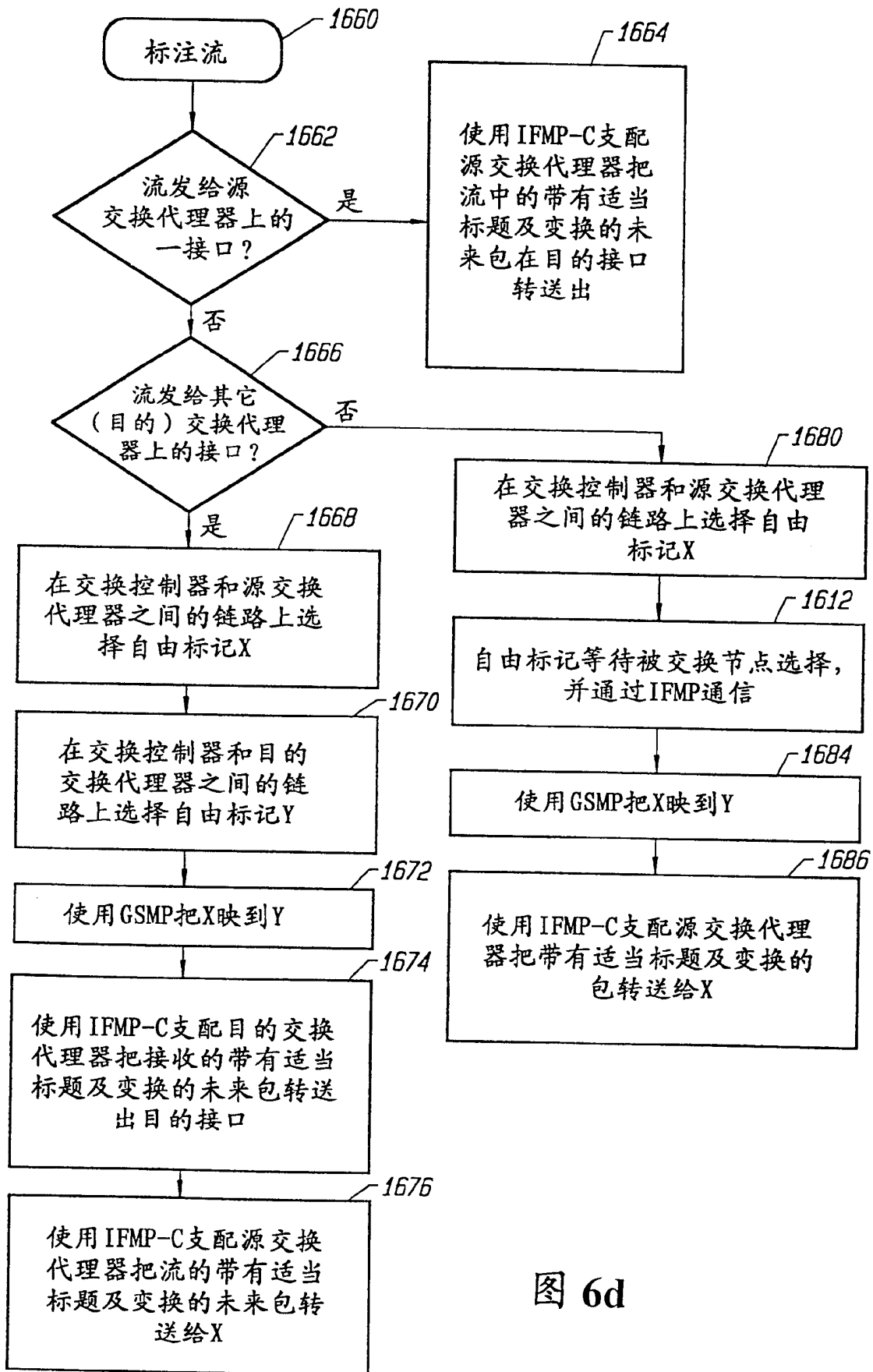


图 6d

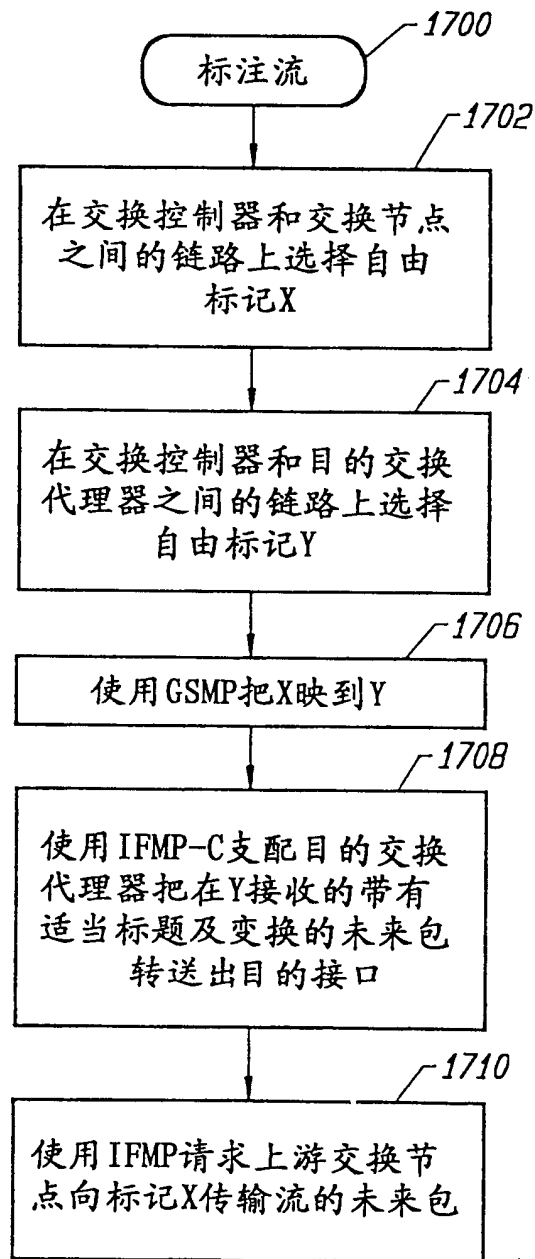


图 6e

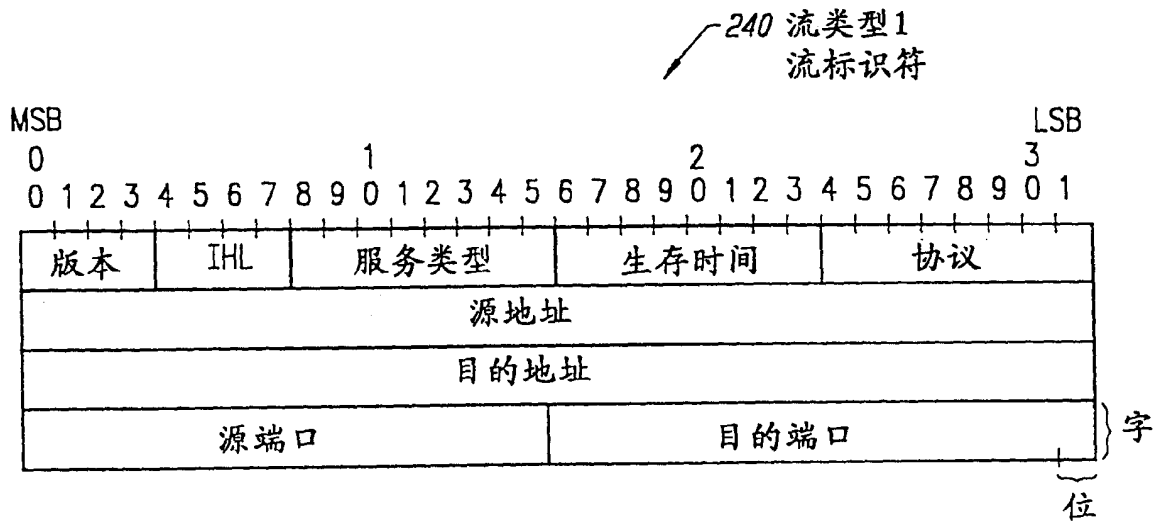


图 7a

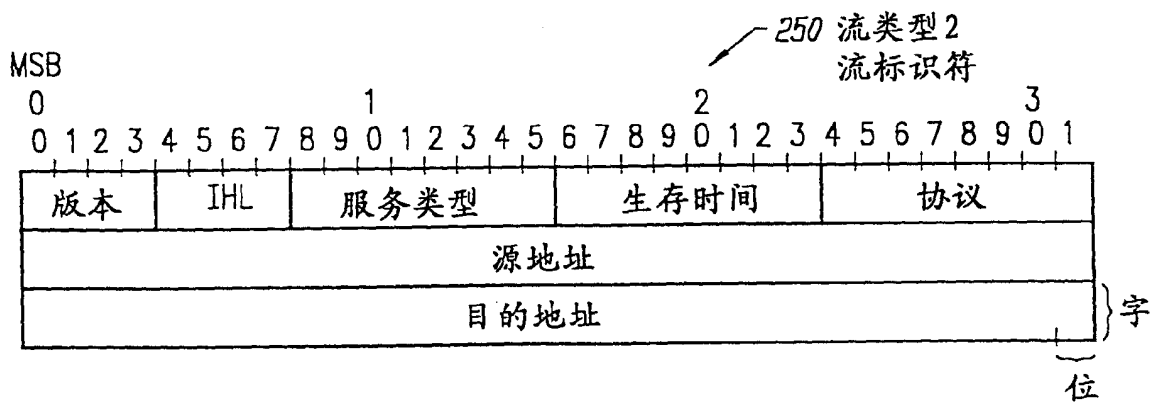


图 7b



300 类属 IFMP 相邻  
协议报文

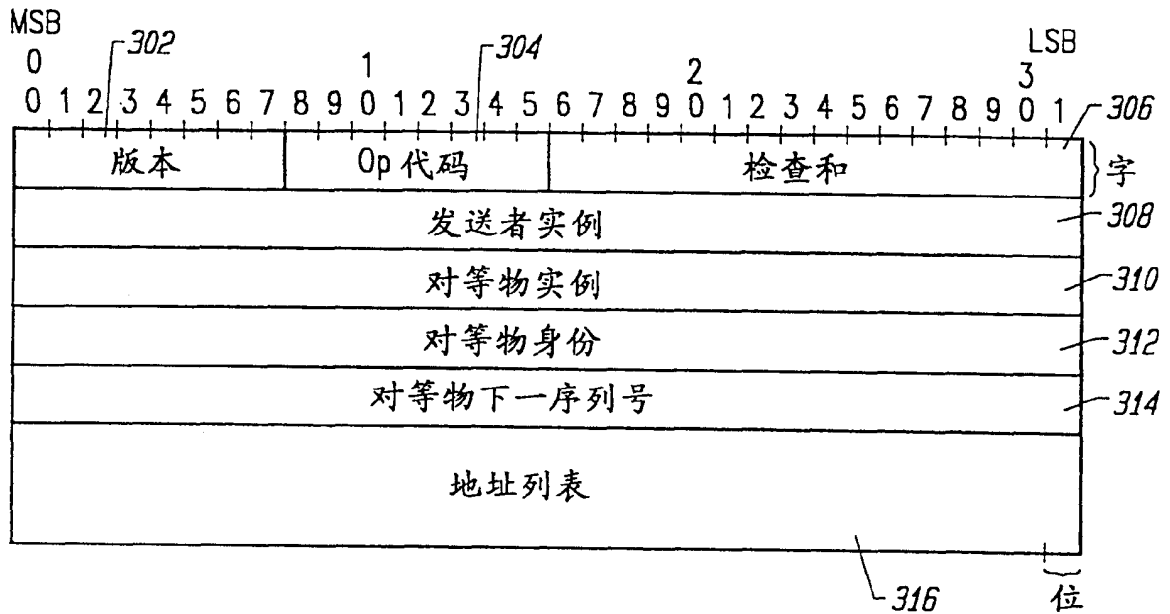
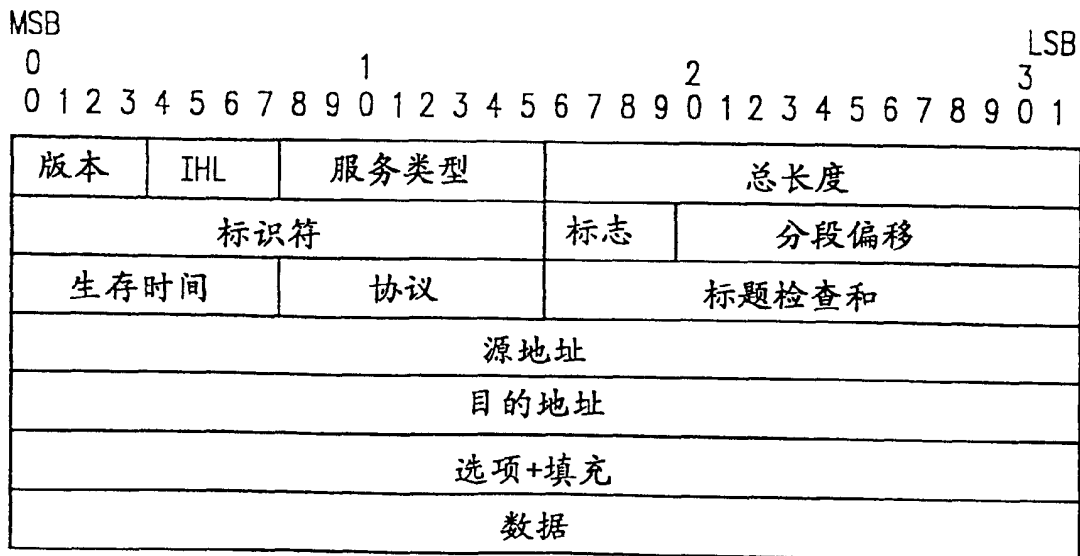


图 8a



类属 IP 包 (IPv4)

图 8b

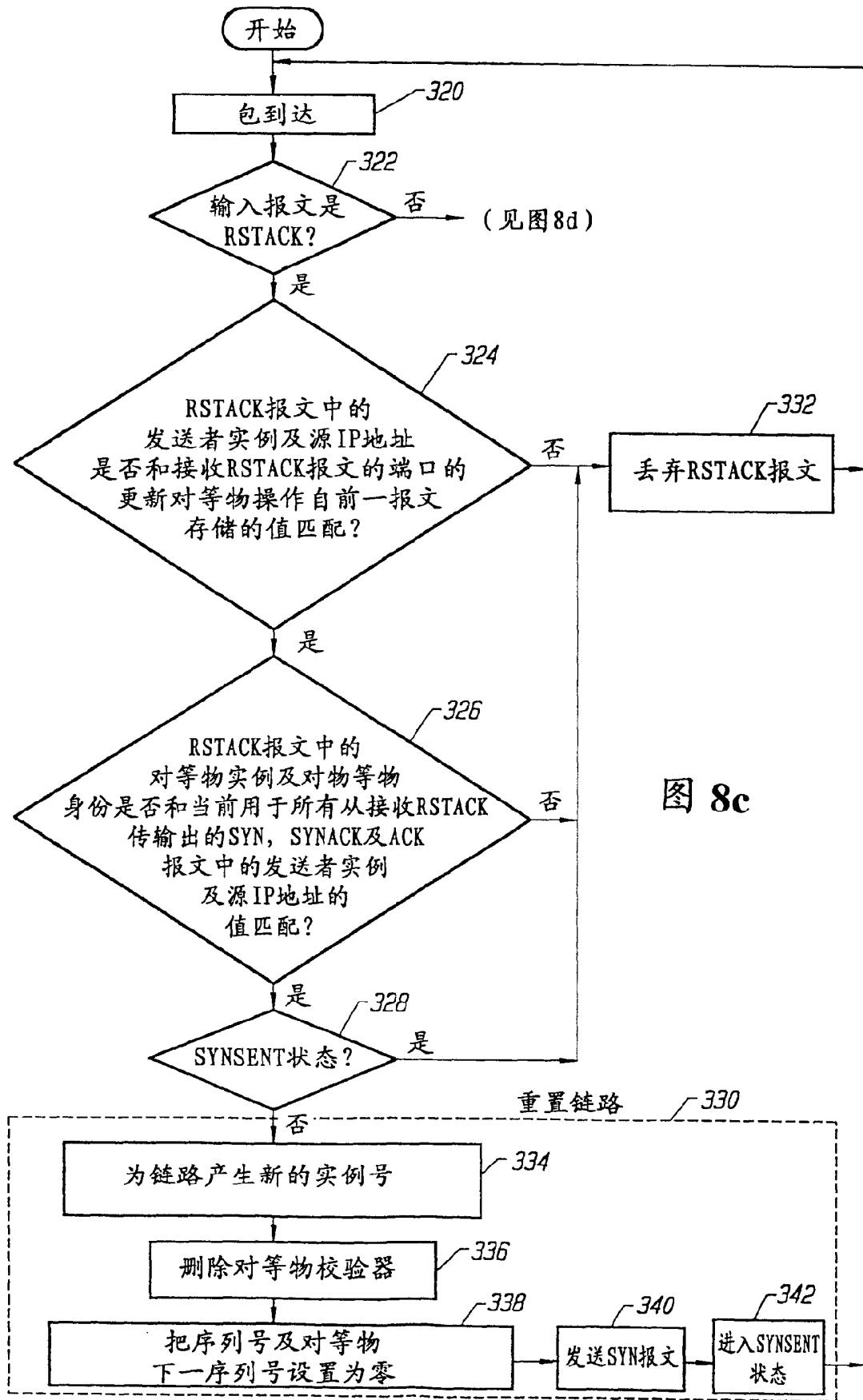
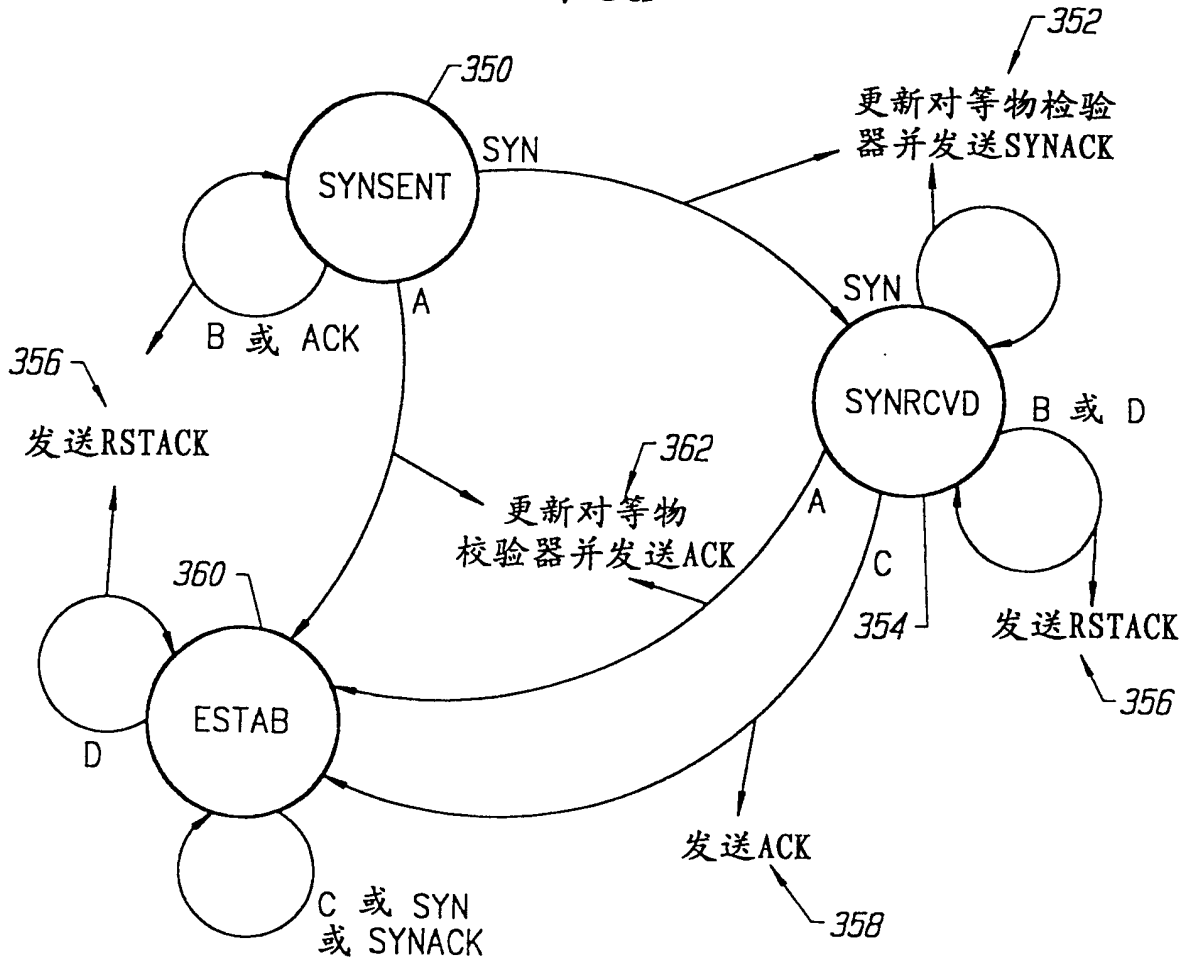


