(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 1836418 B
(45)授权公告日 2010.09.29

(21)申请号 200480023367.X

(22)申请日 2004.07.13

(30)优先权数据
10/640,818 2003.08.14 US

(85)PCT申请进入国家阶段日
2006.02.14

(86)PCT申请的申请数据
PCT/EP2004/051471 2004.07.13

(87)PCT申请的公布数据
WO2005/018194 EN 2005.02.24

(73)专利权人 国际商业机器公司
地址 美国纽约阿芒克

(72)发明人 卡维萨·V·M·巴拉塔克
维尼特·贾恩 瓦萨·瓦拉巴内尼
文卡特·文卡特苏布拉

(74)专利代理机构 北京市柳沈律师事务所 11105
代理人 邸万奎 黄小临

(51)Int.Cl.
H04L 29/06(2006.01)

(56)对比文件
WO 01/97446 A2,2001.12.20, 全文.
CN 1405994 A,2003.03.26, 全文.
US 6252851 B1,2001.06.26, 说明书第5栏第60行至第9栏第23行.
EP 1195966 A2,2002.04.10, 全文.

审查员 姚雅倩

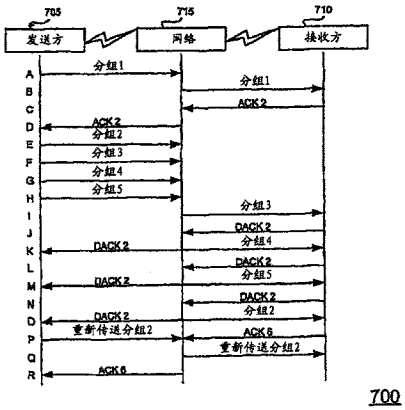
权利要求书 2 页 说明书 9 页 附图 7 页

(54)发明名称

分组重新排序之后改进传输层性能的方法和系统

(57)摘要

一旦重新传送分组，便实现TCP拥塞避免，并且一旦接收到指示分组重新排序的较早确认(ACK)，拥塞避免便回复到原始的拥塞状态，由此消除了不需要的TCP带宽约束。一旦接收到对重新传送的分组的ACK，便基于ACK在发送方的到达时刻，确定ACK是源自原始重新排序的分组还是重新传送的分组的接收。如果用于重新传送的分组的往返时间(RTT)远低于发送方和接收方之间的网络链路的平均或当前计算的往返时间(RTT)，则作为重新排序事件的结果而出现重新传送，并将拥塞窗口恢复回为在重新传送之前的值，由此允许网络链路继续以其原始增加的吞吐量操作。



700

CN 1836418 B

1. 一种用于在网络 (715) 中的分组重新排序之后改进传输层 (312) 性能的方法，其中网络包括发送方 (705) 和接收方 (710) 之间的网络连接，并且其中，网络连接在拥塞状态下操作，所述方法包括以下步骤：

响应 (805) 于接收到预定阈值数目的请求在第一拥塞状态下操作的网络连接上重新传送数据分组的第一确认，在网络连接上将所请求的数据分组从发送方重新传送 (815) 到接收方；

在发送方接收 (820) 指示所请求数据分组已被接收方接收的第二确认；

计算 (825) 接收第二确认的时刻和重新传送所请求的数据分组的时刻之间的差；

其特征在于：

如果该差小于阈值往返时间，则将网络连接的拥塞状态设为 (835) 第一拥塞状态，其中阈值往返时间是用于发送方和接收方之间的网络连接上的数据分组的最小往返时间；以及

如果该差大于阈值往返时间，则将网络连接的拥塞状态设为第二拥塞状态，其中第二拥塞状态小于第一拥塞状态。

2. 如权利要求 1 所述的方法，还包括以下步骤：一旦重新传送所请求的数据分组，便将拥塞状态设为第三拥塞状态。

3. 如权利要求 2 所述的方法，还包括以下步骤：在重新传送所请求的数据分组之前存储 (810) 第一拥塞状态。

4. 如权利要求 2 所述的方法，其中，第三拥塞状态小于第一拥塞状态。

5. 如权利要求 1 所述的方法，其中，预定阈值数目为 3。

6. 如权利要求 1 所述的方法，其中，拥塞状态包括拥塞窗口。

7. 如权利要求 1 所述的方法，其中，拥塞状态包括慢启动阈值。

8. 如权利要求 1 所述的方法，还包括以下步骤 (815)：存储重新传送所请求的数据分组的时刻。

9. 如权利要求 1 所述的方法，其中，网络连接遵循 TCP 协议而操作。

10. 如权利要求 1 所述的方法，其中，阈值往返时间介于网络连接的平均往返时间的 50％和 75％之间。

11. 一种数据处理系统，其提供在网络 (715) 中的分组重新排序之后改进的传输层 (312) 性能，所述系统包括：其中网络包括发送方 (710) 和接收方 (705) 之间的网络连接，并且其中，网络连接包括：

用于响应于接收到预定阈值数目的请求在第一拥塞状态下操作的网络连接上重新传送数据分组的第一确认、将所请求的数据分组从发送方重新传送 (815) 到接收方的装置；

用于接收 (820) 指示所请求的数据分组已被接收方接收的第二确认的装置；

用于计算 (825) 接收第二确认的时刻和重新传送所请求的数据分组的时刻之间的差的装置；

其特征在于：

如果该差小于阈值往返时间、则在第一拥塞状态下操作 (835) 网络连接的装置，其中阈值往返时间是用于发送方和接收方之间的网络连接上的数据分组的最小往返时间；以及

如果该差大于阈值往返时间、则在第二拥塞状态下操作网络连接的装置，其中第二拥

塞状态小于第一拥塞状态。

12. 如权利要求 11 所述的数据处理系统，还包括：一旦重新传送所请求的数据分组、便在第三拥塞状态下操作网络连接的装置。

13. 如权利要求 12 所述的数据处理系统，还包括：在重新传送所请求的数据分组之前存储 (810) 第一拥塞状态的装置。

14. 如权利要求 12 所述的数据处理系统，其中第三拥塞状态小于第一拥塞状态。

15. 如权利要求 11 所述的数据处理系统，其中预定阈值数目为 3。

16. 如权利要求 11 所述的数据处理系统，其中拥塞状态包括拥塞窗口。

17. 如权利要求 11 所述的数据处理系统，其中，拥塞状态包括慢启动阈值。

18. 如权利要求 11 所述的数据处理系统，还包括存储重新传送所请求的数据分组的时刻的装置。

19. 如权利要求 11 所述的数据处理系统，其中，网络连接遵循 TCP 协议而操作。

20. 如权利要求 11 所述的数据处理系统，其中，由数据处理系统发送所请求的数据分组，并且在因特网上从客户机发送确认。

分组重新排序之后改进传输层性能的方法和系统

技术领域

[0001] 本发明一般涉及数据处理系统网络中的数据传送，并且具体地，涉及因特网或类似网络上的数据块的传送。更具体地，本发明涉及在分组重新排序期间改进 TCP 网络中的拥塞控制。

[0002] 背景技术

[0003] 因特网已成为用于数据（文本、代码、图像、视频、音频、或混合）和软件的传送和分配的重要渠道。用户连接到主干网，其具有从 14.4Kb/s 至高于 45Mb/s 的宽泛性能区分层。此外，在因特网和内联网技术中，传输控制协议 / 因特网协议 (TCP/IP) 已成为广泛实现的标准通信协议，其允许客户机、服务器、以及与它们耦接的通信系统之间的宽泛的多相性。因特网协议 (IP) 是网络层协议，而传输控制协议 (TCP) 是传输层协议。在网络层上，IP 提供“数据报”递交服务。相反，TCP 在数据报服务上构建面向连接的传输层服务，以提供两个 IP 主机之间的字节流的确保的顺序递交。将发送到 TCP 的应用数据分为被发送到 IP 的段 (segment)。通过段号码而对段排序。

