(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 107852371 B
(45)授权公告日 2021.06.29

(21)申请号 201680045863.8
(22)申请日 2016.06.16
(65)同一申请的已公布的文献号
申请公布号 CN 107852371 A
(43)申请公布日 2018.03.27
(30)优先权数据
15180017.4 2015.08.06 EP
(85)PCT国际申请进入国家阶段日
2018.02.05
(86)PCT国际申请的申请数据
PCT/EP2016/063867 2016.06.16
(87)PCT国际申请的公布数据
WO2017/021046 EN 2017.02.09
(73)专利权人 英国电讯有限公司
地址 英国伦敦
(72)发明人 R·布里斯科 P·厄德利
(74)专利代理机构 北京三友知识产权代理有限公司 11127
代理人 吕俊刚 师玮
(51)Int.Cl.
H04L 12/801(2006.01)
H04L 12/851(2006.01)
H04L 12/825(2006.01)
H04L 12/835(2006.01)
H04L 12/823(2006.01)
(56)对比文件
CN 103222248 A,2013.07.24
US 7266612 B1,2007.09.04
US 2003007454 A1,2003.01.09
US 2006013241 A1,2006.01.19
US 6188698 B1,2001.02.13
审查员 郭海波

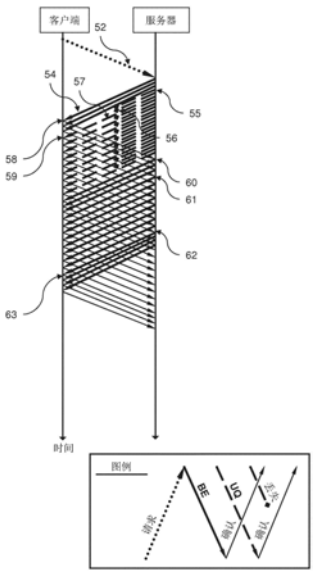
权利要求书2页 说明书12页 附图13页

(54)发明名称

数据分组网络

(57)摘要

本发明涉及在数据分组网络中控制分组的方法，所述方法包括以下步骤：源节点经由中间节点通过数据分组网络向接收方节点发送第一分组集，其中，第一分组集被标记为具有使得在中间节点处存在分组队列的情况下阻止第一分组集被转发到接收方节点的服务类别；源节点从接收方节点接收对所述第一分组集的接收分组的确认；以及所述源节点通过所述数据分组网络向所述接收方节点发送第二分组集。



CN 107852371 B

1.一种在数据分组网络中控制分组的方法,该方法包括以下步骤:

源节点将第一分组集标记为使得在中间节点的缓冲器不是空的情况下所述中间节点不应将所述第一分组集转发到接收方节点并且仅在所述中间节点的所述缓冲器是空的情况下才将所述第一分组集转发到所述接收方节点的服务类别;

所述源节点经由所述中间节点通过所述数据分组网络向接收方节点发送所述第一分组集;

所述源节点从所述接收方节点接收对所述第一分组集的第一接收分组的第一确认,所述第一确认表示所述第一分组集的分组被阻止转发到所述接收方节点;以及

所述源节点通过所述数据分组网络向所述接收方节点发送第二分组集。

2.根据权利要求1所述的方法,该方法还包括以下初始步骤:

所述源节点在所述第一分组集之前经由所述中间节点通过所述数据分组网络向所述接收方节点发送初始分组集,其中,所述初始分组集为使得所述初始分组集能够在所述中间节点处排队的服务类别。

3.根据权利要求2所述的方法,其中,所述源节点发送所述第一分组集的速率高于所述源节点发送所述初始分组集的速率。

4.根据权利要求1至3中任一项所述的方法,其中,所述第一分组集包含虚设数据。

5.根据权利要求1至3中任一项所述的方法,其中,所述第二分组集包括所述被阻止分组的重传。

6.根据权利要求1至3中任一项所述的方法,该方法还包括以下步骤:

所述源节点从所述接收方节点接收对所述第一分组集的第二接收分组的第二确认;

所述源节点基于在所述源节点处接收所述第一确认和所述第二确认的速率来确定所述源节点与所述接收方节点之间的传输速率;

所述源节点基于所确定的传输速率向所述接收方节点发送第二数据分组集。

7.一种在数据分组网络中控制网络节点的方法,所述网络节点包括缓冲器,所述方法包括以下步骤:

从第一外部网络节点接收第一数据分组;

分析所述第一数据分组,以确定所述第一数据分组是否为在所述网络节点的缓冲器不是空的情况下不应将所述第一数据分组转发到接收方节点并且仅在所述网络节点的所述缓冲器是空的情况下才将所述第一数据分组转发到所述接收方节点的服务类别;

如果所述第一数据分组为在所述网络节点的缓冲器不是空的情况下不应将所述第一数据分组转发到接收方节点的服务类别,则确定至少一个数据分组是否被存储在所述网络节点的缓冲器中;以及

如果至少一个数据分组被存储在所述网络节点的缓冲器中,则不发送所述第一数据分组。

8.根据权利要求7所述的方法,其中,如果所述第一数据分组为在所述网络节点的缓冲器不是空的情况下不应将所述第一数据分组转发到所述接收方节点的服务类别,并且在所述网络节点的缓冲器中没有存储其它数据分组,则所述第一数据分组被转发到第二外部网络节点。

9.一种网络节点,该网络节点包括处理器,所述处理器被配置为执行根据权利要求1至

8中任一项所述的方法。

10.一种网络,该网络包括根据权利要求9所述的网络节点。

11.一种存储计算机程序的非暂时性计算机可读存储介质,所述计算机程序在由计算机系统执行时,执行根据权利要求1至8中任一项所述的方法。

数据分组网络

技术领域

[0001] 本发明涉及数据分组网络并且涉及在数据分组网络中控制分组的方法。

背景技术

[0002] 现今使用中的大多数网络使用经由一个或更多个中间节点在发送方节点与接收方节点之间传送的离散数据分组。这些数据分组网络中的常见问题是发送方节点具有很少或不具有关于数据分组网络中的可用容量的信息，并且因此不能立即确定它可以发送数据分组的适当的传输速率。适当的传输速率将是在不引起网络拥塞的情况下可用来发送数据分组的最大速率，否则网络拥塞将使一些数据分组被丢弃并且也可使在其它数据流(例如在共享沿着其相应传输路径的一个或更多个中间节点的其它对节点之间)上的数据分组被丢弃。

[0003] 为了解决此问题，数据分组网络中的节点使用闭环或开环拥塞控制算法。闭环算法依靠被供应给发送方节点的某个拥塞反馈，从而使得它能够确定或估计用来发送将来的数据分组的适当速率。然而，这个拥塞反馈可在非常短的时间内变得无用，因为网络中的其它对节点(共享沿着其传输路径的一个或更多个中间节点)可以随时启动或停止数据流。因此，拥塞反馈可快速地变得过时，并且闭环算法无法准确地预测要发送数据分组的适当速率。这个缺点随着数据分组网络中的链路的容量增加而变得越来越严重，意味着容量和拥塞的大增加或减少可能发生。

[0004] 通常在存在很少或没有来自网络的拥塞信息时开始新数据流时使用开环拥塞控制算法。最常见的拥塞控制算法中的一种是用于网际协议IP网络的传输控制协议TCP“慢启动”算法，该算法具有初始指数增长阶段，然后是拥塞避免阶段。当新的TCP慢启动流开始时，发送方的拥塞窗口(表示网络上的拥塞的估计的值)被设定为一个初始值并且第一分组集被发送到接收方节点。接收方节点针对它接收到的每个数据分组向发送方节点发送回确认。在初始的指数增长阶段期间，发送方节点针对接收到的每个确认分组将其拥塞窗口增加一个分组。拥塞窗口以及因此传输速率因此每往返时间加倍。一旦拥塞窗口达到发送方节点的慢启动阈值(‘ssthresh’)，那么指数增长阶段就结束并且它开始拥塞避免阶段，在该拥塞避免阶段中，拥塞窗口对于它接收到确认的每个往返仅被增加一个分组，而不管接收到多少个确认分组。如果在任何时候确认分组(或其不存在)指示已发生丢失(这很可能是由于网络上的拥塞而导致的)，则发送方节点通过使拥塞窗口减半来响应以试图减少由该特定数据流所引起的拥塞量。然而，发送方节点在其传输速率超过了可用容量之后的一个往返时间内接收此反馈(即指示已发生丢失的确认分组)。截止发送方节点接收到此反馈时，发送方节点将已发送了可用容量两倍快的数据。这被称为“超调(overshoot)”。

[0005] 指数增长阶段可对非TCP业务造成问题。考虑在原本空的1GB/s链路上进行中的低速率(例如64kB/s)恒定比特速率语音流的情况。进一步想象大的TCP流在具有十个1500B分组的初始拥塞窗口和200ms的往返时间的同一链路上开始。流保持每次往返将其拥塞窗口加倍，直到在将近十一次往返之后其窗口是每次往返16,666个分组(1Gb/s)为止。在下一往

返中,它将在它得到检测到暗示它在前一往返超过了网络中的可用容量的丢弃的第一反馈之前加倍到2Gb/s。此下一往返中的分组的大约50％(16,666个分组)将被丢弃。

[0006] 在此示例中,TCP慢启动算法已花费十一个往返时间(超过两秒)来找到它正确的工作速率。此外,当TCP丢弃这样大数量的分组时,它可能花费长时间来恢复,有时导致更多秒的中断。由于在此时段期间语音分组的至少50％被丢弃,语音流也很可能中断达至少200ms并且常常会更长。

[0007] 因此超调问题有两个主要后果。首先,数据流需要很长时间稳定在针对可用网络容量的适当的速率,并且其次,具有共享网络的现在拥塞部分的传输路径的任何数据流发生非常大量的损害。

[0008] 现在将描述数据分组网络的另外的概念。

[0009] 节点通常具有用于接收数据分组的接收器、用于发送数据分组的发送器和用于存储数据分组的缓冲器。当节点在接收器处接收到数据分组时,数据分组被临时地存储在缓冲器中。如果在缓冲器中当前没有存储其它分组(即新分组不在“队列”中),则分组被立即转发到发送器。如果在缓冲器中存在其它分组而使得新分组在队列中,则它必须等待轮到它才被转发到发送器。现在将描述有关节点缓冲器的管理和利用的几个概念。

