



(12) 发明专利申请

(10) 申请公布号 CN 101816153 A

(43) 申请公布日 2010. 08. 25

(21) 申请号 200880107947. 5

(74) 专利代理机构 北京三友知识产权代理有限公司 11127

(22) 申请日 2008. 09. 09

代理人 李辉

(30) 优先权数据

08003010. 9 2008. 02. 19 EP
60/973, 920 2007. 09. 20 US
61/031, 499 2008. 02. 26 US

(51) Int. Cl.
H04L 12/56(2006. 01)

(85) PCT申请进入国家阶段日
2010. 03. 19

(86) PCT申请的申请数据
PCT/EP2008/061928 2008. 09. 09

(87) PCT申请的公布数据
W02009/037152 EN 2009. 03. 26

(71) 申请人 LM 爱立信电话有限公司
地址 瑞典斯德哥尔摩

(72) 发明人 蒂博尔·拉托尼 马蒂斯·尼尔森
马茨·萨格弗斯 亨宁·维曼

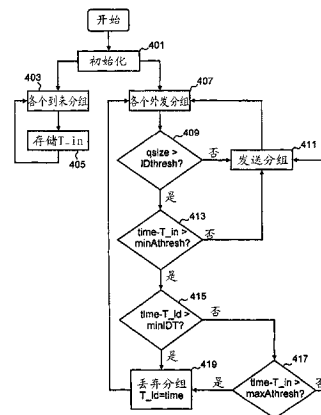
权利要求书 2 页 说明书 7 页 附图 3 页

(54) 发明名称

增强的数据链路利用

(57) 摘要

介绍了通信系统中的数据分组缓冲器的控制。示出了在传输链路中的数据分组队列的控制。在队列中存储了要经由无线接口发送的多个接收到的数据分组。从队列中丢弃所选择的分组，使其不被发送。至少根据分组在队列中存储的时间长度来选择要丢弃的分组。在发生前一次分组丢弃之后至少已经经过了预定时段的情况下，执行所选择的分组的丢弃。这种方案的优点在于，对于传输链路中改变的传输带宽条件，可以实现良好的链路使用和吞吐量，同时使在链路中传输的分组能实现低的端对端延迟。



1. 一种用于控制通信系统 (100) 的传输链路中的数据分组队列的方法, 所述队列存储了要经由无线接口发送的多个接收到的数据分组, 所述方法包括从所述队列中丢弃所选择的分组, 而使所述选择的分组不被发送, 其中, 至少根据分组被存储在所述队列中的时间长度来选择分组, 并且其中在自发生前一次分组丢弃起算至少已经经过了预定时段的情况下, 才执行所选择的分组的丢弃。

2. 根据权利要求 1 所述的方法, 所述方法包括以下步骤:

- 计算自分组被从所述队列中丢弃而不被发送起算的时段,

- 对于所述队列的分组中的至少一个子集的各个分组, 计算所述各个分组被存储在所述队列中的时段,

- 对于所述各个分组, 如果该分组被存储在所述队列中的时段大于第一阈值, 并且自分组被从所述队列中丢弃而不被发送起算的时段大于第二阈值, 则丢弃该分组。

3. 根据权利要求 2 所述的方法, 所述方法还包括以下步骤:

- 对于所述各个分组, 如果该分组被存储在所述队列中的时段大于第三阈值, 则丢弃该分组。

4. 根据权利要求 1 至 3 中任意一项所述的方法, 其中, 在存储在所述队列中的分组的数量小于第四阈值的情况下, 防止丢弃分组。

5. 根据权利要求 2 至 4 中任意一项所述的方法, 所述方法还包括以下步骤:

- 计算作为所述无线接口的特性的往返时间, 并且将至少所计算出的往返时间的分配给所述第一阈值。

6. 根据权利要求 1 至 5 中任意一项所述的方法, 其中, 存储接收到的数据分组包括存储要通过移动通信系统 (100) 的上行链路 (300) 发送的分组。

7. 根据权利要求 1 至 5 中任意一项所述的方法, 其中, 存储接收到的数据分组包括存储要通过移动通信系统 (100) 的下行链路发送的分组。

8. 根据权利要求 2 至 7 中任意一项所述的方法, 其中,

- 所述第一阈值处于 0.25 秒至 0.30 秒的区间中, 和 / 或

- 所述第二阈值处于 0.095 秒至 0.105 秒的区间中, 和 / 或

- 所述第三阈值处于 1.1 秒至 1.3 秒的区间中, 和 / 或

- 所述第四阈值为至少 3 个分组。

9. 一种用于控制与通信系统 (100) 的传输链路相连接的数据分组队列存储器的装置 (201), 所述队列存储器被配置为存储要经由所述通信系统的无线接口发送的多个接收到的数据分组, 所述装置被配置从所述队列存储器中丢弃所选择的分组, 使其不被发送, 其中, 该装置还被配置为使得其能够至少根据分组被存储在所述队列中的时间长度来选择分组, 并且还被配置为使得其能够在自发生了前一次分组丢弃起算已经经过了至少预定时段的情况下, 才丢弃所述选择的分组。

10. 根据权利要求 9 所述的装置, 所述装置被配置成能够:

- 计算自分组被从所述队列中丢弃而不被发送起算的时段,

- 对于所述队列中的分组的至少一个子集的各个分组, 计算该分组被存储在所述队列中的时段,

- 对于所述各个分组, 如果该分组被存储在所述队列中的时段大于第一阈值, 并且自分

组被从所述队列中丢弃而不被发送起算的时段大于第二阈值,则丢弃该分组。

11. 根据权利要求 10 所述的装置,所述装置被配置成能够:

- 对于所述各个分组,如果所述分组被存储在所述队列中的时段大于第三阈值,则丢弃该分组。

12. 根据权利要求 9 至 11 中任意一项所述的装置,所述装置被配置成,当存储在所述队列中的分组的数量小于第四阈值的情况下,防止丢弃分组。

13. 根据权利要求 10 至 12 中任意一项所述的装置,所述装置被配置成能够:

- 计算作为所述无线接口的特性的往返时间,并且将至少所计算出的往返时间的时间分配给所述第一阈值。

14. 根据权利要求 9 至 13 中任意一项所述的装置,所述装置被配置成能够通过移动通信系统 (100) 的上行链路 (300) 发送所述接收到的数据分组。

15. 根据权利要求 9 至 13 中任意一项所述的装置,所述装置被配置成能够通过移动通信系统 (100) 的下行链路发送所述接收到的数据分组。

16. 根据权利要求 10 至 15 中任意一项所述的装置,其中,

- 所述第一阈值处于 0.25 秒至 0.30 秒的区间中,和 / 或
- 所述第二阈值处于 0.095 秒至 0.105 秒的区间中,和 / 或
- 所述第三阈值处于 1.1 秒至 1.3 秒的区间中,和 / 或
- 所述第四阈值为至少 3 个分组。

17. 一种移动通信设备 (206),其包括权利要求 9 至 16 中任意一项所述的装置 (201)。

18. 一种包括软件指令的计算机程序,当在计算机 (210) 中执行所述计算机程序时,所述计算机程序执行权利要求 1 至 8 中任意一项所述的方法。

增强的数据链路利用

技术领域

[0001] 本发明涉及对通信系统中的数据分组缓冲器的控制。

背景技术

[0002] 在基于数据分组的通信系统中,即,在要发送的信息被分成多个分组并且通过通信链路来发送各个分组的系统中,已知在网络的不同点处提供队列缓冲器。缓冲器可以是发送或输入缓冲器(即,用于要通过链路发送的数据分组的缓冲器)、或者接收或输出缓冲器(即,用于已经通过链路发送来的数据分组的缓冲器)。

[0003] 根据特定的语境、使用的特定协议、以及某些其它约定,可以将用于传送数据的分组称作多种名称中的任何一种,诸如,协议数据分组、帧、段、信元等。在本文的语境中,通常将数据的这种的分组称作数据分组。将把数据分组排进队列、使它们在队列中前进、以及将数据分组从队列中除去的过程称作“队列管理”。