[0004] 数据传送的可靠性可被三个事件损害 :数据恶化、数据丢失、以及数据的重新排序。通过超时机制来管理数据丢失。TCP 维持定时器（重新传送定时器），以针对所传送的段，测定从接收方接收确认 (ACK) 的延迟。当 ACK 未在所估算的时间间隔内到达时，假定对应的段丢失，并重新传送对应的段。TCP 通过维持对进入分组进行排队的重新组合队列，而管理数据的重新排序、或作为 IP 数据报而传送的数据的段的无序到达，直到依次重新排列了它们为止。仅在此队列中的数据依次到达时，才将所述数据移动到用户的接收缓冲器，其中可由用户查看所述数据。TCP 通过在段到达接收方时对它们执行校验和，来管理数据恶化。对于校验和，TCP 发送方对分组数据计算校验和，并将此 2 字节值置于 TCP 报头上。校验和算法是 TCP 报头和数据中所有 16 位字的 1 的补码和的 16 位 1 的补码。接收方对所接收的数据（除了 TCP 报头中的 2 字节校验和字段之外）计算校验和，并验证其与报头中的校验和值匹配。校验和字段还包括 12 字节的伪报头，其包含来自 IP 报头的信息（包括 4 字节“src ip”地址、4 字节“dest ip”地址、2 字节有效负载长度、1 字节协议字段）。

[0005] 最初，设计 TCP 协议用来在具有低链路误码率的网络中工作，即，所有的段损失主要是由于网络拥塞造成的。结果，发送方在每次检测到段损失时减小其传送速率。然而，当将重新排序的分组错误地确定为网络拥塞中的分组丢失时，来自拥塞控制的所关联的传送速率的减小引起了不必要的吞吐量的降低。

[0006] 对于给定的并行链路的流行、以及分组重新排序的其它原因，在 TCP 网络中分组重新排序是普通的现象。例如，在聚集了很多真实适配器以形成逻辑适配器的以太信道上，当在这些多个适配器上并行发送分组时，一般会引起分组重新排序。在 TCP 中，在接收方对跟随在已丢失或被重新排序的数据分组之后的任何数据分组进行排队，直到这个丢失的分组到达为止。随后，接收方一起确认所有分组。当接收方等待要重新传送的丢失的分组时，不再发送数据。然而，由于分组的重新排序，TCP 会话将自动实现快速重新传送和恢复，因为 TCP 将错误地推断网络拥塞已引起了分组丢失。当重新排序无意地触发了快速重新传

送和恢复、或通过选择确认选项 (SACK) 的通知时，拥塞窗口被截成两半；并且，当出现超时时，将拥塞窗口设为一个段大小，从而强制慢启动。因为这些机制自动地减小拥塞窗口，所以，这样的分组重新排序无意地引起了网络性能的急剧恶化。期望避免由于分组重新排序而造成的这种不必要的吞吐量降低。

[0007] **发明内容**

[0008] 本发明提供了用于在网络中的分组重新排序之后改进传输层性能的方法，其中网络包括发送方和接收方之间的网络连接，并且其中，网络连接在拥塞状态下操作，所述方法包括以下步骤：在发送方接收预定阈值数目的第一确认，该第一确认请求在操作于第一拥塞状态下的网络连接上传送数据分组；响应于接收预定阈值数目的第一确认，在网络连接上将所请求的数据分组从发送方传送到接收方；在发送方接收指示所请求的数据分组已被接收方接收的第二确认；计算接收第二确认的时刻和传送所请求的数据分组的时刻之间的差；其特征在于，如果该差小于阈值往返时间，则将网络连接的拥塞状态设为第一拥塞状态，其中，阈值往返时间是用于发送方和接收方之间的网络连接上的数据分组的最小往返时间；以及如果该差大于阈值往返时间，则将网络连接的拥塞状态设为第二拥塞状态，其中第二拥塞状态小于第一拥塞状态。

[0009] 本发明还提供了一种数据处理系统，其提供在网络中的分组重新排序之后改进的传输层性能，所述系统包括：其中网络包括发送方和接收方之间的网络连接，并且其中，网络连接包括：用于从接收方接收请求传送数据分组的预定阈值数目的第一确认、同时在第一拥塞状态下操作数据处理系统和接收方之间的网络连接的装置；用于响应于接收预定阈值数目的第一确认、将所请求的数据分组传送到接收方的装置；用于接收指示所请求的数据分组已被接收方接收的第二确认的装置；用于计算接收第二确认的时刻和传送所请求的数据分组的时刻之间的差的装置；其特征在于，如果该差小于阈值往返时间、则在第一拥塞状态下操作网络连接的装置，其中阈值往返时间是用于发送方和接收方之间的网络连接上的数据分组的最小往返时间；以及如果该差大于阈值往返时间、则在第二拥塞状态下操作网络连接的装置，其中第二拥塞状态小于第一拥塞状态。

[0010] 因而，本发明提供了用于在网络中的分组重新排序之后改进传输层性能的改进方法、系统和产品。在本发明的一个优选方法中，发送方和接收方之间的网络连接在拥塞状态下操作，该方法包括：接收预定阈值数目的第一确认，该第一确认请求在操作于第一拥塞状态下的网络连接上传送数据分组；响应于接收预定阈值数目的第一确认，在网络连接上将所请求的数据分组从发送方传送到接收方；接收指示所请求的数据分组已被接收方接收的第二确认；计算接收第二确认的时刻和传送所请求的数据分组的时刻之间的差；如果该差小于阈值往返时间 (round-trip-time)，则将网络连接的拥塞状态设为第一拥塞状态，其中阈值往返时间是用于发送方和接收方之间的网络连接上的数据分组的最小往返时间；以及如果该差大于阈值往返时间，则将网络连接的拥塞状态设为第二拥塞状态，其中第二拥塞状态小于第一拥塞状态。

附图说明

[0011] 现在将通过参照附图、仅以例子的方式描述本发明，其中：

[0012] 图 1 示出了其中可实现本发明的优选实施例的数据处理系统网络；

[0013] 图 2 是可在本发明的优选实施例中利用的服务器 - 客户机系统的典型的软件架构的绘图；

[0014] 图 3 示出了利用 TCP/IP 的 4 层通信架构的例子；

[0015] 图 4 示出了包括通过路由器连接到令牌环网的以太网的因特网的例子；

[0016] 图 5 示出了遍历 TCP/IP 协议堆栈时的数据格式；

[0017] 图 6 示出了 TCP 发送或重试帧、以及 TCP 确认帧的数据结构；

[0018] 图 7 示出了由本发明的优选实施例提供的改进的 TCP 性能的例子的时序图；以及

[0019] 图 8 示出了根据本发明的优选实施例、用于在分组重新排序之后的改进的 TCP 性能的过程的流程图。

具体实施方式

[0020] 现在参照附图、并且具体参照图 1，描述其中可实现本发明的优选实施例的数据处理系统网络。数据处理系统网络 100 包括至少一个服务器系统 104，其经由例如因特网 108 的至少一个网络而耦接到至少一个客户机系统 106。服务器系统 104 和客户机系统 106 之间的数据传送遵循 TCP/IP 规范、以及文件传输协议 (FTP)、超文本传输协议 (HTTP)、或某些类似的通信协议。如将理解的，尽管仅示出了单个服务器系统 104 和单个客户机系统 106，但数据处理系统网络 100 可包括：通过包括因特网 108 的一个或多个连接和网络而互连的任意数目的服务器和客户机系统（未示出）。