[0010] 实现针对其缓冲器的非常基本的管理技术的节点将简单地在其缓冲器中存储任何到达的分组,直到缓冲器达到容量为止。这时,大于缓冲器的剩余容量的任何数据分组将被丢弃。这被称为队尾丢弃。然而,这导致较大分组与较小分组相比被更经常丢弃,所述较小分组可以仍然被添加到缓冲器队列的末尾。对此技术的一个改进是被称为主动队列管理(AQM)的过程,其中,当检测到缓冲器中的分组的队列在开始增长至阈值速率以上时但是在缓冲器满之前,数据分组被丢弃。即便在持续很久的数据流期间,这也给予缓冲器足够的容量以吸收分组的突发。

[0011] 一些节点可以将其缓冲器中的每个数据分组一视同仁,使得数据分组按照它们被接收到的相同顺序被发送(被称为“先进先出”)。然而,节点缓冲器管理技术引入了用不同服务类别来标记数据分组的概念。通过将某些类别定义为比其它类别高,可以使用此技术,并且网络节点然后可实现以较低类别中的分组为代价防止或减轻较高类别中的分组的丢失或延迟的转发功能。使用不同服务类别来管理分组缓冲器的技术的示例包括:

[0012] • (非严格)优先级排序:在此技术中,较高类别分组将被网络节点先于较低类别分组转发,即使较低类别分组更早到达节点。这常常通过向较低类别指派较低权重并与其权重成比例地服务每个类别来实现。

[0013] • 严格优先级排序:与非严格优先级排序类似,但是当在缓冲器中存在较高类别分组时,将不转发较低类别分组。

[0014] • 业务限速器:网络节点可以实施例如对平均速率和最大突发大小指定极限的业务配置文件。不满足配置文件的任何数据流被相应地标记,并且可以在拥塞达到某个水平的情况下被丢弃。

[0015] • 优先丢弃:如果缓冲器被数据分组队列填满,则优先在较高类别分组之前丢弃任何较低类别分组。

[0016] • 选择性分组丢弃:为较高类别分组保留缓冲器的一定比例。较低类别分组可以仅占据缓冲器的较小比例(相对于没有选择性分组丢弃的节点的缓冲器),并且如果该较小

缓冲器满了，则丢弃分组。

[0017] • AQM：如上所述，AQM在检测到缓冲器中的分组队列开始增长至阈值速率以上时丢弃分组。这可被修改为使得被AQM丢弃的分组是较低服务类别的分组。

[0018] 严格优先级排序和优先丢弃的方法都被提出来以确保较低类别分组不能对较高类别分组造成伤害。然而，这些技术仍然存在问题。在严格优先级排序中，一些网络节点可长时间在缓冲器中具有一个或更多个较高优先级分组(许多秒钟或甚至数分钟)，特别是在峰值负载时间期间。这使任何较低类别数据分组长时间地保持在缓冲器中。在此期间，发送/接收节点将可能超时并且数据分组将以较高类别重传(假定较低类别数据分组将被丢弃)。当较高优先级缓冲器中的忙碌时段结束时，较低类别数据分组的缓冲器最终被发送。这仅仅浪费容量，因为数据已从重传的较高优先级分组接收到。

[0019] 网络节点可利用较低类别数据分组来确定网络中的可用容量(称为“探测”)。在优先丢弃中，“符合丢弃条件的”探测数据分组的突发可以填满缓冲器，并且然后仅触发优先丢弃。在探测期间，即使新到达的探测业务被丢弃，符合丢弃条件的分组也将使队列达到丢弃阈值。因此，探测将不是非侵入性的，因为来自已经建立的流的较高类别业务将经历增加的延迟。

[0020] 因此期望减轻以上问题中的一些或全部。

发明内容

[0021] 根据本发明的第一方面，提供了一种在数据分组网络中控制分组的方法，该方法包括以下步骤：源节点经由中间节点通过所述数据分组网络向接收方节点发送第一分组集，其中，所述第一分组集具有使得如果在所述中间节点处存在分组队列则所述中间节点不应把所述第一分组集转发到所述接收方节点的服务类别；所述源节点从所述接收方节点接收对所述第一分组集的第一接收分组的第一确认；以及所述源节点通过所述数据分组网络向所述接收方节点发送第二分组集。

[0022] 根据本发明的第二方面，提供了一种在数据分组网络中控制网络节点的方法，所述网络节点包括缓冲器，所述方法包括以下步骤：从第一外部网络节点接收第一数据分组；分析所述第一数据分组以确定所述第一数据分组是否具有被视为可排队的或不可排队的服务类别；如果所述第一数据分组不可排队，则确定在所述网络节点的缓冲器中是否存储至少一个数据分组；以及如果在所述网络节点的缓冲器中存储有至少一个数据分组，则不发送所述第一数据分组。

[0023] 本发明提供了针对数据分组的新服务类别，其中，如果在发送节点与接收节点之间的中间节点的缓冲器中已存在数据分组，则可以不使所述数据分组排队。如果在所述缓冲器中存在另一数据分组，则所述中间节点可以丢弃所述不可排队分组。可以在发送节点与接收节点之间的数据传送开始时(例如在TCP慢启动算法的指数增长阶段的早期阶段)发送这些分组，并且然后可以在代表发送节点与接收节点之间的传输路径中的瓶颈的所述中间节点处丢弃这些不可排队分组。因此，响应于所述不可排队分组从所述接收节点发送回到所述发送节点的任何确认分组具有将未来分组的传输速率限制为网络的瓶颈速率的效果。本发明因此提供了一种在数据传送中比TCP慢启动早得多地建立适当的传输速率的方法，并且在这样做时，较少的数据分组被丢弃。

[0024] 所述方法还可以包括以下步骤:所述源节点经由所述中间节点通过所述数据分组网络向所述接收方节点发送初始分组集,其中,所述初始分组集具有使得它们能够在所述中间节点处被排队的服务类别。所述源节点发送所述第一分组集的速率可以高于所述源节点发送所述初始分组集的速率。以这种方式,所述源节点接收到所述确认分组由的速率更接近于所述瓶颈速率。

[0025] 所述第一数据分组集可以包含虚设数据。这些第一分组集因此可以是不包含请求的数据的探测分组,并且因此在被丢弃的情况下不必被重传。另选地,所述第一分组集可以包含请求的数据,并且所述确认可以表示阻止所述第一分组集的分组被转发到所述接收方节点;以及所述第二分组集可以包括所述被阻止分组的重传。

[0026] 所述源节点可以是包含所请求的数据的节点(并且因此是传输的原始源)或所述原始源与所述接收节点之间的中间节点。数据分组网络中的任何两个节点可以使用本发明的方法以便建立传输路径的瓶颈速率,并且可以随时这样做。在一个实施方式中,如在下面所描述的,在新数据流开始时执行本发明的方法。另选地,可以在新数据开始之前执行本发明的方法,并且然后可以使用计算出的瓶颈速率来配置新数据流的传输速率。因此,所述方法还可以包括以下步骤:所述源节点从所述接收方节点接收对所述第一分组集的第二接收分组的第二确认;所述源节点基于在所述源节点处接收所述第一确认和第二确认的速率来确定所述源节点与所述接收方节点之间的传输速率;所述源节点基于所确定的传输速率向所述接收方节点发送第二数据分组集。

[0027] 还提供了被配置为执行本发明的第一方面和第二方面所述的方法的网络节点和计算机程序。还提供了一种包括所述网络节点的网络。

[0028] 根据本发明的第三方面,提供了一种在数据分组网络中传输的数据分组,其中,所述数据分组可由所述数据分组网络中的节点识别为具有使得如果在所述节点处存在分组队列则不应转发所述数据分组的服务类别。所述数据分组还可以包括表示所述数据分组是不可排队的标识符。

附图说明

[0029] 为了可以更好地理解本发明,现在将参照附图仅通过示例来描述本发明的实施方式,在附图中:

[0030] 图1是本发明的实施方式的通信网络的示意图;

[0031] 图2a是例示了图1的网络的路由器的示意图;

[0032] 图2b例示了本发明的实施方式的数据分组;

[0033] 图3是例示了在第一场景中图2b的数据分组由图1的通信网络的路由器来处理的示意图;

[0034] 图4是例示了在第二场景中图2b的数据分组由图1的通信网络的路由器来处理的示意图;

[0035] 图5a是例示了处理图2b的数据分组的方法的流程图;

[0036] 图5b是例示了处理图2b的数据分组的另选方法的流程图;

[0037] 图6是表示现有技术的TCP慢启动算法的定时图;

[0038] 图7是例示了本发明的方法的实施方式的定时图;

[0039] 图8是例示了本发明的自计时特性的示意图；

[0040] 图9a是表示针对大数据流的现有技术的TCP慢启动算法的定时图；

[0041] 图9b是表示针对大数据流的图7的方法的实施方式的定时图；

[0042] 图10是例示了中间盒的本发明的第二实施方式的通信网络的示意图；以及

[0043] 图11是表示本发明的方法的第二实施方式的定时图。

具体实施方式

[0044] 现在将参照图1至图2b来描述本发明的通信网络10的第一实施方式。通信网络10是具有客户端11、服务器18、多个客户边缘路由器13、17、多个提供商边缘路由器13、16以及多个核心路由器15的数据分组网络。客户端11经由路径12向服务器18发送数据分组，所述路径穿过多个客户边缘路由器、提供商边缘路由器和核心路由器。技术人员应理解，其它客户端和服务器可以连接到客户边缘路由器，并且其它客户边缘路由器可以连接到提供商边缘路由器。

[0045] 当客户端11沿着路径12发送数据分组时，数据分组最初被转发到第一客户边缘路由器13，第一客户边缘路由器13继续将数据分组转发到第一提供商边缘路由器14。第一提供商边缘路由器14将数据分组转发到核心路由器15，核心路由器15进而继续将数据分组转发到第二提供商边缘路由器16(其可以经由一个或更多个其它核心路由器)。第二提供商边缘路由器16将数据分组转发到第二客户边缘路由器17，第二客户边缘路由器17继续将数据分组转发到服务器18。

[0046] 在图2a中更详细地示出了核心路由器15(并且技术人员应理解，来自图1的任何其它路由器包括类似构造)。核心路由器15包括被适于接收数据分组的接收器15a、处理器15b、包括用于存储等待传输的数据分组的缓冲器20的第一存储器15c以及用于发送数据分组的发送器15d。路由器15的所有模块通过总线15e连接。