图 8c

图 8d



图例

- A: SYNACK与%*C*
- B: SYNACK与非(%*C*)
- C: ACK与%*B*与%*C*
- D: ACK与非(%*B*与%*C*)

其中%*B*: 输入报文中的发送者实例与源IP地址和接收输入报文的端口的更新对等物操作自前一报文存储的值匹配

其中%*C*: 输入报文中的对等物实例及对等物身份和当前用于所有从接收输入报文的端口传输的SYN, SYNACK及ACK报文中的发送者实例及源IP地址的值匹配

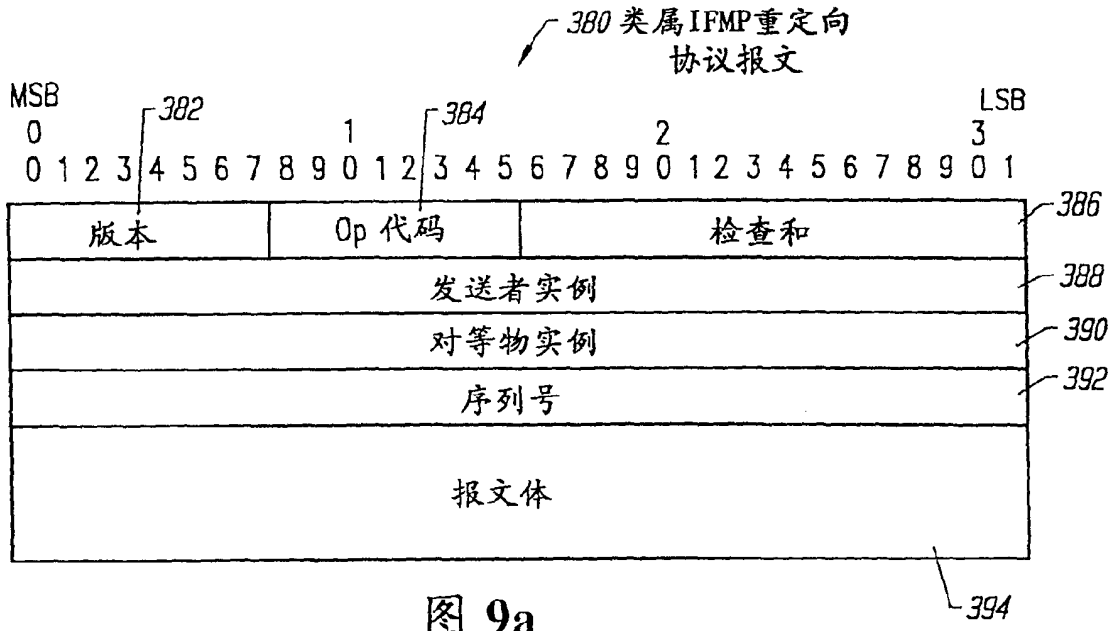


图 9a

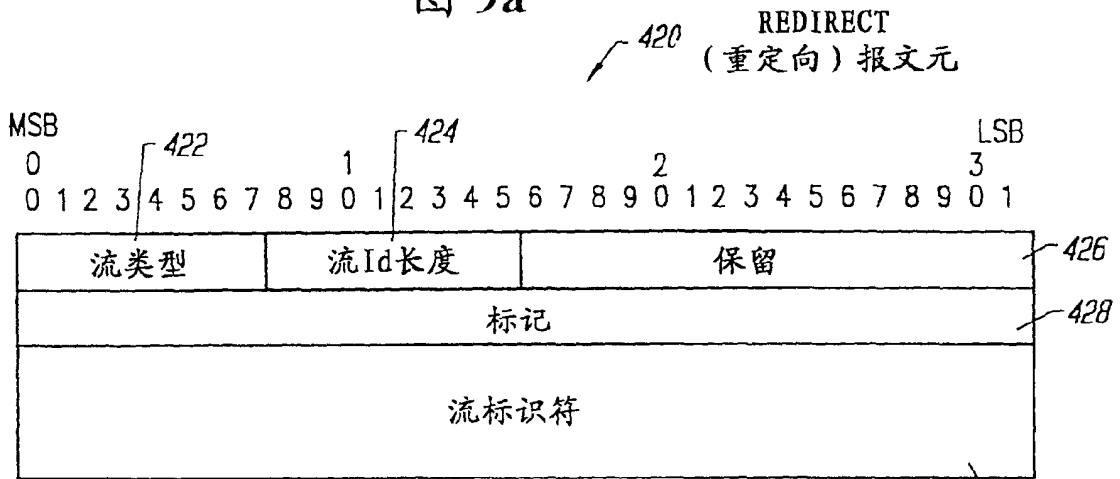


图 9c

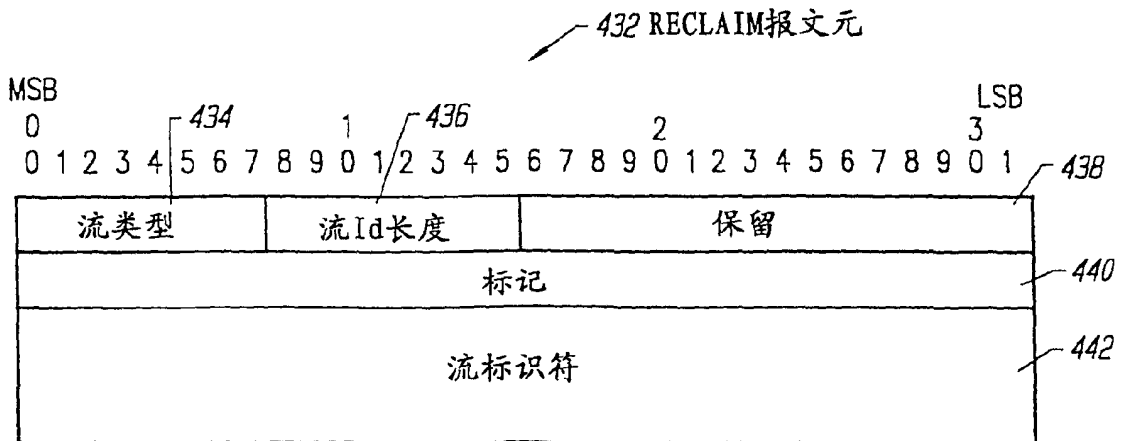


图 9d

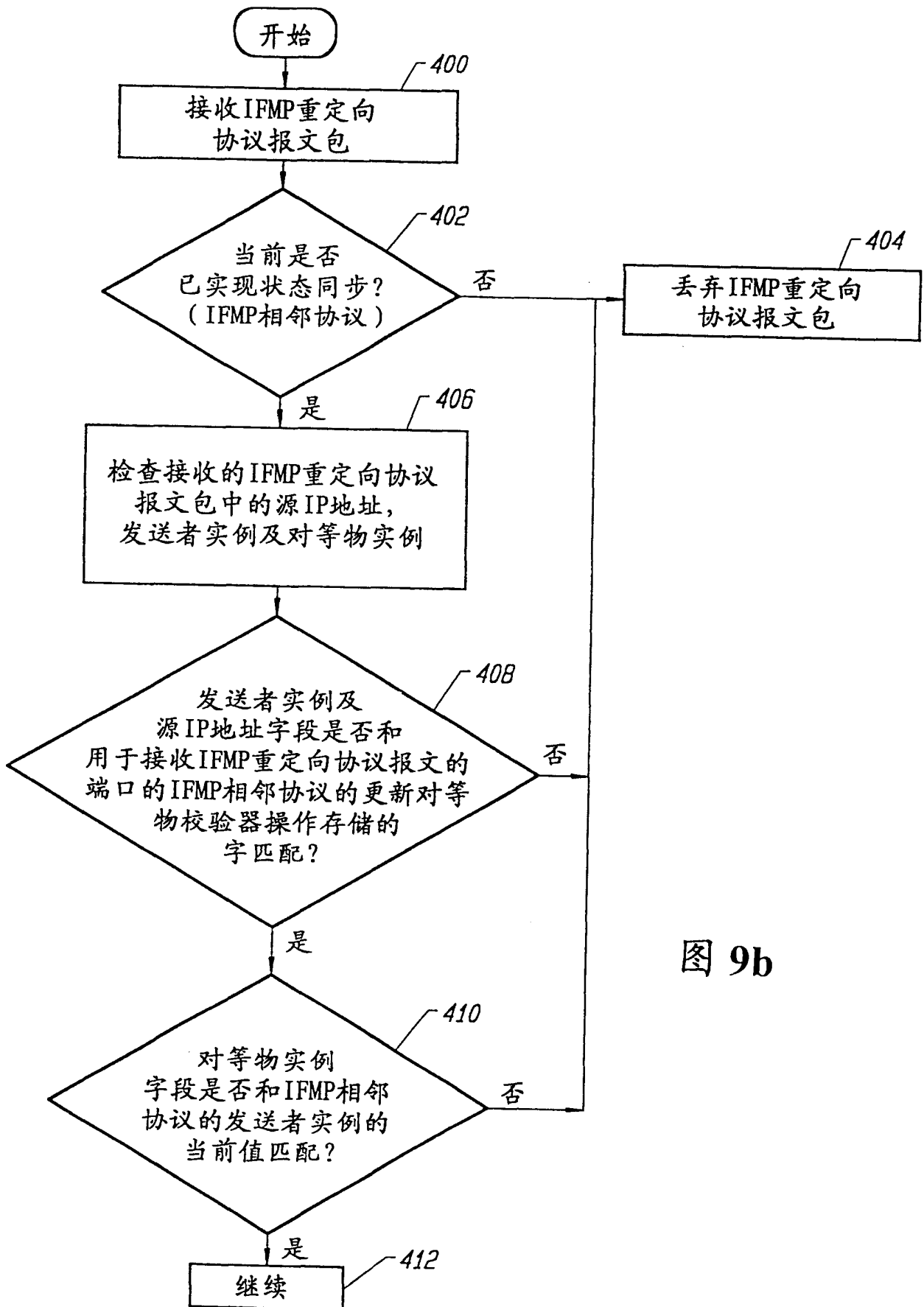


图 9b

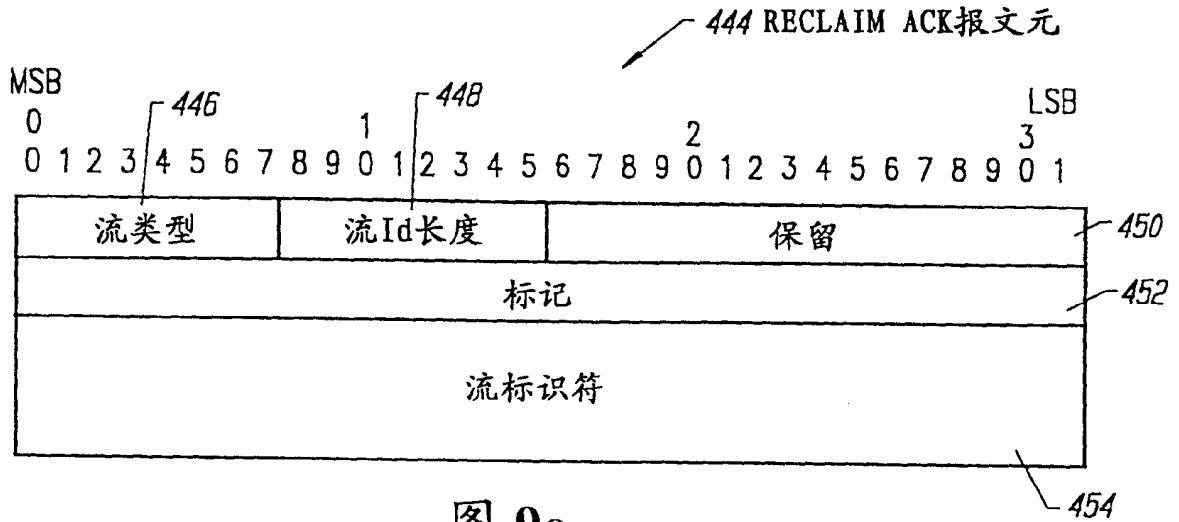


图 9e

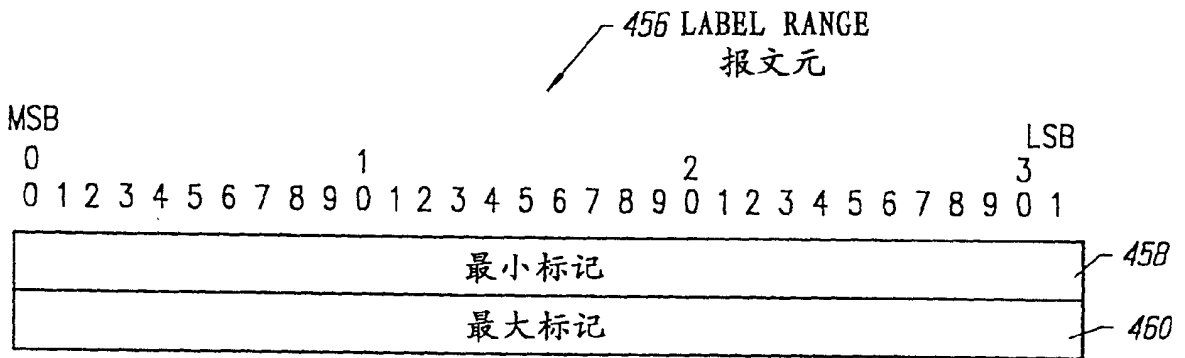


图 9f

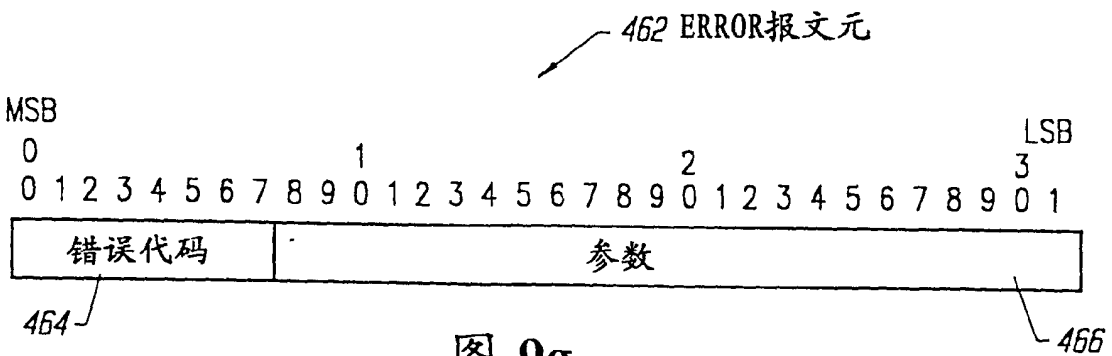