[0021] 为了在网络上传送数据，有必要具有一组规则，以便适当地执行传送序列的每一个部分。这些规则中的每一个被称为协议，并且，一组规则被称为协议组。通过 TCP/IP（传输控制协议 / 因特网协议）协议组提供在因特网和诸如 LAN（局域网）和 WAN（广域网）的各种其它网络上传送数据时使用的最常用的协议组。TCP/IP 协议组允许运行不同操作系统的多种不同类型的计算机彼此通信。TCP/IP 形成全球因特网（即几乎跨越全球的多于一百万台计算机的广域网）的基础。除了 TCP/IP 组之外还存在很多其它网络协议组，其包括 IPX/SPX（因特网分组交换 / 顺序分组交换）、以及 NetBios。尽管最初由独立研究组开发，但多数网络协议是开放（非私有）的标准，它们中的很多作为一系列用数字排序的 RFC（请求注解）文件而出版。例如，IP 协议是 RFC791。可在因特网上或各种图书馆中容易地得到 RFC 文件。尽管有所不同，但这些网络协议组中的每个在结构上是类似的，其包括一组层，每层负责通信任务的不同方面。为了简化起见，下面的讨论将主要关于在使用 TCP/IP 协议时的本发明的使用。然而，本领域的技术人员应认识到，尽管关于 TCP/IP 协议而描述本发明的原理，但也可将本发明应用于各种其它的网络协议。

[0022] 流程控制是处理在接收方和发送方之间的处理和缓冲容量中的失配的过程，以最佳地利用由传送介质提供的带宽。TCP 流程控制机制在终端站处排它地操作，以限制 TCP 端点发出数据的速率。然而，TCP 缺少显式数据速率控制。基本流程控制机制为被叠加在超越最后被显式确认的字节之外的字节的范围上的“滑动窗口”。滑动窗口限制最近从服务器发送的字节和尚未从客户机接收到接收确认的最早字节之间的连续字节的最大数目。此滑动操作限制 TCP 端点可发出的未确认的可发送数据的量。接收主机在连接建立阶段期间向发送主机通知与分组的“最大”数目有关的其缓冲器容量，其中，所述“最大”数目可为未定（未确认的）任意给定时间。这是接收方或滑动窗口大小，rwnd。发送方维持传送窗口，

其当前大小 wnd 是对可将多少分组送入网络而用不等待确认 (ACK) 的估算。wnd 的上界是 rwnd。

[0023] 在超过滑动窗口限制时，各种算法自动重新发送分组，并较慢地重新启动数据传送。由此，如果服务器和客户机之间的链路在数据集合的传送的中间关闭，则服务器将在客户机确认的上一个分组的一个滑动窗口内停止发送分组。滑动窗口的这种使用在本质上限制了通过网络的数据传送的带宽。

[0024] TCP 中的慢启动 (SS) 算法将窗口大小设为 0，并在启动或重新启动连接时测定传送路由的未知容量。数据路径的带宽延迟积 (product) 是传送路由的容量的粗略近似。在 SS 算法的每一次迭代中，将窗口大小加倍，从而到达延迟带宽积的估算值。

[0025] TCP 拥塞避免算法通过以受控方式减小传送速率，而对付网络中的拥塞。拥塞是由在切换点（例如，在网关或路由器）处的数据报过载而引起的严重延迟的情形，其导致分组的丢弃。发送方通过重新传送受影响的分组而对付延迟和分组丢失。这增加了随后可能快速加剧的拥塞，从而导致现在所谓的拥塞崩溃。“拥塞避免”在如通过超时检测的原始 ACK 丢失时、或在接收方响应于接收无序分组而传送重复的确认 (DACK) 时开始。在其出现时，将在拥塞避免算法的先前迭代中计算的传送路由容量估算值存储在变量 ssthresh 中。由此，使用此估算值作为阈值，直到执行了慢启动、并在之后重新测定传送路由容量为止。在 ssthresh 的窗口大小之外，还采用窗口大小的线性增加。使用另一个变量 cwnd 来存储如通过拥塞而控制的窗口大小的运行估算值。可以看出，将传送窗口限制为拥塞窗口和滑动窗口中的较小者（即，wnd ＝ min(cwnd，rwnd)）。另外，如果通过超时指示拥塞，则将 cwnd 设为一个段大小——强制慢启动。

[0026] 用于增强的 TCP 性能的另一个机制是快速重新传送和恢复 (FRR) 算法。当为特定传送而接收的 DACK 的数目到达了阈值 (typ3) 时，TCP 重新传送丢失的分组，并且，窗口减小到二分之一，而不是接近 1。一旦通过超时（拥塞的强指示）或 DACK（拥塞的弱指示）而检测到拥塞，便将当前拥塞窗口 (cwnd) 的一半存储在 ssthresh 中，以强制拥塞避免。这是拥塞控制算法的乘法减小部分。此外，为了防止在最终确认重新传送时传送分组的突发组 (burst)，在执行拥塞避免时，利用每个 DACK 传送新分组。在丢失了超过传送窗口 (wnd) 的连续数目分组或传送中的上一个分组时，FRR 失败。

[0027] 图 2 是可在本发明的优选实施例中利用的服务器－客户机系统的典型软件架构的绘图。服务器 104 和客户机 106 分别利用软件架构 200 来构建。在最低层，利用操作系统 205 来将高层功能提供给用户和其它软件。典型地，这样的操作系统包括 BIOS（基本输入输出系统）。通信软件 210 通过直接调用操作系统功能、或间接绕过操作系统来访问用于在网络上通信的硬件，经由物理通信链路，通过到例如因特网的网络的外部端口而提供通信。应用编程接口 215 允许作为单机或软件例程的系统的用户使用标准相容接口调用系统功能，而不用关心如何实现特定功能。因特网软件 220 表示可用于给计算机配备因特网功能的几种标准商用软件包中的任一种。应用软件 225 表示任意数目的软件应用，其被设计用来通过通信端口对数据起作用，以提供用户搜寻的期望功能。此层上的应用可包括处理数据、视频、图形、图片或文本所需的那些应用，它们可被因特网的用户访问。

[0028] 如图 3 所示，通过包括应用层 310、传输层 312、网络层 314、以及链路层 316 的用于网络的 4 层通信架构 300，来利用 TCP/IP 和类似协议。每层负责处理如下的各种通信任务。

链路层 316（也被称为数据链路层、或网络接口层）通常包括操作系统中的设备驱动器、以及计算机中对应的网络接口卡。它们一起处理与正在使用的网络介质（例如，以太网线缆等）物理对接的所有硬件细节。

[0029] 网络层 314（也被称为因特网层）处理网络各处的数据分组的移动。例如，网络层处理在网络上传送的各种数据分组的路由。TCP 组中的网络层由几种协议组成，包括 IP（因特网协议）、ICMP（因特网控制消息协议）、以及 IGMP（因特网组管理协议）。

[0030] 传输层 312 提供网络层 314 和应用层 310 之间的接口，其帮助数据在两个主计算机之间的传送。传输层关心这样的事情，如将从应用传递给它的数据划分为用于下面的网络层的适当大小的数据块 (chunk)、确认所接收的分组、设置超时以确定被发送的其它结束确认分组，等等。在 TCP/IP 协议组中，存在两种明显不同的传输协议 :TCP（传输控制协议）和 UDP（用户数据报协议）。TCP 提供可靠性服务，以确保在两个主机之间正确地传送数据，该服务包括丢失检测和重新传送服务。相反，UDP 仅通过将称为数据报的数据分组从一个主机发送到另一个，而向应用层提供简单得多的服务，而不提供用于确保正确地传送数据的任何机制。在使用 UDP 时，应用层必须执行可靠性功能。

