



(12) 发明专利申请

(10) 申请公布号 CN 102859921 A

(43) 申请公布日 2013. 01. 02

(21) 申请号 201180020856. X

(74) 专利代理机构 北京信慧永光知识产权代理有限公司 11290

(22) 申请日 2011. 03. 03

代理人 褚海英 武玉琴

(30) 优先权数据

12/718, 650 2010. 03. 05 US

(51) Int. Cl.

H04L 1/00 (2006. 01)

(85) PCT申请进入国家阶段日

H04L 1/08 (2006. 01)

2012. 10. 25

(86) PCT申请的申请数据

PCT/IB2011/050917 2011. 03. 03

(87) PCT申请的公布数据

W02011/107963 EN 2011. 09. 09

(71) 申请人 生活服务集团有限公司

地址 加拿大安大略

(72) 发明人 马修·R·威廉斯

莫汉·克里希纳·韦米拉帕利

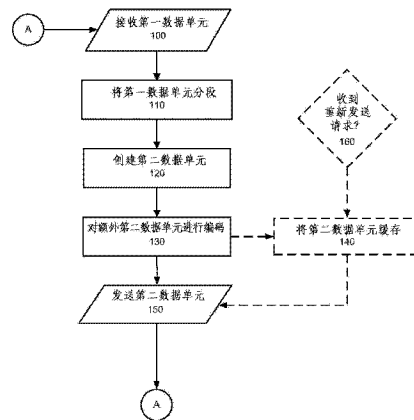
权利要求书 6 页 说明书 15 页 附图 11 页

(54) 发明名称

用于实现加速的吞吐量的系统和方法

(57) 摘要

本发明公开了一种经由编码通道在两个端点之间用于传输数据的系统和方法。在逻辑上位于所述端点之间的编码组件处接收来自源网络的数据传输单元(数据单元)。这些第一数据单元被细分为第二数据单元并且通过所述传送网络被发送至目标网络。还发送编码或额外第二数据单元,使得即使一些所述第二数据单元丢失也能够重建所述原始第一数据单元。这些编码第二数据单元可以仅仅是被发送的所述第二数据单元的拷贝、奇偶校验位第二数据单元或者是已经使用纠错编码来编码的第二数据单元。在接收端点处,接收所述第二数据单元并且用来重建所述原始第一数据单元。



1. 一种在不可靠网络上加速数据通信的方法,所述方法包括:
  - 提供与两个端点中的每一个相关联的编码组件以进行通信;
  - 在所述编码组件之间建立用于在所述端点之间进行通信会话的编码通道;
  - 在一个编码组件处拦截与所述通信相关的多个数据包;
  - 将所述多个数据包中的每一个分段并且打包,从而提供用于传输至另一编码组件的具有至少一个编码数据段和至少一个额外编码段的已分段和打包的数据包;
  - 在所述一个编码组件处存储所述已分段和打包的数据包;
  - 当满足预定的条件时,将所述已分段和打包的数据包经由所述编码通道传输至所述另一编码组件;
  - 在所述另一编码组件处,基于接收到的所述编码数据段和接收到的所述额外编码段将所述多个数据包解码并重组;并且
  - 将已重组的数据包从所述另一编码组件发送至其相应的端点。
2. 根据权利要求1所述的方法,其中,所述预定的条件是定时器到期。
3. 根据权利要求2所述的方法,其中,当所述多个数据包中的相应的一个被拦截时,所述定时器被激活。
4. 根据权利要求1所述的方法,其中,所述预定的条件是拦截到含有实时敏感数据的数据包。
5. 根据权利要求1所述的方法,其中,所述预定的条件是存储了预定数量的所述已分段和打包的数据包。
6. 根据权利要求1所述的方法,其中,所述多个数据包的所述编码数据段和所述至少一个额外编码段以保留它们的顺序的方式被交织并且传输。
7. 根据权利要求6所述的方法,其中,所述已分段和打包的数据包根据它们各自的大小被依次存储,并且依次从各所述已分段和打包的数据包传输各个数据段。
8. 根据权利要求1所述的方法,其中,所述多个数据包中的每一个的所述编码数据段和所述至少一个额外数据段被以随机顺序传输。
9. 根据权利要求1所述的方法,还包括步骤:如果在预定的时段内在所述另一编码组件处未接收到一个编码数据段,则判定所述一个编码数据段丢失。
10. 根据权利要求9所述的方法,其中,所述预定的时段是由当接收到给定数据包的第一个编码数据段时所设定的计时器的超时确定的。
11. 根据权利要求9所述的方法,其中,所述预定的时段是通过参考环形缓冲来确定的。
12. 根据权利要求10所述的方法,其中,所述环形缓冲跟踪丢失的编码数据段的序列号,并且当所述环形缓冲循环返回给定的编码数据段的序列号时,则判定所述给定的编码数据段丢失。
13. 根据权利要求9所述的方法,还包括步骤:当判定来自给定的所述已分段和打包的数据包的预定数量的编码数据段已经丢失时,将来自所述多个数据包的数据段交织。
14. 根据权利要求1所述的方法,其中,将数据包分段并打包包括将所述数据包分段并打包为  $n$  个编码数据段,其中  $n > 1$ 。
15. 根据权利要求14所述的方法,其中,  $n$  是根据所述数据包的大小而动态调整的。

16. 根据权利要求 15 所述的方法,其中,n 被选择为使得所述编码数据段的大小不超过预定的最大传输单元大小。

17. 一种在将数据包从第一端点输送至第二端点的不可靠网络上加速数据通信的方法,所述方法包括:

在第一编码组件与第二编码组件之间建立用于在所述两个端点之间进行通信会话的编码通道,该步骤是通过下列步骤完成的:

在所述第一编码组件处检测从所述两个端点中的所述第一端点发起并且发往作为所述两个端点之另一端点的所述第二端点的第一个所述消息数据包,

通过向被检测到的所述第一个数据包添加标记,创建发往所述另一端点的被标记的消息,所述被标记的消息具有表示所述第一端点能够根据对于两个编码组件都是已知的协议对所述数据包进行分段和编码的标记,其中所述标记选自多个可用的标记,

将所述被标记的消息数据包转发至所述另一端点,

接收在预选的时间段内是否从所述第二编码组件接收到对所述被标记的消息数据包的应答的判断,并且如果在所述时间段内未接收到应答,则在所述第一编码组件处检测从所述第一端点发起的并且发往所述第二端点的第二个所述数据包,向检测到的所述第二个所述数据包添加选自所述多个可用的标记的不同的标记,并且将所述被不同标记的数据包转发至所述另一端点,并且

响应于在所述时间段内接收到来自所述第二编码组件的应答,交换控制消息以在所述编码组件之间建立通信通道;

在所述第一编码组件和所述第二编码组件之一处拦截与所述通信相关的数据包;

将所述数据包分段并且打包,从而提供用于传输至所述第一编码组件和所述第二编码组件中的另一个的编码数据段;

将所述编码数据段和至少一个额外编码段经由所述编码通道传输至所述第一编码组件和所述第二编码组件中的另一个;

在所述第一编码组件和所述第二编码组件中的另一个处,基于接收到的所述编码数据段将所述数据包解码并重组,以补偿在所述不可靠网络上发生的所述数据段的丢失;并且

将已重组的数据包从所述第一编码组件和所述第二编码组件中的另一个发送至其相应的端点。

18. 根据权利要求 16 所述的方法,其中,建立所述编码通道还包括存储唯一识别所述通信会话的信息。

19. 一种在不可靠网络上加速数据通信的方法,所述方法包括:

在第一编码组件与第二编码组件之间建立用于在两个端点之间进行通信会话的编码通道,该步骤通过下列步骤完成:

在所述第一编码组件处检测从所述两个端点中的第一端点发起并且发往所述两个端点中的另一端点的消息,

创建发往所述另一端点的被标记的消息,所述被标记的消息具有表示所述第一端点能够根据对于两个编码组件都是已知的协议对所述数据包进行分段和编码的标记,其中所述标记选自多个可用的标记,

将所述被标记的消息转发至所述另一端点,

接收在预选的时间段内是否从所述第二编码组件接收到对所述被标记的消息的应答的判断,并且如果在所述时间段内未接收到应答,则创建具有选自所述多个可用标记的不同标记的新被标记的消息,并且将所述新被标记的消息转发至所述另一端点,并且

响应于在所述时间段内接收到来自所述第二编码组件的应答,交换控制消息以建立所述编码组件之间的通信通道;

在所述第一编码组件与所述第二编码组件之一处拦截与所述通信相关的数据包;

将所述数据包分段和打包,从而提供用于向所述第一编码组件与所述第二编码组件中的另一个传输的编码数据段;

将所述编码数据段和至少一个额外编码段经由所述编码通道传输至所述第一编码组件与所述第二编码组件中的另一个;

在所述第一编码组件与所述第二编码组件中的另一个处,基于接收到的所述编码数据段对所述数据包进行解码和重组;

将已重组的数据包从所述第一编码组件与所述第二编码组件中的另一个传输至其相应的端点;

如果在预定的时间内没有接收到应答,则通过从所述多个可用的标记中再选择一个标记以再创建被标记的消息;并且

如果对于所述多个可用的标记中的任何一个都没收到应答,则告知网络管理者所述第二端点不能与所述第一端点进行加速通信。

20. 根据权利要求 16 所述的方法,还包括存储在后续的通信会话中使用的所述标记的标识。

21. 根据权利要求 16 所述的方法,还包括如果在预定的超时时段内没有传输编码数据包,则进入拆除状态。

22. 一种在不可靠网络上加速数据通信的方法,所述方法包括:

在第一编码组件与第二编码组件之间建立用于在两个端点之间进行通信会话的编码通道,该步骤由下列步骤完成:

在所述第一编码组件处检测从所述两个端点中的第一端点发起并且发往所述两个端点中的另一端点的消息,

创建发往所述另一端点的被标记的消息,所述被标记的消息具有表示所述第一端点能够根据对于两个编码组件都是已知的协议对所述数据包进行分段和编码的标记,其中所述标记选自多个可用的标记,

将所述被标记的消息转发至所述另一端点,

接收在预选的时间段内是否从所述第二编码组件接收到对所述被标记的消息的应答的判断,并且如果在所述时间段内未接收到应答,创建具有选自所述多个可用标记的不同标记的新被标记的消息,并且将所述新被标记的消息转发至所述另一端点,并且

响应于在所述时间段内接收到来自所述第二编码组件的应答,交换控制消息以建立所述编码组件之间的通信通道;

在所述第一编码组件与所述第二编码组件之一处拦截与所述通信相关的数据包;

将所述数据包分段和打包,从而提供用于向所述第一编码组件与所述第二编码组件中的另一个传输的编码数据段;

将所述编码数据段和至少一个额外编码段经由所述编码通道传输至所述第一编码组件与所述第二编码组件中的另一个；

在所述第一编码组件与所述第二编码组件中的另一个处，基于接收到的所述编码数据段对所述数据包进行解码和重组；

将已重组的数据包从所述第一编码组件与所述第二编码组件中的另一个传输至其相应的端点；

如果在预定的超时时段内没有传输编码数据包，则进入拆除状态；并且

在所述拆除状态期间，新收到的数据包不被分段和打包。

23. 根据权利要求 16 所述的方法，其中，所述第一编码组件和所述第二编码组件相对于其他编码组件具有优先权。

24. 一种在不可靠网络上加速数据通信的方法，所述方法包括：

在第一编码组件与第二编码组件之间建立用于在两个端点之间进行通信会话的编码通道，该步骤由下列步骤完成：

在所述第一编码组件处检测从所述两个端点中的第一端点发起并且发往所述两个端点中的另一个端点的消息，

创建发往所述另一端点的被标记的消息，所述被标记的消息具有表示所述第一端点能够根据对于两个编码组件都是已知的协议对所述数据包进行分段和编码的标记，其中所述标记选自多个可用的标记，

将所述被标记的消息转发至所述另一端点，

接收在预选的时间段内是否从所述第二编码组件接收到对所述被标记的消息的应答的判断，并且如果在所述时间段内未接收到应答，创建具有选自所述多个可用标记的不同标记的新被标记的消息，并且将所述新被标记的消息转发至所述另一端点，并且

响应于在所述时间段内接收到来自所述第二编码组件的应答，交换控制消息以建立所述编码组件之间的通信通道；

在所述第一编码组件与所述第二编码组件之一处拦截与所述通信相关的数据包；

将所述数据包分段和打包，从而提供用于向所述第一编码组件与所述第二编码组件中的另一个传输的编码数据段；

将所述编码数据段和至少一个额外编码段经由所述编码通道传输至所述第一编码组件与所述第二编码组件中的另一个；

在所述第一编码组件与所述第二编码组件中的另一个处，基于接收到的所述编码数据段对所述数据包进行解码和重组；

将已重组的数据包从所述第一编码组件与所述第二编码组件中的另一个传输至其相应的端点；

其中，所述第一编码组件和所述第二编码组件相对其它编码组件具有优先权；

在所述第一编码组件处接收来自除了所述第一编码组件和所述第二编码组件之外的编码组件的被标记的数据包；并且

优先权是这样主张的：

从所述被标记的数据包中去除标记；并且

在所述第一编码组件和所述第二编码组件之间建立所述编码通道。

25. 一种在不可靠网络上加速数据通信的方法,所述方法包括:

在第一编码组件与第二编码组件之间建立用于在两个端点之间进行通信会话的编码通道,该步骤由下列步骤完成:

在所述第一编码组件处检测从所述两个端点中的第一端点发起并且发往所述两个端点中的另一个端点的消息,

创建发往所述另一端点的被标记的消息,所述被标记的消息具有表示所述第一端点能够根据对于两个编码组件都是已知的协议对所述数据包进行分段和编码的标记,其中所述标记选自多个可用的标记,

将所述被标记的消息转发至所述另一端点,

接收在预选的时间段内是否从所述第二编码组件接收到对所述被标记的消息的应答的判断,并且如果在所述时间段内未接收到应答,则创建具有选自所述多个可用标记的不同标记的新被标记的消息,并且将所述新被标记的消息转发至所述另一端点,并且

响应于在所述时间段内接收到来自所述第二编码组件的应答,交换控制消息以建立所述编码组件之间的通信通道;

在所述第一编码组件与所述第二编码组件之一处拦截与所述通信相关的数据包;

将所述数据包分段和打包,从而提供用于向所述第一编码组件与所述第二编码组件中的另一个传输的编码数据段;

将所述编码数据段和至少一个额外编码段经由所述编码通道传输至所述第一编码组件与所述第二编码组件中的另一个;

在所述第一编码组件与所述第二编码组件中的另一个处,基于接收到的所述编码数据段对所述数据包进行解码和重组;

将已重组的数据包从所述第一编码组件与所述第二编码组件中的另一个传输至其相应的端点;

其中,所述第一编码组件和所述第二编码组件配置为忽略从指定的编码组件所接收到的被标记的数据包,并且透明地传输所述被标记的数据包。

26. 根据权利要求 21 所述的方法,其中,所述编码组件在非激活时段还发送保持存活的消息。

27. 根据权利要求 16 所述的方法,其中,所述端点之一被指定为主端点,并且所述方法还包括所述主端点判定是否指示拆除所述编码通道。

28. 根据权利要求 27 所述的方法,其中,如果在预定时段内没有接收到新的数据包,则所述主端点判定指示拆除。

29. 一种用于对数据进行编码而设定编码率的方法,该方法用于在分别与编码组件相关联的两个端点之间的不可靠网络上加速数据通信,所述方法包括:

在所述编码组件之间建立用于在所述端点之间进行通信会话的编码通道;

在一个编码组件处拦截与所述通信相关的数据包;

对所述数据包进行分段和打包,从而提供向另一个编码组件传输的编码数据段;

将所述编码数据段和至少一个额外数据段经由所述编码通道传输至所述另一个编码组件;

在所述另一个编码组件处,基于接收到的所述编码数据段对所述数据包进行解码和重

组；

将已重组的数据包从所述另一个编码组件传输至其相应的端点；

计算所述通信会话的加权丢失率；并且

基于所述加权丢失率调整所述编码率。

30. 根据权利要求 29 所述的方法，其中，计算所述加权率包括根据之前的通信会话的平均丢失率计算所述加权率。

31. 根据权利要求 29 所述的方法，其中，计算所述加权率包括根据重新发送请求计算所述加权率。

32. 根据权利要求 29 所述的方法，还包括根据所述加权丢失率计算目前丢失等级。

33. 根据权利要求 32 所述的方法，其中，将所述加权丢失率归一化。

34. 根据权利要求 32 所述的方法，其中，所述目前丢失等级被传送至所述另一个编码组件。

35. 根据权利要求 34 所述的方法，其中，在一个所述编码数据段的报头部分中传送所述目前丢失等级。

36. 根据权利要求 35 所述的方法，其中，所述报头部分还识别期望的保护等级。

37. 根据权利要求 36 所述的方法，其中，所述期望的保护等级是依照丢失指标确定的，所述丢失指标是根据网络策略配置的。

38. 根据权利要求 34 所述的方法，其中，通过控制通道传送所述目前丢失等级。

## 用于实现加速的吞吐量的系统和方法

[0001] 相关申请的交叉参考

[0002] 本申请是 2008 年 8 月 18 日提交的美国专利申请 12/193,345 号、即 2010 年 6 月 22 日公布的当前美国专利 7,742,501 号的部分继续申请案,12/193,345 是 2004 年 8 月 6 日提交的美国专利申请 10/912,200 的部分继续申请案,因此将该美国专利的全部内容以引用的方式并入本文。

### 技术领域

[0003] 本发明涉及通信数据网络。更加具体地,本发明涉及用于通过传送网络提高数据传输的吞吐量的系统和方法,就像从传送网络的边缘看到的。

### 背景技术

[0004] 大多数(如果不是全部)传送网络是基于数据包的网络,其将数据流分为更小的数据包,并随后将数据包从第一源网络或端点经由第二传送网络发送至第三目标网络或端点。然而,由于拥塞和别的网络限制,不是所有的包都能够成功到达目标网络。所以传送网络的性能对于源网络和终端目标网络而言是重要的。从位于终端网络的应用的角度来看,理想的是传送网络必须完美地不会丢失数据包。然而,优选的是能够比租用高性能传送网络的通常成本以更低的价钱获得这样的性能。

[0005] 因此,需要能够使用低成本通信传送网络给终端网络应用提供高性能的传送网络的系统和方法。

[0006] 已经尝试了一些方法来解决上述问题。在一种方法中,在端点处安装自定义协议栈(custom protocol stack)来改善对于丢失和延迟的响应。然而,该方法需要两个终端网络根据相同的自定义协议进行通信,这通常需要大量的重新编程。

[0007] 另一种方法使用代表远端元件拦截标准协议并且发送协议响应的网元(network element)。随后在拦截网元之间使用自定义协议。该方法受到 TCP/IP 应用的限制并增加了复杂度,尤其是在解决网络问题方面。

### 发明内容

[0008] 在第一方面,提供了一种在不可靠网络上加速数据通信的方法。所述方法包括如下步骤:提供与两个端点中的每一个相关联的编码组件以进行通信;在所述编码组件之间建立用于在所述端点之间进行通信会话的编码通道;在所述编码组件之一处拦截与所述通信相关的多个数据包;将所述多个数据包中的每一个分段并且打包,从而提供用于传输至另一编码组件的具有至少一个编码数据段和至少一个额外编码段的已分段和打包的数据包;在所述编码组件之一处存储所述已分段和打包的数据包;当满足预定的条件时,将所述已分段和打包的数据包经由所述编码通道传输至另一编码组件;在所述另一编码组件处,基于接收到的所述编码数据段和接收到的所述额外编码段将所述多个数据包解码并重组;并且将已重组的数据包从所述另一编码组件发送至其相应的端点。

[0009] 所述预定的条件可以是定时器到期,例如,当所述多个数据包中的相应的一个被拦截时,所述定时器可被激活。所述预定的条件还可以是拦截到含有实时敏感数据的数据包,或者是存储了预定数量的已分段和打包的数据包。

[0010] 所述多个数据包的所述编码数据段和所述至少一个额外编码段可以保留它们的顺序的方式被交织发送。例如,所述已分段和打包的数据包可以根据它们各自的大小被依次存储,并且可以依次从各所述已分段和打包的数据包发送各个数据段。或者,所述多个数据包中的每一个的所述编码数据段和所述至少一个额外数据段以随机顺序发送。

[0011] 所述方法还可以包括如果在预定的时段内在所述另一编码组件处未接收到一个编码数据段,则判定所述一个编码数据段丢失,所述预定的时段例如是当接收到关于给定的数据包的第一个所述编码数据段时设定的计时器超时的时候,或者参考环形缓冲。在后一种情况下,所述环形缓冲能够跟踪丢失的编码数据段的序列号,并且当所述环形缓冲循环返回给定的编码数据段的序列号时,判定所述给定的编码数据段丢失。所述方法还可以包括当判定来自给定的所述已分段打包的数据包的预定数量的编码数据段已经丢失时,将来自所述多个数据包的编码数据段交织。

[0012] 将数据包分段并打包可以包括将所述数据包分段并打包为  $n$  个编码数据段,其中  $n > 1$ 。在此情况下,可以根据所述数据包的大小对  $n$  进行动态调整;例如, $n$  可以被选择为使得所述编码数据段的大小不超过预定的最大传输单元大小。

[0013] 在另一方面,提供了一种在不可靠网络上加速数据通信的方法。所述方法包括如下步骤:在第一编码组件与第二编码组件之间建立用于在两个端点之间进行通信会话的编码通道,该步骤通过下列步骤完成:在所述第一编码组件处检测从所述两个端点中的第一端点发起并且发往所述两个端点中的另一端点的消息,创建发往所述另一端点的被标记的消息,所述被标记的消息具有表示所述第一端点能够根据对于两个编码组件都是已知的协议对所述数据包进行分段和编码的标记,其中所述标记选自多个可用的标记,将所述被标记的消息转发至所述另一端点,从所述第二编码组件接收对所述被标记的消息的应答,并且交换控制消息以建立所述编码组件之间的通信通道;在所述第一编码组件与所述第二编码组件之一处拦截与所述通信相关的数据包;将所述数据包分段和打包,从而提供用于向所述第一编码组件与所述第二编码组件中的另一个传输的编码数据段;将所述编码数据段和至少一个额外编码段经由所述编码通道传输至所述第一编码组件与所述第二编码组件中的另一个;在所述第一编码组件与所述第二编码组件中的另一个处,基于接收到的所述编码数据段对所述数据包进行解码和重组;将已重组的数据包从所述第一编码组件与所述第二编码组件中的另一个传输至其相应的端点。

[0014] 根据本方面,建立所述编码通道还包括存储唯一识别所述通信会话的信息。能够存储在所述编码组件之间的后续通信会话中使用的所述标记的标识。如果在预定的时间内没有接收到应答,能够通过从所述多个可用的标记中再选择一个标记以再创建被标记的消息。如果对于所述多个可用的标记中的任一个都没收到应答,则告知网络管理者所述第二端点不能与所述第一端点进行加速通信。

[0015] 所述方法还包括如果在预定的超时时段内没有发送编码数据包,则进入拆除状态。在所述拆除状态期间,新收到的数据包不被分段打包。所述编码组件还能够非激活期间发送保持存活消息。还能够将所述端点之一指定为主端点,并且所述方法还包括所述

主端点判定是否指示拆除所述编码通道。例如,如果在预定时段内没有接收到新的数据包,则所述主端点能够判定指示拆除。

[0016] 所述第一编码组件和所述第二编码组件相对于其他编码组件具有优先权。如果它们具有优先权,则所述方法还包括在所述第一编码组件处接收来自除了所述第一编码组件和所述第二编码组件之外的编码组件的被标记的数据包;优先权是这样主张的:从所述被标记的数据包中去除标记;并且在所述第一编码组件和所述第二编码组件之间建立所述编码通道。所述第一编码组件和所述第二编码组件还能够被配置为忽略从指定的编码组件接收到的被标记的数据包,并且透明地发送所述被标记的数据包。

[0017] 在另一方面,提供了一种用于对数据进行编码而设定编码率的方法,该方法用于在分别与编码组件相关联的两个端点之间的不可靠网络上加速数据通信,所述方法包括以下步骤:在所述编码组件之间建立用于在所述端点之间进行通信会话的编码通道;在一个所述编码组件处拦截与所述通信相关的数据包;对所述数据包进行分段和打包,从而提供向另一个编码组件传输的编码数据段;将所述编码数据段和至少一个额外数据段经由所述编码通道传输至所述另一个编码组件;在所述另一个编码组件处,基于接收到的所述编码数据段对所述数据包进行解码和重组;将已重组的数据包从所述另一个编码组件传输至其相应的端点;计算所述通信会话的加权丢失率;并且基于所述加权丢失率调整所述编码率。计算所述加权率例如包括根据之前的通信会话的平均丢失率或者根据再传输请求计算所述加权率。

[0018] 所述方法还可以包括根据所述加权丢失率计算目前丢失等级。所述加权丢失率可以被归一化。所述目前丢失等级能够通过控制通道、或者在一个所述编码数据段的报头部分中传递。所述报头部分还能够识别期望的保护等级,能够依照丢失指标确定所述期望的保护等级,而所述丢失指标是根据网络策略配置的。

[0019] 下面结合附图,在阅读了对本发明的具体实施例的说明的基础上,本发明的其它方面和特征对于本领域普通技术人员而言将变得显而易见。

## 附图说明

[0020] 现在将参照附图仅以举例的方式说明本发明的实施例,其中:

[0021] 图 1 是可以实施本发明的环境的框图。

[0022] 图 2 是图示了在图 1 中使用的服务器的组件的框图。

[0023] 图 3 是图示了图 1 中使用的另一服务器的组件的框图。

[0024] 图 4 示出了本发明实施例的触发模式的示例。

[0025] 图 5 示出了本发明实施例的多级触发模式的示例。

[0026] 图 6 是图示了当收到第一数据单元并转换为第二数据单元以向传送网络传输时执行的步骤的流程图。

[0027] 图 7 是图示了当从传送网络接收到第二数据单元并且转换为第一数据单元以向端网络传输时执行的步骤的流程图。

[0028] 图 8 图示了本发明实施例的示例性系统和连接协商协议。

[0029] 图 9 示出了本发明实施例的具有多个编码组件的网络示例。

[0030] 图 10 图示了本发明的数据包的一般的分段和编码。

- [0031] 图 11 示出了本发明实施例的数据包的示例性分段和编码。
- [0032] 图 12 示出了本发明实施例的用来评价丢失的第一示例算法。
- [0033] 图 13 示出了本发明实施例的考虑了平均丢失的示例。

### 具体实施方式

[0034] 参照图 1, 图示了可以实施本发明的环境的框图。第一端点 10 通过网络 20 与第二端点 30 通信。第一端点 10 和网络 20 通过编码 / 解码组件 40 通信, 同时网络 20 与第二端点 30 通过编码 / 解码组件 50 通信。编码 / 解码组件 40 和编码 / 解码组件 50 (下文中称为编码组件) 可位于端点处, 也可位于诸如服务器等中间设备中。第一端点 10 和第二端点 30 可以是诸如通过调制解调器、局域或广域网络连接的计算机或者是可以访问网络 20 的其它设备或系统等终端。网络 20 可以是例如因特网或其它通信网络。

[0035] 本文中所使用的数据单元是被用来在网络上以数字的形式传输数据的任何数据的单元。这样的数据单元可以是数据包、数据胞、数据帧或任何其它数据被封装在单元内或者作为有效载荷(payload)而被传送的单元。术语“数据单元”适用于实现特定的协议、标准或传输方案的任何和所有的数据包和数据帧。一般地, 将携带有路由、控制和身份识别信息的数据头添加到所述数据有效载荷。

[0036] 本系统提供了用于将端点与变化的网络 20 隔离开的手段。编码组件 40 和编码组件 50 设置有通过处理数据单元的编码和解码以及数据单元通过网络 20 的传输而将端点 10 和端点 30 与网络 20 隔离开的装置。端点 10 和端点 30 能够使用任意的协议通信, 编码组件 40 和编码组件 50 也是如此。编码组件 40 和编码组件 50 接收来自端点 10 和端点 30 的一个协议中的原始数据单元, 并且通过如下步骤对原始数据单元进行编码, 即将原始数据单元细分为优选地更小的数据单元, 创建额外的数据单元以在通过传送网络所传输的一些数据单元丢失的情况下能够用来重建或重构原始数据单元, 并且通过传送网络传输细分的数据单元和额外数据单元之前根据当前的协议将它们再打包。

[0037] 一旦在与远端点相关联的编码组件处接收到通过传送网络 20 传输的数据单元, 来自发送端点的原始数据单元就被重建或重构。通过将收到的数据单元重新排序, 如果必要且如果在传输过程中丢失了一些数据单元, 利用额外数据单元来重建丢失的数据单元, 则可以完成上述重建或重构。在接收到的额外数据单元不足以重建原始数据单元的情况下, 接收编码组件能够可选地要求再次传输之前发送的数据单元。

[0038] 为了易于说明, 将源自端点或在端点处接收的数据单元称为第一数据单元或数据包, 并且作为第一类型的数据单元。将通过传送网络传输和 / 或从传送网络接收的数据单元称为第二数据单元或数据段, 并且作为第二类型的数据单元。第二数据单元也被称为编码数据单元。

[0039] 图 2 示出了在编码组件 40 内部实现本发明所必须的模块的示例性实施例以及它们各自的数据流。所述模块可能位于单个设备中, 或者可能分布于多个设备中。如图中所见, 编码组件 40 具有第一接口 60、第二接口 70、重组模块 80 和分段模块 90。第一接口 60 将第一数据单元发送至端点 10 并且从端点 10 接收第一数据单元。服务器 40 的第二接口 70 将第二数据单元发送至传送网络 20 并且从传送网络 20 接收第二数据单元。

[0040] 重组模块 80 接收来自第二接口 70 的第二数据单元并且生成用于经由第一接口 60

向第一网络 10 传输的第一数据单元。另一方面,分段模块 90 接收来自第一接口 60 的第一数据单元并且产生用于经由第二接口 70 向传送网络 20 传输的第二数据单元。

[0041] 一旦第二数据单元已经通过传送网络进行了传输,位于另一端点处的编码组件将接收到该第二数据单元。为了对此进行说明,图 3 示出了与传送网络和端点 30 通信的编码组件 50 的模块和数据流的示例性实施例。

[0042] 编码组件 50 中的模块与编码组件 40 中的模块是相同的,并且实际上具有相同的功能。编码组件 50 中的第二接口 70A 也与传送网络通信并且发送和接收第二数据单元。第一接口 60A 也与端点(在此例中为端点 30)通信并且发送和接收第一数据单元。重组模块 80A 接收第二数据单元并且产生第一数据单元,同时分段模块 90A 接收第一数据单元并且产生第二数据单元。

[0043] 如上所述,第一数据单元是端网络使用的数据单元,而第二数据单元是传送网络和编码组件 40、50 彼此间相互传输数据时使用的数据单元。第二数据单元由第一数据单元获得。第一数据单元的有效载荷可以分为更小的单元,并且各更小的单元可以是第二数据单元中的有效载荷。这样,各第二数据单元可以比原始第一数据单元更小,其中第二数据单元由原始第一数据单元获得。例如,可以将 10kB 的第一数据单元细分为 5 个 2kB 单元。这些 2kB 单元可以是五个第二数据单元的有效载荷,这五个第二数据单元中的每一个可以小于 10kB。从第一数据单元生成第二数据单元的这种功能是通过分段模块 90 和分段模块 90A 完成的。

[0044] 为了协助重建原始第一数据单元(第二数据单元从所述原始第一数据单元获得),分段模块也生成额外第二数据单元。这些额外第二数据单元能够从第一数据单元和第二数据单元中获得。在一个或多个第二数据单元经由传送网络传输的过程中丢失的情况下,额外第二数据单元协助重组模块 80 和重组模块 80A 重建或重组原始第一数据单元。

[0045] 额外第二数据单元能够采取多种形式。最简单的实施例或许是,额外第二数据单元仅仅是之前发送的选中的第二数据单元的拷贝。例如,如果第一数据单元被分割或者分段为四个第二数据单元(例如 DU1、DU2、DU3、DU4),那么额外第二数据单元可能是 DU2 和 DU3 的拷贝。这样,如果 DU2 和 DU3 在传输过程中丢失,那么仍然能够重建原始第一数据单元。被复制的第二数据单元的数量和身份识别信息可以是预定的,或者留给系统管理员来处理。根据网络 20 中实际的丢失程度或预料的丢失程度,能够通过包含更多的复制的第二数据单元在系统中建立更多的冗余,或者反之通过包含更少的复制的第二数据单元在系统中建立更少的冗余。在最简单的情况下,能够复制所有的第二数据单元以在根本上确保各第二数据单元的两份拷贝被发送到目标编码组件。

[0046] 还能够将奇偶校验位(Parity)数据单元用作第二额外数据单元。本领域众所周知的是,能够利用 XOR 运算来形成奇偶校验位数据单元。能够对由原始第一数据单元生成的不同的第二数据单元的各个位进行 XOR 运算,从而得到能够被存储于额外第二数据单元中的位值。如果任一个第二数据单元(不包括额外第二数据单元)在传输过程中丢失,能够使用另一个接收到的第二数据单元和额外第二数据单元来重建丢失的第二数据单元。对接收到的第二数据单元和额外第二数据单元进行 XOR 运算将重建丢失的第二数据单元。

[0047] 应当注意的是,可以使用其它的纠删编码(eraser correcting code)来对额外第二数据单元进行编码。例如,如果为单个第一数据单元生成 n 个第二数据单元,

可以生成  $m$  个额外第二数据单元,从而使得丢失的第二数据单元能够被重建。如上所述,可以将  $m$  个额外第二数据单元视为“冗余”的第二数据单元,并且如果使用的是单纯的复制,则  $m \leq n$ , 并且  $m=n$  时得到完全的复制。然而,如果使用纠删编码,通过  $m=2$  就能够以如下的方式对冗余信息进行编码,即能够丢失任意两个第二数据单元而重组模块仍能够重构丢失的第二数据单元。可以使用诸如 Reed-Solomon 码、前向纠删技术和 Bose-Chaudhuri-Hochquenghem (BCH) 码等公知的方法和编码技术以及许多其它的技术。

[0048] 虽然额外第二数据单元可以有助于应对丢失一些第二数据单元的效果,但是丢失太多第二数据单元不能完全被补偿。这样,丢失超过门限值的大量第二数据单元可能选择性地导致重组模块要求重新传输一包第二数据单元或一组第二数据单元。例如,如果额外第二数据单元能够恢复数据单元 25% 的丢失,并且存在着由单个第一数据单元生成的四个第二数据单元,那么单个第二数据单元的丢失将不会触发再传输请求。然而,在丢失两个第二数据单元(即,50% 丢失)的情况下,重组模块会请求再传输。如果再传输被启用,那么理想的是,再传输门限与额外或冗余第二数据单元使用的编码的纠错或纠删能力相关。因为重组模块将需要正确地对第二数据单元的有效载荷进行排序,所以重组模块会跟踪记录为各个已被分段的第一数据单元接收的第二数据单元的数量。

[0049] 如上所述,重组模块 80 和重组模块 80A 对接收到的第二数据单元进行解码和重组从而形成原始第一数据单元。对接收到的第二数据单元和额外第二数据单元进行跟踪,以确定是否已经接收到了足够的数量来重建原始第一数据单元。如果还没有接收到足够的数量,能够选择性地请求再传输。如果已经丢失了一些第二数据单元,那么重组模块能够重建或重构丢失的第二数据单元。如上所述,该过程取决于使用的编码和采用的整体策略。这样的解码和纠错处理对于本领域技术人员而言是公知的。

[0050] 一旦接受到了所需数量的第二数据单元,就取出并使用它们的有效载荷来重构原始第一数据单元,其中第二数据单元是从所述原始第一数据单元获得的。这可以像将第二数据单元的有效载荷连结起来以获得重构的第一数据单元那样简单。然而,如上文中关于解码的部分所述,重构过程将取决于用来分段或分割原始第一数据单元的过程。一旦原始第一数据单元被重构,它就能够在被发送到与接收端点通信的适当接口。

[0051] 关于分段模块,这些模块进行将第一数据单元分段或分割并且将各个数据段“再打包(repackaging)”为第二数据单元的任务。分段模块如上所述也对额外第二数据单元进行编码。从第一数据单元和额外第二数据单元得到的第二数据单元随后被传递至与传送网络通信的接口模块。为了使可选的第二数据单元的再传输变得更容易,分段模块还能够对第二数据单元进行缓存。例如,如果 5 个第一数据单元被分段为 20 个第二数据单元和 5 个额外第二数据单元,分段模块能够对与被编码的最后三个第一数据单元相对应的最后三组第二数据单元进行缓存。因此,分段模块将会缓存 12 个第二数据单元和 3 个额外第二数据单元。

[0052] 分段模块还能够被设置为以交织的方式来传输第二数据单元,从而将丢失多个第二数据单元的风险分散到不同的被分段的第一数据单元。因此,来自不同的第一数据单元的第二数据单元能够相互交织,而不是依次发送多组第二数据单元从而使得各组第二数据单元对应于单个第一数据单元。图 4 图示了交织模式的示例。在此示例中分别地,假设第一数据单元 A 被分段为第二数据单元 A1、A2,第一数据单元 B 被分段为第二数据单元 B1、B2、

B3, 第一数据单元 C 被分段为第二数据单元 C1、C2、C3 和 C4。第二数据单元 A2、第二数据单元 B3 和第二数据单元 C4 是额外第二数据单元。这些第二数据单元能够通过多组 k 个第一数据单元将它们交织, 而不是发送按照它们各自的第一数据单元编组的第二数据单元。在图 4 的示例中, k=3。编码器将对应于 k 个第一数据单元的第二数据单元存储到缓存 1303 中。或者, 编码器存储在预定的时段内或者直到具有时间敏感信息的数据包被检测到为止 (例如, 通过深度数据包检测) 所到达的多个数据包。为了避免重新排序, 第二数据单元组左端对齐, 使得各第一数据单元的最后第二个第二数据单元处于尾部并且最后被发送, 而各第一数据单元的第二个第二数据单元处于头部并且将被首先传输。编码组件在物理接口 1304 上按照从上到下的顺序或者按照将使发送来自同一第一数据单元的两个连续的第二数据单元最少化的任意顺序发送位于头部的数据单元。在图 4 的示例中, 第二数据单元可能按照下列顺序 C1、B1、C2、A1、B2、C3、A2、B3、C4 发送。最后发送所有的额外第二数据单元能够使延迟最小化。交织也可以是随机进行的。将要进行交织的数据单元组可以包括从接口传输的所有第二数据单元, 或者交织的组可以包括相同应用 (例如, 视频会议的同一频道)、相同的目的地或预设分组的第二数据单元。在图 5 中, 通过将已交织的第二数据单元的不同交织组 1401、1402、1403 的第二数据单元进行交织, 能够获得另一层级的交织。通过这样的交织, 能够将数据单元的任何较大的丢失的影响最小化, 并且根据所采用的编码和策略, 这种类型的丢失是可恢复的。