图 9g

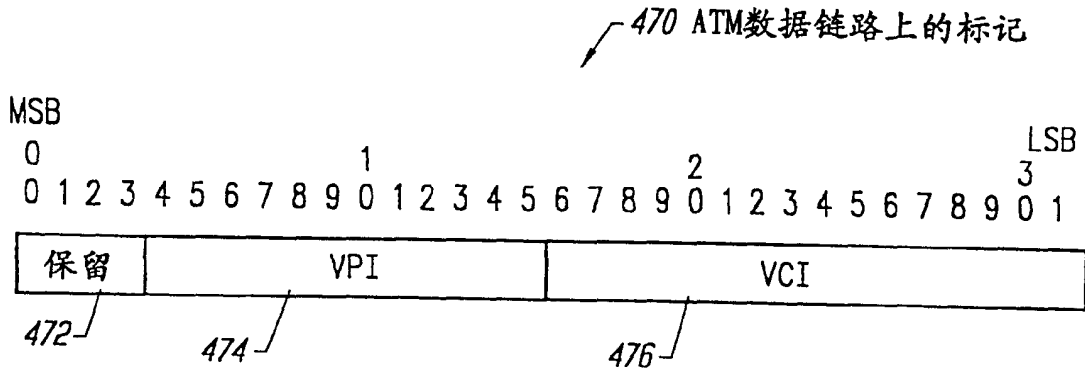


图 10a

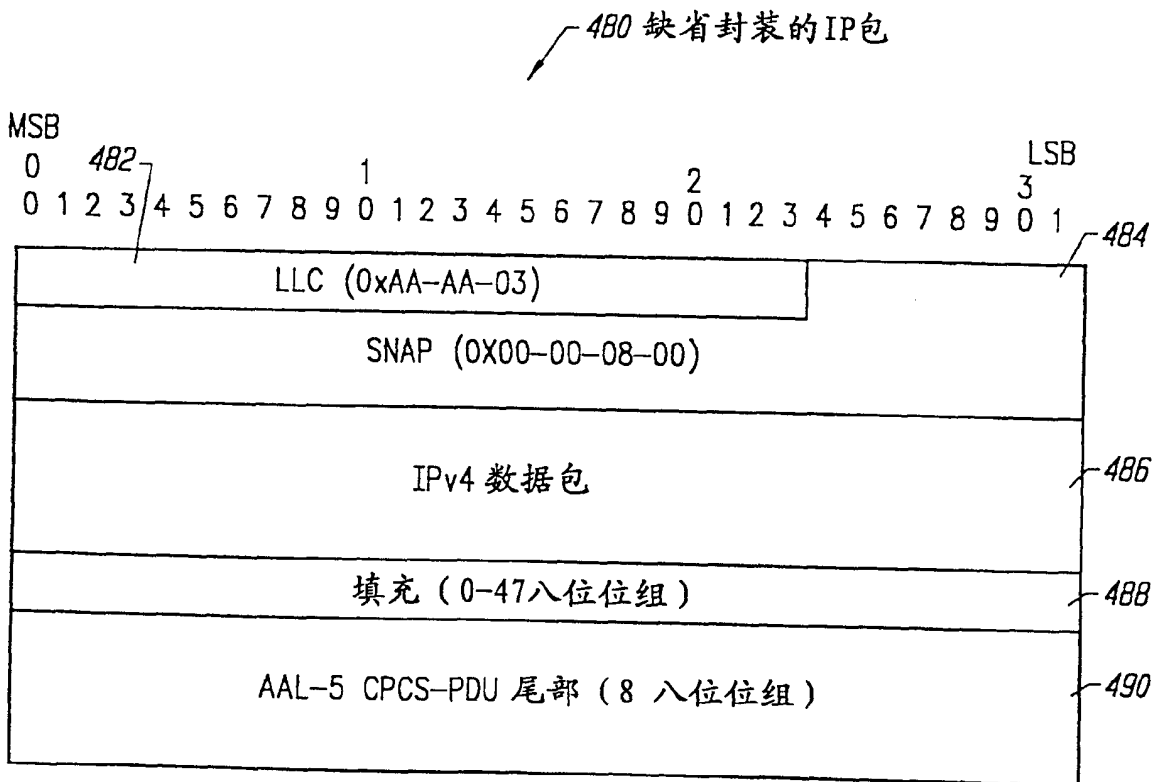


图 10b

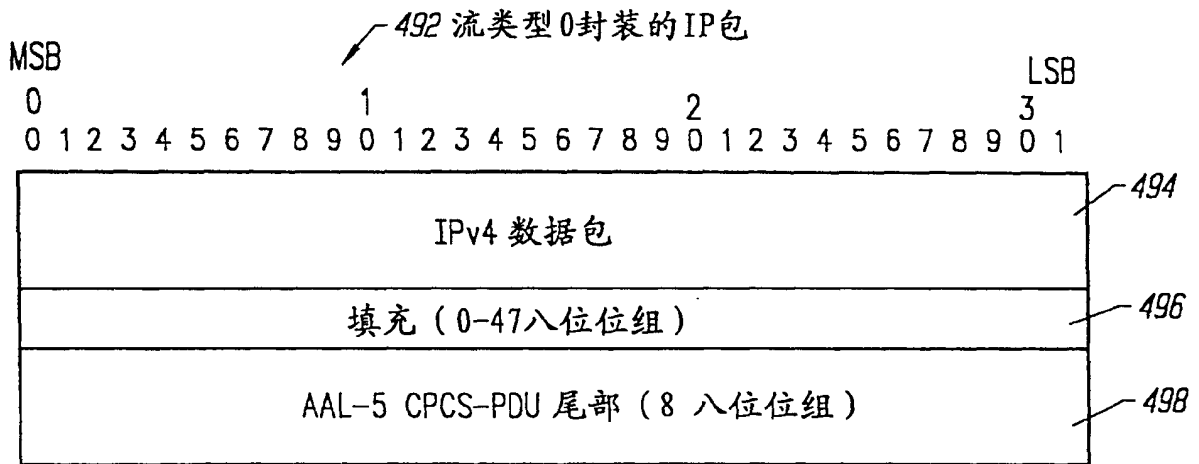


图 10c

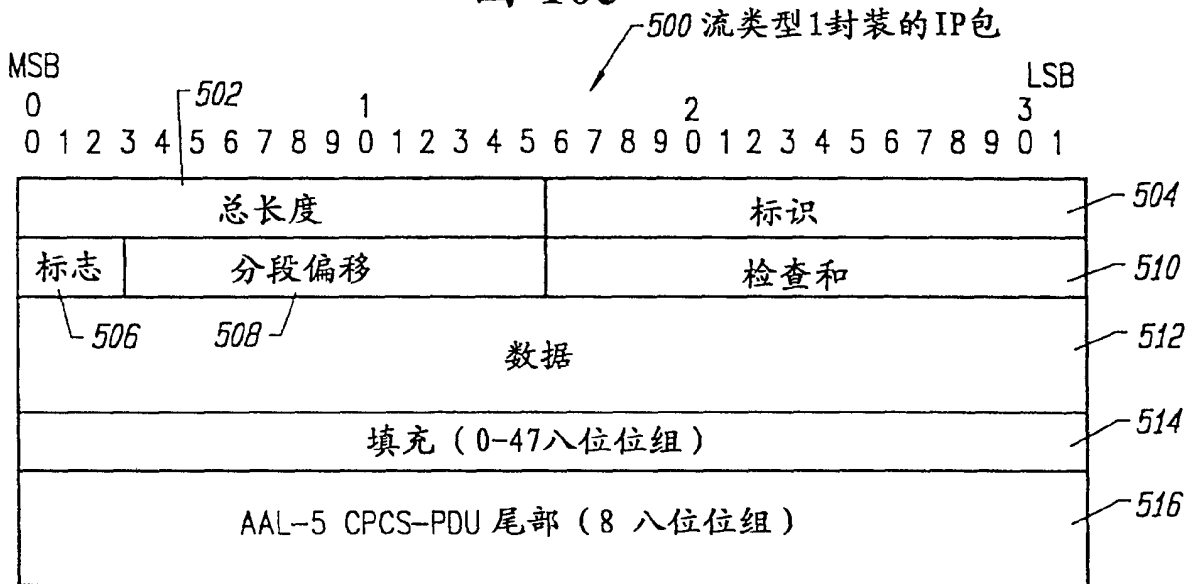


图 10d

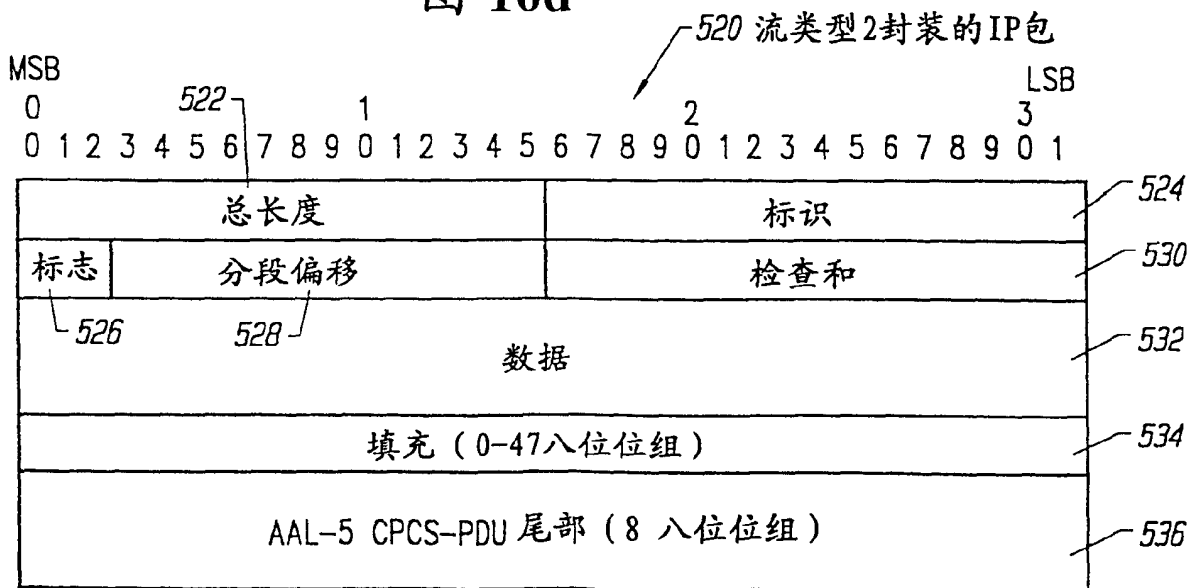


图 10e



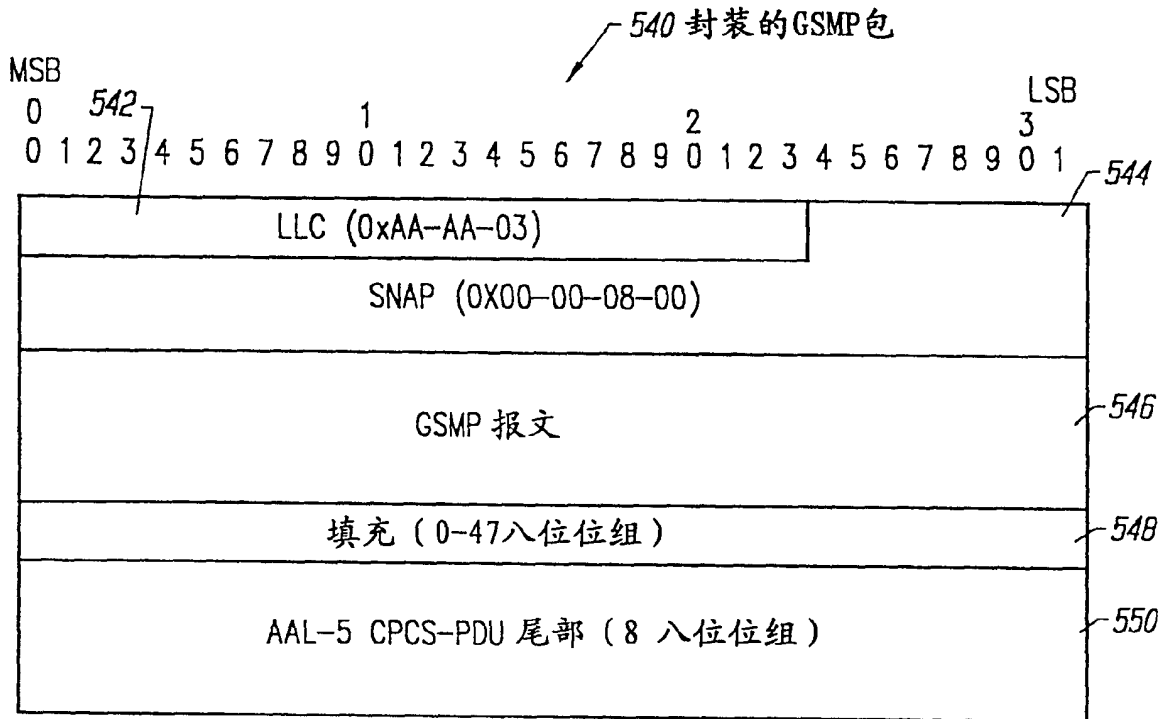


图 11a

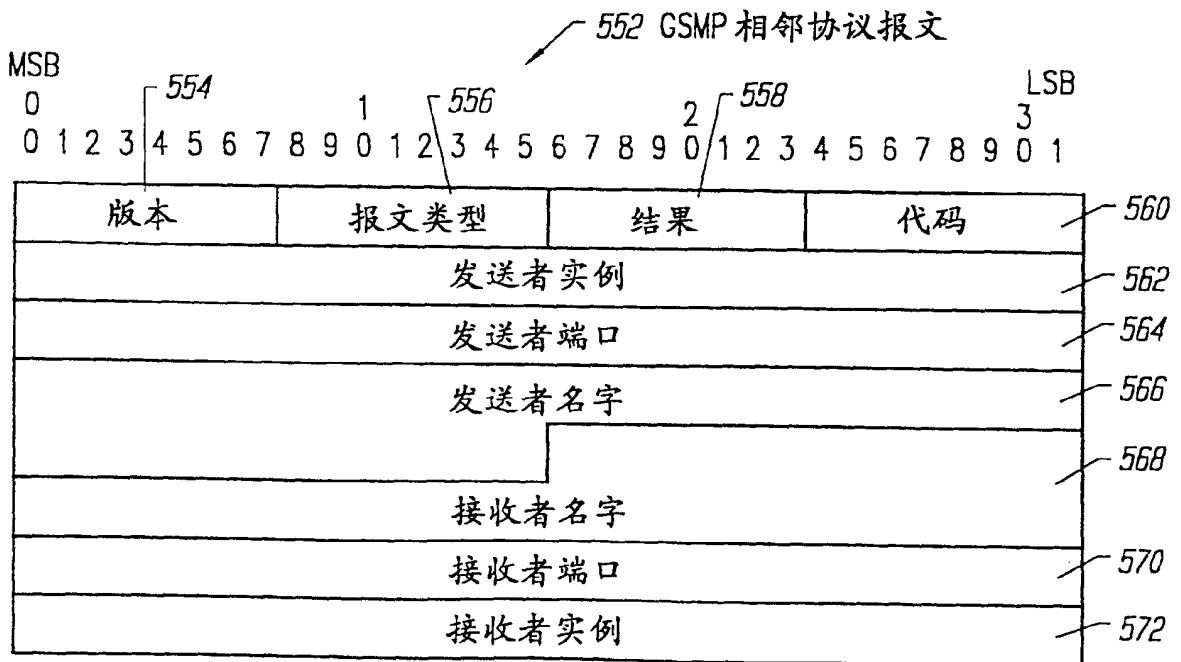


图 11b

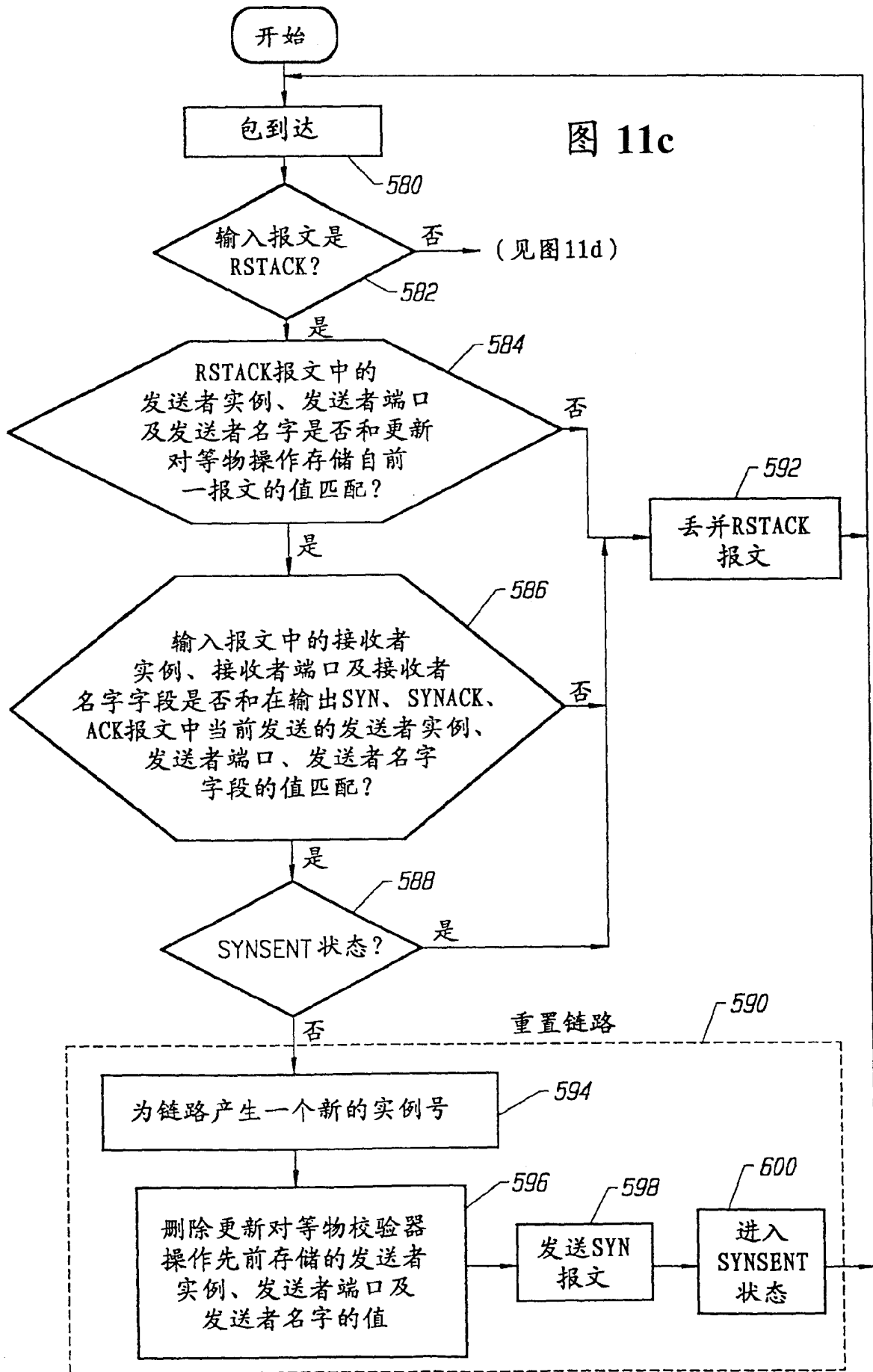
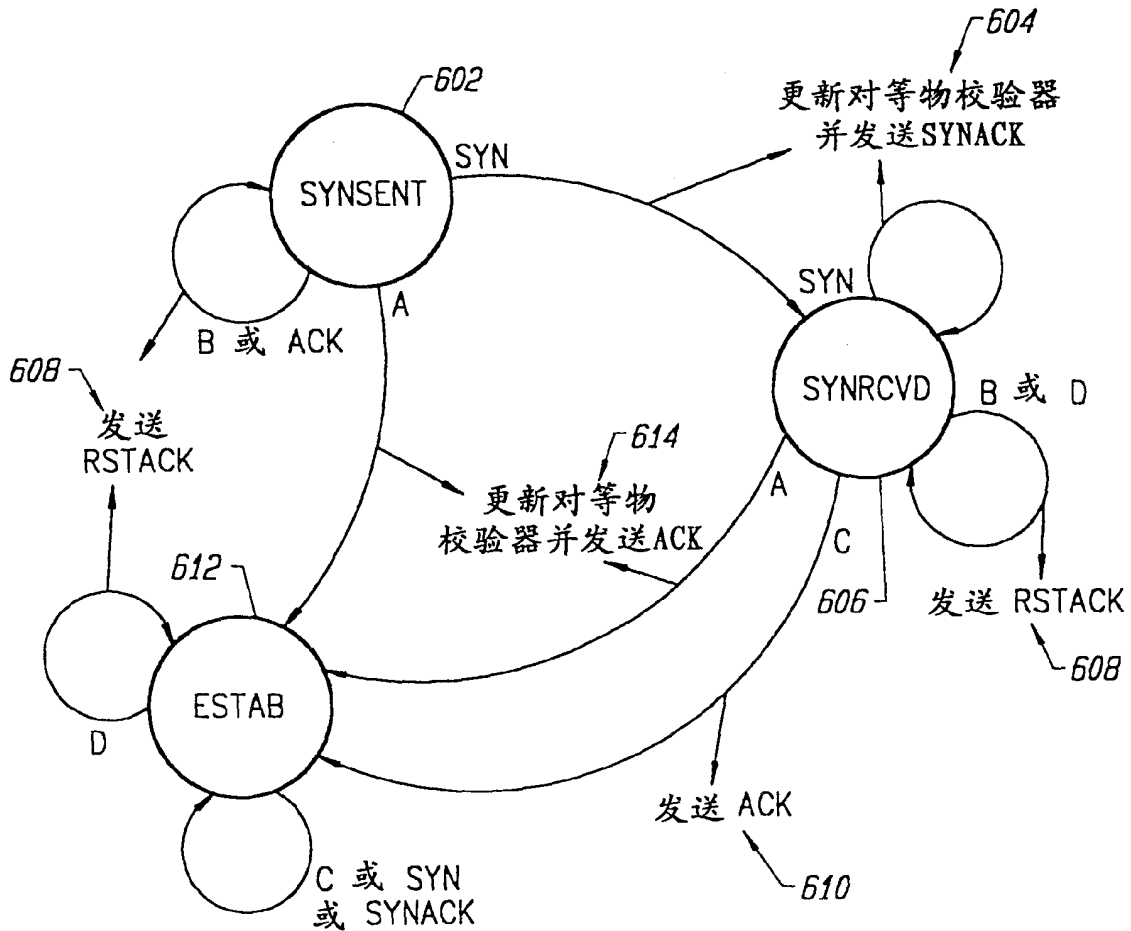


图 11d



图例

- A: SYNACK 与 %C
- B: SYNACK 与非 (%C)
- C: ACK 与 %B 与 %C
- D: ACK 与非 (%B 与 %C)