[0031] 应用层 310 处理特定应用的细节。存在几乎每一个实现所提供的很多常见 TCP/IP 应用，其包括 :(1) 用于远程登录的 Telnet ;(2)FTP，即文件传输协议 ;(3)SMTP，即用于电子邮件的简单邮件传输协议 ;以及 (4)SNMP，即简单网络管理协议。

[0032] 通过各自将两个或更多网络连接在一起的路由器，互连例如因特网的网络。典型的路由器包括具有输入和输出连接、以及专用硬件的特定用途硬件盒，和 / 或允许连接很多不同类型的物理网络（如以太网、令牌环、点对点链接等）的嵌入软件。图 4 示出了因特网 400，其包括通过路由器 436 连接到令牌环网络 434 的以太网 432。尽管图 4 仅示出了通信中的两个主机，但以太网上的任意主机可与其上的任意主机、或与令牌环网络上的任意主机通信，并且，反之亦然。

[0033] 如图 4 所示，路由器 436 包括网络层模块 438（在此情况下是 IP 模块）、以及用于连接到主机网络的适当的网络驱动器，即以太网驱动器 440 和令牌环驱动器 442。在应用层，网络包括 FTP 客户机 420 和 FTP 服务器 422。将大多数网络应用设计为使得一端是客户机，而另一端是服务器。服务器向各种客户机提供某些类型的服务，在此情况下为访问服务器主机上的文件。每层具有用于与其在相同层上的同位体 (peer) 通信的一个或多个协议。这些通信协议包括应用层上的 FTP 协议 444、传输层上的 TCP 协议 446、网络层上的 IP 协议 448、以及链路层上的以太网协以 450 和令牌环协议 454。对于应用层来说常见的是处理用户过程，同时在诸如 UNIX 或 Windows 操作系统的操作系统的内核中实现较低的三个层（传输、网络和链路）。例如，网络接口层的目的在于处理通信介质（以太网、令牌环等）的细节，而应用层的目的在于处理一个特定的用户应用 (FTP、Telnet 等）。

[0034] 应用层和传输层使用端到端协议 (FTP 协议 444、TCP 协议 446)。网络层提供路程段到路程段 (hop-to-hop) 协议，其在两个端系统以及之间的每个中间系统上使用（为了清楚起见，这里仅示出了一个中间系统）。例如，通过 IP 协议 448 而将路由器 436 的 IP 模块 438 连接到两个主机。还存在专用于连接到路由器的各种类型的主机网络的链路层协议，以在链路层上处理网络和路由器之间的通信。由此，使用以太网协议 450 来处理以太网 432 上的路由器 436 中的以太网驱动器 440 和主机的以大网驱动器 452 之间的通信，而使用令

牌环协议 454 来处理令牌环网络 434 上的路由器 436 的令牌环驱动器 442 和主机的令牌环驱动器 456 之间的通信。

[0035] 在 TCP/IP 协议组中，网络层（即 IP）提供不可靠的服务。它将数据分组从源移动到目的地，但它不提供用于确保递交、或者甚至能够确定是否已发生了正确传送的机制。TCP 提供可靠服务，以保证在两个主机之间正确地传送了数据，该服务包括丢失检测和重新传送服务。

[0036] 路由器具有两个或更多网络接口层（由于其连接两个或更多网络）。具有多个接口的任意系统被称为多重初始地址 (multi-homed)。主机也可以是多重初始地址，但是，除非它具体地将分组从一个接口转发到另一个，否则它不能被称为路由器。并且，路由器不需要仅在因特网各处移动分组的特定硬件盒。大多数 TCP/IP 实现允许多重初始地址主机用作路由器，但是主机需要被具体地配置为支持此用途。在这样的实例中，系统为主机（在使用诸如 FTP 或 Telnet 的应用时）或路由器（在它将分组从一个网络转发到另一个时）。连接网络的另一种方式是通过网桥。网桥在链路层上连接网络，而路由器在网络层上连接网络。网桥使多个 LAN 对上层呈现为单个 LAN。

[0037] 当应用使用 TCP/IP 来发送数据时，将数据通过每个层而向下发送到协议堆栈，直到通过网络将其作为比特流而发送了为止。如图 5 所示，每层通过将报头预先附加 (prepend) 到其接收的数据上而将信息添加到数据上。例如，在应用层，将应用报头 580 预先附加到用户数据 582 上，以形成应用数据 584。在传输层，将传输协议报头预先附加到应用数据上。在图 5 的情况中，传输层是 TCP，并且因此，将 TCP 报头 586 预先附加到应用数据 584 上，由此形成 TCP 帧 588，其被发送到网络层 IP。TCP 报头 586 包括 20 个字节。类似地，在网络层，将网络层报头预先附加到传输层数据上。在 TCP/IP 的情况下，将 IP 报头 590 预先附加到 TCP 帧 588，以形成 IP 数据报 592。IP 报头 590 也包括 20 个字节。最后，在链路层，将例如以太网报头 594 的介质报头添加到从网络层接收的数据上，以形成数据帧。在例如当介质是以太网的某些实例中，还将介质尾部 (trailer) 附加到数据的末尾。例如，在图 5 中，将介质尾部 596 附加到以太网报头 594 和 IP 数据报 592 上，以形成以太网帧 598。以太网帧包括与原始应用消息数据相对应的流过网络的比特流。报头底部的数 (14、20、20、4) 是字节形式的报头的典型大小，例如，以太网报头 94 包括 14 个字节，等等。帧的大小将受正在用于传送数据分组的网络类型的最大传送单位 (MTU) 限制。例如，以太网的 MTU 是 1500 字节。网络层自动执行分段 (fragmentation)（将数据报分割为较小的块），使得每个段小于网络的 MTU。

[0038] 当客户机检测到某些数据帧从数据传送流丢失时，客户机将通过在确认帧中发送所丢失的帧的第一字节的序列号，而请求服务器重新传送丢失的帧。如在图 6 中看到的，典型地，TCP 发送或重试消息 610 包括介质报头 612、协议报头 614、所接收的序列号字段 616、发送序列号字段 618、以及消息体 620。介质报头 612 对网络的类型（例如用于以太网的以太网报头等）将是具体的。协议报头 614 将取决于所使用的传输和网络层协议，如 TCP/IP、IPX/SPX、Netbios 等。所接收的序列号字段 616 向由计算机可靠地接收的上一个序列号提供标识符。发送序列号字段 618 对应于消息的相对序列号。消息体 620 包含应用数据，其在源和目的计算机之间被发送。TCP 确认帧 622 包括介质报头 624、协议报头 626、所接收的序列号字段 628、以及第发送列号字段 630。这些字段类似于共享相同名称的上述字段。通

过接收计算机而发送确认帧 622，以确认发送或重试消息的接收。根据 TCP，一旦接收到指示丢失帧的三个连续的确认帧，服务器便会正常地“快速重新传送”从丢失的序列号开始的丢失帧。由于服务器缩减其滑动窗口并减小其发送的数据量，所以这样的重新传送对性能带来负面影响。

[0039] 现在参照图 7，其中示出了由本发明的优选实施例提供的改进的 TCP 性能的例子的时序图。图 7 中图解的例子示出了在网络 715 上将 5 个数据分组传送到接收方 710 的发送方 705 的时序图 700。在时刻 A，将分组 1 发送到网络 715。在时刻 B，将分组 1 从网络 715 传送到接收方 710。在时刻 C，接收方 710 确认分组 1 的接收，并请求分组 2(ACK2)。在时刻 D，将 ACK2 从网络 715 传送到发送方 705。