[0047] 图2b例示了本发明的数据分组100。数据分组100包括报头部分110和数据净荷部分120。报头部分110被修改为包括数据分组100具有不可排队服务类别的标识符115。路由器15(并且为了完整，通信网络10中的任何其它节点)的处理器15b适于对数据分组的报头部分110解码并且确定数据分组是不可排队的(与可以排队的其它服务类别(诸如最大努力(BE))相反)。如果当前在缓冲器20中未存储其它数据分组，则通信网络10的路由器15可以因此仅存储此UQ分组。

[0048] 技术人员应理解，标识符可以被存储在IPv4或IPv6分组的6比特差异化服务字段(DSfield)、以太网帧的3比特802.1p服务类别(CoS)字段或MPLS帧的3比特业务类别字段中。技术人员也应理解，能使用其它标识符或码点，只要网络中的相关节点理解此标识符/码点指示数据分组是不可排队的即可。现在将参照图3和图4所例示的两个场景对比进行说明。

[0049] 在图3中例示了根据本发明的由核心路由器15处理数据分组的概要的示意图。数据分组经由接收器15a到达核心路由器15。在该图中，根据本发明数据分组具有可排队类别(例如，常规的最大努力(BE)数据分组)或不可排队(UQ)类别。在图3中描绘的场景中，第一分组23到达接收器15d，并且管理功能22(处理器15b的，其通常对到达的数据分组进行分类、调度和排队)确定第一分组具有不可排队类别，而且确定在缓冲器20中不存在数据分

组。管理功能22因此将第一分组23存储在缓冲器20中，同时出队功能21(也由处理器15b实现)将第一分组23转发给发送器15d。

[0050] 当第一分组23被转发到发送器15d时，第二分组24到达接收器15a。管理功能22确定第二分组24是可排队BE分组。在此场景中，第一分组23尚未被完全发送并因此仍存在于缓冲器20中。第二分组24因此在缓冲器20中被存储在第一分组23后面。第三分组25然后到达接收器15a，同时第一分组23和第二分组24仍然存在于缓冲器20中。管理功能22确定第三分组25是UQ分组并且在缓冲器20中已经有数据分组。在这种情况下，管理功能20丢弃该数据分组(即，它被阻止被发送到服务器18)。最后，第四分组26到达，并且再次被确定为可排队BE分组并因此被存储在缓冲器20中。

[0051] 在图4中例示了第二场景，其中，当缓冲器20空时到达接收器15a的第一分组27是可排队BE分组。在出队功能21将该分组转发到发送器15d的同时，管理功能22将此分组存储在缓冲器20中。当第一分组27被转发时，UQ类别的第二分组28到达。管理功能22确定第二分组28是UQ类别的并且缓冲器不是空的。第二分组28因此被丢弃。当另外的可排队BE分组(例如分组29)到达时，即使第一分组27仍在被转发，管理功能也可以将它们存储在缓冲器20中。

[0052] 在以上两个场景中，数据分组在发送器完成它对该分组的最后字节的传输时被视为已离开缓冲器。一旦这个最后字节完成传输，则缓冲器就可以存储不可排队分组。

[0053] 在图5a中示出表示处理器15b的管理功能22的第一实施方式的流程图。在此图的步骤S1中，检查新数据分组以确定它是否是UQ类别的。这可以通过处理器15b对数据分组的报头部分110进行解码并且确定标识符/码点是否与该UQ类别的已知标识符/码点匹配来实现。如果处理器15b确定新数据分组是可排队类别的，则处理器15b将该新数据分组传递给入队功能并且该新数据分组被存储在缓冲器20中(步骤S2)。然而，如果处理器15b确定新数据分组是不可排队的，则处理器15b确定缓冲器20是否为空。如果缓冲器是空的，则处理器15b再次将新数据分组传递给入队功能并且新数据分组被存储在缓冲器20中(步骤S3)。另选地，如果处理器15b确定缓冲器不是空的，则处理器15b丢弃该分组(步骤S4)。

[0054] 在图5b中例示了处理器15b的管理功能22的第二实施方式的流程图。在此实施方式中，确定缓冲器是否为空并且确定分组是否是不可排队的步骤是相反的。

[0055] 不可排队的服务类别可被发送方节点11/接收方节点18对利用，以便确定要在通信网络10中使用的适当传送速率(即，在不使任何分组被丢弃或者使相同传输路径的数据流共享部分上的分组被丢弃的情况下可用来发送数据的最大速率)。在描述此算法的实施方式之前，将参照图6来呈现常规的TCP慢启动过程及其对应的定时图的概要。

[0056] 图6是两个时间轴从客户端(例如客户端11)和服务器(例如服务器18)向下延伸的定时图。各种数据分组由在两个时间轴之间延伸的箭头表示，箭头表示数据分组由客户端或服务器发送或接收(使得箭头的根部表示客户端/服务器发送分组的时间，并且箭头的头部表示客户端/服务器接收到数据分组的时间)。数据分组通常将穿过多个客户边缘路由器、提供商边缘路由器和核心路由器(如图1所例示)，但是为了简单起见仅例示了两个终端系统。在TCP慢启动过程中，客户端向服务器发送针对数据的初始请求。服务器通过缓冲要发送到客户端的数据分组流来响应并且将其初始拥塞窗口设定为三个分组的当前标准TCP大小。因此，服务器从缓冲器向客户端发送三个数据分组(由粗的实箭头表示)，所述分组全

部被标记为BE服务类别。

[0057] 在此示例中，这三个数据分组不经历任何拥塞并且全部被客户端以及时方式接收。客户端因此向服务器发送针对三个数据分组中的每一个的确认分组(由细的实箭头来表示)。服务器收到这些确认，并且作为响应，增加拥塞窗口(针对接收到的每个确认增加一个分组)。服务器因此在下一次传输中发送六个数据分组。在图6中，在服务器完成将所有数据分组从其缓冲器传送之前，存在数据分组被发送和确认被接收的四次往返。算法因此保持在指数增长阶段中直到它完成传送为止。

[0058] 技术人员应理解，如果数据流大得多，则TCP慢启动算法将针对接收到的每个确认将其拥塞窗口增加一个分组，直到它达到其慢启动阈值为止。一旦达到了此阈值，然后拥塞窗口就在它一个往返时间内(即在发生超时之前)接收到确认的情况下增加一个分组，而不管在那时接收到多少确认。该算法因此从指数增长阶段移动到线性拥塞避免阶段。技术人员也应理解，如果在没有接收到任何确认的情况下发生超时，或者接收到指示分组已被丢弃的确认，则拥塞窗口被减半。

[0059] 现在将参照图7来描述本发明的方法的实施方式。图7也是两个时间轴从客户端11和服务器18向下延伸的定时图，其中各种数据分组由在两个时间轴之间延伸的箭头来表示，箭头例示数据分组由客户端11或服务器18发送或者接收(使得箭头的根部表示发送方/接收方节点发送分组的时间，并且箭头的头部表示发送方/接收方节点接收到分组的时间)。再次，数据分组穿过多个客户边缘路由器13、17、提供商边缘路由器14、16和核心路由器15(如图1所例示)，但是为了简单起见，仅示出了两个端系统。

[0060] 本发明的方法的初始步骤与上面概述的慢启动方法非常相似。客户端11向服务器18发送针对数据的初始请求52。服务器18通过缓冲要发送到客户端11的数据分组流来响应并且将其初始拥塞窗口设定为三个分组的当前标准TCP大小。因此，服务器18从缓冲器向客户端11发送三个数据分组54，三个数据分组全部被标记为BE服务类别(由粗的实箭头表示)。

[0061] 在这点上，本发明的方法不同于常规的慢启动算法。紧跟在初始的三个BE数据分组之后，服务器18继续从缓冲器向客户端11发送另外的数据分组55。这些另外的数据分组中的每一个被标记为UQ(例如，报头部分包含通信网络10中的所有节点识别为具有不可排队类别的标识符/码点)，并且在此实施方式中，被以比前三个BE分组高的传输速率发送。这些UQ数据分组在图7中由虚线箭头表示。

[0062] 初始BE数据分组以及随后的UQ数据分组的突发以服务器的发送器的最大速率离开服务器18。在此示例中，这通过服务器18上的网络接口与第二客户边缘路由器17之间的1GB/s连接(例如1Gb/s以太网链路)。一旦这些BE和UQ分组到达第二客户边缘路由器17，它们就被转发到第二提供商边缘路由器16。在此示例中，这通过500Mb/s接入链路。因此，当第一UQ分组到达第二客户边缘路由器17时，第二客户边缘路由器17的相对较慢的输出速率(即，相对于接收来自服务器18的分组的传输速率，向第二提供商边缘路由器16转发分组的较慢传输速率)表示通信网络10中的瓶颈。第二客户边缘路由器17的缓冲器20因此将必须根据早先描述的管理功能22对接收到的数据分组进行排队。

[0063] 因此，前三个BE分组到达第二客户边缘路由器17。所有这些BE分组的报头部分被解码，并且管理功能22确定它们全部是可排队BE分组。在此示例中，在缓冲器20中最初没有

其它数据分组。因此，所有三个BE分组均被存储在缓冲器20中并且这些BE分组中的第一个分组被转发到发送器。

[0064] 如上所述，在这些初始三个BE分组之后，从服务器18向第二客户边缘路由器17发送UQ分组的流。这些UQ分组中的第一个分组到达第二客户边缘路由器17并且报头部分被解码。管理功能22确定该分组是UQ分组。管理功能22还确定缓冲器20不是空的(因为当第一UQ分组到达时，三个BE分组尚未全部被发送)并因此丢弃第一UQ分组。所丢弃的UQ分组由具有在图7中的服务器18与客户端11之间的区域中终止的菱形头部(而不是箭头头部)的线来表示。

[0065] UQ分组中的第二个分组到达第二客户边缘路由器17并且报头部分被解码。管理功能22再次确定该分组是UQ分组并且也再次确定缓冲器20不是空的。第二UQ分组因此被丢弃。

[0066] 最后，所有三个BE分组被成功地发送到第二提供商边缘路由器16并且第二客户边缘路由器17的缓冲器20是空的。第三UQ分组然后到达第二客户边缘路由器17并且报头部分被解码。再次，管理功能22确定该分组是UQ分组但是现在确定缓冲器20是空的。第三UQ分组因此被存储在缓冲器20中并被转发到发送器57，以供继续传输到提供商边缘路由器16(最终传输到客户端11)。这在图7中被例示为从服务器18的时间轴(表示第三UQ分组)向客户端11延伸的第三虚线。