[0053] 图 6 中示出了在编码组件 40 和编码组件 50 之一中实施的根据本发明的示例性方法。处理是以从源端接收第一数据单元的步骤 100 开始的。在被接收之后, 第一数据单元随后被分割或分段 (步骤 110), 并且数据段被打包为第二数据单元 (步骤 120)。一旦生成了第二数据单元, 就编码和创建额外或冗余第二数据单元 (步骤 130)。然后, 第二数据单元能够被可选地缓存 (步骤 140) 并且发送至传送网络 (步骤 150)。该方法随后经由连接器 A 返回至步骤 100。如果再传输被启用, 能够异步进行对于再传输请求的可选检查 (步骤 160)。如果收到了这样的请求, 那么决策流程返回至步骤 150, 并且能够发送之前在可选步骤 140 中缓存的被请求的第二数据单元。

[0054] 图 7 示出了接收来自传送网络的第二数据单元的编码组件所执行的步骤的示例性实施例。处理在步骤 180 处开始, 在此步骤中, 服务器接收到来自传输网络的第二数据单元。判定步骤 190 随后判定是否已经收到了用来重构第一数据单元的所有第二数据单元, 其中第二数据单元是从第一数据单元获得的。如果接收第二数据单元的编码组件在预定量的时间之后未接收到第二数据单元和 / 或如果接收到了具有更高序列号的预定量的第二数据单元, 接收第二数据单元的编码组件就判定第二数据单元丢失。另一种判定第二数据单元是否丢失的技术是在等待第二数据单元到达的同时利用环形缓冲来跟踪丢失的第二数据单元的序号。当缓冲周期返回仍没有被接收到的第二数据单元时, 那么就认为该第二数据单元丢失了。如果已经接收到了所有的第二数据单元, 那么在步骤 200 中重建原始第一数据单元。一旦第一数据单元已经被重建, 重建后的数据单元被发送至目标端点 (步骤 205), 并且控制流通过连接器 D 返回步骤 180。

[0055] 返回判定步骤 190, 如果没有收到所有的第二数据单元, 那么进行判定以确定是否已经收到额外第二数据单元 (步骤 220)。如果没有收到额外第二数据单元, 能够可选地请求再传输 (步骤 230)。在可选的再传输请求之后, 控制流通过连接器 D 返回至步骤 180。如果

收到了额外第二数据单元,判定步骤 240 判定是否收到了足够用来重建原始第一数据单元的额外第二数据单元和第二数据单元。如果收到的数量不足,原始数据包可被丢弃,或者可选地,控制流能够如连接器 C 所示返回至步骤 230 请求再传输。如果收到了足够数量的第二数据单元和额外第二数据单元,那么能够使用额外第二数据单元来重建或重构丢失的第二数据单元(步骤 260)。随后,连接器 B 使控制流返回至步骤 200。

[0056] 一些具体的实施例将有助于理解本系统的操作和方法。图 8 示出了具有端点 A 和端点 B 的示例性系统。端点 A 通过编码组件 302 与诸如因特网的网络 300 通信。端点 B 通过编码组件 304 也与网络 300 通信。编码组件 302 和编码组件 304 包括参照图 2 和图 3 所述的接口、分段模块和重组模块,并且能够完全以软件的方式实现、或者能够被实现为预编程硬件单元、其它的相关软件、或者被实现为硬件部分与软件部分的结合。能够使用以太网或者其它适合的连接以将各端点连接至它们各自的编码组件。

[0057] 关于本示例的目的,假设在端点 A 和端点 B 之间存在双向通信,并且端点 A 发起通信。目前的通信协议对于端点 A 和端点 B 是不可见的;所有的功能位于编码组件 302 和编码组件 304 中。为了在端点 A 与端点 B 之间建立会话,并且为了判定两个端点是否都与编码组件相关联从而允许根据目前的协议进行通信,可以使用图 8 中所示的连接协商协议。端点 A 发送发往端点 B 的数据包 P。编码组件 302 检测到或注意到数据包 P 及其目的地(306),并且编码组件 302 存储唯一标识该通信会话的信息。编码组件 304 也检测到或注意到数据包 P (308)。编码组件 304 存储唯一标识与该数据包相关的通信会话的信息(310)并且将该数据包发送至端点 B。当端点 B 向端点 A 发送应答包 R 时,应答包 R 被编码组件 304 拦截,与之前存储的标识通信会话的信息进行匹配并被标记(312)。上述唯一标识通信会话的信息可以是例如源 IP 地址、目的 IP 地址、IP 数据包中包含的协议(例如 UDP 或 TCP)以及源端口和目的端口。对于以太网帧而言,还能够使用 VLAN ID。

[0058] 被标记的数据包 R<sup>n</sup> 被用来发送至编码组件 302,编码组件 304 能够接收和发送根据本发明被分段和编码的数据包。标记数据包可以例如由如下步骤组成:在数据包的 IP 报头中设定选项,和/或将 IP 标识字段设定为已知值。公共网络 300 中的路由器和其它装置可以在传输时保持 IP 标识字段不变。能够使用诸如在时间戳中设定 IP 地址标志等其它合适的标记或信号发送方案,只要它们优选地对于不能够依照目前协议进行通信的设备是无损的且无破坏性的。

[0059] 编码组件 304 随后将被标记的数据包 R<sup>n</sup> 经由公共网络 300 转发至端点 A。编码组件 302 检测到被标记的数据包 R<sup>n</sup>,更新之前存储的标识通信会话的信息(316),并且将数据包发送至端点 A。编码组件 302 能够在将数据包转发至端点 A 之前可选地去除标记。当端点 A 发送发往端点 B 的另一数据包 P<sub>2</sub> 时,编码组件 302 将数据包 P<sub>2</sub> 与之前存储的标识通信会话的信息进行匹配(318)。确认了端点 B 设置有能够依照目前协议通信的编码组件,编码组件 302 不仅发送数据包 P<sub>2</sub> 还发送“hello”消息。

[0060] 一旦从编码组件 302 收到“hello”消息,编码组件 304 用“hello”回复进行应答。编码组件 302 随后向编码组件 304 发送回复确认(“replyack”),并且开始采用目前协议对发往端点 B 的数据包进行分段和编码。当编码组件 304 从编码组件 302 接收到回复确认之后,编码组件 304 也开始对发往端点 A 的数据包进行分段和编码,以提供上文所述的第二数据单元。因此,编码组件 302 与编码组件 304 之间的编码通道被成功地自动检测和协商。该

编码通道能够被用来携带数据和控制信息。一旦在能够使用目前协议的两个编码组件之间已经协商了编码通道,就能够发送和接收被分段和 / 或被编码的信息。

[0061] 编码组件 304 可选地保持可能的不同类型标记的表。如果因为编码组件 302 未识别出标记,或者因为路径上的路由器或其它设备去除了所述标记,编码组件 302 未检测到已标记的数据包  $R^n$ ,则不会有“hello”回复被发送回编码组件 304。在一定的预定延迟之后,编码组件 304 将来自不同类型标记的表中的不同类型的标记对另一数据包  $R^m$  进行标记。编码组件能够连续改变标记,直到编码组件 302 发送“hello”消息以确认收到该标记为止。编码组件 304 能够跟踪哪个标记成功到达另一编码组件,并当该标记在将来试图再次到达那一编码组件的时候首先使用那个标记。在所有类型的标记都无法成功建立通道之后,编码组件 304 能够向网络管理系统或者接口标明给定的编码组件 302 无法理解该标记或者在网络中不存在编码组件 302。编码组件 304 或外部系统能够收集关于另一编码组件 302 多久会成为编码通道的一部分的统计数据,从而安排安装更多编码软件的先后顺序。

[0062] 编码通道保持激活,直到经过预定的超时时段为止,在超时时段内编码组件没有接收任何其它数据包。然后两侧拆除编码通道。由于在通信通道两端的计时器不是同步的,所以一个端点会在预定的超时时段后进入“拆除”状态。在“拆除”状态期间,继续对到来的数据包进行解码,但是不再对发出的数据包进行编码。“拆除”状态在预定的时段内是激活的,或者直到不再接收到编码的数据包为止是激活的。端点能够发送“保持存活”消息,使得在没有通信量的时候保持通道存活。在没有通信量发送的预定时段之后,保持存活可以停止,并且在预定的超时时段后拆除通道。或者,能够将端点中的一个指定为主端点(端点 A 或端点 B),并且能够通过发送拆除消息明确地拆除该通道。一旦收到拆除通知,非主端点停止对来自主端点的数据包进行解码,并且停止对发往主端点的数据包进行编码。也可以考虑在两个端点间协商编码通道的其它方法。例如,能够通过两个端点之间的已有控制通道来协商编码通道以进行通信。

[0063] 应当注意的是,端点和编码组件 302、304 之间的所有通信都发生于已建立的或现有的通信通道上。该通道能够实现在 IP 基础上的诸如通用数据报协议(Universal Datagram Protocol, UDP)和传输控制协议(Transmission Control Protocol, TCP)等已知的通信协议。如果连接是 UDP 连接,本发明的消息被直接嵌入 UDP 数据有效载荷中。

[0064] 如果通过已有的 TCP 连接发送数据,能够将序号已被修改的原始 TCP 报头附加至各段。因此,源端口和目的端口保持不变。第一段被赋予原始数据包的序号,而后续段具有新的序号。在目前优选的实施例,编码段或额外段重新使用与其相关的段之一的序号。这使得额外段能够以与重新发送的数据包相同的方式通过防火墙。数据偏移和校验和保持不变,但是标志或控制位除了原始段之外都改变。对于未分段信息(即,控制信息),剩余的消息字节紧跟着修改后的 TCP 报头。对于分段后的信息,段有效载荷跟着修改后的 TCP 报头,以例如提供诸如序列号等身份识别信息来识别数据段和与之相关的其它段,并且使得能够重组原始数据包。随后跟着该数据段的数据有效载荷。

[0065] 正如将要被理解的,如上所述被标记或被创建的数据包将顺利地通过防火墙和网络地址转换(Network Address Translation, NAT)转换器。事实上,编码通道建立编码组件 302 与编码组件 304 之间的控制通道,并且数据包能够从任一端通过,而不被诸如防火墙或 NAT 等任何中间组件识别或查验。

[0066] 现在参照图 9, 在路径中存在多个编码组件: 1001、1002、1003、1004。在本发明的一个实施例中, 仅在编码组件 1002 与编码组件 1003 之间的最内侧路径中进行编码。由于编码组件 1001 不知道其它具有优先权的编码组件的存在, 所以编码组件 1001 如上所述标记数据包以建立通道。被设置为具有优先权的编码组件 1002 去除上述信息, 并且向编码组件 1003 建立编码会话。编码组件 1003 被设置用来终止编码会话并且如上所述进行解码。编码通道将在内部编码组件 1002 与内部编码组件 1003 之间激活。编码组件能够设置为取得优先权, 使得网络的特定部分或路径总是被编码。

[0067] 编码组件 B 1002 和编码组件 C 1003 能够被设置为忽略从特定物理端口输入的标记并且透明地携带着该标记, 在此情况下, 编码存在于编码组件 A 1001 与编码组件 D 1004 之间的整个路径中。编码组件还能够了解网络中其它编码组件的拓扑结构, 并且基于网络策略, 忽略到达特定物理端口的标记或对该标记做出反应。

[0068] 现在将参照图 10 说明根据本发明实施例的数据包的分段和编码。为了简化讨论, 从原始数据包和分段数据包中省略报头。根据本发明, 在编码组件处接收原始数据有效载荷 400。原始数据有效载荷具有  $x$  个字节, 并且被分为  $n$  个段 402。 $n$  被选择为  $x$  的整数因数, 如图所示,  $n$  段中的每个将具有  $x/n$  字节。另外, 创建  $m$  个附加段。在图 10 中所示的示例中,  $m=1$ , 并且附加或额外数据包 404 是通过对  $n$  个段之一进行 XOR 函数而创建的奇偶校验段。编码组件能够考虑到数据包的大小, 并且当编码头加入第二数据单元时, 能够自动增大  $n$  以避免发送大小超过网络的最大传输单元 (Maximum Transfer Unit, MTU) 的第二数据单元。此功能在巨型帧 (jumbo frame) 进入对其无法进行处理的网络前切分该巨型帧也是有用的。

[0069] 将原始数据有效载荷 400 分为偶数段不总是可能的或期望的。例如, 在  $n$  不是  $x$  的因数的情况下可能需要填充最后一段以确保所有的段具有相等的长度。例如, 如果  $x=17$  且  $n=2$ , 则最接近  $x/n$  的整数值为 9。具有因数 4 或 8 的段大小通常是优选的, 因此在此情况下的段大小被选择为大于或等于  $x/n$  的下一个因数 4 (例如, 12 字节)。如图 11 所示, 原始的 17 字节数据有效载荷 410 被分为两个 12 字节的分段有效载荷 412、414。分段有效载荷 412 含有原始数据有效载荷 410 的前 12 个字节, 而分段有效载荷 414 含有来自原始数据有效载荷 410 的其余 5 个字节, 分段有效载荷 414 可选地被填充至 12 个字节。此外, 额外段 416 是段 412 的奇偶校验段, 其形成序列中的尾段。应当注意的是, 与单个原始数据包或帧相关的所有段均具有相同的系列号, 从而使得在接收端能够对它们进行识别和重组。