[0040] 在时刻 E、F、G 和 H，发送方 705 分别将分组 2、3、4 和 5 传送到网络 715。然而，网络 715 对从发送方 705 接收的分组重新排序，并且，如在时刻 I 所看到的，首先将分组 3 传送到接收方 710。响应于接收分组 3，接收方 710 在时刻 J 传送重复的 ACK2(DACK2)，其指示接收方 710 尚未接收到分组 2，并且无序地接收了分组 3。在时刻 K，将 DACK2 从网络 715 传送到发送方 705。并且，在时刻 K，网络 715 将分组 4 传送到接收方 710。作为响应，接收方 710 在时刻 L 传送第二 DACK2，其指示已无序地接收了分组 4 并请求分组 2。在时刻 M，将此第二 DACK2 从网络 715 传递到发送方 705，并且，将分组 5 从网络 715 传递到接收方 710。在时刻 N，接收方 710 再次响应分组 5 的接收，重新传送第三 DACK2，确认分组 5 的接收，并再次请求分组 2。在时刻 O，在发送方 705 处从网络 715 接收此第三 DACK2。并且，在时刻 O，接收方 710 最终从网络 715 接收分组 2。在时刻 P，如由 TCP 协议所规定的，一旦接收到第三 DACK2，发送方 705 便“快速重新传送”分组 2。因为 TCP 将第三 DACK 的接收视为网络 715 上的拥塞的结果，所以，一旦重新传送分组 2，发送方 705 便将拥塞窗口减小 50％。并且，在时刻 P，接收方 710 现在已接收了分组 2、3、4 和 5，并通过请求分组 6 的 ACK 而做出响应。作为网络 715 中分组重新排序的结果，在网络 715 中同时存在 ACK6 和重新传送的分组 2 两者。（将会理解，此定时仅代表例子，而在此过程中可出现其它定时。用于本发明的方法的适当操作的仅有的定时需求是针对在接收方 710 接收重新传送的分组 2 之前要从接收方 710 传送的 ACK6。）在时刻 Q，接收方 710 从网络 715 接收重新传送的分组 2。在时刻 R，发送方 705 从网络 715 接收 ACK6。

[0041] 在这一点上，在重新传送分组之后的时刻 R 接收的 ACK6 是响应于分组 2 的原始传送还是重新传送是不确定的。在现有技术中，假定 ACK 是接收方 710 接收重新传送的分组 2 的结果，于是，根据 TCP，发送方 705 继续通过减小的拥塞窗口而操作。然而，本发明的优选实施例向发送方 705 提供用于在作为实际分组丢失事件的结果而接收的 ACK 和作为分组重新排序的结果而接收的 ACK 之间进行区分的方法。如果发送方 705 根据优选实施例而确定作为重新排序的结果而出现了快速重新传送，那么，不执行拥塞控制和避免，由此消除了对 TCP 带宽的不需要的约束。

[0042] 本发明的优选实施例是在通过“快速重新传送”而重新传送分组并减小拥塞窗口之前、发送方 750 将当前拥塞状态保存为变量 CCS 时开始的过程，该变量 CCS 表示 SSThresh 和 cwnd。一旦接收到确认所传送的分组的后续 ACK（在图 7 的例子中是 ACK6），TCP 发送方 705 便基于 ACK 在发送方 705 的到达时刻，而确定 ACK 是源自原始重新排序的分组还是重新传送的分组的接收。如果 ACK 在所计算的用于重新传送的分组的往返时间的时间阈值

(THRESH-RTT)的期满之前到达，则发送方705假定已出现了网络715中的重新排序，而不是分组丢失。在此情况下，使用CCS来将拥塞状态恢复到所保存的值，由此，因为并未作为网络拥塞的结果而出现重新传送，所以允许系统在其原始的增加的吞吐量下继续操作。

[0043] 为计算阈值重新传送RTT(THRESH-RTT)，TCP发送方705存储重新传送分组的时刻(Tretransmitted)。在重新传送确认到达时，发送方705通过存储确认到达的时刻(Tcurrent)、并随后计算重新传送和到达时刻之间的差(即Tcurrent-Tretransmitted)来计算RTT。如果该差远低于用于发送方和接收方之间的链路的平均或当前计算的往返时间(RTT)，那么，存在作为重新排序事件的结果而出现的重新传送的强指示。下面的等式描述了优选实施例中的此计算：

[0044] (Tcurrent-Tretransmitted)<alpha*RTT

[0045] 其中，“alpha”为0和1之间的数，且为用于估算发送方和接收方之间的最小可能RTT的因子，并且，其补偿网络中的RTT和无关的急剧下降的RTT中的变化。在优选实施例中，在0.5和0.75(即50-75％)的范围内设置alpha，并且，优选为0.6。一旦基于被确认的以上计算而检测到重新排序事件，发送方将把拥塞状态恢复回到执行快速重新传送之前的系统值(即，回到所存储的用于CCS的值)。如将理解的，优选实施例的机制在发送方一例实现，并且不需要对接收方一侧逻辑的改变。这确保了此技术将与不实现此性能增强的所有现有和未来的TCP实现来兼容工作。

[0046] 如将理解的，在一个优选实施例中，一旦重新传送分组，便实现拥塞避免，并且，一旦接收到指示分组重新排序的较早的ACK，拥塞避免便回复到原始的CCS。在另一个优选实施例中，拥塞避免的实现在快速重新传送之后被延迟，直到在接收了对重新传送的及时响应ACK之后、确认网络拥塞引起了分组丢失为止。

[0047] 现在参照图8，其中示出了根据本发明的优选实施例、用于在分组重新排序之后的改进的TCP性能的过程的流程图。当在发送方705接收的重复确认的数目超过了传输协议中的阈值数目、而触发所指示的分组的快速重新传送和恢复(FRR)时，过程800在步骤805开始。在TCP协议中，典型地，FRR在发送方705接收到3个DACK时出现。之后，过程转到步骤810，其中，发送方705存储当前拥塞状态(CCS)，其包括拥塞窗口和慢启动阈值，并且，通过将慢启动阈值(SSTHRESH)减小到其当前值的一半而实现拥塞避免。随后，过程转到步骤815，其中，发送方705重新传送所指示的丢失的分组，并将出现重新传送的时刻存储在变量Tretransmission下。随后，过程转到步骤820，在该点处，发送方705接收已被重新传送的分组的确认，并将此确认的到达时刻存储为变量Tcurrent。之后，过程转到步骤825，其中，通过确定重新传送的时刻和分组的确认到达时刻之间的差，而计算用于重新传送的分组的往返时间(rrt)。随后，过程转到判定块830，其中，确定重新传送的分组的往返时间是否比发送方705和接收方710之间的路由的最小预计往返时间小预定量，其中，预定量是网络715中的链路特性和RTT中的理论上可能急剧下降的函数。通过用“alpha”表示的比例因子而实现此预定量，并将其设为0和1之间，优选是在0.5-0.75的范围中。由此，可通过等式rrt<alpha*RTT来表示步骤830的确定。如果步骤830的确定为否，则过程转到步骤840，其中，因为已确定响应于丢失的分组而正确地实现了拥塞避免，过程结束。如果步骤830的确定为是，则过程转到步骤835，其中，将发送方705中的拥塞状态恢复到在步骤810存储的CCS，并且之后，过程在步骤840结束。