其中%B: 输入报文中的发送者实例、发送者端口及发送者名字和更新对等物校验器操作存储自前一报文的值匹配

其中%C: 输入报文中的接收者实例、接收者端口及接收者名字和当前在输出SYN、SYNACK及ACK报文中发送的发送者实例、发送者端口及发送者名字的值匹配

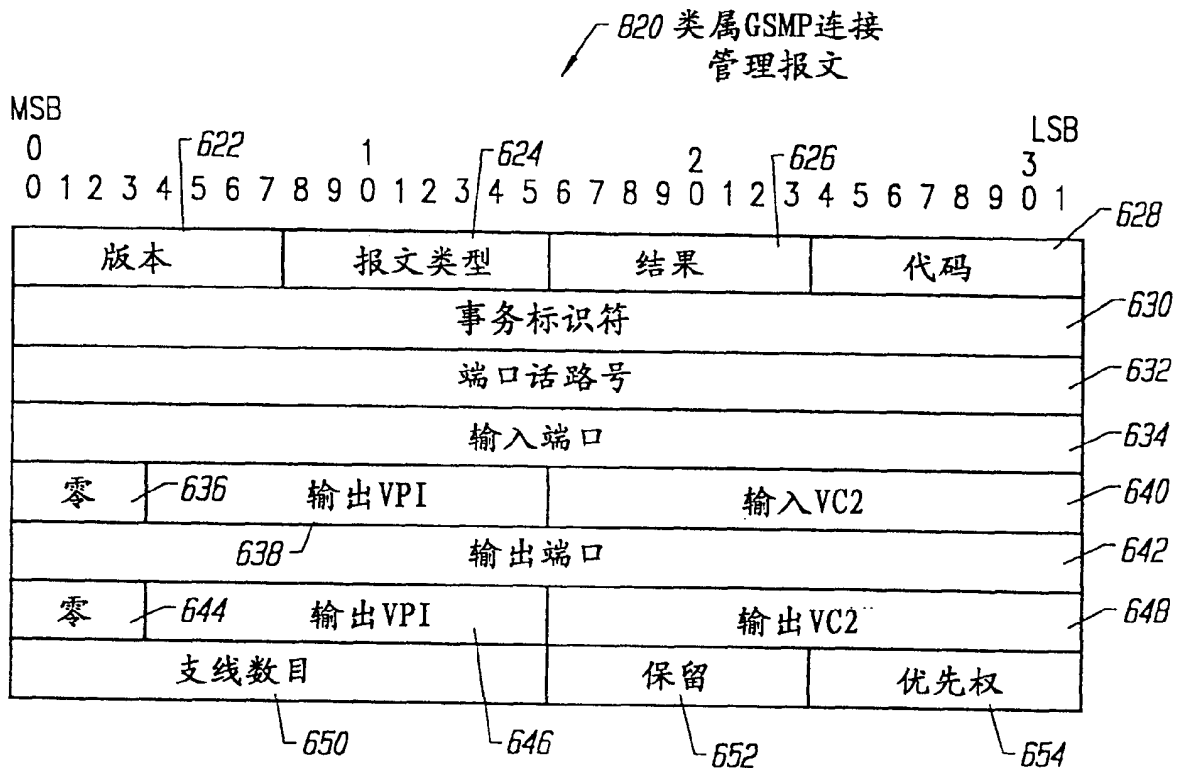


图 12

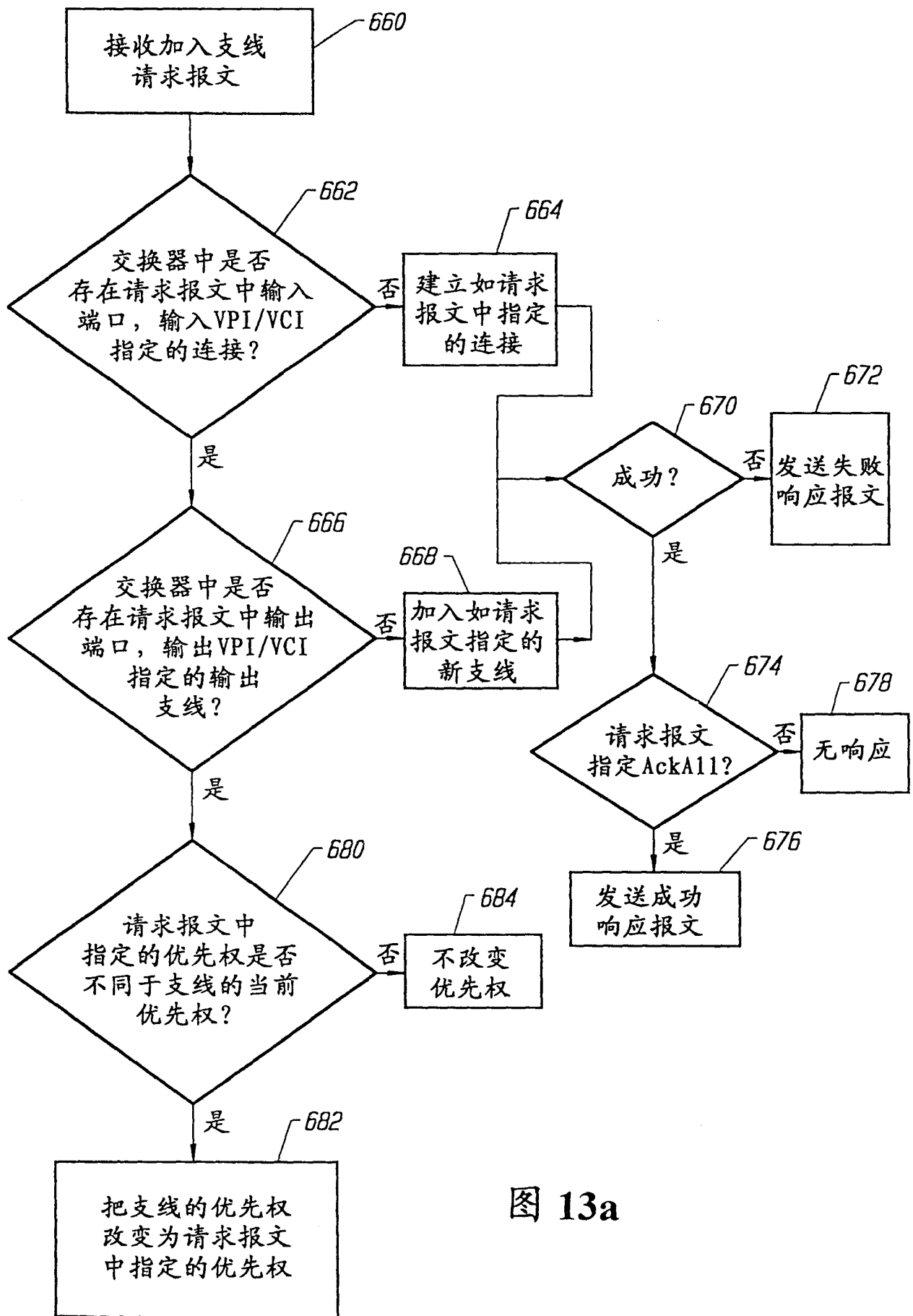


图 13a

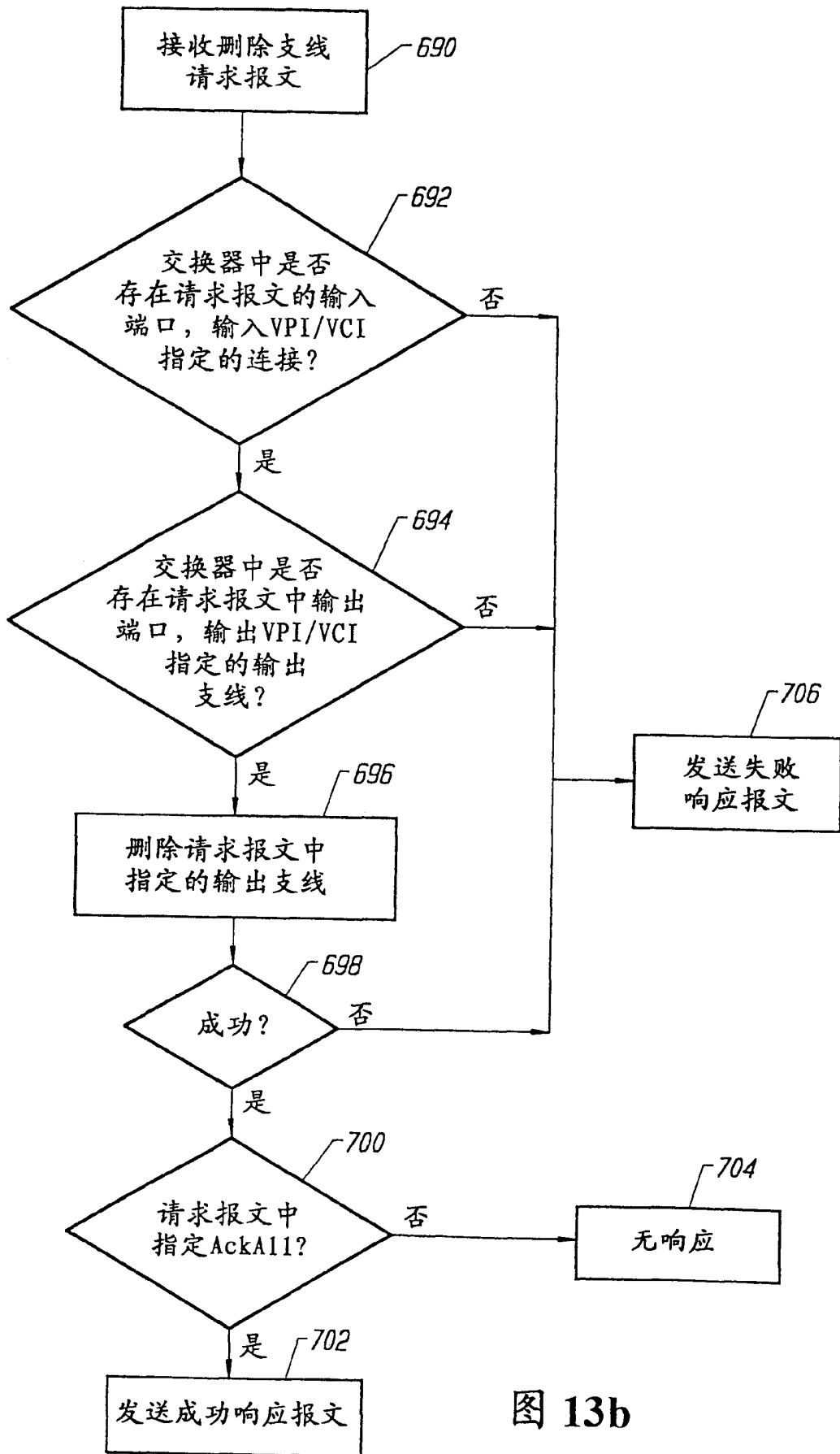


图 13b

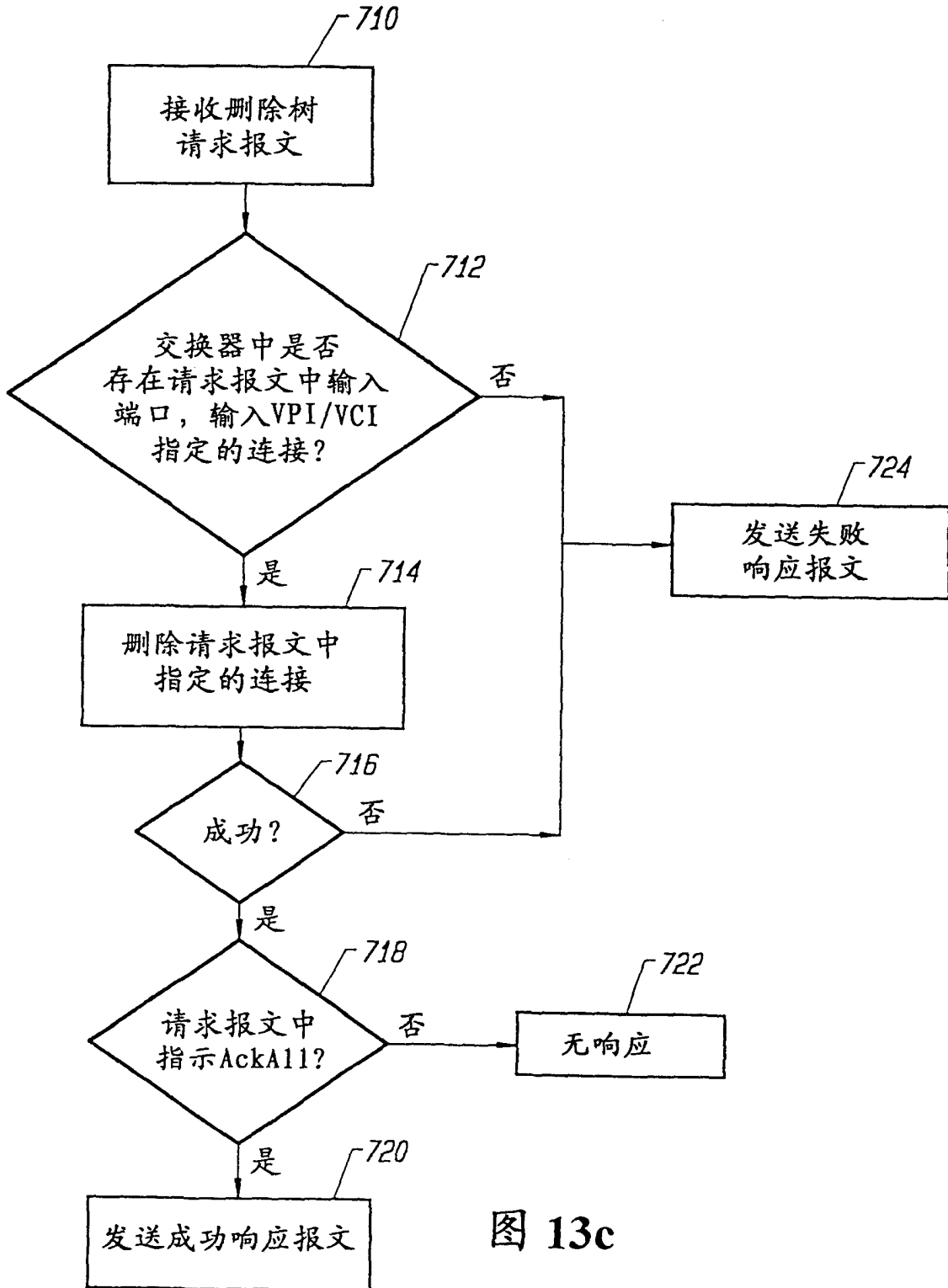


图 13c

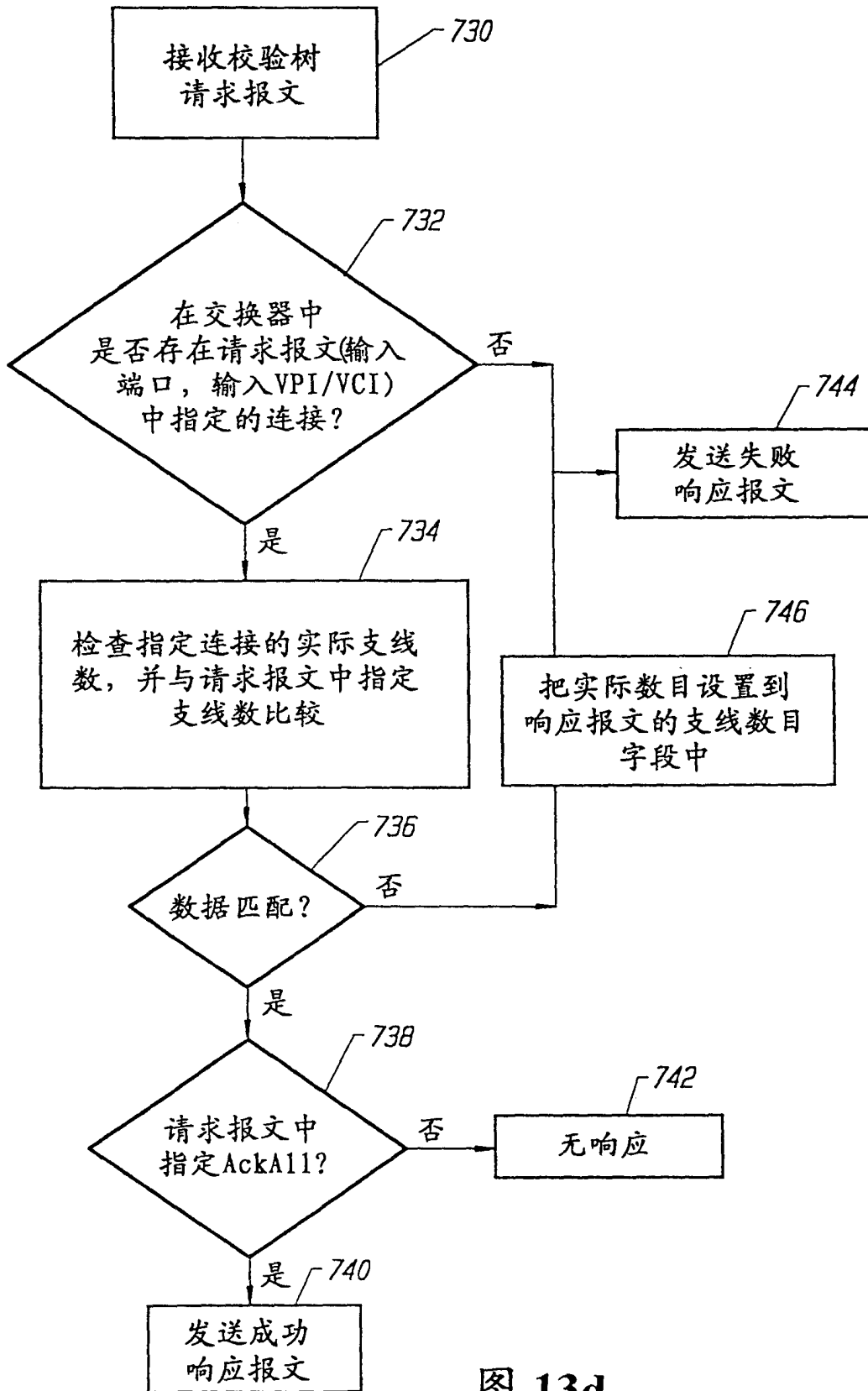


图 13d