[0067] 当第三UQ分组被发送时，第四UQ分组到达并且报头部分被解码。管理功能22确定该分组是UQ分组并且缓冲器不是空的(因为第三UQ分组在它被发送时被存储在缓冲器20中)。第四UQ分组因此被丢弃。

[0068] 同时，如图7所示，初始的三个BE分组到达客户端11。作为响应，客户端11向服务器18发送回表示BE分组被成功接收到的三个BE确认消息58。注意，出于描述的目的，术语BE确认消息和UQ确认消息分别用于区分响应于BE或UQ消息而发送的确认消息，但是不一定暗示消息本身之间的任何差异。

[0069] 当这些BE确认消息穿过通信网络10到服务器18时，服务器18继续向客户端11发送UQ分组。如以上所指出的并如图7所示，这些UQ分组中的一些分组成功地穿过通信网络10并到达客户端11，同时一些分组被中间节点(例如第二客户边缘路由器17，第二客户边缘路由器17丢弃UQ分组，因为在前一个的UQ分组被发送时，前一个UQ分组存在于缓冲器20中)丢弃。每当UQ分组成功地到达客户端11时，客户端11发出UQ确认消息59。

[0070] 如图7所示，客户端11发出BE确认消息(即响应于初始的BE分组)的速率大于客户端11响应于UQ分组而发出UQ确认消息的速率。这是由于一些UQ分组被丢弃，只要分组经历了服务器18与客户端11之间的中间节点处的、伴随有由服务器发送的每个UQ分组之间的小时间延迟的队列。如现在将描述的，这对服务器18如何为通信网络10确定适当的传输速率具有重要的后果。

[0071] 当第一BE确认消息到达服务器18时，服务器18停止向客户端11发送UQ数据分组。服务器18被配置为在接收到此BE确认消息时结束其启动阶段并进入拥塞避免阶段。像常规的TCP慢启动算法一样，本发明的此实施方式的算法是“自计时”，使得新数据分组响应于服务器接收到的每个确认而从服务器18向客户端11发送。在此实施方式中，在从客户端11接收到第一BE确认分组之后，服务器18开始向客户端11发送第二批BE分组60。此第二批的前

三个BE分组被以与它接收前三个BE确认消息的速率相对应的传输速率发送。然而，从图7中将看到，服务器18然后开始接收UQ确认消息(由客户端11响应于成功接收到的UQ分组而发送)61。这些UQ确认消息中的每一个具有修改第二批BE分组中的下一个BE分组的传输速率的效果。在此示例中，如以上所指出的，UQ确认分组的速率低于初始BE确认消息的速率，并且算法的自计时性质因此将下一个BE分组的传输速率降低至瓶颈速率。此新速率略低于瓶颈速率，因为每当存在队列时并且由于由服务器发送的连续UQ分组之间的小时间延迟，UQ分组被中间节点丢弃。然而，如果此时间延迟被最小化(例如，通过使用需要最小处理的小UQ分组)，则可将两个速率之间的这个差异减小至可忽略的量。

[0072] 可使用图8所示的示意图来说明这种自计时性质。此图例示了通过相对低的带宽链路连接的高带宽网络上的发送方和接收方节点(在该图中，垂直维度表示带宽并且水平维度表示时间)。发送方节点向发送方连续地发送一连串分组(每个用交叉阴影示出)。当分组穿过低带宽链路时，分组必须及时传播(因为每个分组中的比特的数量保持不变)。时间P_b表示路径中的最慢链路上的最小分组间距(即瓶颈)。当分组离开瓶颈进入接收方的相对高的带宽网络时，分组间间隔保持不变(即$P_r=P_b$)。接收方节点然后按照与分组被接收的速率相同的速率向发送方节点发送确认分组(假定处理时间是可忽略的)，所以这些确认分组之间的间距与分组间间隔相同(即，$A_r=P_r=P_b$)。确认分组通常小于原始分组，所以确认分组应该穿过低带宽链路，而在间隔方面没有任何变化(即$A_b=A_r$)。因此，如果来自发送方的任何后续分组是仅响应于接收到确认分组而发送的，则发送方的后续分组之间的间距将与网络中的最慢链路上的瓶颈速率完全匹配。

[0073] 因此，如图7所示，服务器18继续按照此瓶颈速率发送第二批BE分组。这发生了，直到它从客户端11接收到来自第二批BE分组中的第一分组的确认为止。因此，从图7中的服务器18发送的最后几个分组根据在服务器18处接收到这些确认消息的速率来发送。

[0074] 技术人员应理解，要到达服务器18的第一UQ确认消息将表示一些数据尚未到达客户端11(由于一些UQ分组被丢弃了)。服务器18因此通过将数据包括在第二批BE分组中来重传此数据。此行为因此修复了UQ分组中的所有数据的丢失。一旦所有此丢失数据被重传，服务器18就将发出任何剩余的新数据，直到它缓冲的数据已全部发送为止。服务器然后将终止连接(未示出)。

[0075] 本发明的方法因此使用新的UQ分组来探测网络，并且更快速地建立通过网络的端到端路径的适当传输速率。这在本发明的算法与用于较大数据流的TCP慢启动相比较时是清楚的，如图9a和图9b所示。

[0076] 图9a例示了针对较大数据流的TCP慢启动算法。该算法按照与针对图6所描述的相同的方式开始，使得客户端11向服务器18发送请求并且传输速率进入指数增长阶段。当在两个终端系统之间有更多的数据分组要传送时，图9a从图6继续。由服务器18接收到的每个确认分组使其拥塞窗口增加两个分组，从而增加服务器18与客户端11之间的传输速率。在来自服务器18的分组的第四传输中，传输速率大于网络中(例如在第二提供商边缘路由器处)的瓶颈速率，并且瓶颈路由器的缓冲器是满的。因此，瓶颈路由器丢弃分组(由具有菱形头部的虚线(71)表示)。随后，当传输速率达到瓶颈速率的两倍时，所有发送的数据分组的大约一半将被丢弃。这将继续，直到在第一被丢弃的分组之后的第一个分组的确认到达服务器18为止。这时，服务器18确定已发生丢失并且通过使拥塞窗口减半来响应。它然后将在

恢复发送新数据之前重传来自丢失的数据分组的数据，直到传送完成为止。

[0077] 从图9a中将看到，当传送速率在TCP慢启动算法的指数增长阶段末期超越瓶颈速率时，这种行为导致BE类别中的大量数据丢失。如果其它数据流在网络的这个现在拥塞的部分中共享同一传输路径，则此超调将使其它数据流遭受与所例示的流相同比例的丢失。

[0078] 图9b例示了相同的数据传送，但是使用本发明的方法。通过将此数据流与图9a的TCP慢启动算法相比较，可看到本发明的方法在数据传送开始时遭受数据分组的丢失。然而，在第一轮数据分组之后，服务器18确立网络的瓶颈率并且(由于算法的自计时性质)，以此瓶颈速率发出将来的分组。本发明的方法因此比TCP慢启动算法更快地确立网络的瓶颈速率，并且在未显著地超越瓶颈速率的情况下实现这个(否则，这对于该数据流并对于其它数据流将引起分组的显著丢弃)。

[0079] 现在将参照图10和图11来描述本发明的第二实施方式。此实施方式与第一实施方式基本上相同。在此实施方式中，中间盒83(其可以作为广域网(WAN)加速器的一部分被提供)被提供并连接到第二客户边缘路由器17。即使客户端81和服务器85尚未适于按照本发明的第一实施方式中描述的方式发送分组，此中间盒83也使得网络能够利用不可排队服务类别。现在将参照图11对此进行说明，图11也是三个时间轴从客户端81、中间盒83和服务器85向下延伸的定时图。再次，各种数据分组由在三个时间轴之间延伸的箭头来表示，箭头例示了数据分组正由客户端81、中间盒83或服务器85发送或者接收(使得箭头的根部表示发送方/接收方节点发送分组的时间，并且箭头的头部表示发送方/接收方节点接收到分组的时间)。再次，数据分组穿过多个客户边缘路由器13、17、提供商边缘路由器14、16和核心路由器15(如图1所例示)，但是为了简单起见，仅例示了客户端、中间盒和服务器。

[0080] 客户端81向服务器85发送针对数据传送的请求分组82。在此实施方式中，中间盒83拦截此请求分组82(例如，通过监视穿过第二客户边缘路由器17的所有数据分组并确定任一个分组是否是请求分组)，并且打开回到客户端81的连接。中间盒83还不能发送客户端81向服务器请求的数据，因为中间盒83未存储所述数据。中间盒83因此将请求继续转发(84)到服务器85。服务器85然后开始到中间盒83的传统的TCP数据传送。

[0081] 在此实施方式中，不需要以任何方式修改服务器85。服务器85与中间盒83之间的数据传送因此可根据传统的TCP慢启动算法进行，这被例示在图11中(参照86)。在此实施方式中，服务器85与中间盒83极为接近。因此，数据传送与使用TCP慢启动算法的广域网上的数据传送相比在少得多的时间内加速到全速。

[0082] 然而，如图11中可看到的，一旦中间盒83从服务器85接收到数据分组的流，中间盒83就可以发起不可排队服务类别数据传送，如以上在第一实施方式中所描述的。也就是说，中间盒83可以将任何数据分组重新分类为不可排队分组(例如，通过给予分组不可排队服务类别标记，代替由服务器85应用于分组的标记)，并且发送后面有UQ分组流的三个数据分组，如以上描述的那样。

[0083] 第二实施方式的优点是服务器85与中间盒83之间的传统TCP慢启动交换可以在相对短的时间内加速到非常快的速率(与通过WAN的传统TCP交换相比)，然后数据传送被转换成不可排队服务类别数据传送以确立WAN上的瓶颈速率。也可以在不对服务器85进行任何修改的情况下实现这个，使得仅从客户边缘路由器起的节点(由网络运营商维护)需要能够区分不可排队分组和任何其它服务类别的分组。

[0084] 技术人员应理解,网络能实现第二实施方式的两个中间盒,使得一个中间盒与服务器相关联,并且另一个中间盒与客户端相关联,使得能在正向和反向二个方向上均实现本发明的优点。

[0085] 在以上实施方式的增强中,客户端与服务器之间的任何中间节点能以比它正常的传输速率略低的速率使分组出队。以这种方式,较大数量的UQ分组将被中间节点丢弃,并且因此被返回到服务器的UQ确认分组的速率降低。当这些UQ确认分组时钟输出来自服务器的另外分组时,可以将新传输速率人为地降低至通过以上概述方法确立的速率。因此,这可提供更安全的传输速率,该传输速率仅小于网络的瓶颈速率。