[0048] 将理解，因为在步骤 825 计算的往返时间指示响应于到达接收方 710 的重新路由的原始分组、而不是响应于重新传送的分组的接收而接收了确认，所以，拥塞状态返回到在发送方 705 启动快速传送之前存在的 CCS。因此，根据优选实施例，通过将拥塞状态重新置为 CCS，取消从重新传送的分组的重新排序而导致的拥塞避免的错误触发，由此通过增加吞吐量而提升性能，否则，吞吐量会通过 TCP 拥塞避免的实现而被扼制。

[0049] 尽管已通过参照优选实施例而具体地示出并描述了本发明，但本领域的技术人员将理解，可在其中做出各种形式和细节上的改变，而不背离本发明的精神和范围。例如，可使用计算机编程软件、固件或硬件的任意组合来实现本发明。作为实践本发明或构造根据本发明的设备的预备步骤，典型地，根据本发明的计算机编程代码（软件或固件）将被存储在一个或多个机器可读存储介质中，如固定（硬）驱动器、磁盘、光盘、磁带、诸如 ROM 和 PROM 的半导体存储器等，由此产生根据本发明的产品。通过直接从诸如硬盘、RAM 等的存储装置执行代码、通过将代码从存储装置复制到另一个存储装置、或通过传送用于远程执行的代码，而使用包含计算机编程代码的产品。可通过组合具有用来运行代码的适当的标准计算机硬件的、包含代码的一个或多个机器可读存储装置，而产生用于实践本发明的方法。用于实践本发明的设备可为一个或多个计算机和存储系统，其包含或具有对根据本发明而编码的计算机程序（多个）的网络访问。

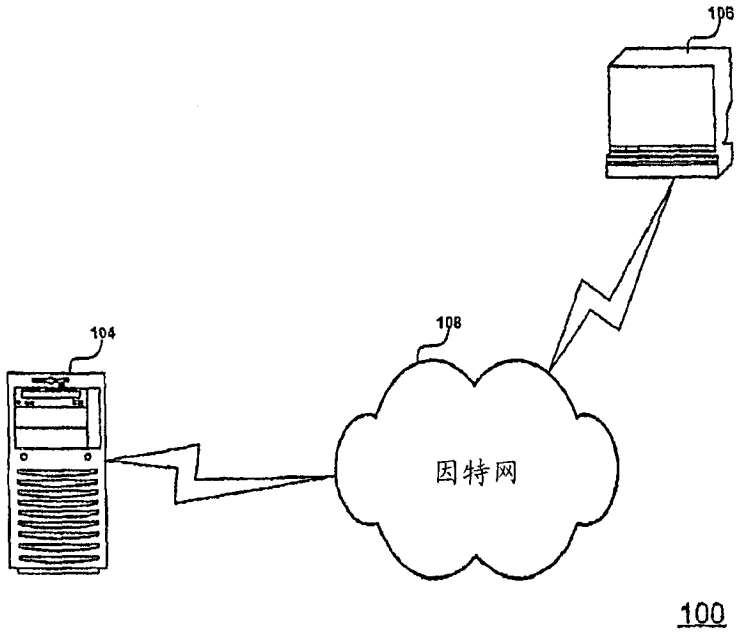


图 1

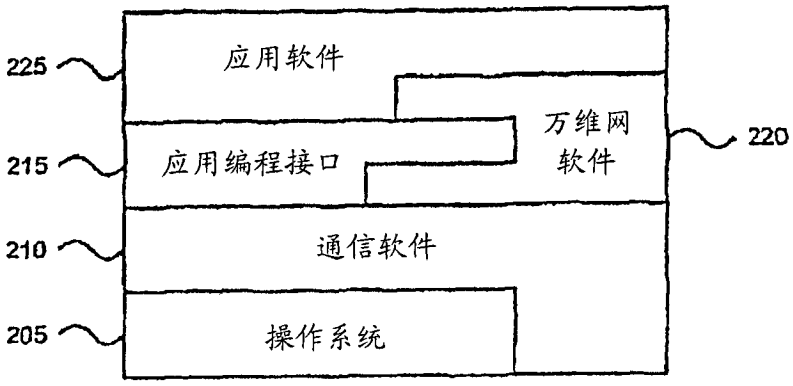

225
应用软件
万维网
软件
220
215
应用编程接口
210
通信软件
205
操作系统
200

图 2

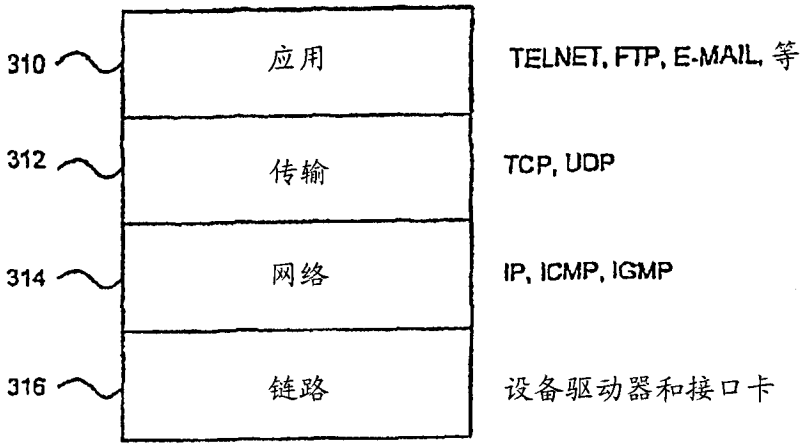

310
应用
TELNET, FTP, E-MAIL, 等
312
传输
TCP, UDP
314
网络
IP, ICMP, IGMP
316
链路
设备驱动器和接口卡
300