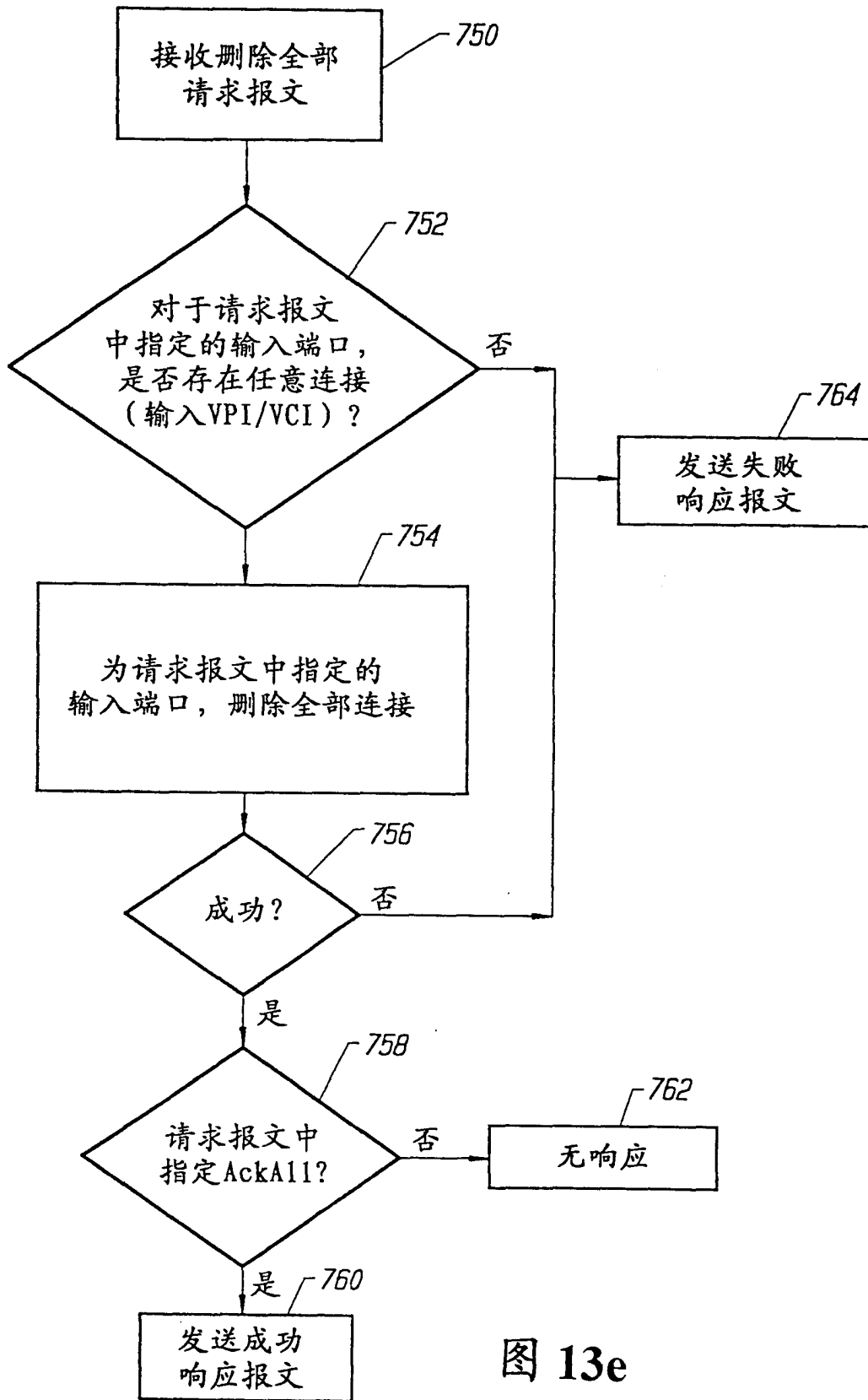


图 13e

770 GSMP CM  
移动根报文

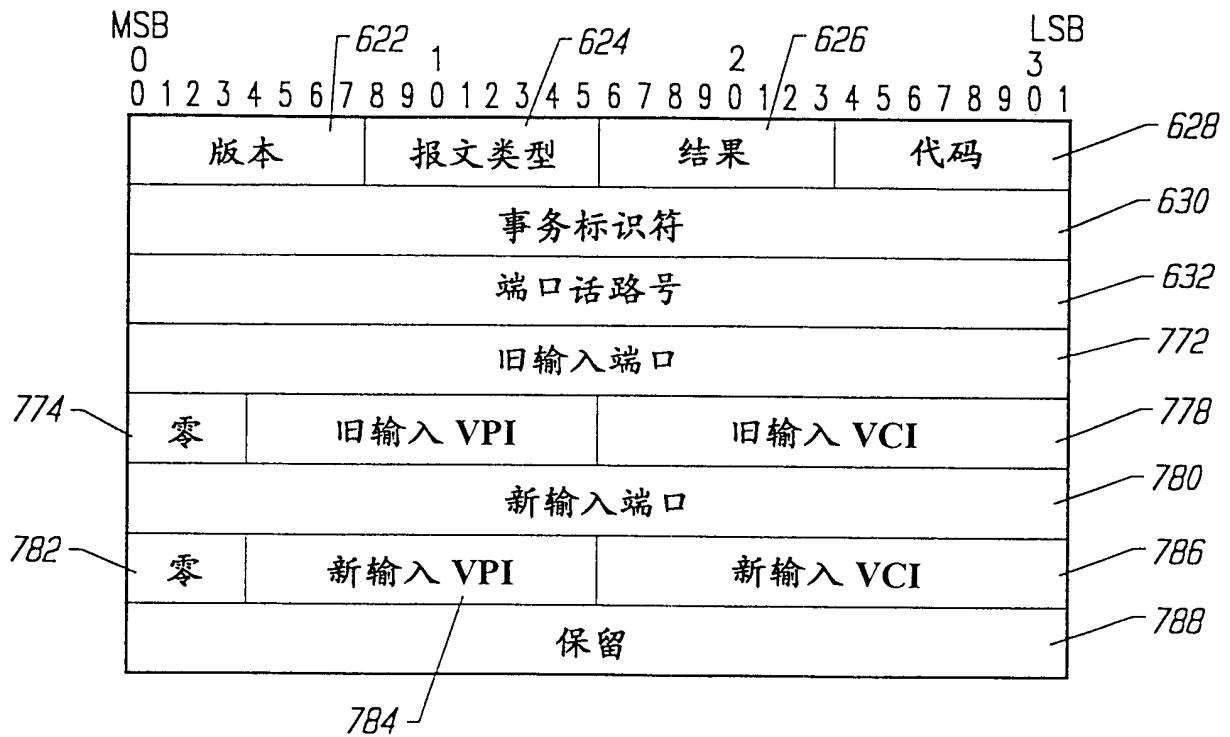
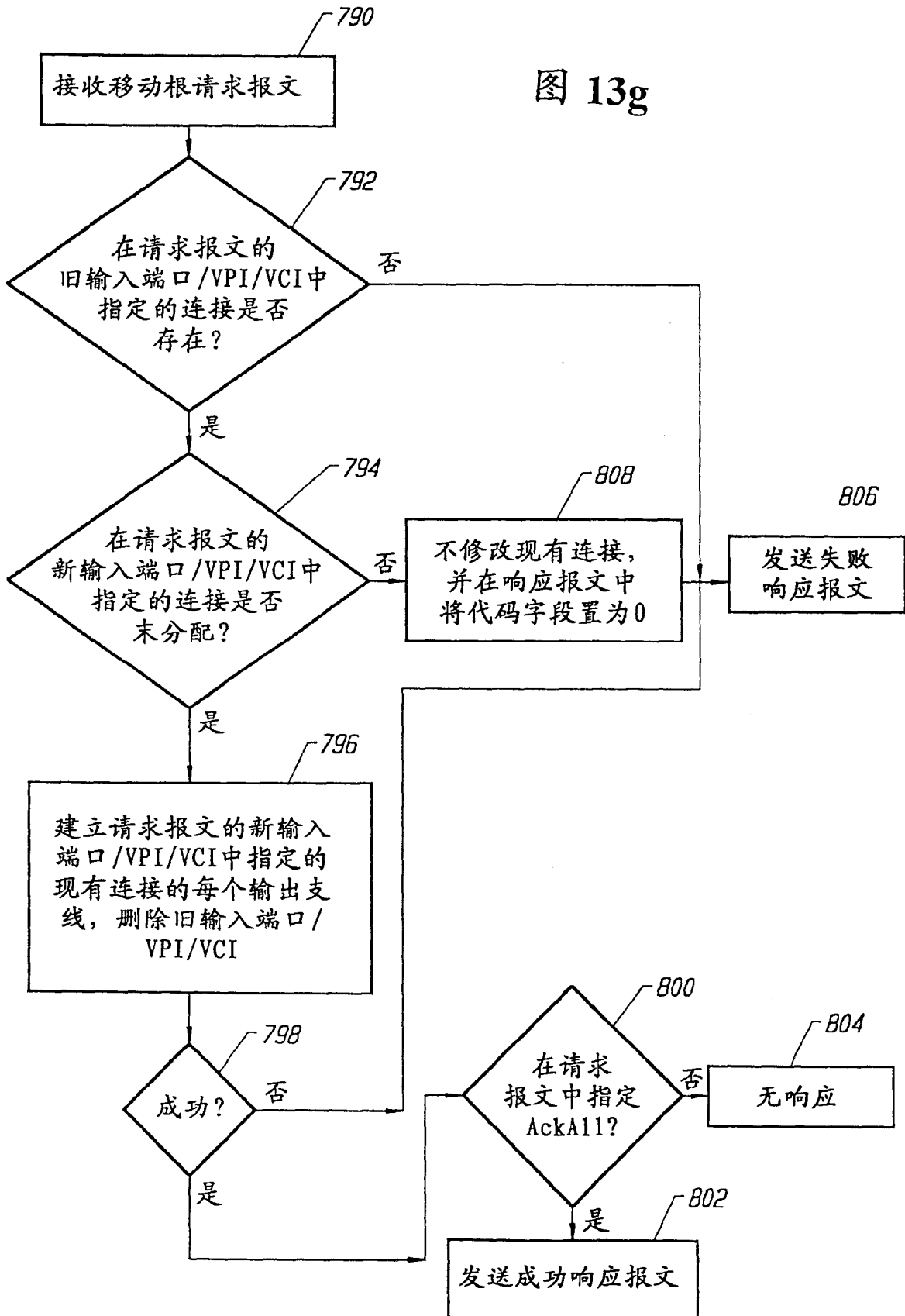


图 13f

图 13g



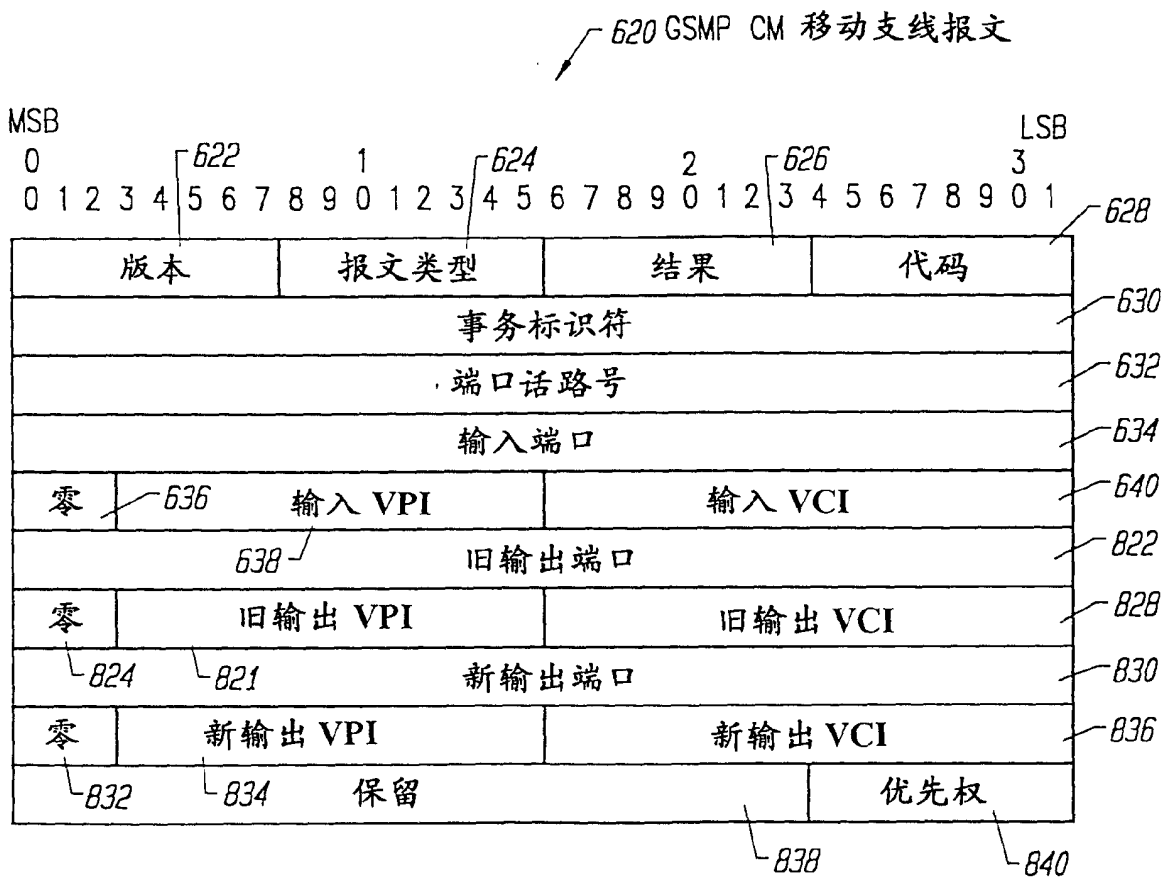
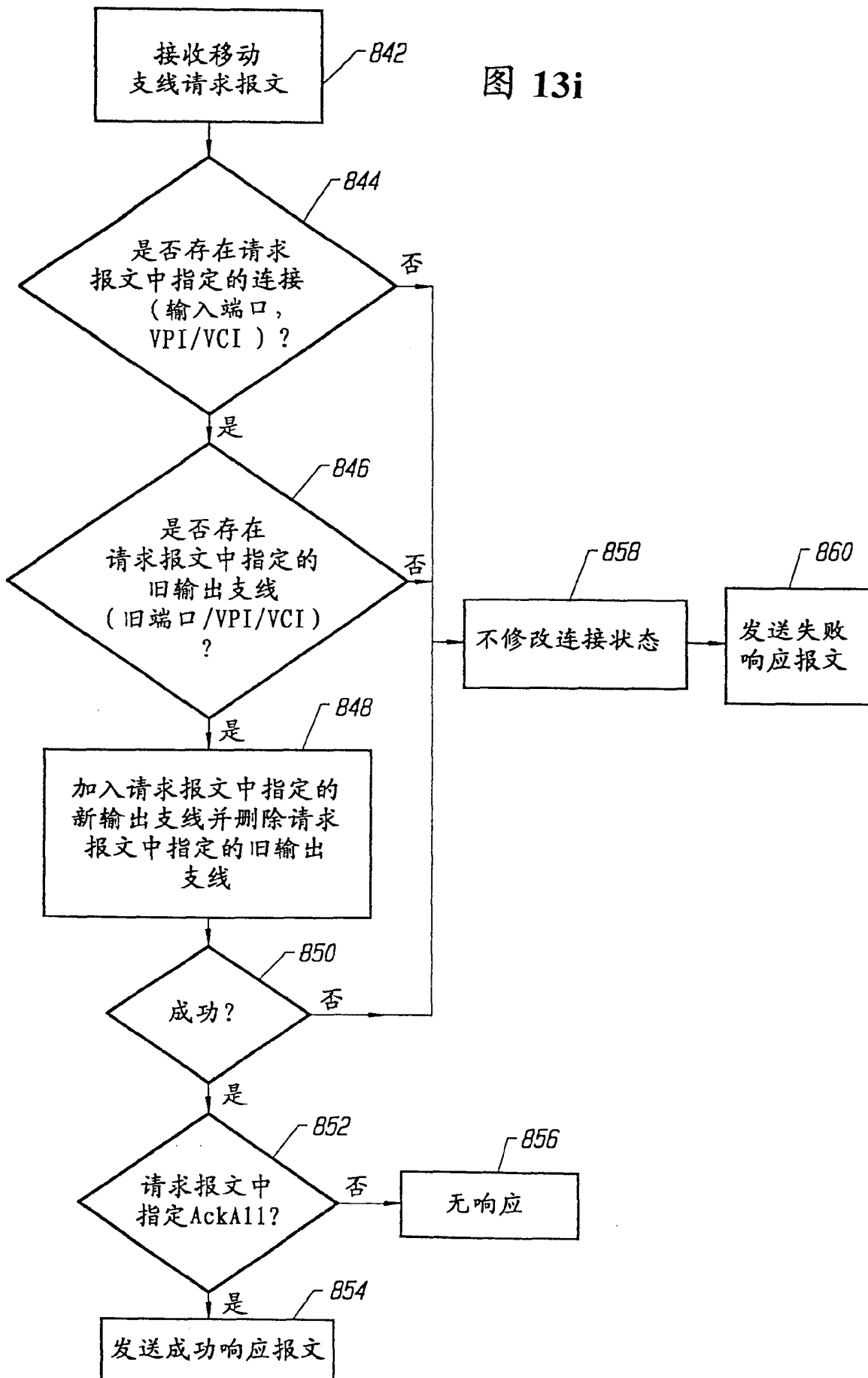


图 13h

图 13i



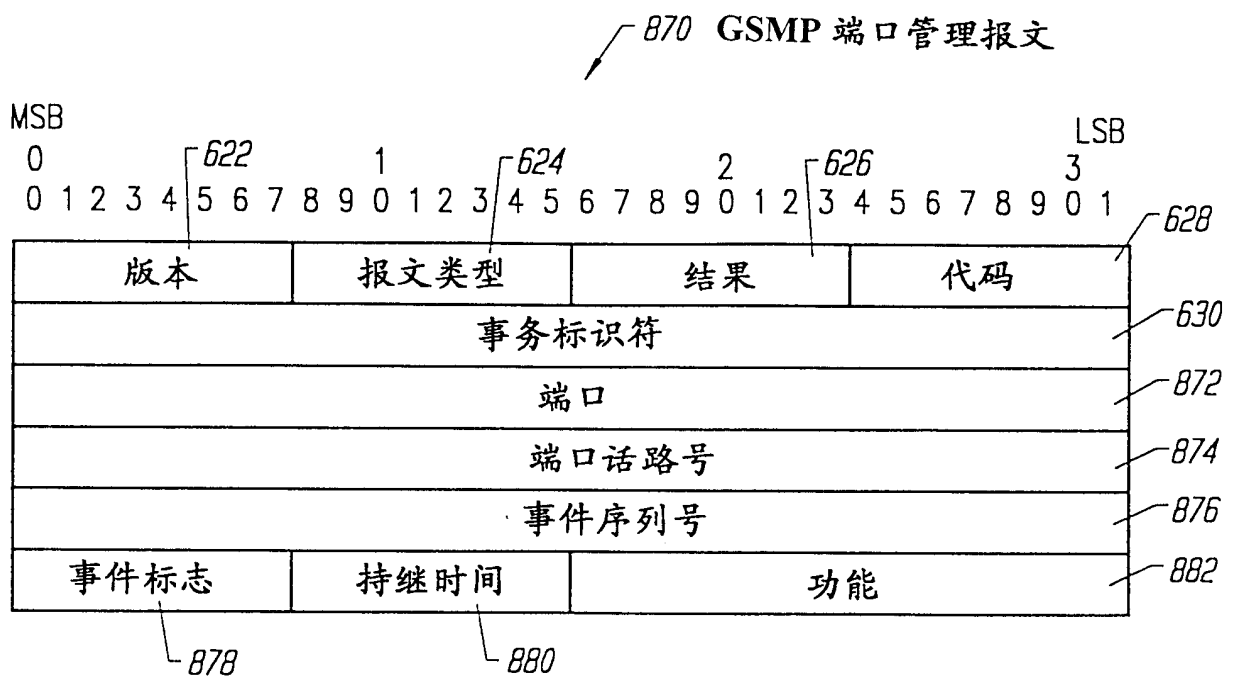
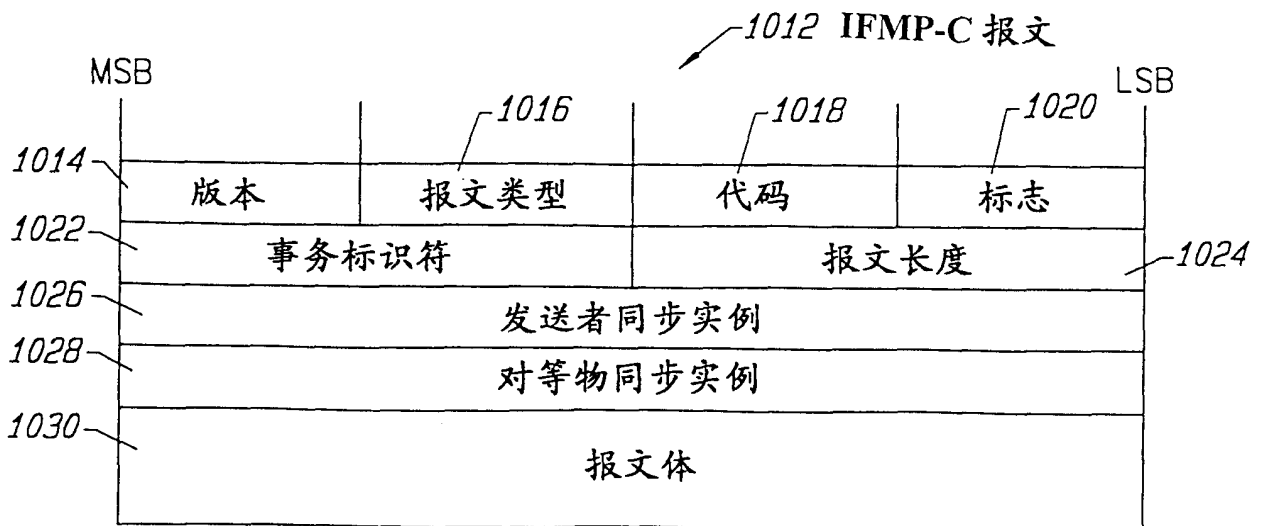
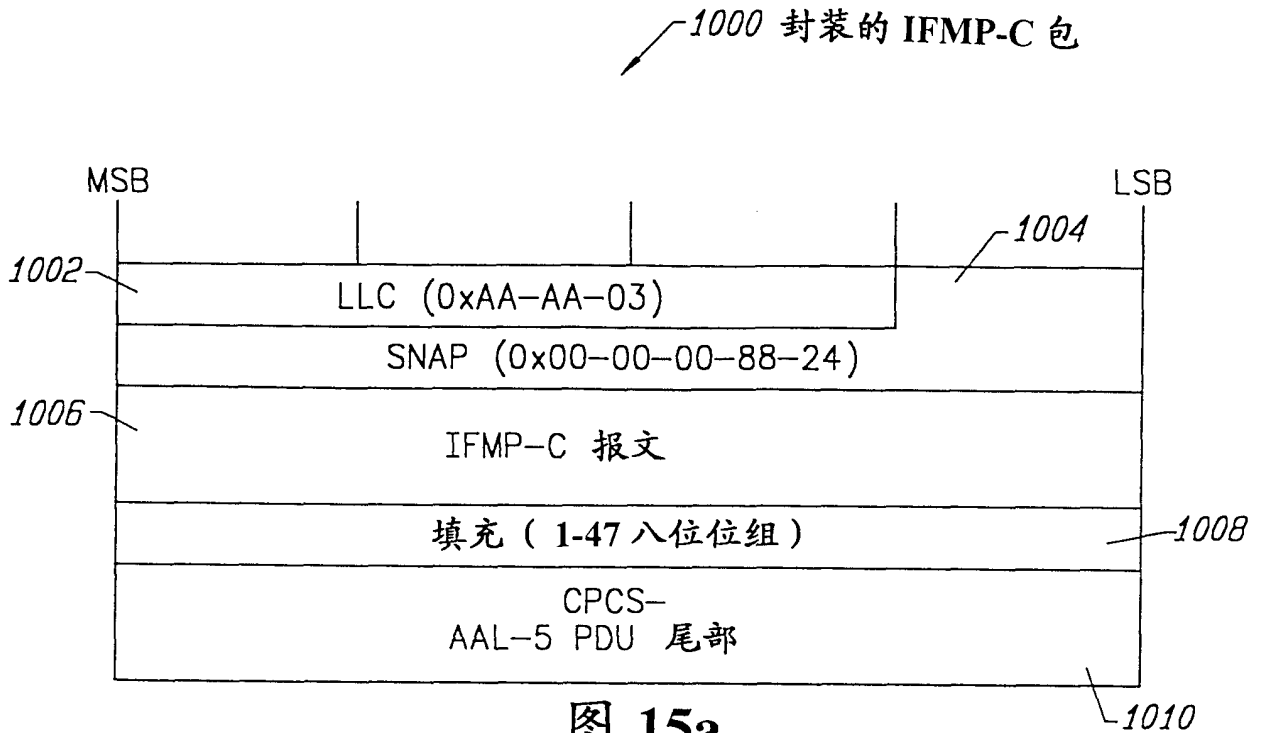