[0086] 在另一增强中,管理实体可以连接到网络中的节点(优选地,提供商边缘节点),管理实体可以监视通过该节点的数据分组以确定正在以不可排队服务类别发送的分组的比例。这可以通过节点的带报头解码器功能的接口以及适当的记录机制来实现。另选地,能使用深度分组检测技术。管理实体使得网络运营商能够确定不同客户端对不可排队服务类别的使用,并且因此可有助于部署规划。

[0087] 在以上实施方式中,服务器18经由客户边缘路由器和提供商边缘路由器向核心网络路由器发送分组。然而,这是非必要的,并且技术人员将理解可以在经由至少一个中间节点进行通信的任何两个网络节点之间实现本发明。例如,服务器可以直接连接到核心路由器15(例如,可以是服务器用于流行视频流网站的高带宽存储服务器的情况)。在这种情况下,瓶颈节点很可能在更远的中间节点(诸如与客户端关联的提供商边缘路由器)处,并且可通过此节点丢弃UQ分组来确立瓶颈速率。此外,实现本发明的两个网络节点可以处于对等布置中,而不是以上详述的服务器/客户端布置。

[0088] 在以上实施方式中,UQ分组通过分组的报头部分中的特定标识符被标记为不可排队的。然而,技术人员应理解,确保分组为不可排队的这种方法是非必要的。也就是说,分组可以通过使用分组中的任何点处的标识符而被标记为不可排队的,只要网络中的任何节点能够对此标识符进行解码即可。此外,这个标记不一定需要是一致的,因为节点可以使用深度分组检查来确定服务类别,而不必对标识符进行解码。技术人员应理解,UQ分组根本不需要任何标记可标识为具有不可排队服务类别。替代地,可以从分组的特定特性(诸如其协议、它被寻址到特定地址范围等)推导不可排队服务类别。中间节点然后可基于此推导而将分组视为不可排队的。因此,技术人员应理解,如果在节点中存在分组队列,则“不可排队”数据分组是网络节点通常理解为不应被排队的数据分组。

[0089] 在以上实施方式中,UQ分组包括作为要从服务器发送到客户端的数据的一部分的数据,并且作为丢弃的UQ分组的结果而丢失的任何数据被服务器重新发送。然而,UQ分组可以替代地包括虚设数据(即,不是由客户端请求的数据的一部分并且通常只是随机的比特合集的数据)。以这种方式,需要由服务器重传的数据分组更少。

[0090] 技术人员也应理解,TCP协议的使用是非必要的,并且本发明可以被应用在实现拥塞控制的许多其它传输协议(诸如数据报拥塞控制协议上的流控制传输协议或实时传输协议)中。

[0091] 以上实施方式描述了在新数据流开始时在服务器与客户端之间操作的本发明。然而,技术人员应理解,可以为了确立网络中的瓶颈速率随时使用本发明。例如,服务器可能已与多个客户端建立数据流,并且这些数据流中的一个可以终止。服务器然后可以使用本

发明的方法来快速地探测网络并且针对其剩余的数据流确立新瓶颈速率。此外，技术人员应理解，在核心网络的入口点和/或出口点处提供中间盒的本发明的方法的第二实施方式可以用于探测网络以确定瓶颈容量。此后，当从新流从与该中间盒关联的客户端开始时，可基于此信息来设定传输速率。

[0092] 在以上实施方式中，中间节点被配置为一旦最后分组的数据的最后字节离开发送器就确定其缓冲器是空的。然而，技术人员应理解，发送器也可以实现缓冲器以在分组被发送时临时地存储它们。因此，在确定节点缓冲器是否是空的时并因此确定新的UQ分组是否可被排队时，节点可以忽视存储在此临时发送器缓冲器中的任何分组。

[0093] 技术人员应理解，存在可以实现“不可排队”分组的多个方式。在以上实施方式中，不可排队分组被中间节点接收，并且中间节点确定分分组是可排队的还是不可排队的，并且如果确定是不可排队的，则确定在节点处是否存在分组队列。如果在那时存在分组队列，则分组被丢弃(例如被删除)。然而，丢弃数据分组不是必要的。在更被动的布置中，分组可以只是从不被转发到其目的地。