图 3

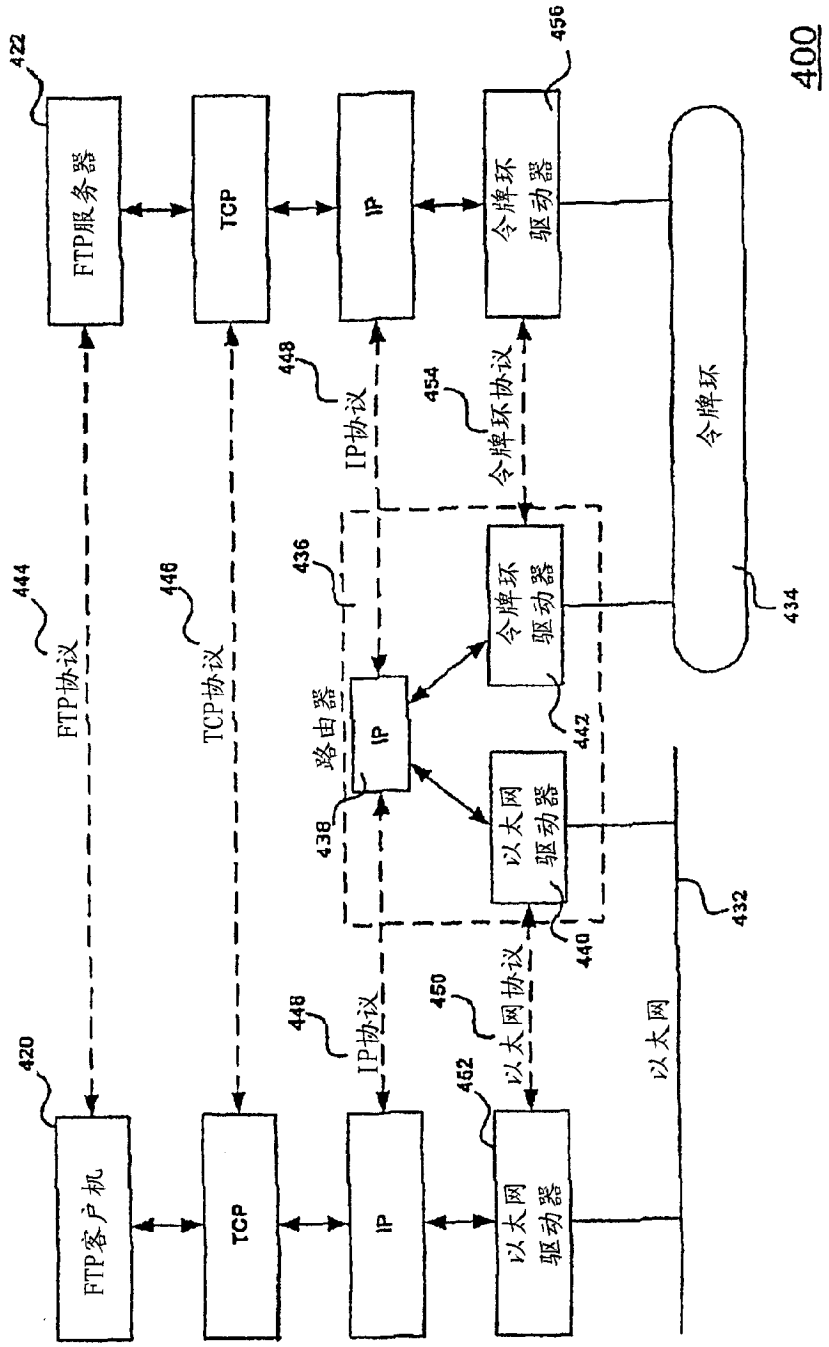

400
422
FTP服务器
TCP
IP
令牌环驱动器
456
448
IP协议
454
令牌环协议
令牌环
434
444
FTP协议
446
TCP协议
436
路由器
IP
令牌环驱动器
442
438
以太网驱动器
440
432
448
IP协议
450
以太网协议
以太网
420
FTP客户机
TCP
IP
452
以太网驱动器

图 4

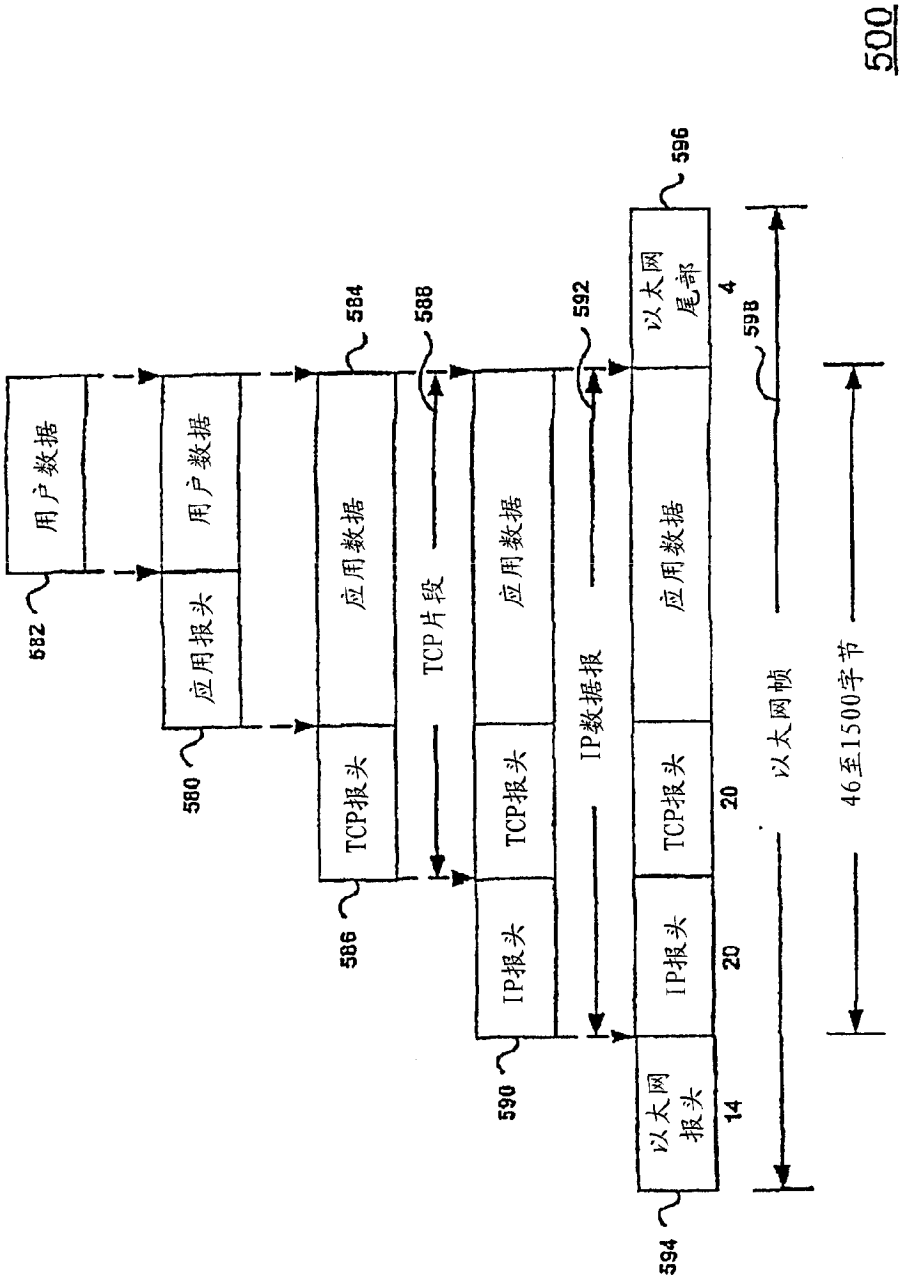

500
用户数据
582
用户数据
应用报头
580
应用数据
TCP报头
586
584
TCP片段
588
应用数据
TCP报头
IP报头
590
IP数据报
592
应用数据
TCP报头
IP报头
以太网报头
594
以太网尾部
596
4
598
20
20
14
以太网帧
46至1500字节

图 5

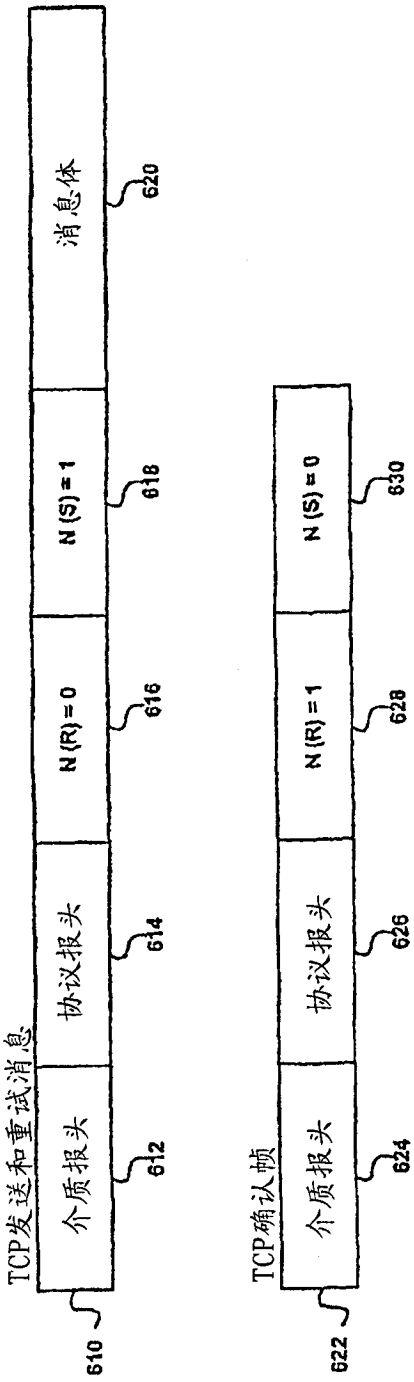

TCP发送和重试消息
介质报头
协议报头
N(R) = 0
N(S) = 1
消息体
610
612
614
616
618
620
TCP确认帧
介质报头
协议报头
N(R) = 1
N(S) = 0
622
624
626
628
630
600