[0070] 编码率是由  $n$  和  $m$  确定的。能够将编码率设定为预定值 (例如, 如图 11 中所示的  $n=2$ ,  $m=1$ ), 或者能够对于在与给定的端点相关联的编码组件处终止的所有编码通道、基于观察到的和 / 或报告的网络性能调整该编码率。

[0071] 参照图 12。说明了关于编码组件 A 1106 与编码组件 B 1107 之间的编码通道 (更加具体地看在编码组件 A 至编码组件 B 1107 的方向上的通道) 的丢失等级的判定。在“无丢失”的状态下, 能够将编码设定为  $n=1$  且  $m=0$ , 以避免在不必要时使用网络中的额外带宽。编码组件 B 1107 的接收处理 1109 在长度为  $W$  个单位时间 (例如,  $W=8$  秒) 1100 的间隔时段内对接收到的数据包  $P_x$  的数量和丢失的数据包  $LP_x$  的数量进行计数。或者,  $LP_x$  可以代表重新传输请求的数量, 从而使得成功恢复的丢失不被算作丢失率的一部分。在步骤 1101 中利用例如如下的比率来计算丢失率  $L_x$  :

[0072]  $L_x = LP_x (P_x + LP_x)$  (1)

[0073] 为了避免对随机事件作出反应,能够计算出最后  $z$  个丢失测量的平均丢失 1102。为了考虑到最近的网络状态,能够根据下面的等式进行加权平均以对最近的测量进行加权:

[0074] 
$$WL_x = \frac{\sum_{i=(x-z) \dots x} (L_i \cdot w_i)}{\sum_{i=1 \dots z} w_i}$$
 (2)

[0075] 其中,  $WL_x$  代表对于间隔  $x$  的最近的丢失测量的加权平均。对于  $i < j \leq x$ , 权重  $w_i$  使得  $w_i < w_j$ 。

[0076] 为了避免跟踪很多丢失测量,能够使用之前的加权丢失来评价新的加权丢失:

[0077] 
$$WL_x = \frac{(WL_{x-1} \cdot w_{old}) + (L_x \cdot w_{new})}{w_{new} + w_{old}}$$
 (3)

[0078] 其中,  $w_{new}$  和  $w_{old}$  是通常被设定为使得  $w_{old} < w_{new}$  的权重。

[0079] 加权丢失率能够可选地被归一化 1103, 以仅使用整数简化查询。能够使用下面的等式计算归一化的丢失  $NL_x$ :

[0080] 
$$NL_x = \frac{N \cdot WL_x}{P_x}$$
 (4)

[0081] 其中,  $N$  是归一化因数(例如,  $N=10000$ )。

[0082] 随后  $NL_x$  被用来索引丢失等级表 1104, 以取得目前的丢失等级。下面示出了设置了 8 个丢失等级的丢失等级表的示例, 其中  $INT\_max$  代表最大整数或最大值(例如 10000)。

[0083]

$NL_x$ min	$NL_x$ max	目前丢失等级
0	0	0
1	75	1
76	145	2
146	185	3
186	210	4
211	260	5
261	370	6
371	$INT\_max$	7

[0084]

[0085] 编码组件 B 1107 的发送处理 1111 随后将目前的丢失等级 1105 嵌入为了该会话被发送至编码组件 A 1106 的各第二数据单元的报头中。编码组件 A 1106 中的接收处理 1112 从接收到的数据包中读取目前的丢失等级。随后用目前的丢失等级索引新的参数表以获得  $n$  和  $Max\_n$  的值。 $Max\_n$  代表给定目前的丢失等级的情况下  $n$  应该被设定的最大值。能够使用多个新的参数表, 来基于应用要求和可接收的开销实现不同的丢失指标。下面示

出了这样的使用 8 个丢失等级的新的参数表的示例。

[0086]

丢失等级	Max_n	m
0	Int_max	0
1	Int_max	1
2	Int_max	1
3	3	1
4	2	1
5	2	1
6	1	1
7	1	1

[0087] 其中, INT\_max 代表最大整数(无穷大)。在编码组件 A 1106 的发送处理 1108 对大小为  $s$  的数据包进行编码之前, 其以  $s$  索引预设的数据包大小表, 从而获得  $rec_n$ ,  $rec_n$  在给定数据包大小的情况下代表  $m$  的推荐值。下面示出了数据包大小表的示例:

[0088]

数据包长度 (字节)	Rec_n
< 88	1
89 至 264	2
265 至 528	3
>528	4

[0089]

[0090] 利用该表, 具有  $s < 88$  字节的数据包返回  $Rec_n = 1$ 。如果  $s > 528$  字节, 那么  $Rec_n = 4$ 。随后用来对数据包编码的  $n$  的值被确定为  $n = \min(Rec_n, Max_n)$ 。

[0091] 在本发明另一实施例中, 接收处理 1109 根据上面的等式 (2) 或等式 (3) 计算加权丢失率  $WL_x$ 。它能够可选地按照等式 (4) 进行归一化来计算  $NL_x$ 。准确的丢失值 ( $WL_x$  或  $NL_x$ ) 以固定的间隔 (例如每秒) 被发送至接受处理 1112, 进入到被插入到用于会话的编码通道中的控制消息中。接收处理 1112 从上述控制消息中提取准确的丢失值。发送编码组件 1108 使用该准确的丢失值来索引新的参数表以获得  $Max_n$  和  $m$ 。

[0092]

$NL_x$ min	$NL_x$ max	Max_n	m
0	0	Int_nax	0
1	75	Int_nax	1
76	145	Int_nax	1
146	185	3	1
186	210	2	1
211	260	2	1
261	370	1	1
371	10000	1	1

[0093] n 的值为  $n = \min(\text{Rec}_n, \text{Max}_n)$ 。能够设置多个新的参数表来反映不同的丢失指标,并且发送处理 1108 基于应用的丢失指标使用合适的表。发送实际的丢失率而不是发送目前的丢失等级,使得能够仅在编码侧配置参数表,因此简化了配置。

[0094] 图 13 中示出了本发明的另一实施例的示例。在该实施例中,编码组件估计在网络 1201 中丢失发生于何处,以确定是否改变编码参数,网络 1201 可能包括例如因特网、专用网络或虚拟专用网络的结合。编码组件 A 1202 与编码组件 D 1203 之间的编码会话 1204 被用来说明该实施例,更加具体地请看从 A 1202 到 D 1203 方向上的流程。

[0095] 如前所述,编码组件 D 1203 的接收处理计算它接收到的各编码流的丢失等级。在此例中,编码组件 D 1203 的接收处理计算分别从编码组件 A 1202、编码组件 B 1207 和编码组件 C 1208 发出的流 1204、1205 和 1206 的丢失等级。接收编码组件不仅返回所测得的流的丢失等级,还计算来自在间隔时段 x 内发送至该接收编码组件的 f 个流的加权丢失率的平均值  $AL_x$ 。

[0096] 间隔时段可以与用于计算单个流的丢失率的间隔时段相同。例如能够如下计算平均丢失率:

$$[0097] \quad AL_x = \frac{\sum_{i=1}^f WL_i}{f} \quad (5)$$

[0098] 能够按照上面的等式(2)或(3)计算  $WL_i$ 。随后能够依照上面的等式(4)将  $AL_x$  归一化,以创建归一化的平均丢失率  $NAL_x$ 。然后在丢失等级表中索引  $NAL_x$ ,从而获得远端的平均丢失等级( $ALFE_x$ )。该丢失等级表可以与上面的相同或者预设有不同的数量。将  $ALFE_x$  与上面计算出的各个流的目前丢失等级一起添加至由编码组件 D 1203 分别发送至编码组件 1202、编码组件 1207、编码组件 1208 的第二数据单元的数据包报头。

[0099] 如果数据包报头中包含汇总信息,那么发送编码组件 1202、发送编码组件 1207、发送编码组件 1208 能够使用该信息来决定是否改变其编码等级。编码组件 A 1202 的接收处理也计算近端处的平均丢失( $ALNE_x$ ),  $ALNE_x$  是在返回通道中从在测量时段 x 内激活的 g 个编码会话 1211、1212、1204 收到的目前丢失等级的平均值,

$$[0100] \quad ALNE_x = \frac{\sum_{i=1..g} CurrentLossLevel_i}{g} \quad (6)$$

[0101] 在图 12 的示例中,在编码组件 A 处计算的  $ALNE_x$  代表从编码组件 D 1203、编码组件 E 1209 和编码组件 F 1210 接收到的目前丢失等级的平均值。

[0102] 如果会话的目前丢失等级与  $ALFE_x$  之间的差别在预定的阈值以下,就依据上面的说明使用目前丢失等级来设定  $m$  和  $n$ 。在此情况下,由于目前丢失等级比计算出的  $ALFE_x$  好或者差一点,所以不大可能认为会话将上游网络拥塞。

[0103] 如果目前丢失等级与  $ALFE_x$  之间的差别等于或大于预定的阈值,并且目前丢失等级减去  $ALFE_x$  大于或等于预定的阈值,那么目前丢失等级被忽略,并且仅根据数据包大小表设定  $m$  和  $n$ ,从而通过为给定数据包选择最有效的带宽编码方法将带宽使用最小化。预定的阈值可以是不同的,并且可以根据网络策略和拓扑结构设定。

[0104] 如果丢失等级增大为超过了预定的交织阈值,发送编码组件能够切换至交织模式以尝试进一步减少丢失。当丢失等级在另一阈值以下时,能够关闭交织模式以减小延迟。或者,可以对预定数量的第二数据单元在同一数据包中被丢失的间隔时间一直进行计数,并且当这个计数超过预定的阈值,编码组件切换为交织模式。当该计数在预定量的时间内低于预定的阈值时,编码组件返回至非交织模式以减少延迟。

[0105] 上述事例一般涉及端点间的双向通信,其中编码通道的协商是可行和期望的。然而,本发明也能够应用在基本上是单向的或者串流应用中以提高它们的性能,以及应用在协商编码组件间的连接是不实际或者是不需要的其它应用中。在这样的应用中,与各端点相关联的编码组件之间不需要协商。通过识别呼叫信令消息(call signaling message)来创建编码通道,并且随后的深度数据包分析使得与通信会话相关的消息能够被识别。

[0106] 例如,本发明的编码/解码组件能够被用来利用实时流协议(RTSP)的有状态的性质来建立并监控流的连接。RTSP 命令的默认端口是端口 554。因此,本发明的编码组件能够分析数据包报头,来判定它们是否前往或来自 RTSP 默认端口,并且由此识别 RTSP 数据包。如果检测到 RTSP 消息,能够分析该消息以判定其是否包含 SETUP 请求、SETUP 响应、TEARDOWN 请求或普通 RTSP 响应。来自用户的 SETUP 请求包含媒体流 URL 和用于接收 RTP 数据(音频或视频)的本地端口。RTSP 服务器答复确认被选择的参数,并且提供服务器的选择端口。通过分析来自 RTSP 服务器的答复,编码组件能够确定分配给数据流的 RTSP 会话身份识别、消息的 RTSP 序列号以及客户端和 RTSP 服务器已经为会话分配的 RTP 端口。以这样的方式,能够在编码组件之间建立用于流的编码通道。接着,通过诸如深度数据包分析能够分析客户与服务器之间的后续数据单元。如果端口与 SETUP 应答中的那些设定匹配,那么可以如上所述在不需要任何协商来建立编码通道的情况下截获消息用于分段和编码,或者用于解码和重组。

[0107] 类似地,对于 H. 323,编码组件能够为表示 H. 323 呼叫正在建立的消息来分析数据包。例如,编码组件能够对 H. 225/Q. 931 呼叫建立消息和 / 或 H. 245 协商和路径建立消息进行检测并进行深度数据包分析,从而识别将用于相关 H. 323 会话的被叫方端口和主叫方数据端口。随后建立根据本协议的编码通道,以拦截如上所述的数据流并对其进行编码。

[0108] 在给定会话的上下文内,能够以不同的丢失指标来处理不同的流。能够使用 DPI 来检测应用的类型,并且根据对于该类型的应用所配置的网络策略来设定丢失指标。最终,

各数据包基于识别需要的保护等级的报头中提供的信息能够具有自己的丢失指标(例如 SVC-可扩展的视频编解码器(scalable video codec))。

[0109] 在本发明的另一实施例中,  $n$  值的增大是逐渐进行的, 以避免大幅增加开销。当使用该表推荐  $n$  的更高值时, 仅被应用于随后的第一数据单元的子集。使用本实施例, 在接下来输入的  $w$  个第一数据单元中仅有  $v$  个第一数据单元使用增大的  $n$  值, 而其它的  $w-v$  个将使用之前的较小的  $n$  值。如果测得的丢失率继续增大或者由于测得的丢失率达到了下一个等级,  $v$  和  $w$  的值也能够改变。