[0094] 技术人员应理解，特征的任何组合在如要求保护的本发明的范围内是可能的。

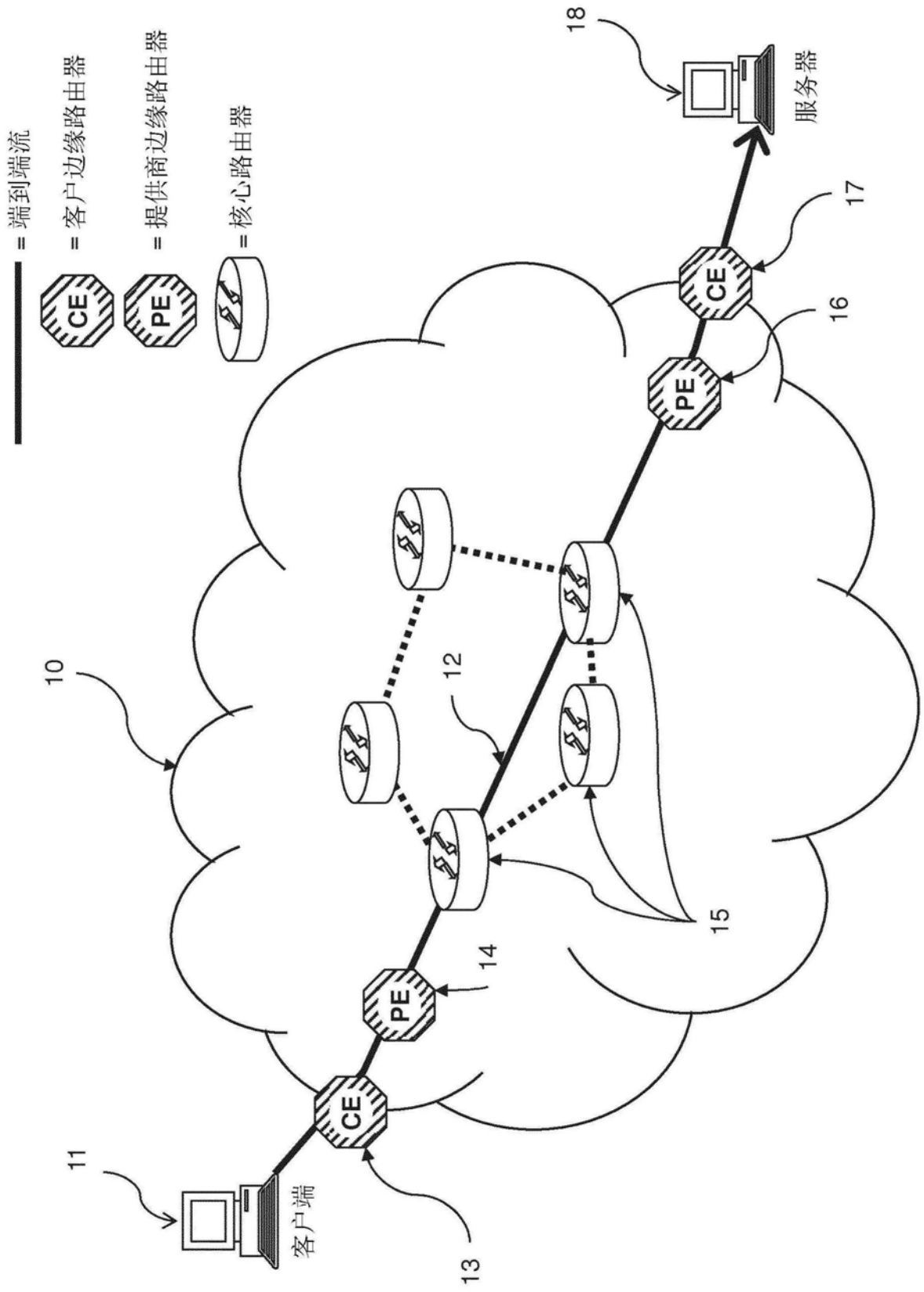

= 端到端流
= 客户边缘路由器
= 提供商边缘路由器
= 核心路由器
CE
PE
10
11
客户端
12
13
14
15
16
17
18
服务器

图1

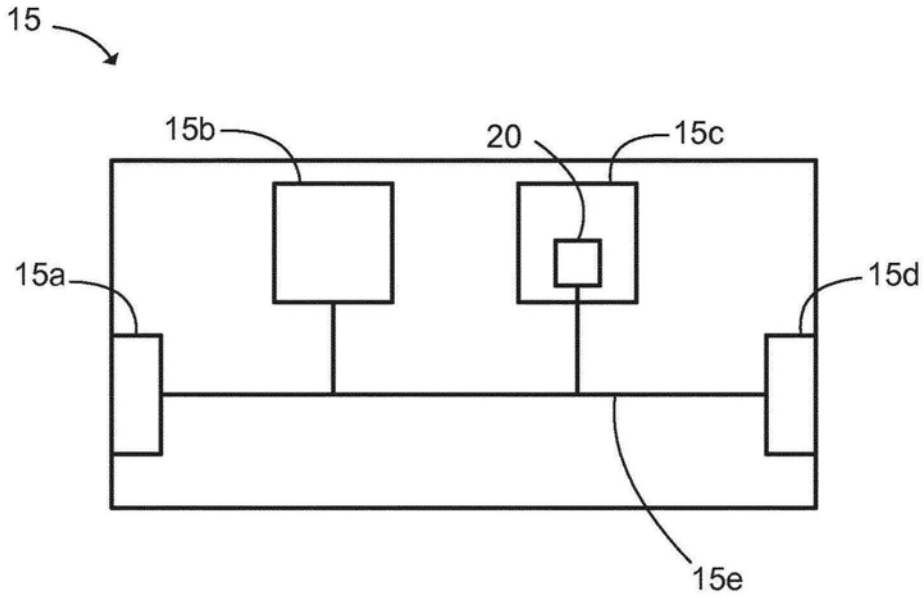


图2a

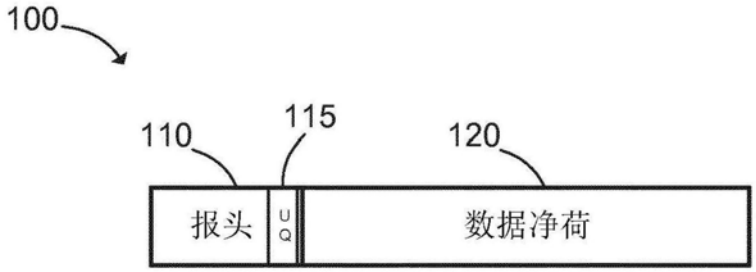


图2b

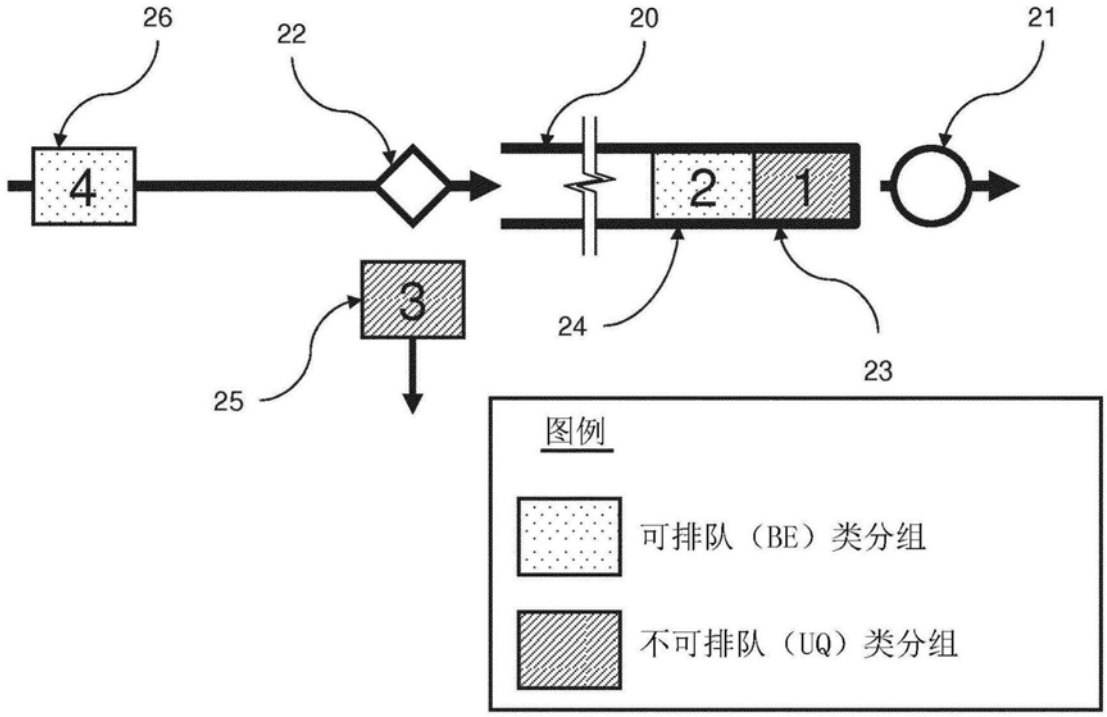