图 14



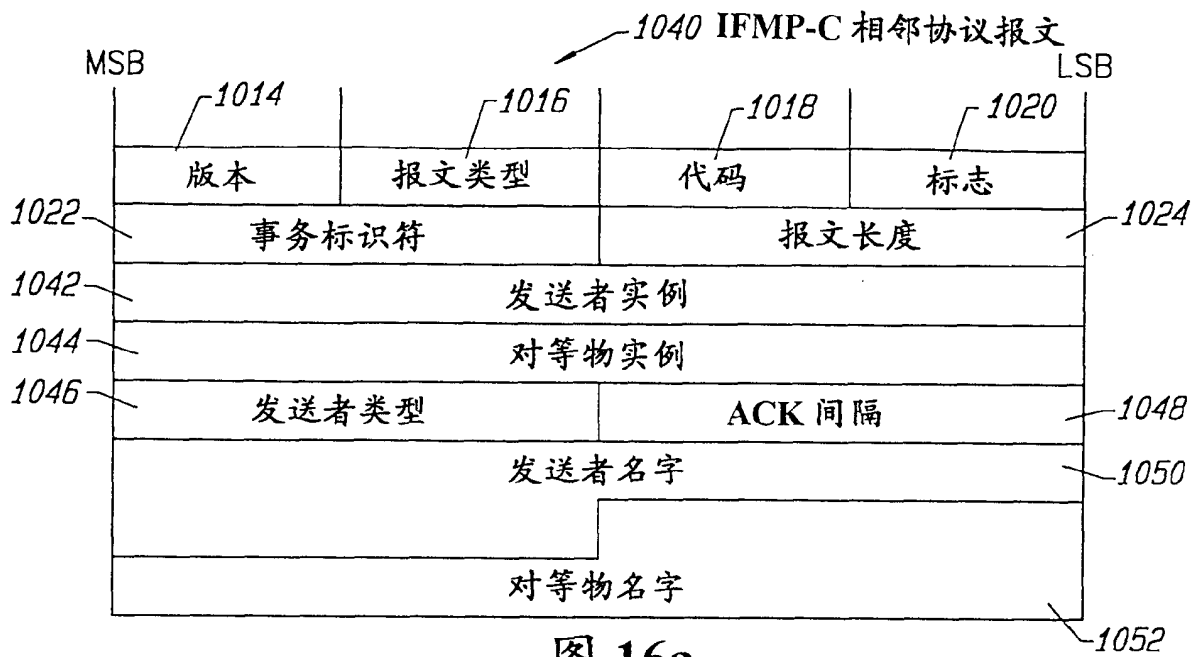
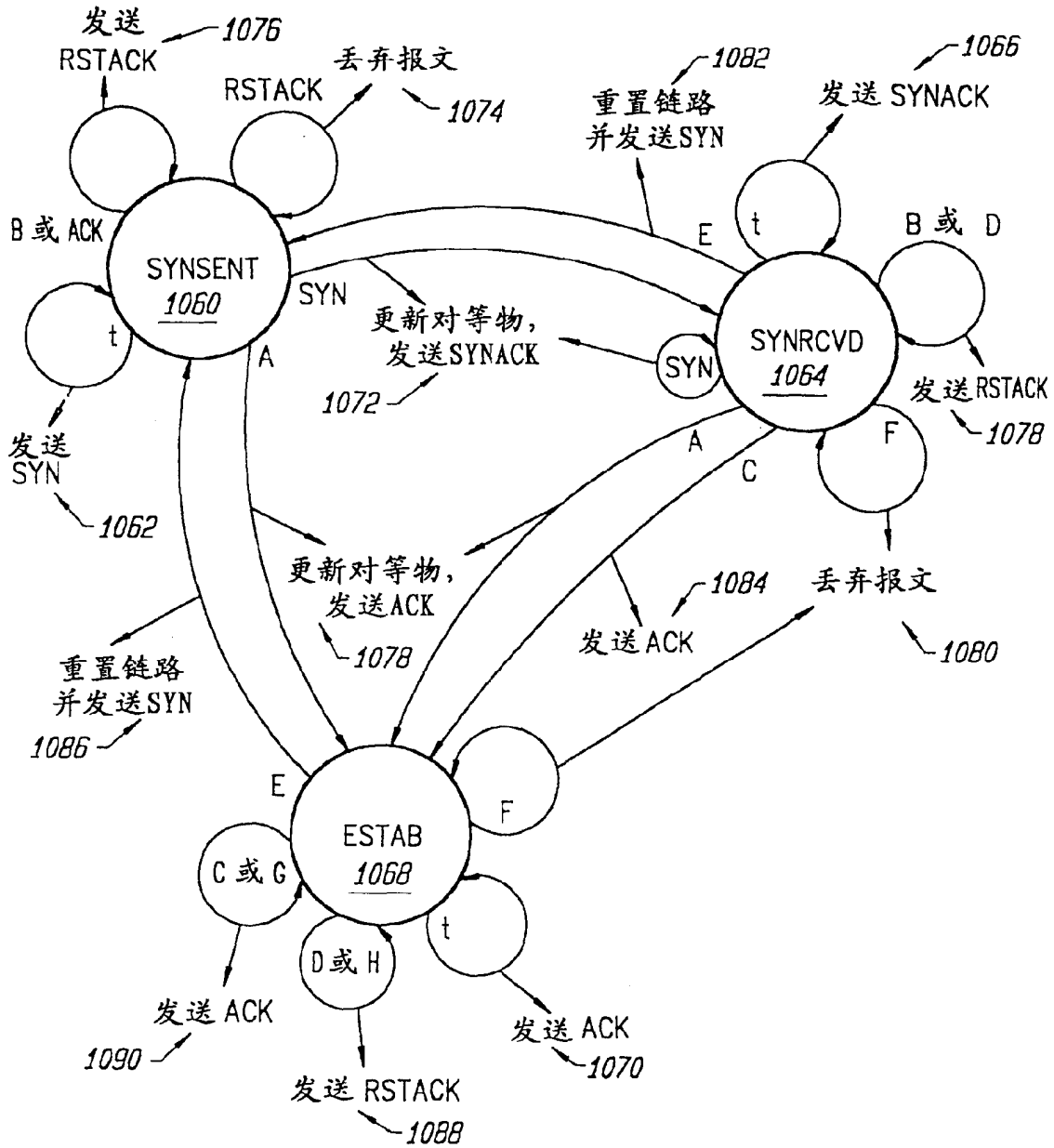


图 16a





- A: SYNACK 与 %X
- B: SYNACK 与非 (%X)
- C: ACK 和 %X 与 %Y
- D: ACK 与非 (%X 与 %Y)
- E: RSTACK 与 %X 与 %Y
- F: RSTACK 与非 (%X 与 %Y)
- G: (SYN 或 SYNACK) 与 (%X)
- H: (SYN 或 SYNACK) 与非 (%X)

其中%X=输入报文中的对等物实例及  
对等物名字和与链路相关的  
实例及名字的本地值匹配  
%Y=输入报文中的发送者实例及  
发送者名字和为对等物实例  
及对等物名字存储的值匹配

图 16b

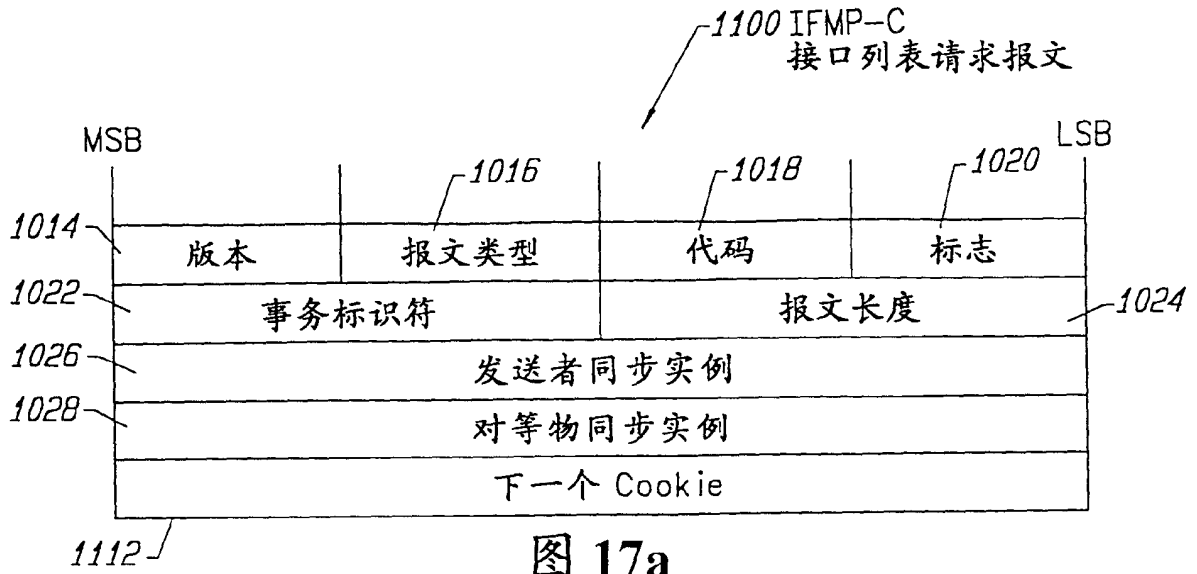


图 17a

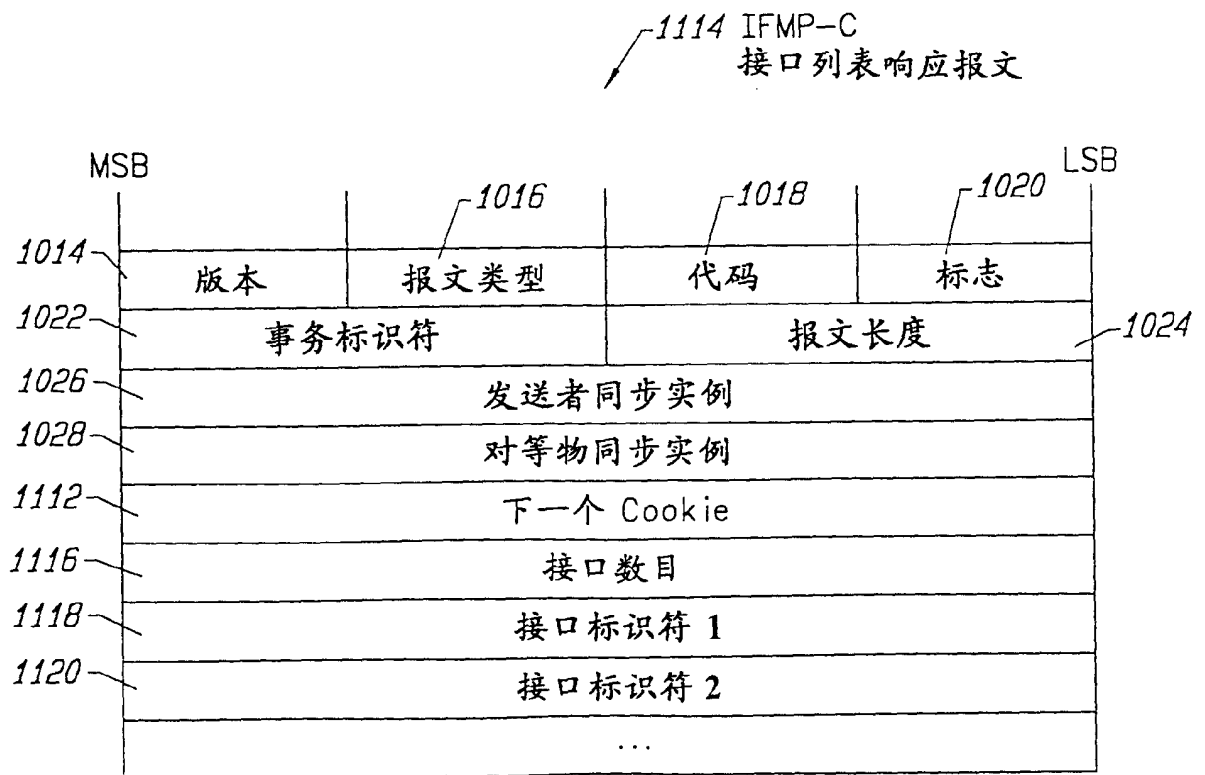
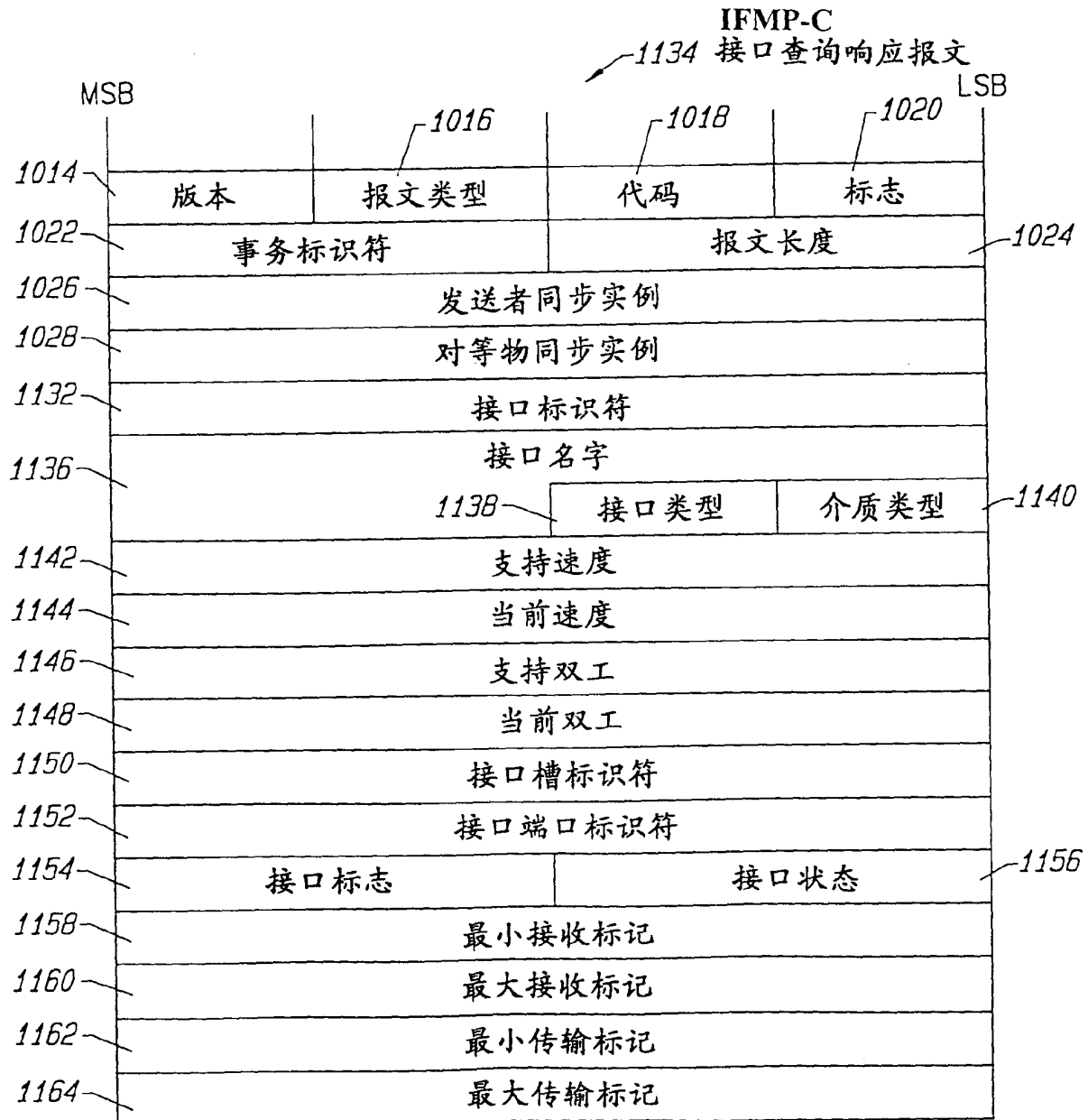
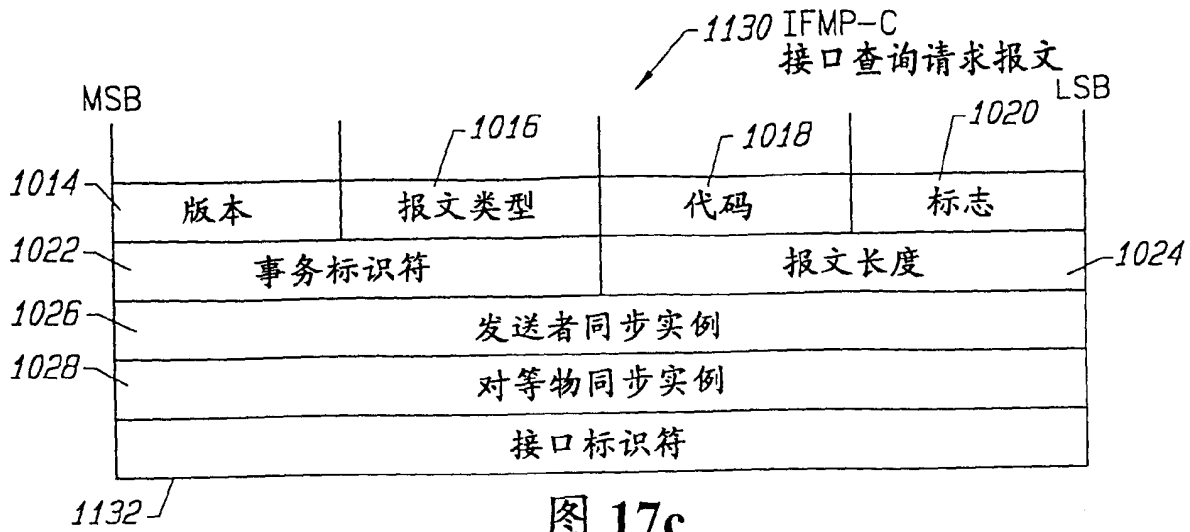