[0110] 例如, 如果测得的丢失(加权的或归一化的)为 0%, 那么  $n=1$ ,  $m=0$  并且  $v=w=1$ 。因此, 所有的数据包以  $n=1, m=0$  编码。如果测得的丢失增加至大于 0% 且小于 0.05%, 那么  $n=4$ ,  $m=1$ , 但是  $v=1$  而  $w=3$ , 使得 3 个数据包中只有一个以  $n=4$  编码, 而其它的数据包使用之前的编码等级  $n=1, m=0$ 。当测得的丢失超过 0.05% 但低于 0.1% 时, 那么变为  $v=1$  且  $w=2$ , 从而使得每第二个数据包以  $n=4$  编码, 而其它的数据包使用之前的编码等级  $n=1, m=0$ 。当测得的丢失超过 0.1% 但低于或等于 0.2% 时, 那么使用  $v=1$  并且  $w=1$ , 从而使得每个数据包以  $n=4$  编码。 $v$  和  $w$  的不同的值能够被设置用来使得在不同预定丢失等级处的开销增加变得平滑。该能力能够显著平滑不同丢失率之间的转换功能。

[0111] 在下面的说明中, 出于说明的目的, 为了提供对本发明的彻底理解而陈述了很多细节。然而, 对于本领域技术人员显而易见的是, 这些具体细节不是实施本发明所必须的。在其它一些情况下, 为了使得本发明更加清楚, 以框图的形式示出了公知的电气结构和电路。例如, 并没有提供关于本文中说明的本发明的实施例可以实现为软件程序、硬件电路、固件或它们的组合的具体细节。

[0112] 本发明实施例可以在具有诸如服务器或其它计算设备等端点和相关联的编码组件的网络中实施。所述编码组件以及所述方法能够以硬件、软件或者硬件和软件的组合来实施。以软件形式实施的那些部分能够表示为存储于机器可读介质(也称为计算机可读介质、处理器可读介质或者其中含有计算机可读程序代码的计算机可用介质)中的软件产品。机器可读介质可以是任何适当的有形介质, 包括包含软盘、光盘只读存储器(CD-ROM)、存储装置(易失性或非易失性)或者类似的存储机构的磁、光、电存储介质。机器可读介质可以包含不同组的指令、代码序列、配置信息或者其它的数据, 当它们被执行时使得处理器执行本发明实施例的方法中的步骤。本领域技术人员应当理解, 实施本发明所需的其它的指令和操作也可以存储在机器可读介质中。从该机器可读介质运行的软件可以与电路连接以进行上述任务。