图3

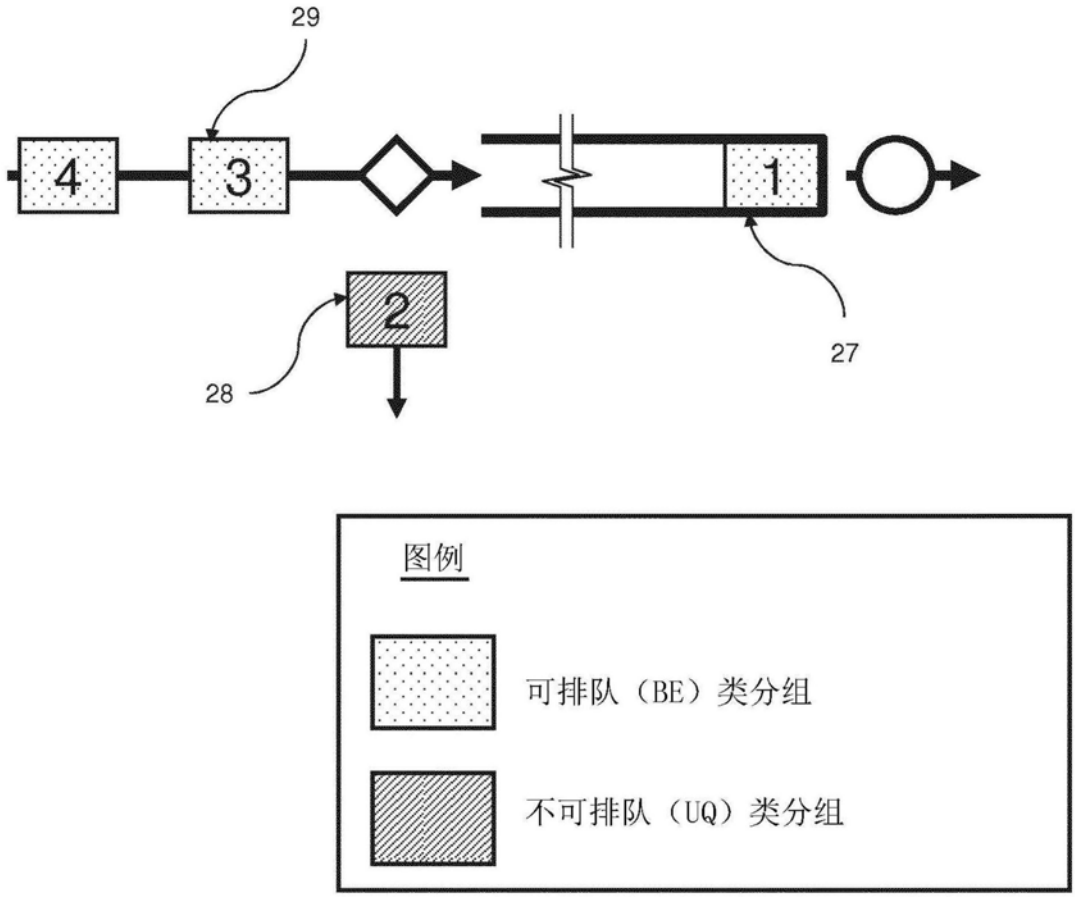


图4

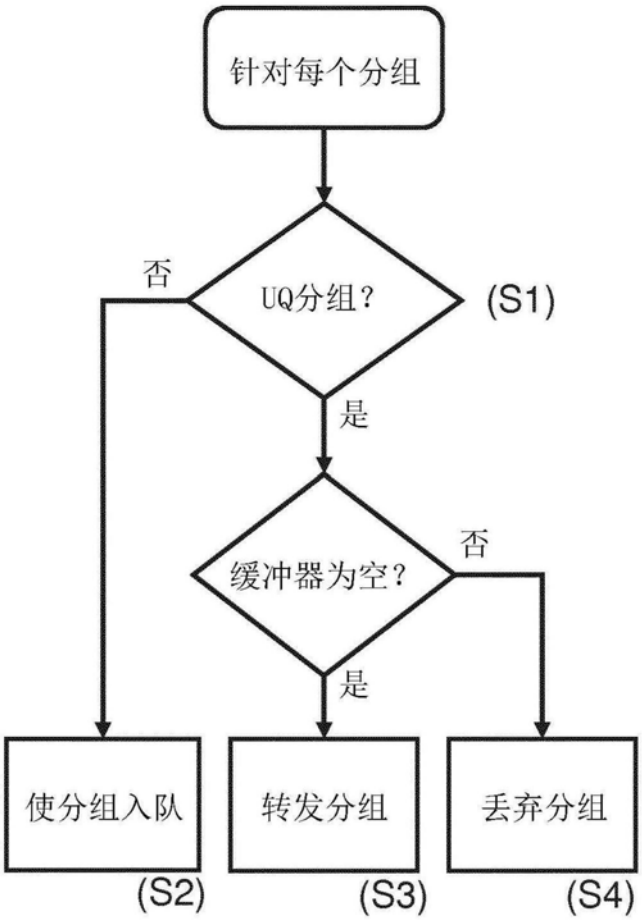


图5a

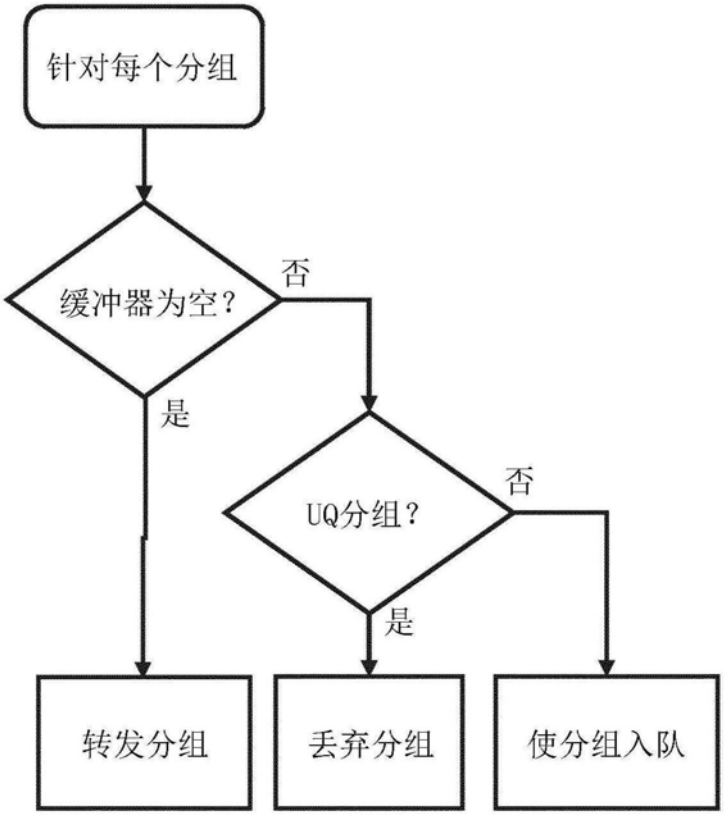


图5b

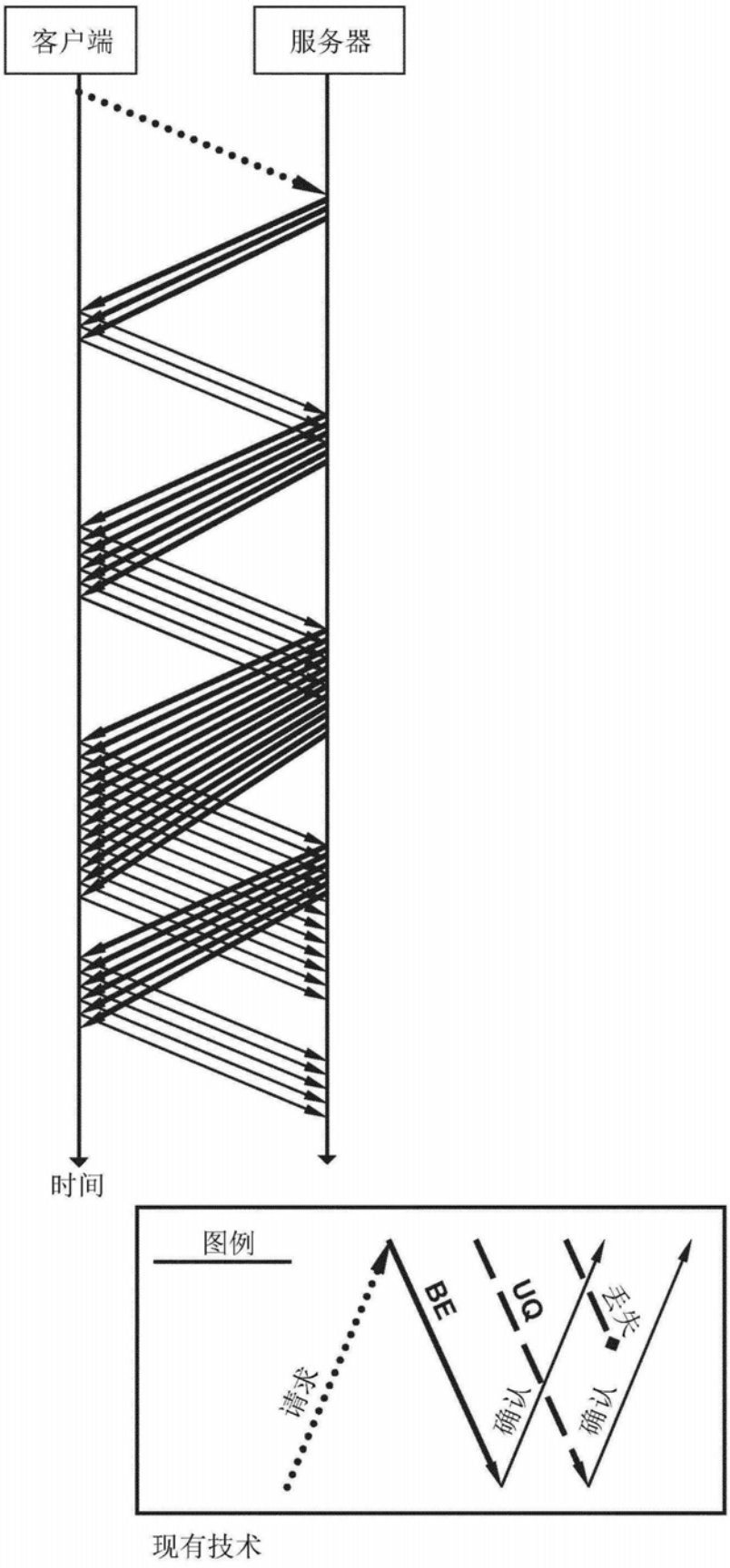


图6

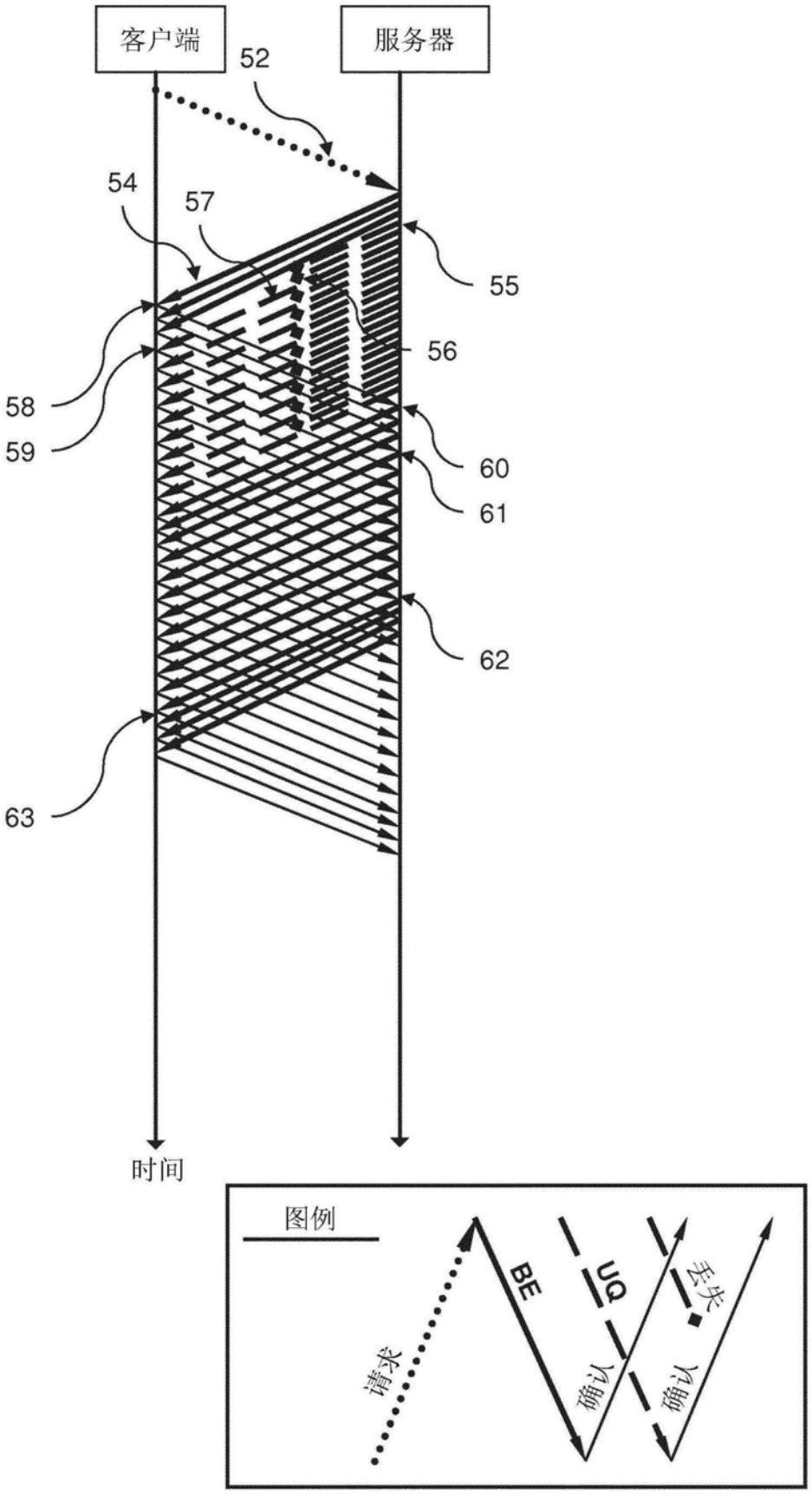


图7

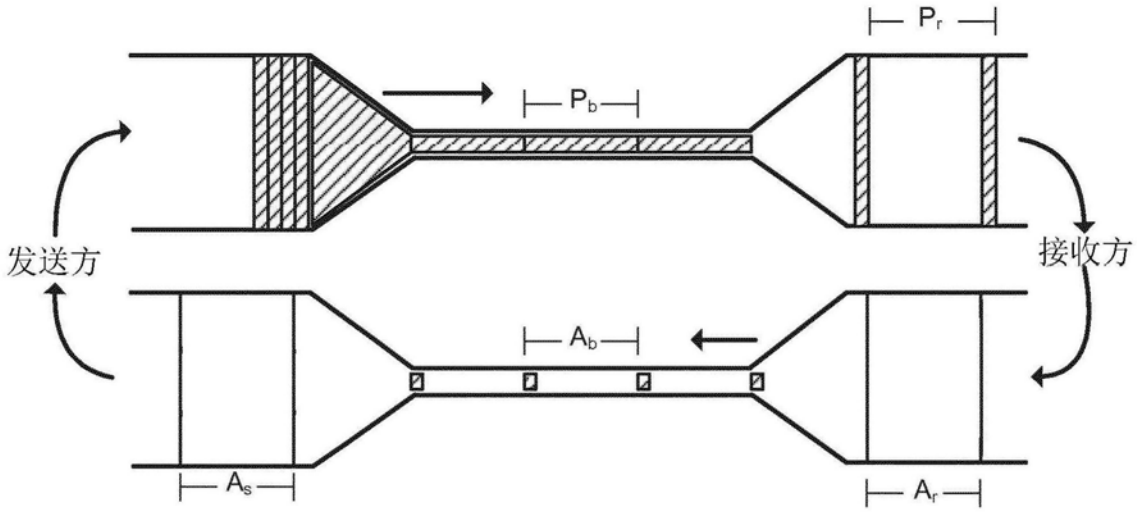