图 17b



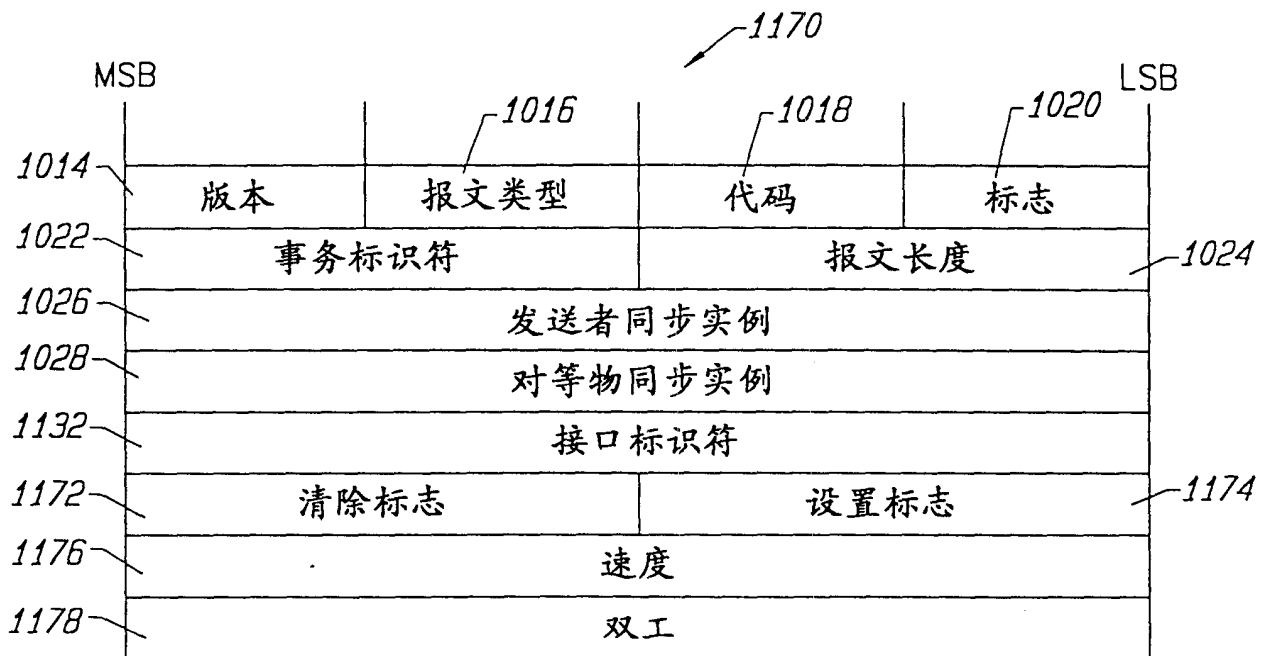


图 17e

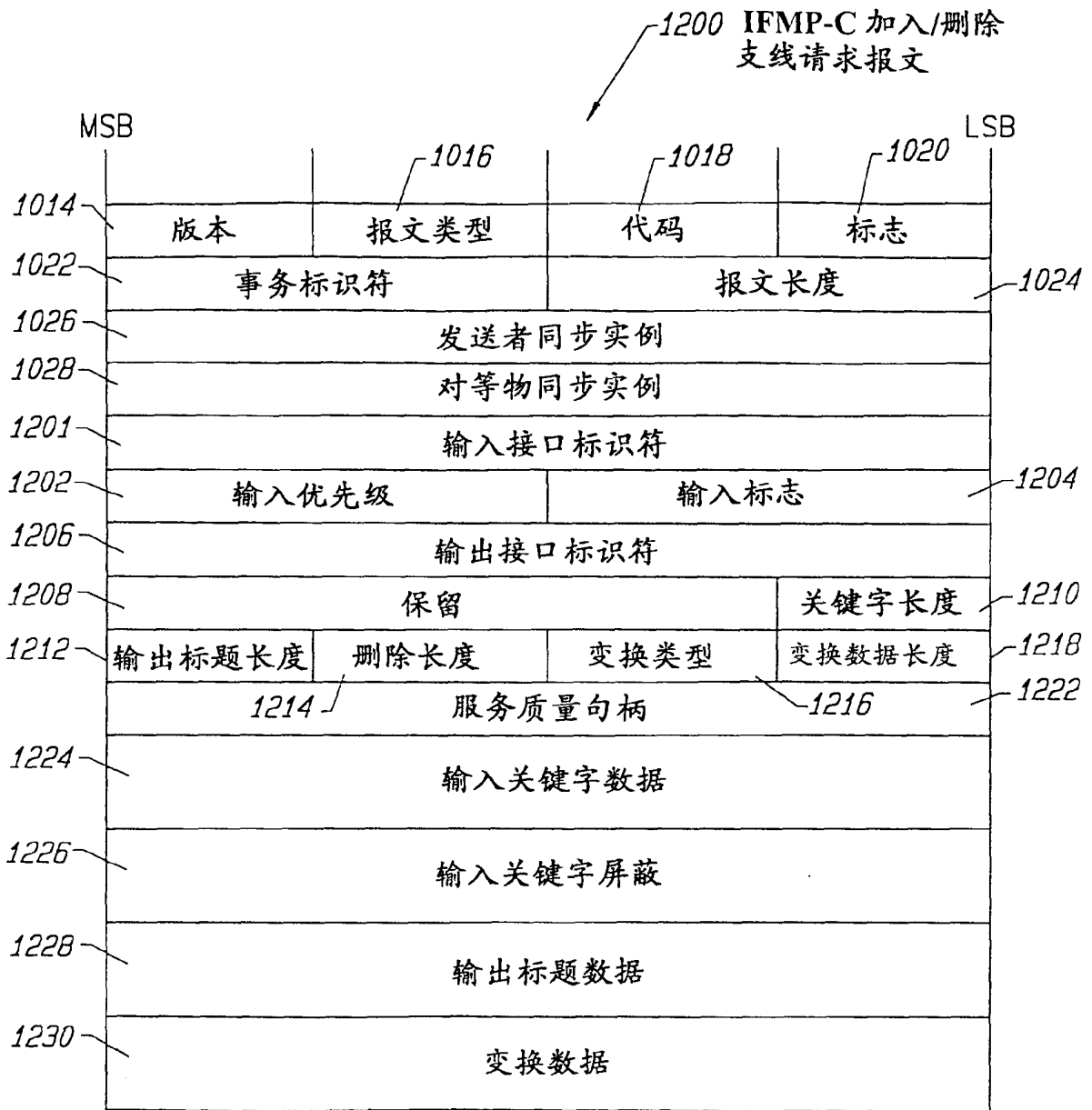


图 18a

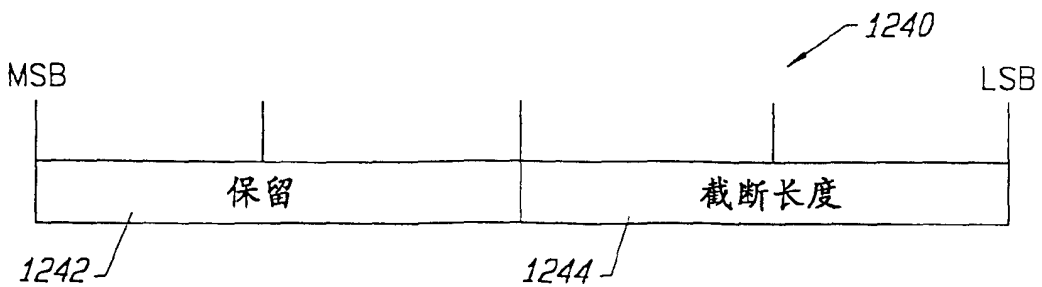


图 18b

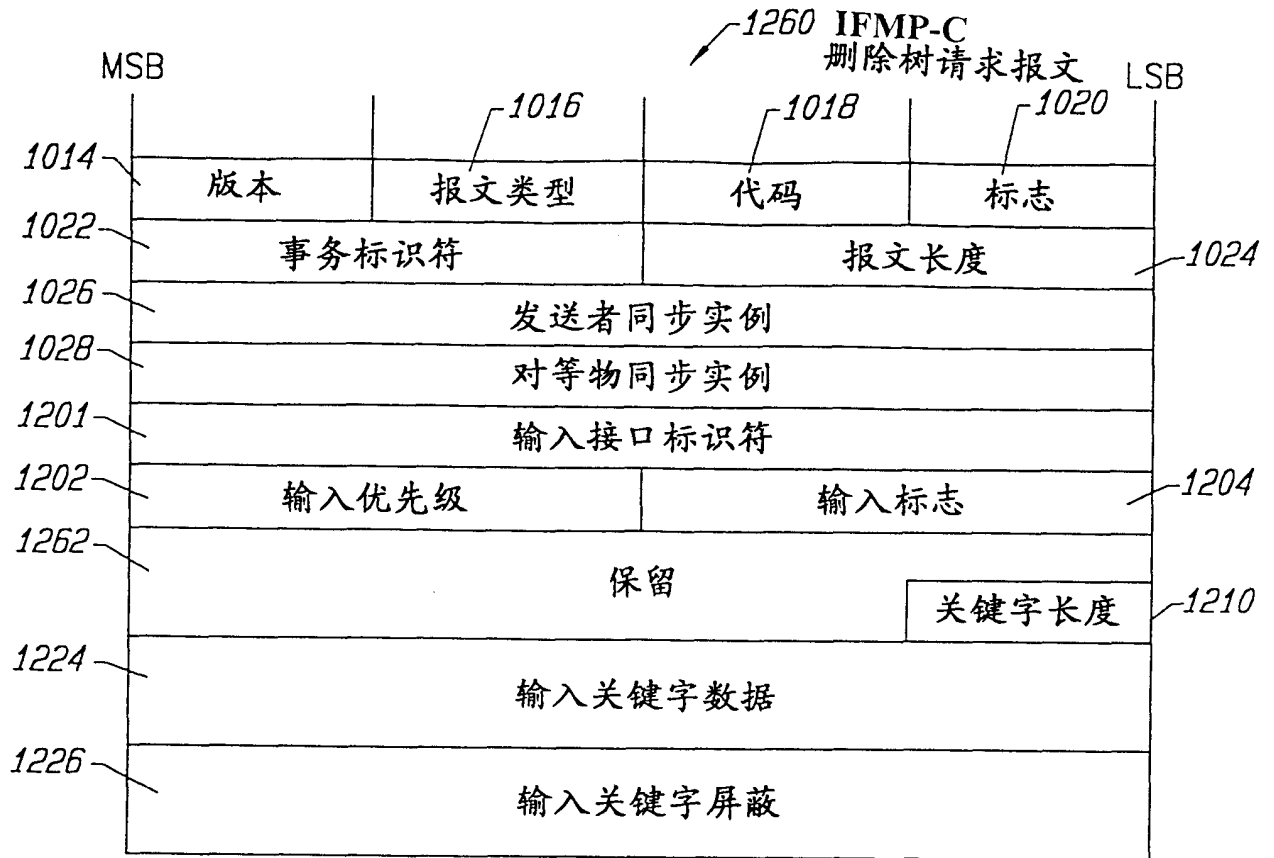


图 18d

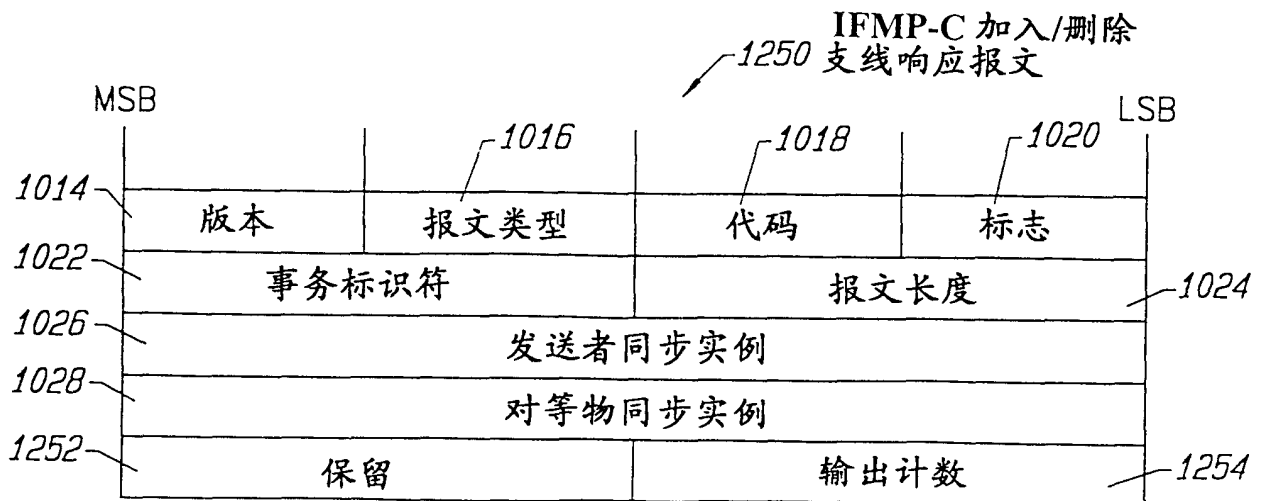


图 18c

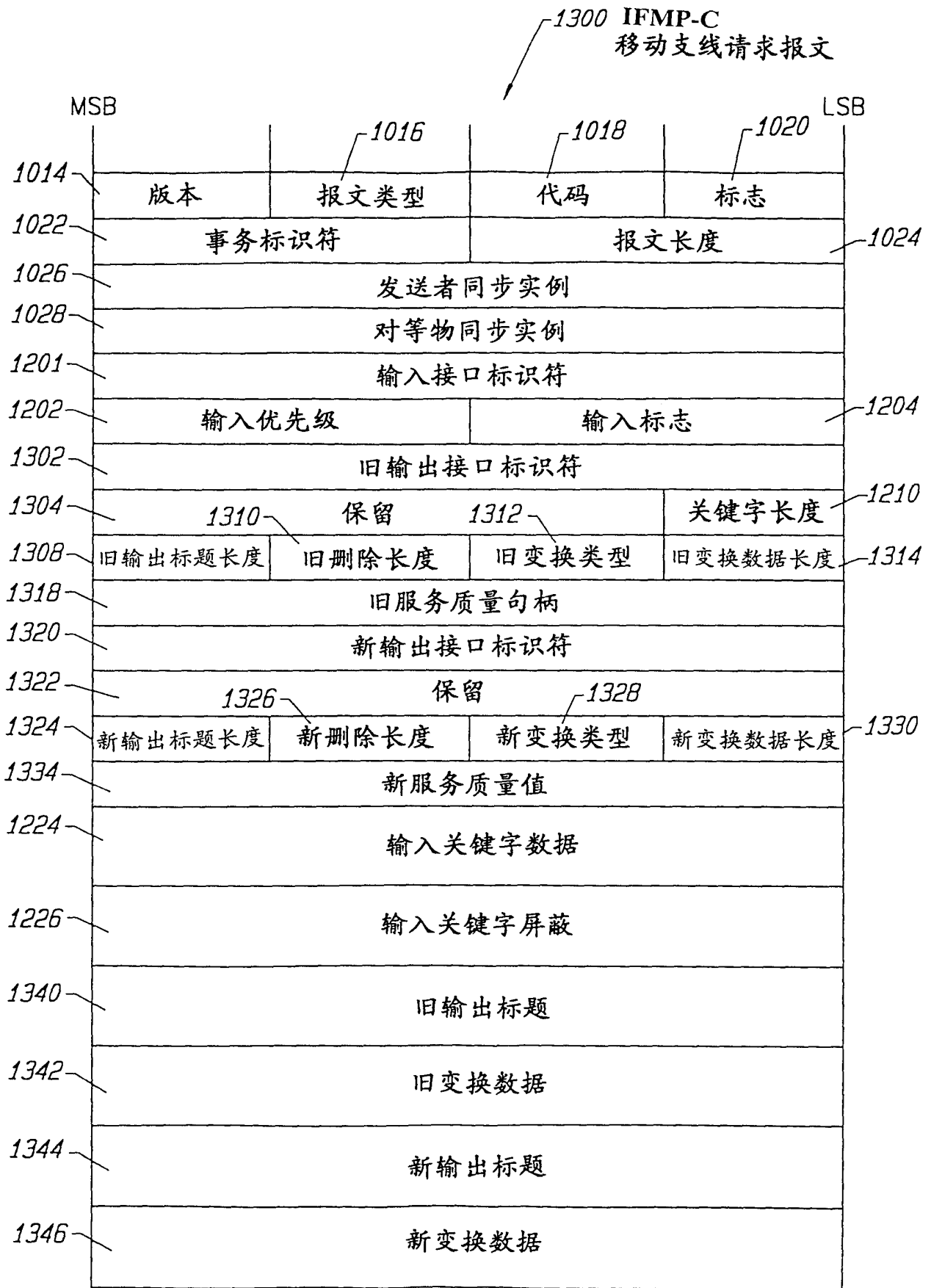


图 18e

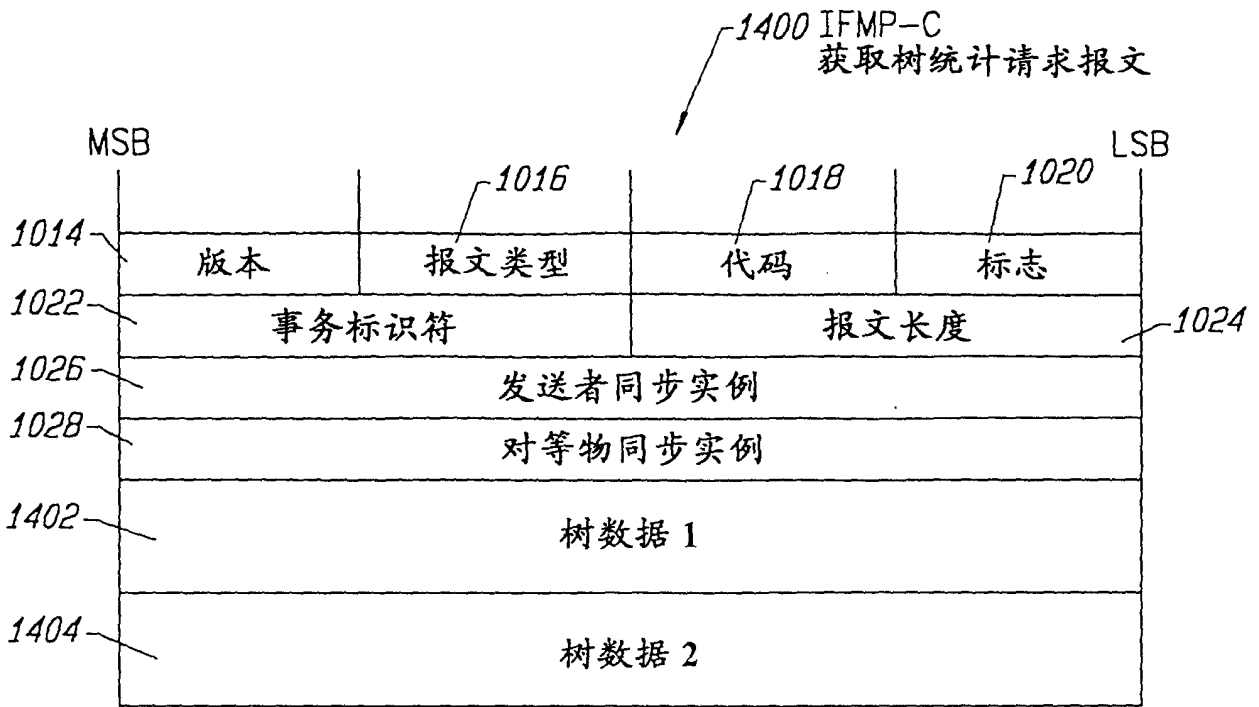


图 19a

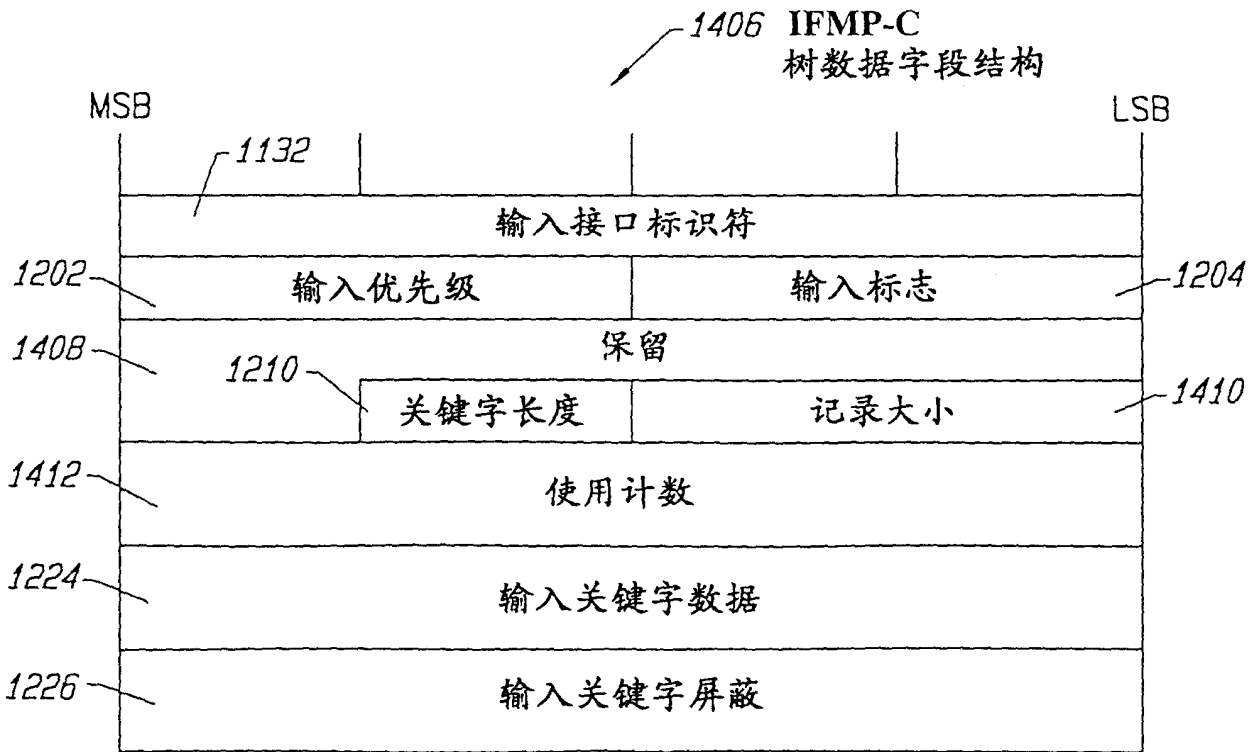


图 19b