[0113] 本发明的上述实施例仅旨在作为示例。在本发明的范围内本领域技术人员可以对具体实施例进行改变、修改和变化, 本发明仅由所附的权利要求书限定。

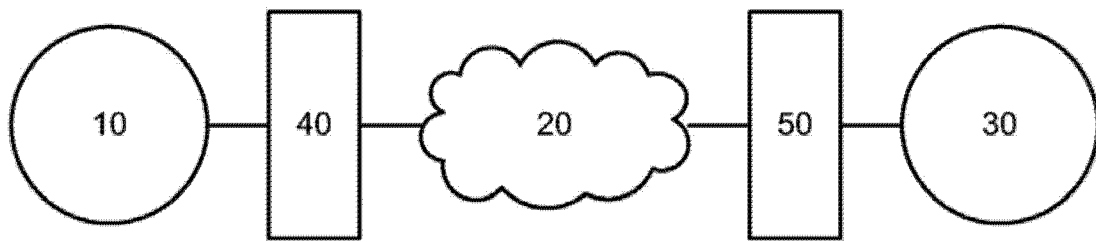


图 1

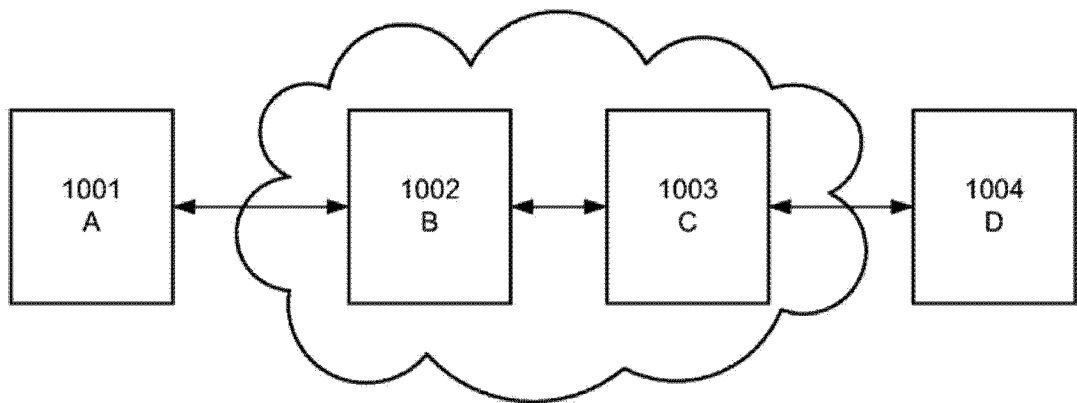


图 9

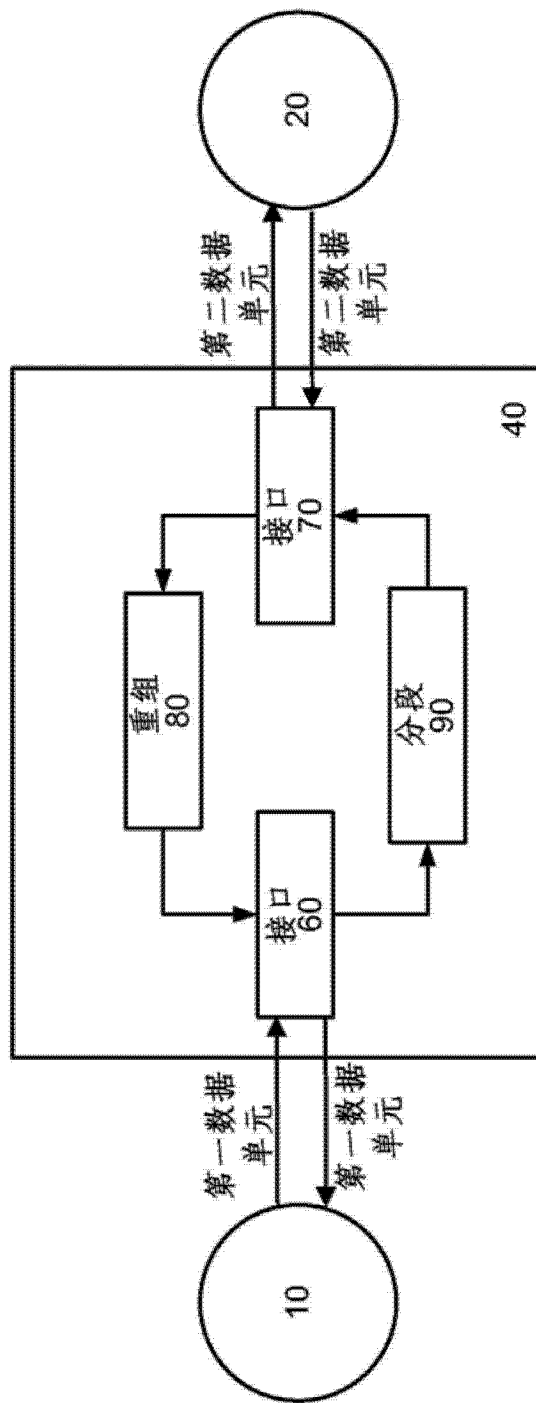


图 2

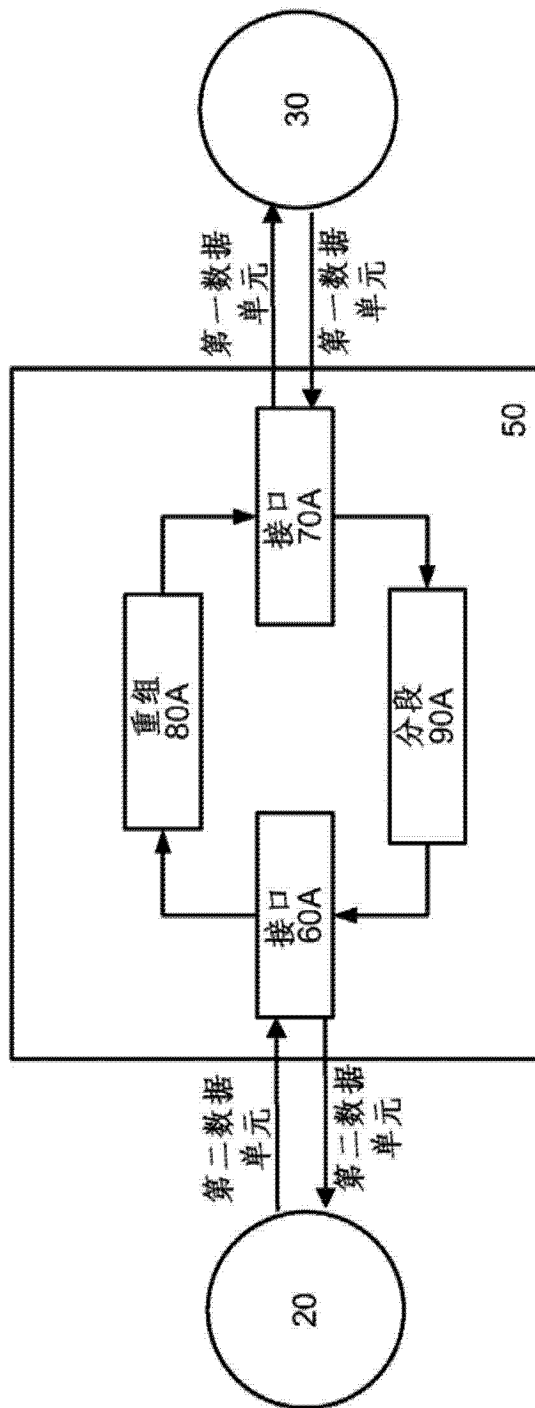


图 3

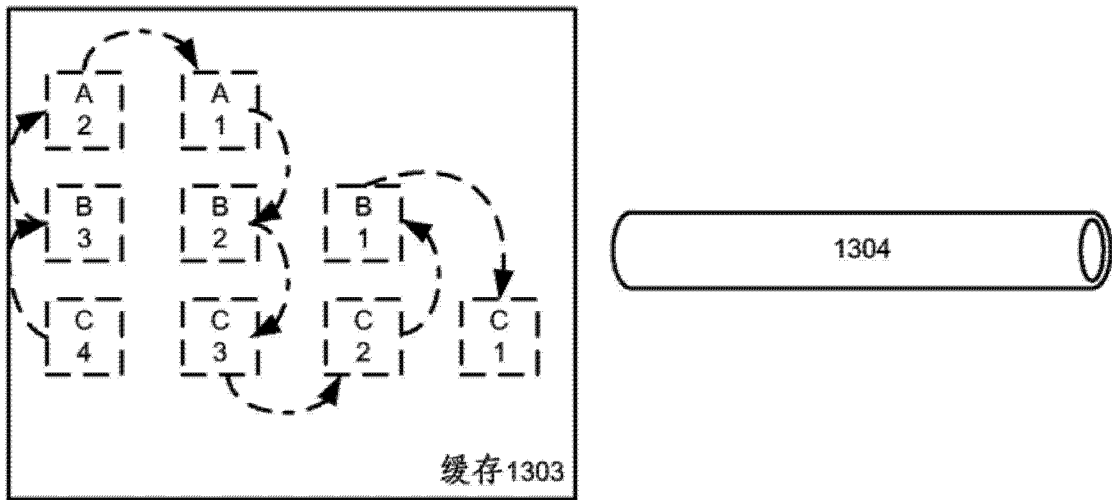


图 4

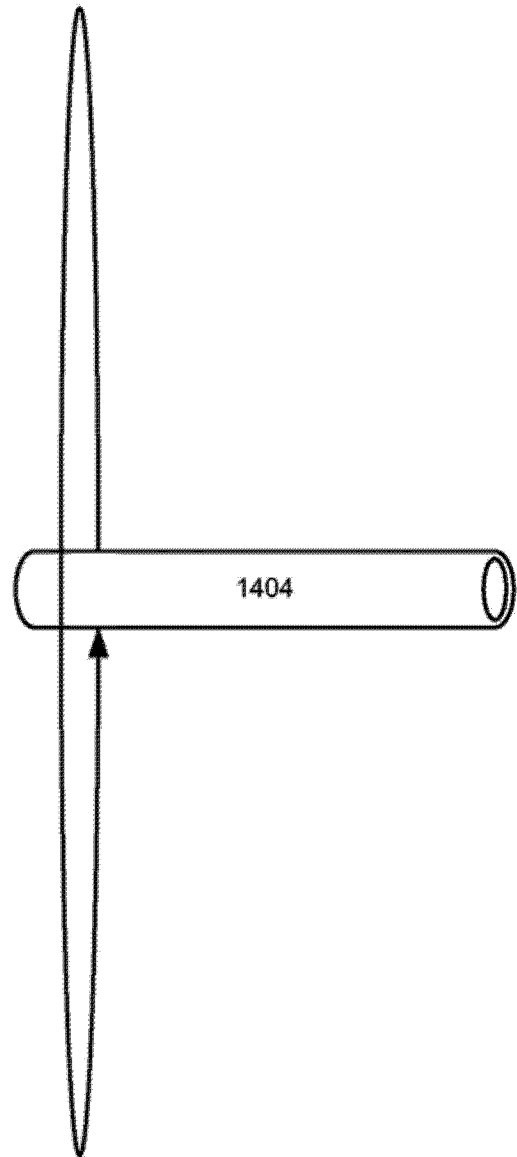
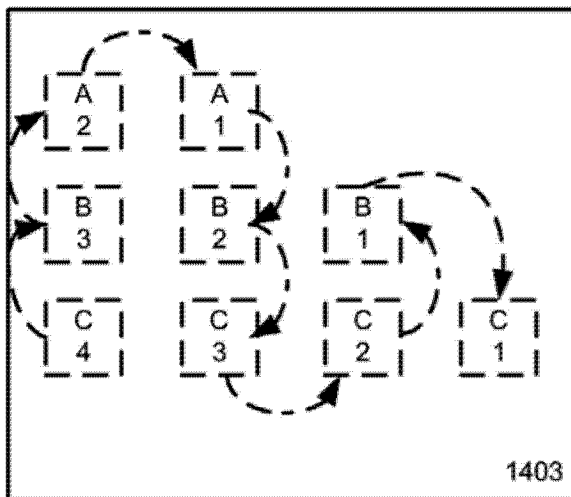
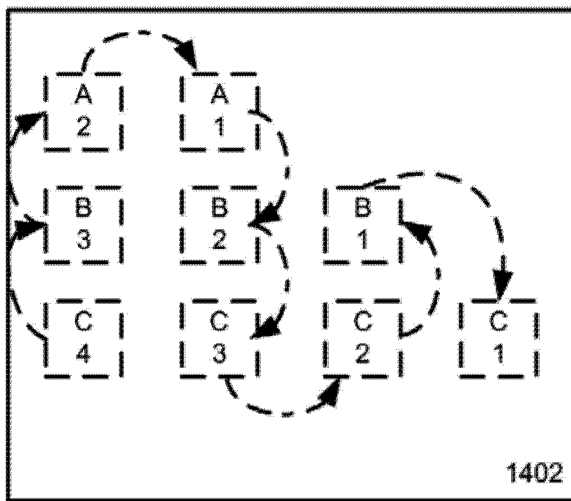
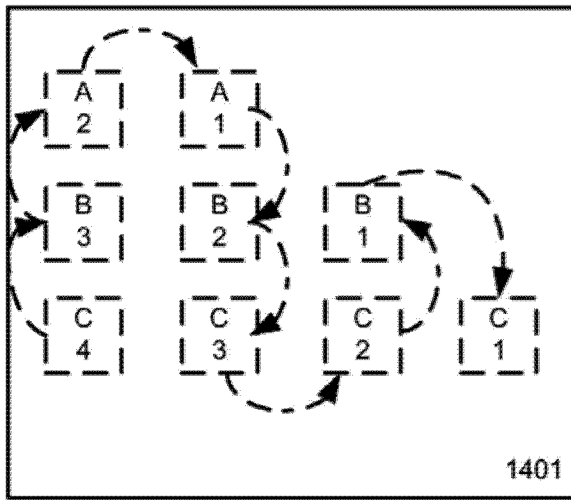