图8

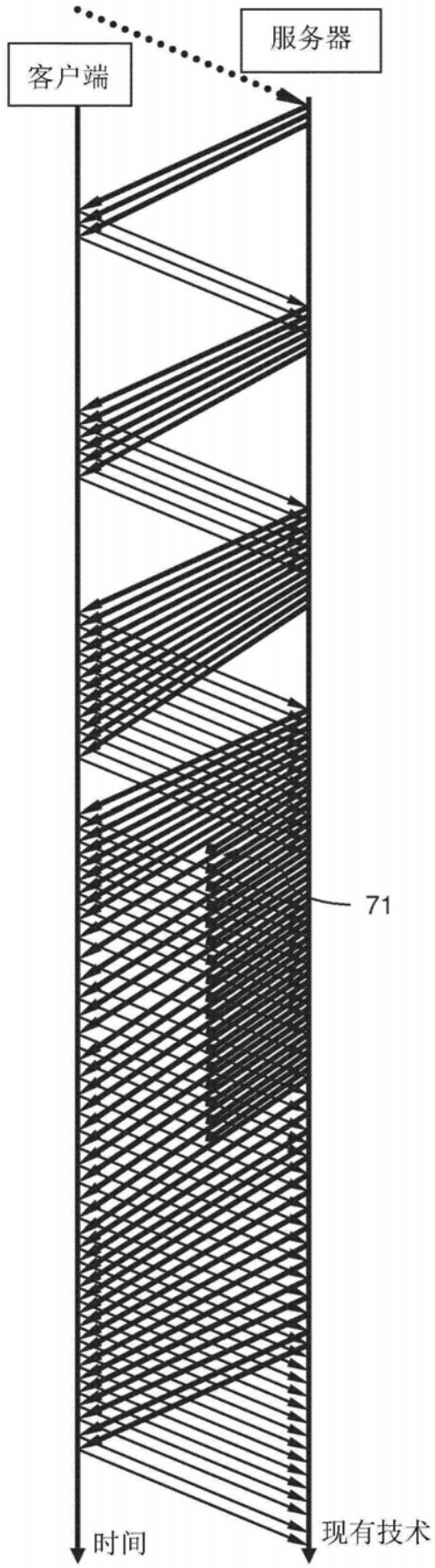


图9a

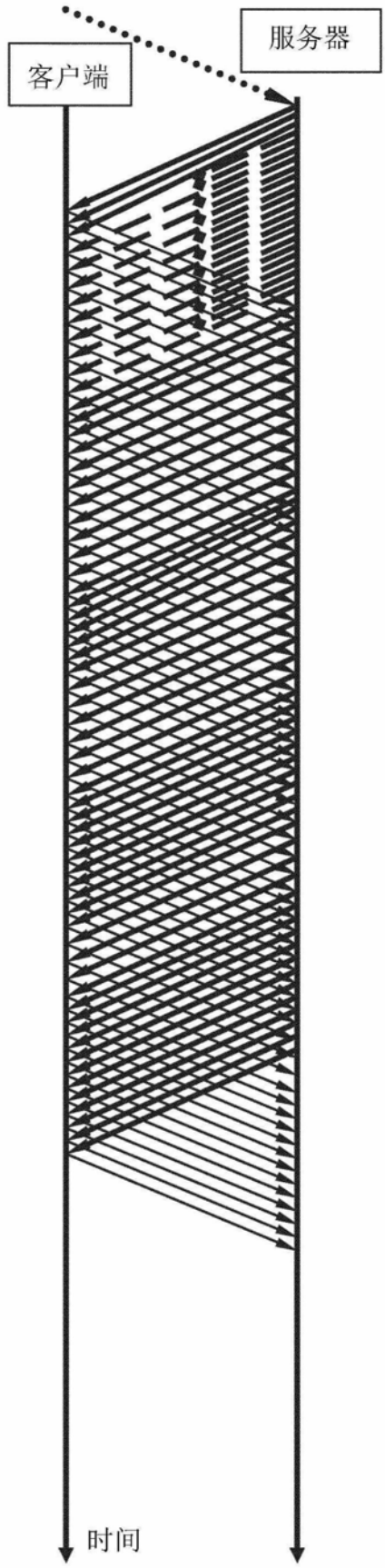


图9b

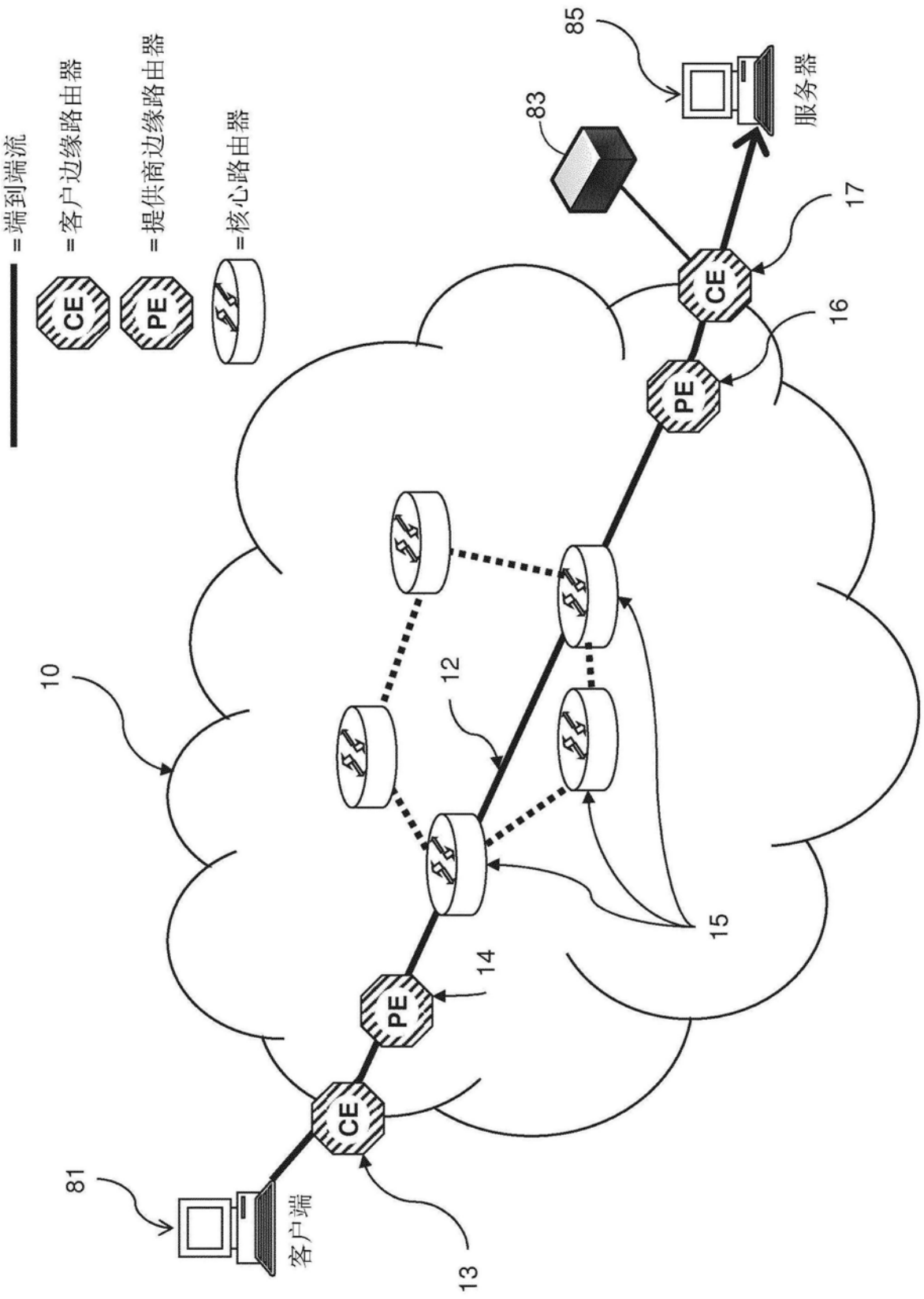

=端到端流
=客户边缘路由器
=提供商边缘路由器
=核心路由器
CE
PE
10
12
13
14
15
16
17
81
客户端
83
85
服务器

图10

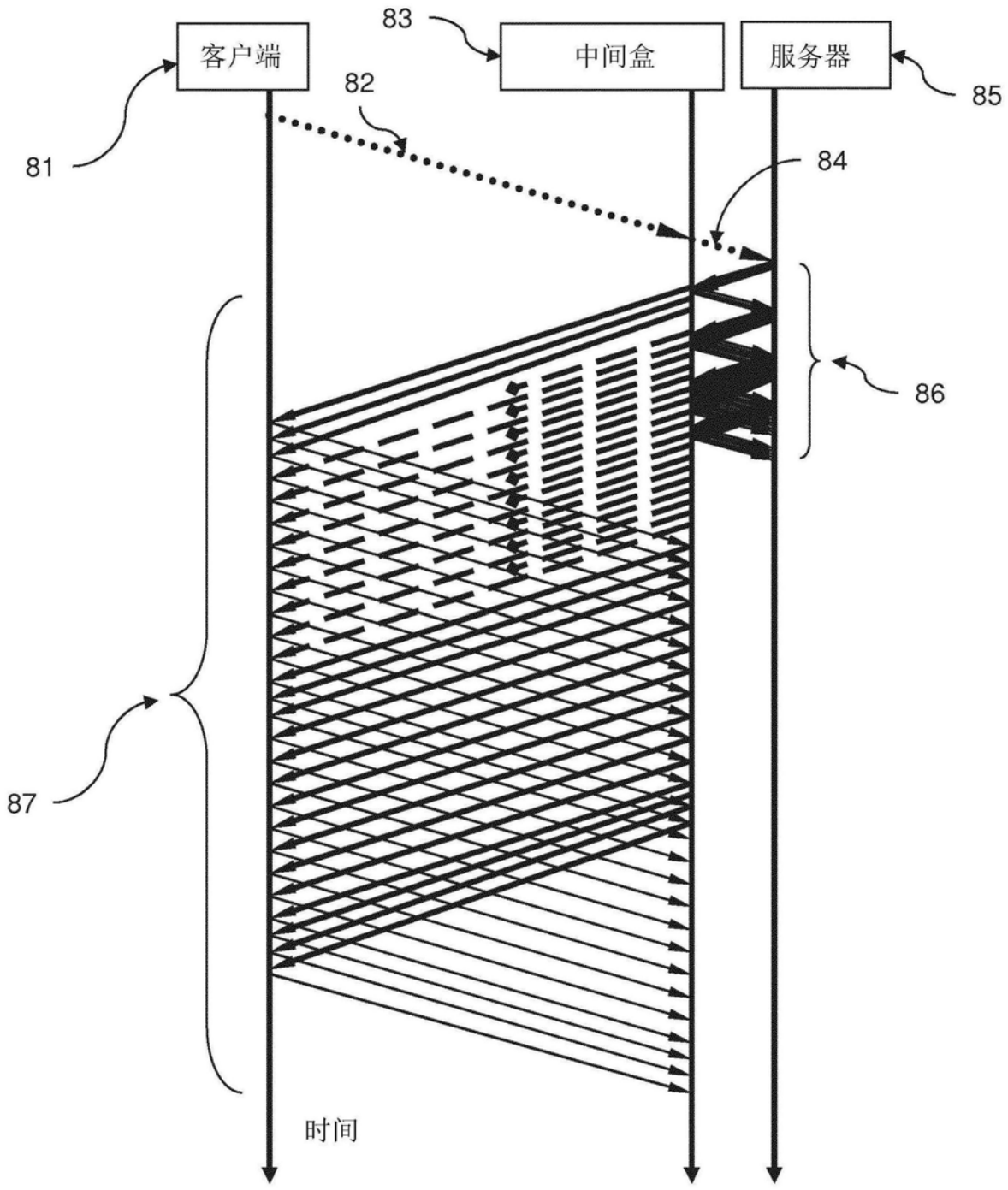


图11