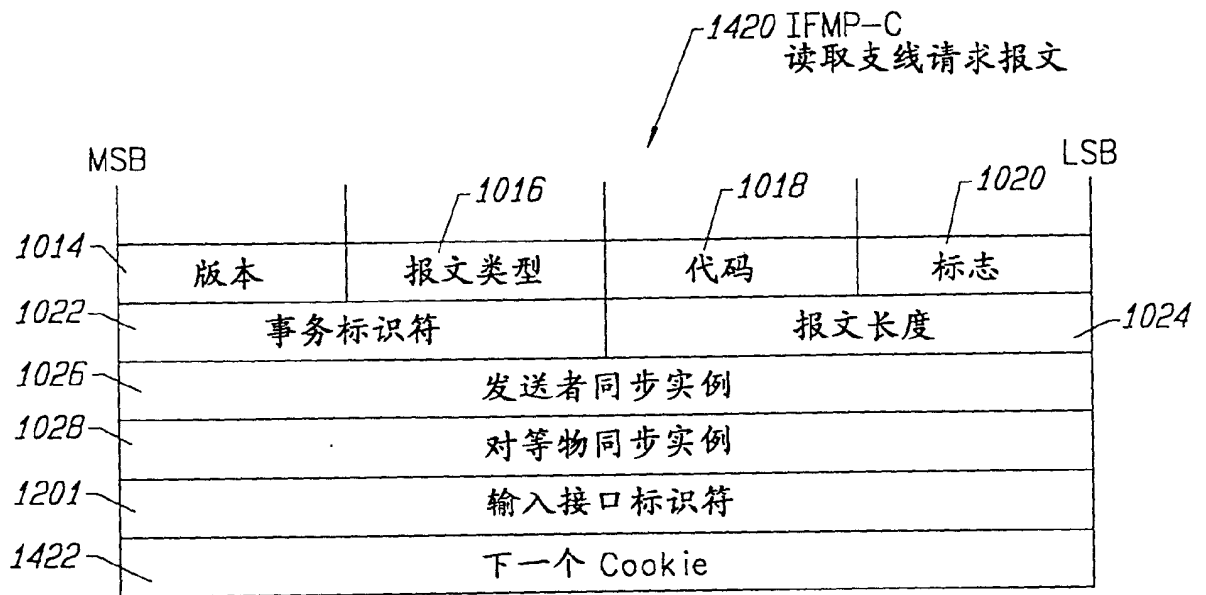


图 20a

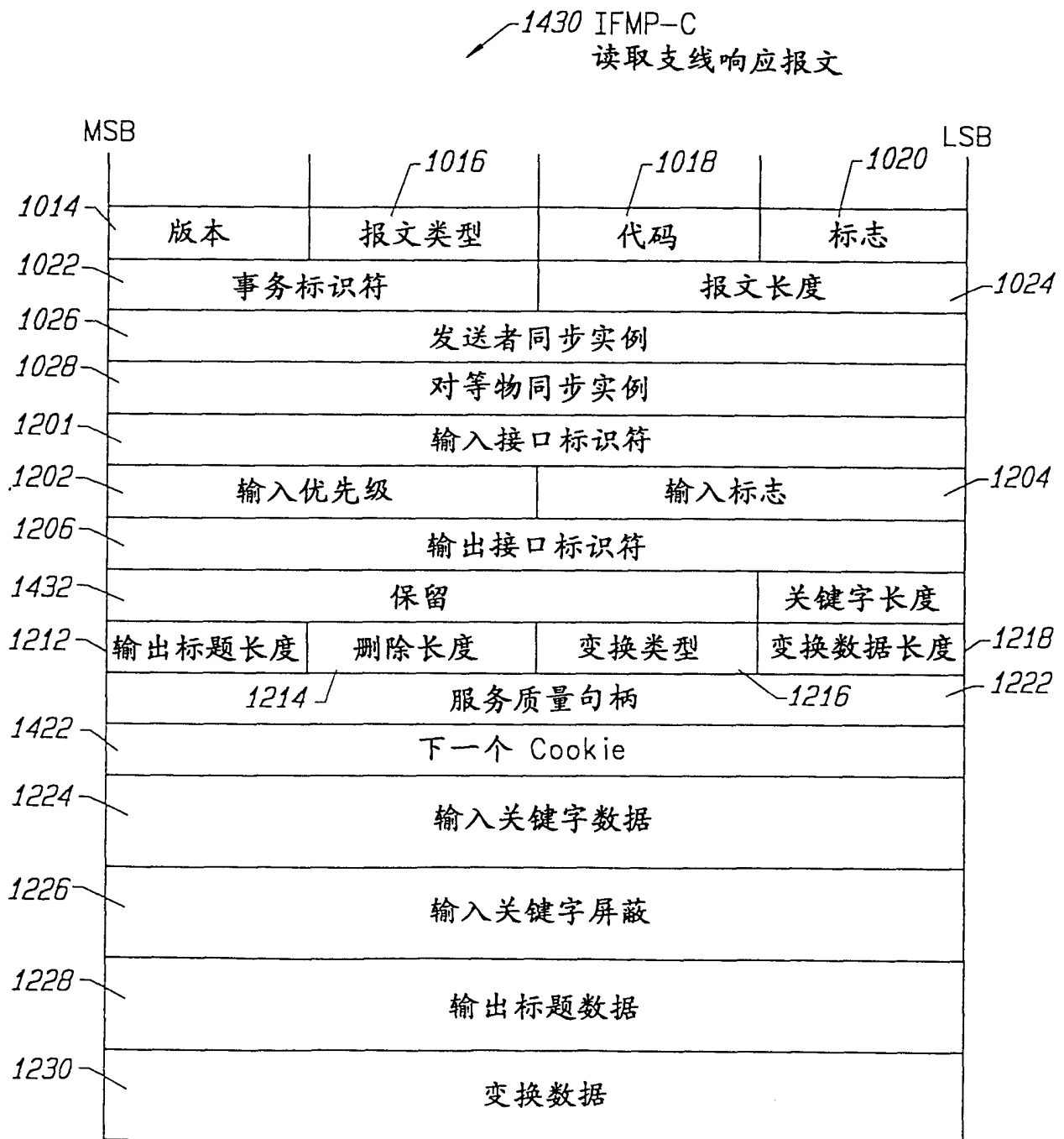
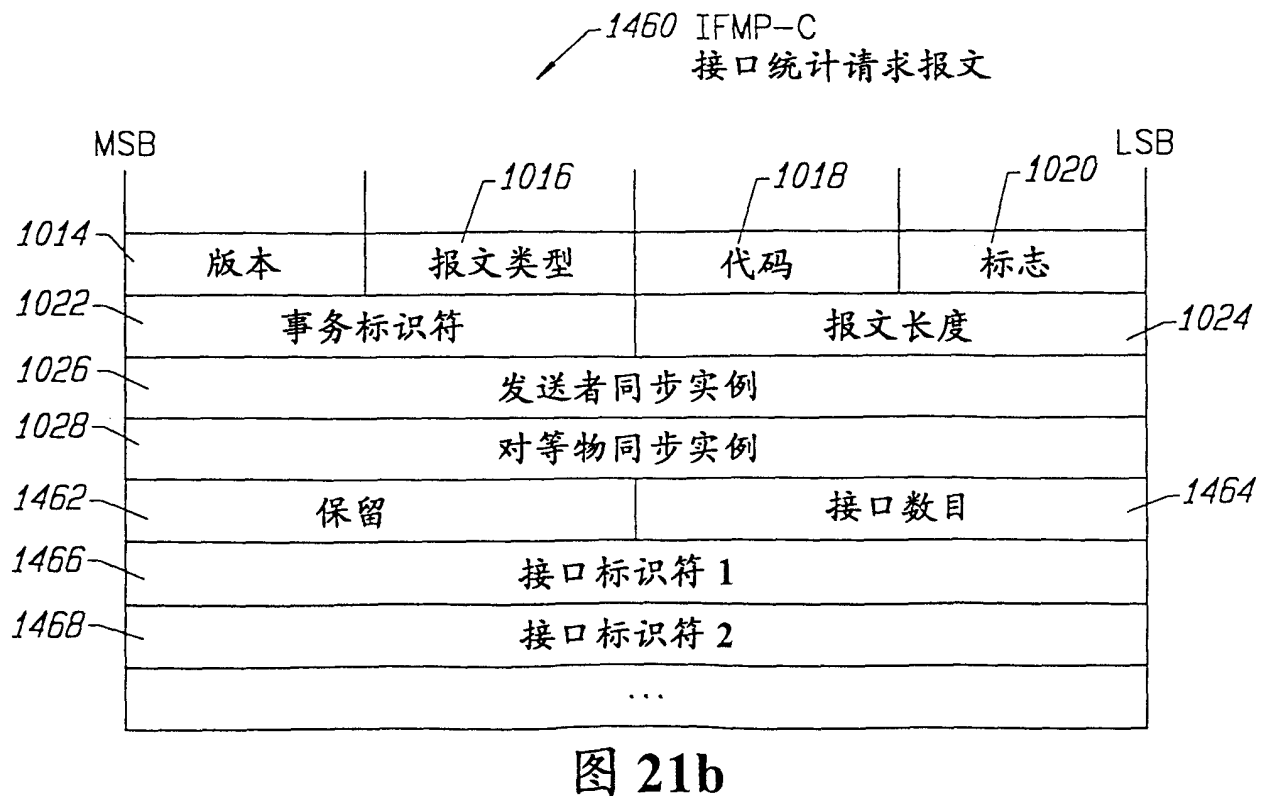
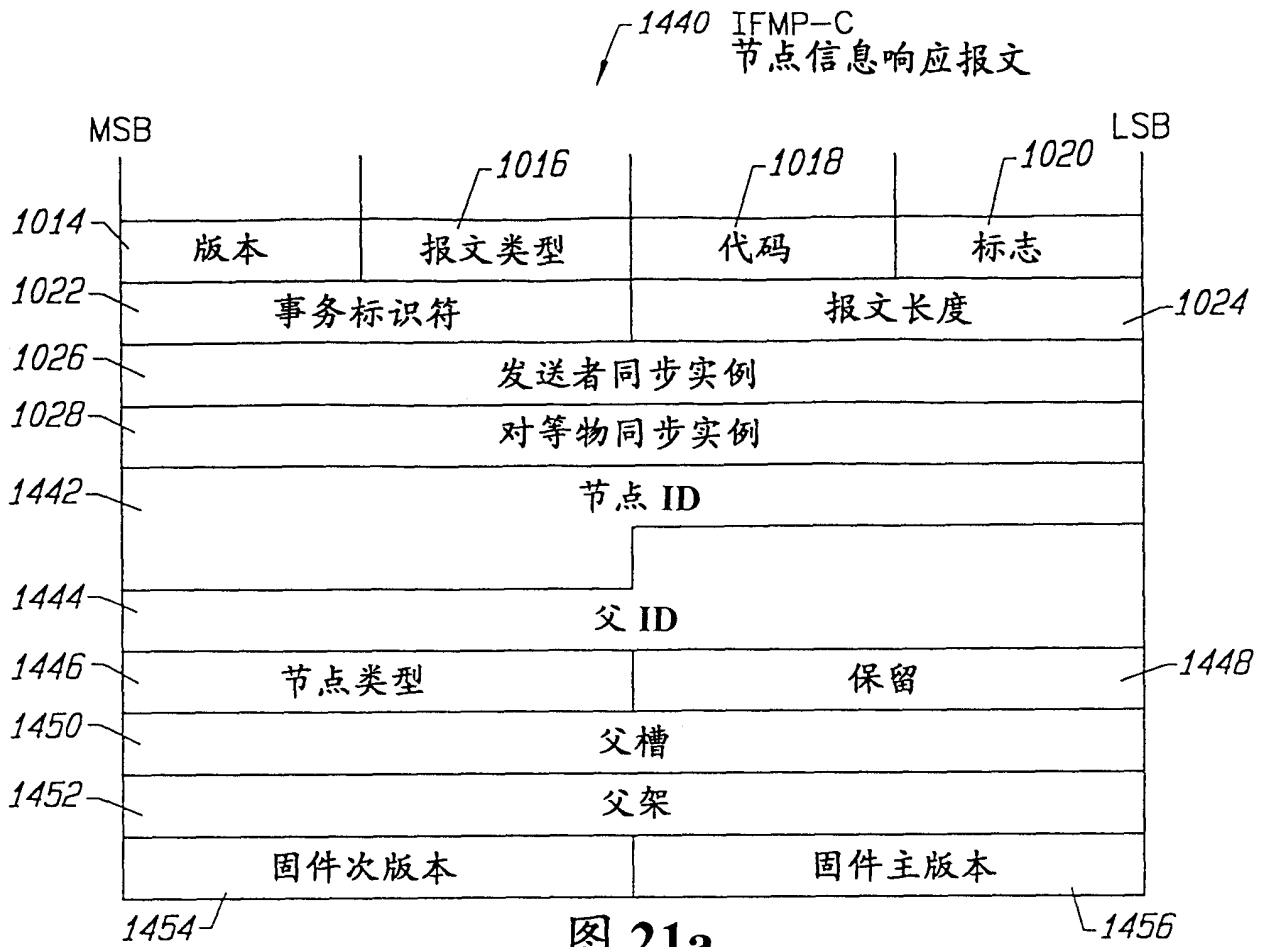


图 20b



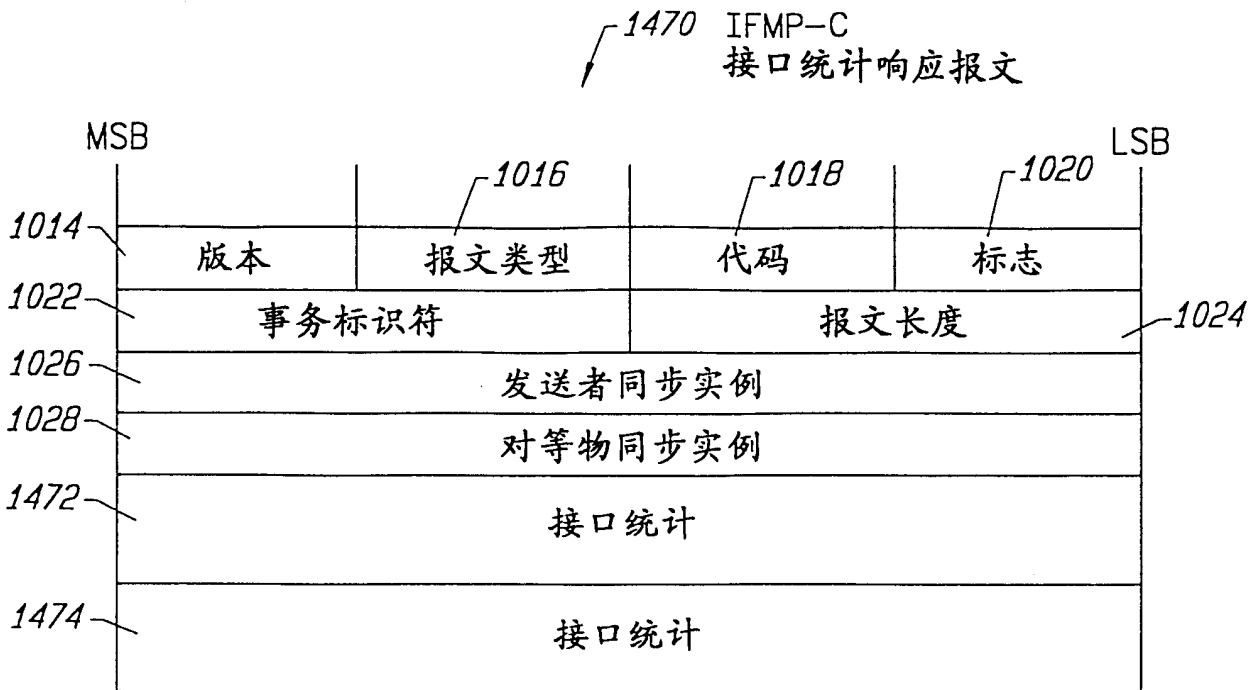


图 21c

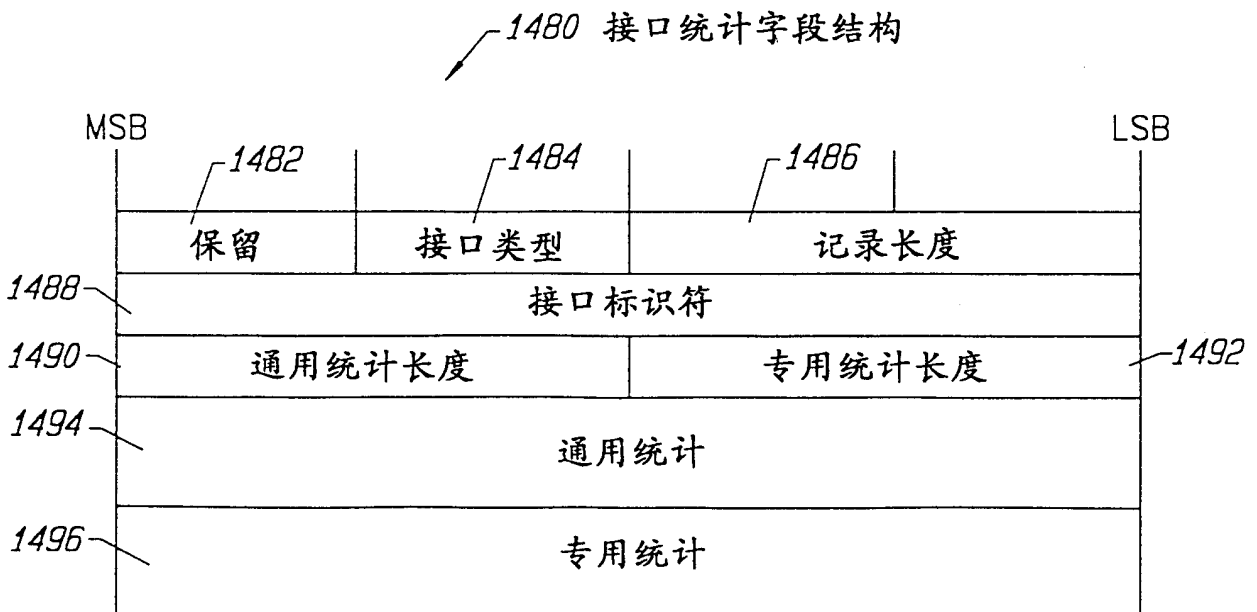


图 21d

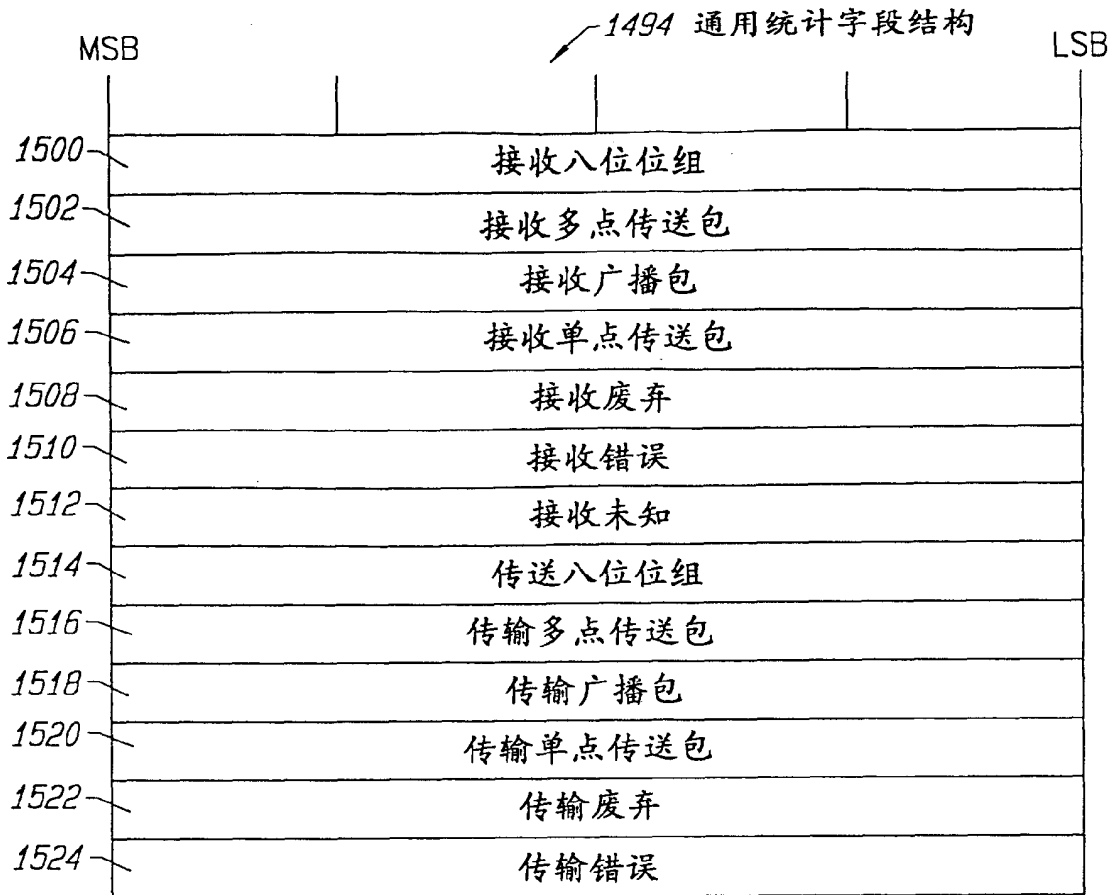


图 21e

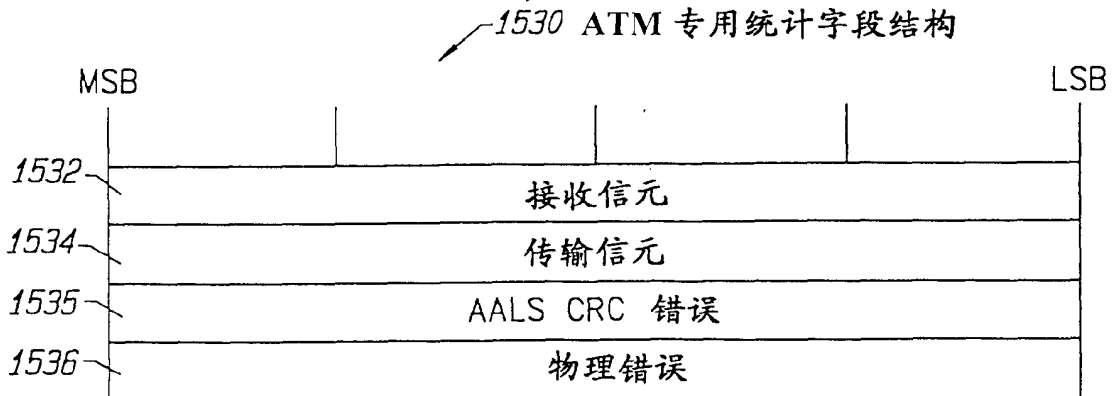


图 21f

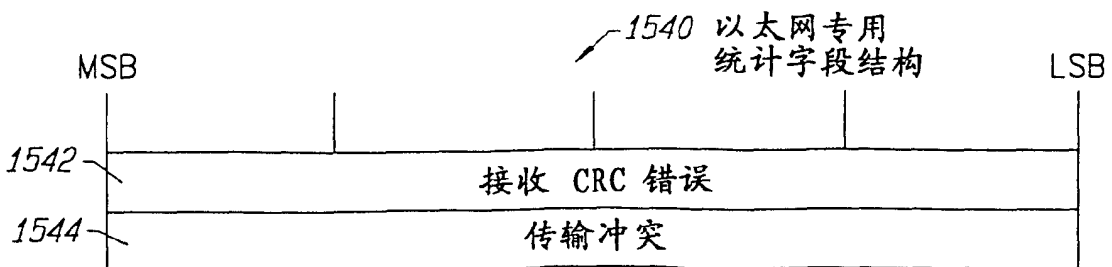


图 21g