图 5

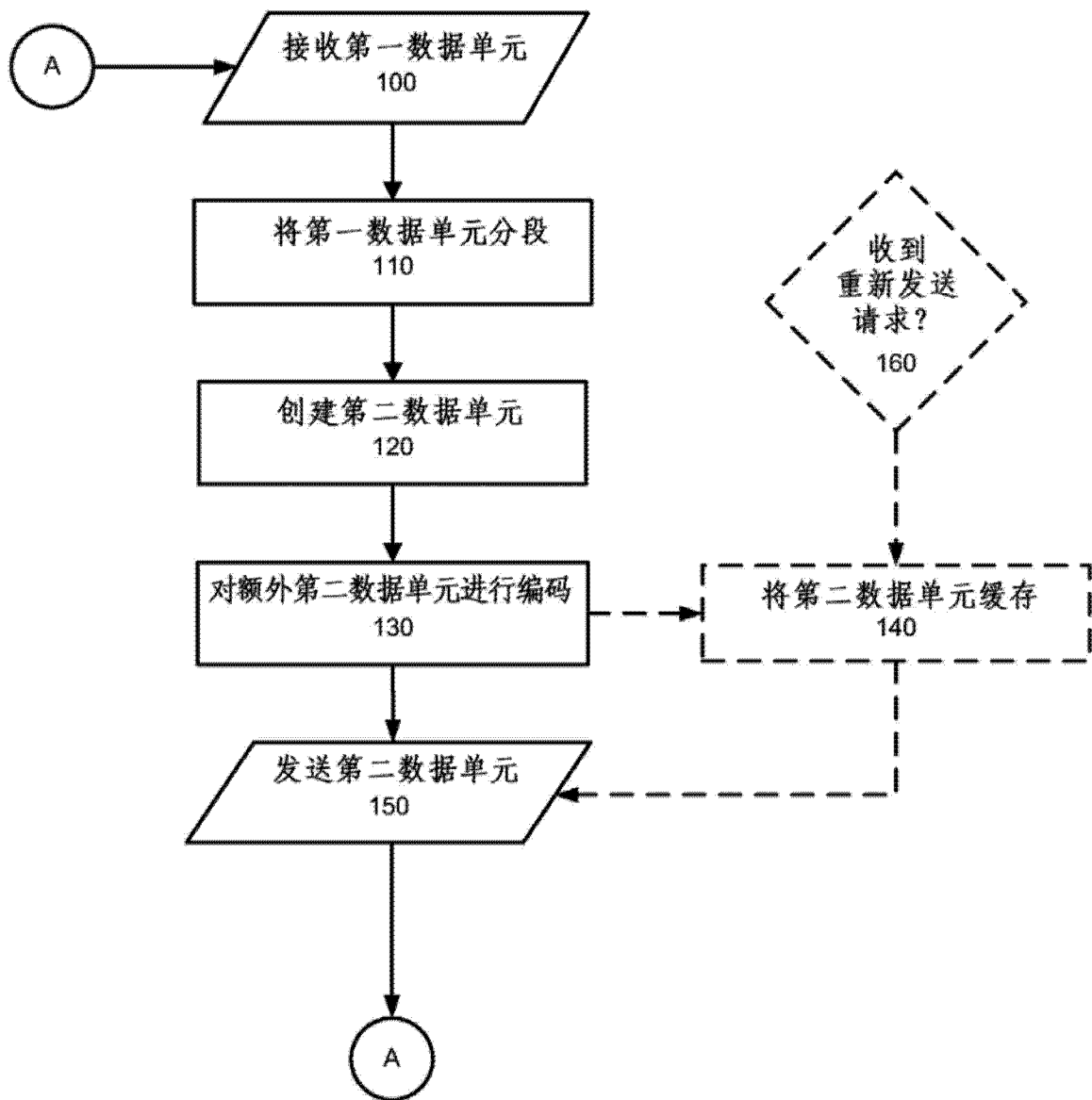


图 6

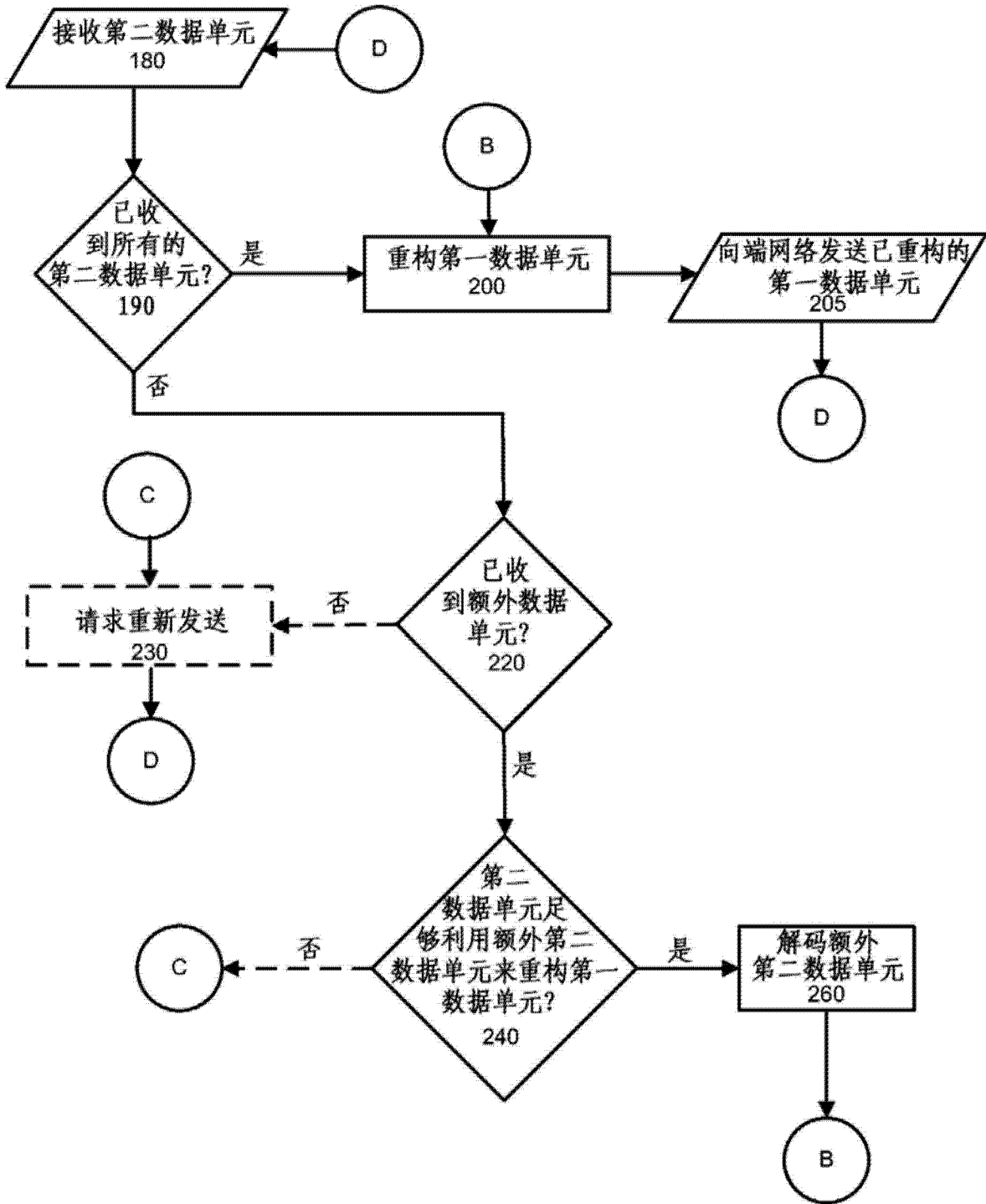


图 7

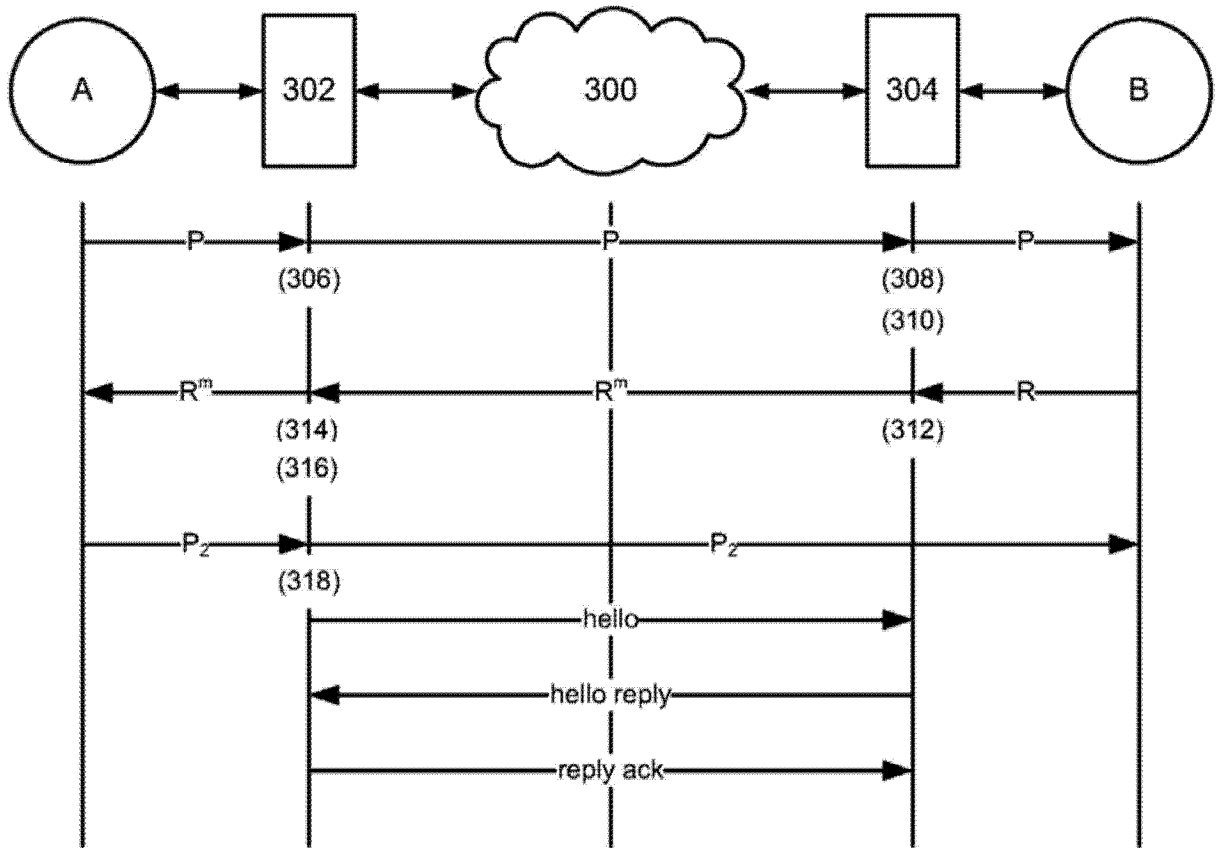


图 8

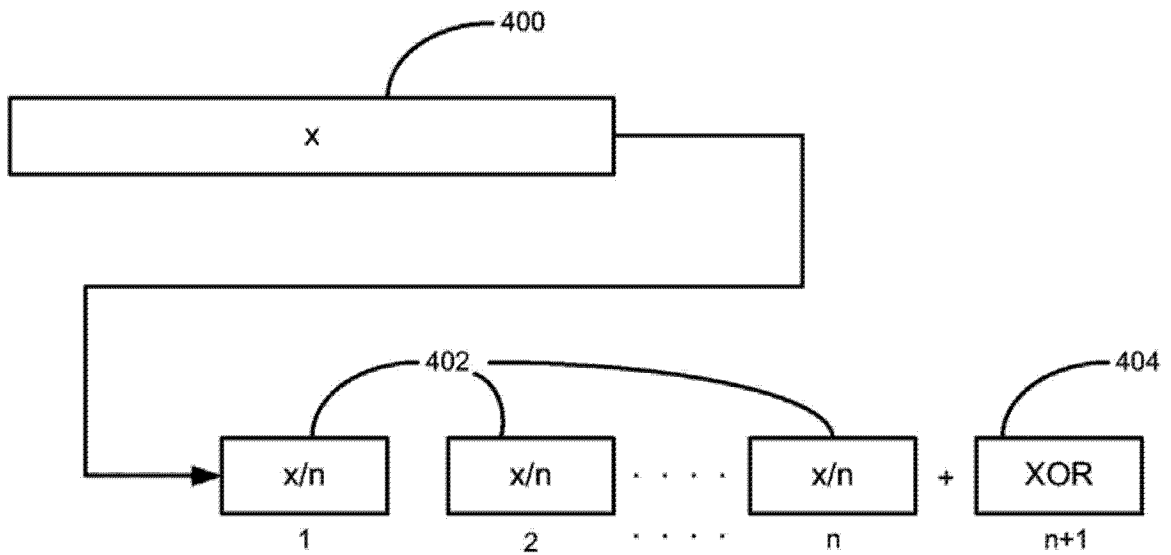


图 10

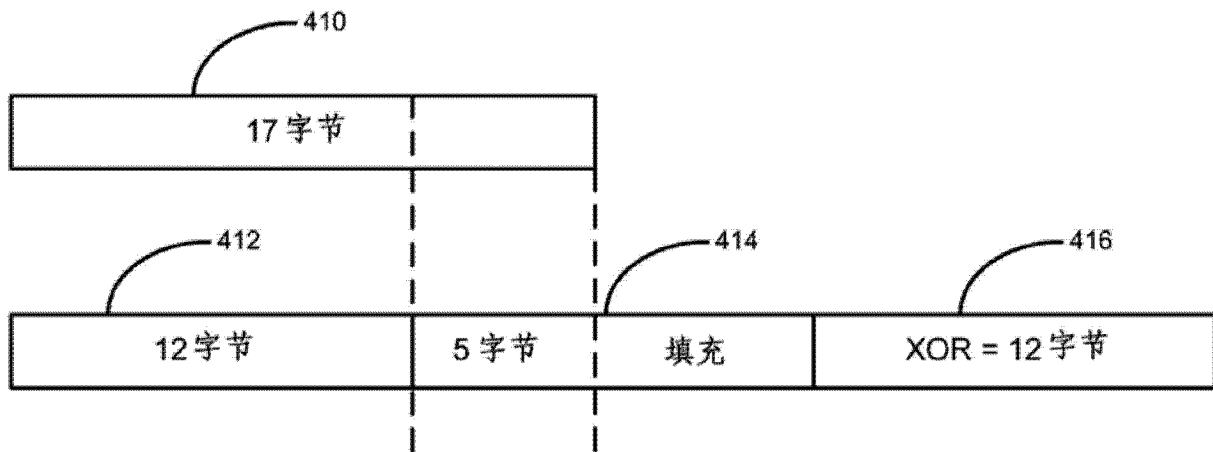


图 11

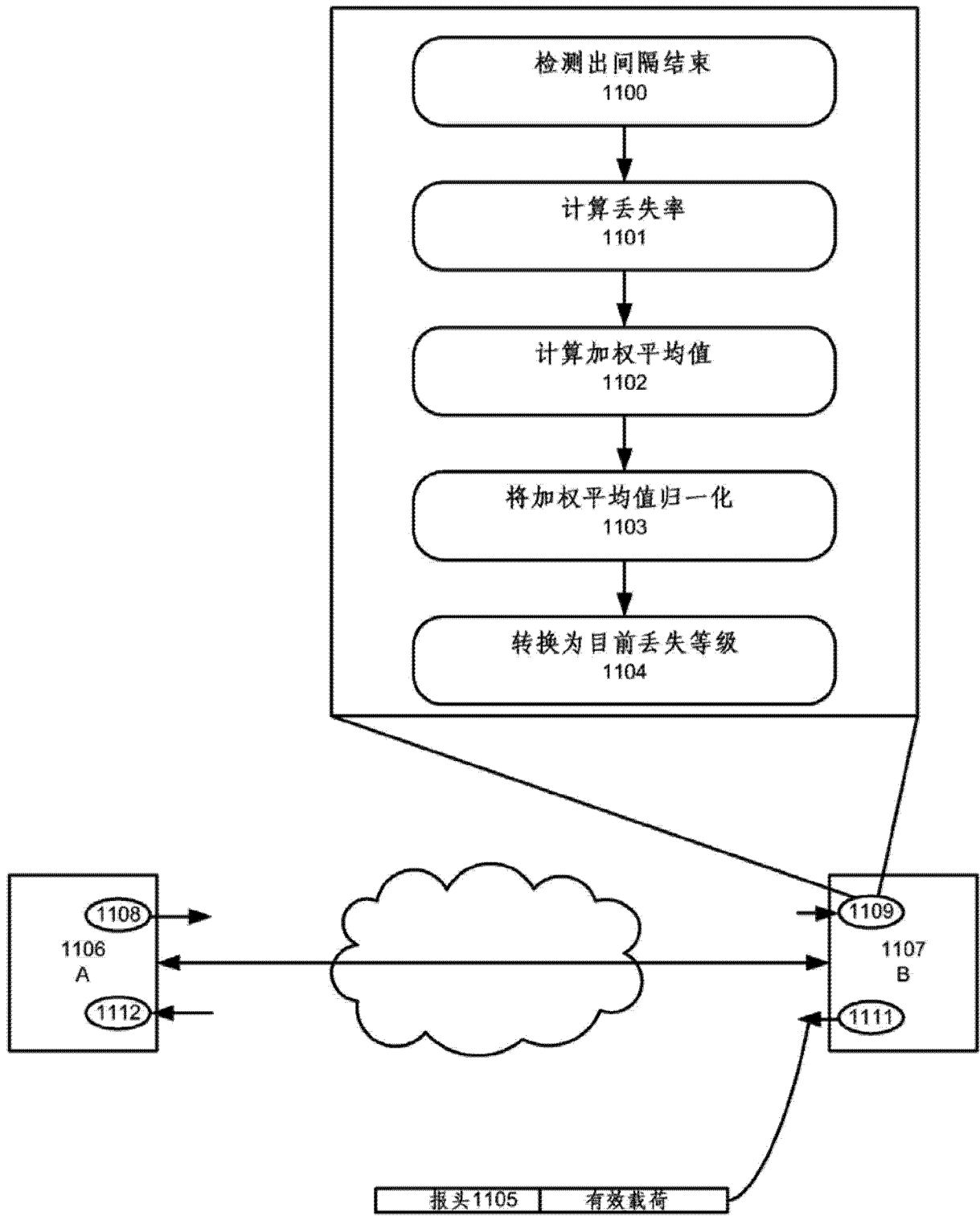


图 12

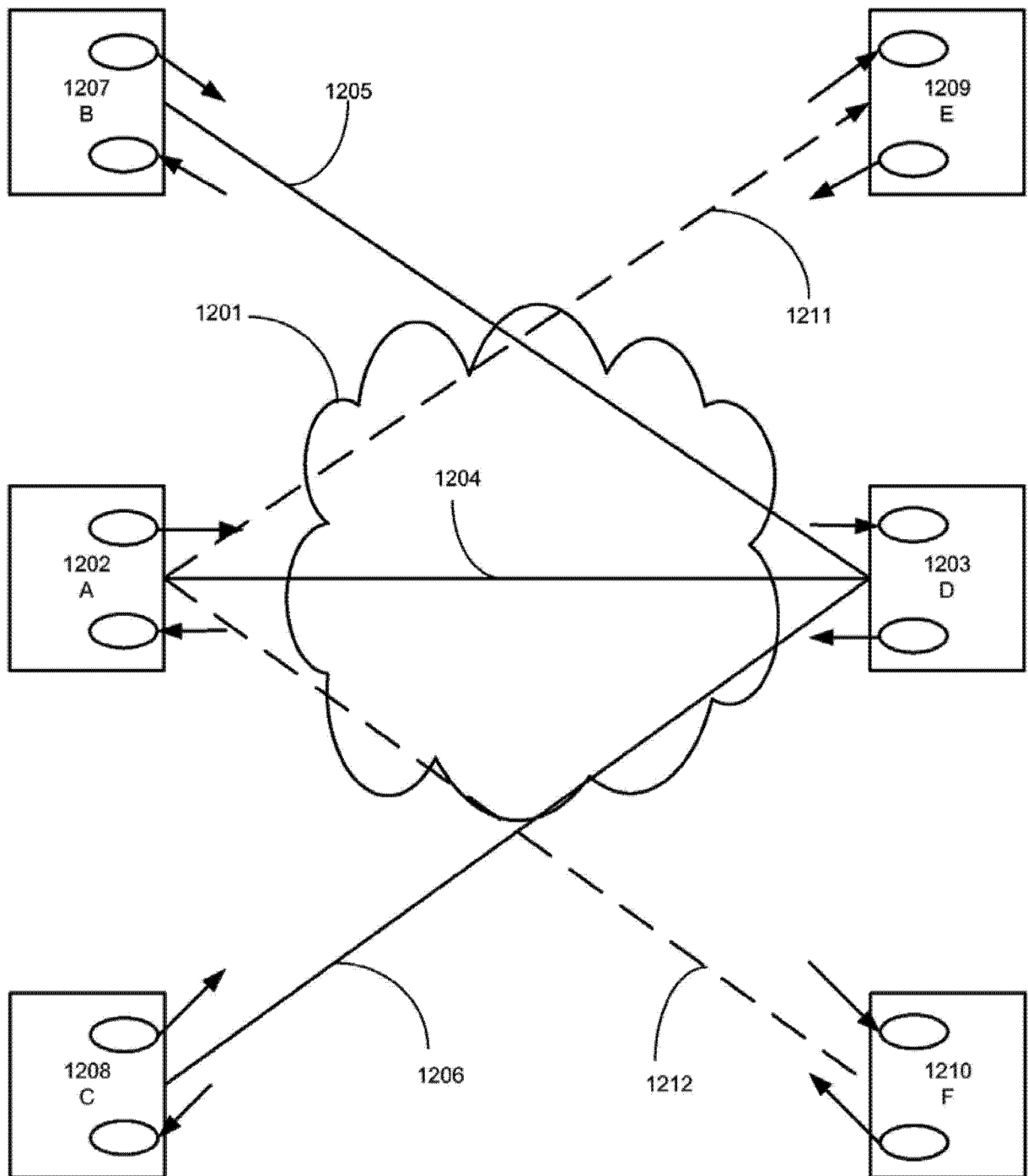


图 13