图 6

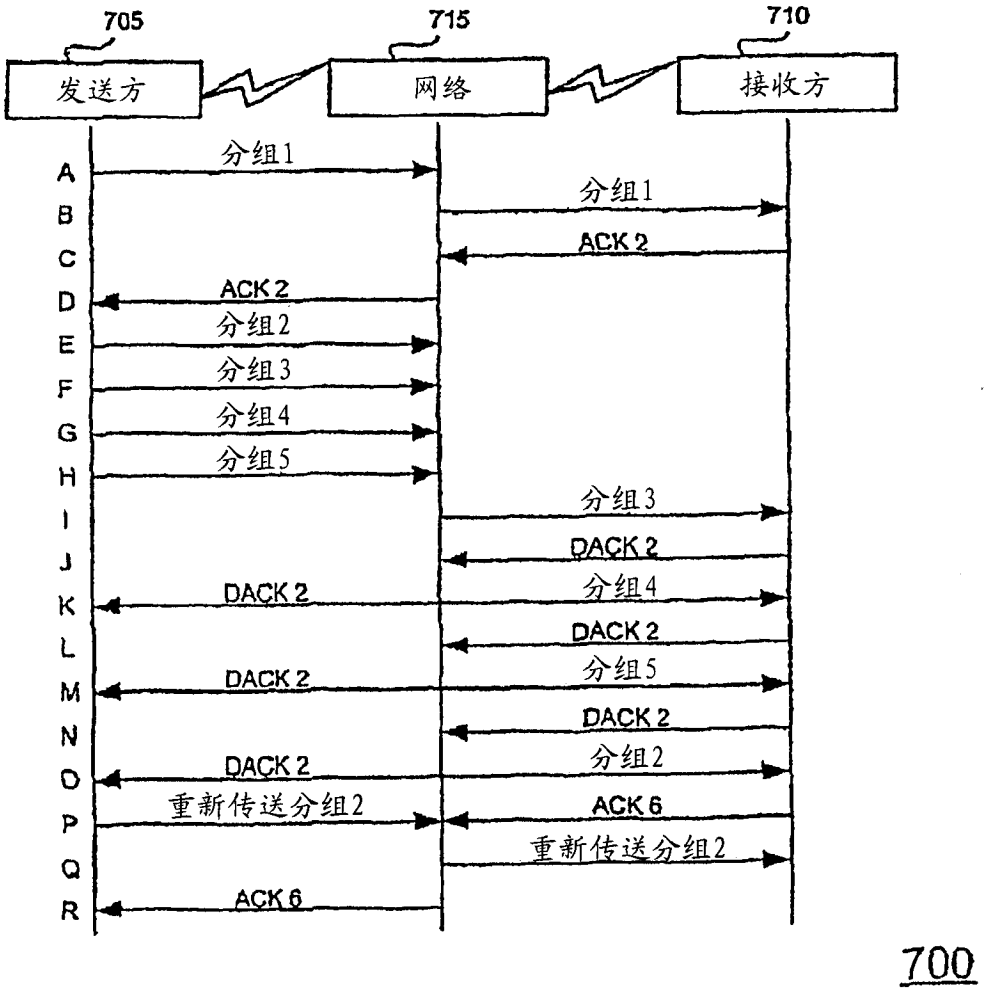

705
715
710
发送方
网络
接收方
A
B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
Q
R
分组1
分组1
ACK 2
ACK 2
分组2
分组3
分组4
分组5
分组3
DACK 2
DACK 2
分组4
DACK 2
DACK 2
分组5
DACK 2
DACK 2
分组2
重新传送分组2
ACK 6
重新传送分组2
ACK 6
700

图 7

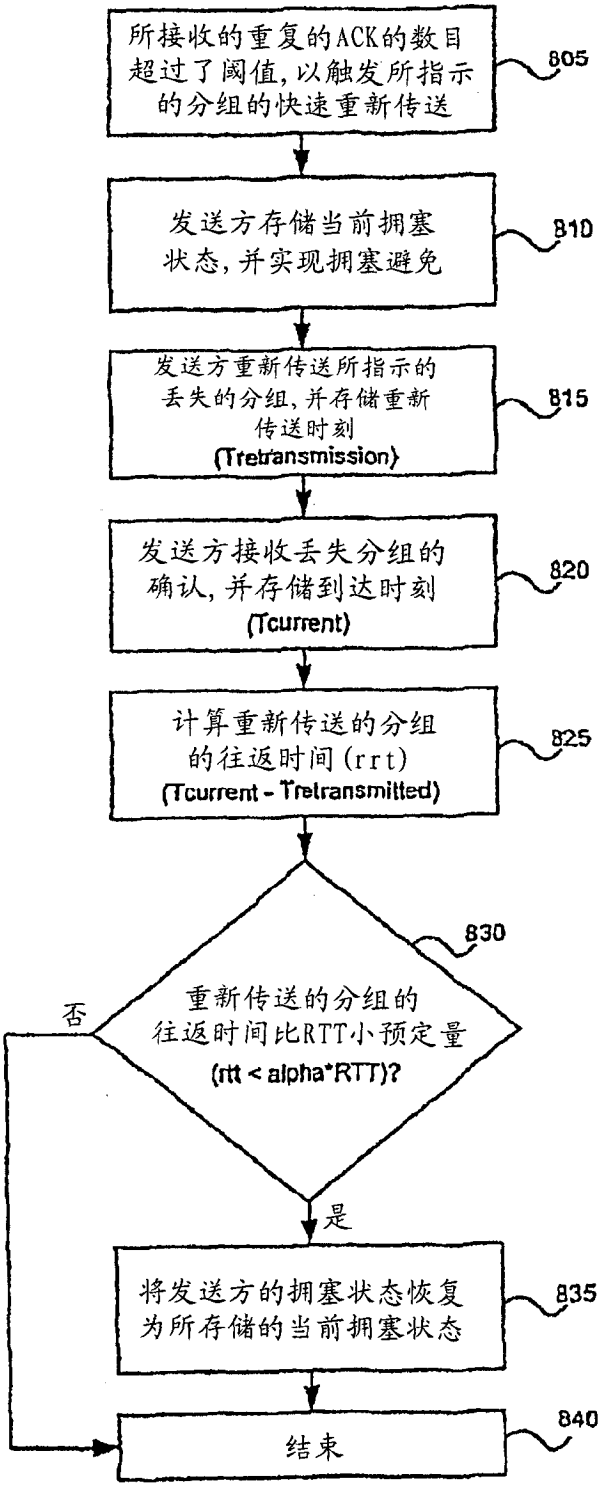

所接收的重复的ACK的数目超过了阈值，以触发所指示的分组的快速重新传送
805
发送方存储当前拥塞状态，并实现拥塞避免
810
发送方重新传送所指示的丢失的分组，并存储重新传送时刻
(Tretransmission)
815
发送方接收丢失分组的确认，并存储到达时刻
(Tcurrent)
820
计算重新传送的分组的往返时间(rrt)
(Tcurrent - Tretransmitted)
825
重新传送的分组的往返时间比RTT小预定量
(rtt < alpha*RTT)?
830
否
是
将发送方的拥塞状态恢复为所存储的当前拥塞状态
835
结束
840
800

图 8