[0004] 数据分组传输网络中公知的一种现象是拥塞。拥塞表示无法迅速地处理要求通过该连接或链路进行传输的大量数据分组的状态。作为给定链路处的拥塞结果,与所述链路相关联的队列缓冲器中的数据分组的数量将增加。响应于拥塞情况,公知的是实施一种被称作“满则丢弃”的数据分组丢弃机制。根据该机制,一旦队列缓冲器接收到数据分组,就将队列长度有关参数(诸如,实际队列长度或平均队列长度)与预定的阈值进行比较。如果超过了预定的阈值,则丢弃数据分组。该阈值表示队列的“满”状态。

[0005] 被丢弃的数据分组可以是新到达的分组,在这种情况下该机制被称作“尾部丢弃(tail-drop)”。除了尾部丢弃技术,还公知执行所谓的“随机丢弃”,其中根据随机函数来选择已经在队列中的数据分组,或者所谓的“前端丢弃”,其中将队列中的第一个数据分组丢弃。这种满则丢弃的机制不仅用于减少拥塞链路上的负载,还用于对数据分组的源和/或目的地进行隐式的拥塞通知。

[0006] 这种队列管理将在本文中讨论,并且具体描述的示例将涉及移动宽带环境。当前,进行了第三代(3G)宽带码分多址(WCDMA)技术的升级以为下行信道和上行信道两者提供更高的数据速率。第一阶段主要目标是高达14Mbps的较高的下行速率。其已经在商业网络中实现并被称作高速下行分组接入(HSDPA)。不久将由高速上行分组接入(HSUPA)提供更高的上行速率(高达6Mbps),高速上行分组接入(HSUPA)也被称作增强型上行链路(EUL)。HSDPA和HSUPA的组合被统称为高速分组接入(HSPA)。

[0007] 当前正在进行进一步升级3G系统的标准化工作以提供更高的数据速率并减少传输延迟。这可以通过对仍基于WCDMA的进一步增强HSPA(e-HSPA)来实现。长期演进(LTE)允许使用更宽的频带。这些技术的共同之处是存在由小区中的全部移动终端共享的高速无线链路。由根据网络专用算法来工作的调度器通过网络来控制在该共享信道上的传输。调度器发送信道接入授权给小区中的终端以控制允许谁使用该共享信道。该接入授权处理信令非常快,并且接入授权可以每秒几次在用户之间进行交换。调度器算法、激活终端的数量以及小区中当前的无线电资源情况对于移动终端是未知的。这导致从移动终端观察的无线

链路可以具有大速率的变化,并且在最差的情况下可以每秒几次地从几 Mbps 变化到几百 Kbps。

[0008] 与这些升级所提供的增强的数据速率无关地,无线链路很可能仍然是端对端连接的瓶颈。在无线链路中的无线电情况发生变化以及带宽发生变化时,移动终端中的上行缓冲器的队列长度会随之改变。因此,需要某种缓冲器的管理以实现良好的链路利用和低延迟。最直接的方法是使全部到来数据都被缓冲而不管链路的情况怎样。但是,这种方法具有许多缺点。首先,缓冲器容量物理受限。此外,大的队列存在许多问题,诸如过大的端到端分组延迟、竞争流之间的不公平、对共享同一缓冲器的其它业务造成的延迟、以及网络冲浪的慢反应。为了保持队列长度“适当”,因此需要管理队列缓冲器的方案。

[0009] 如 **Sågfors M.**, Ludwig R., Meyer M., Peisa J.. “Queue Management for TCP Traffic over 3G Links”, IEEE, March 2003, 和 **Sågfors M.**, Ludwig R., Meyer M., Peisa J., “Buffer Management for Rate-Varying 3G Wireless Link Supporting TCP Traffic”, IEEE, April, 2003 中所示,现有解决方案包括用于传统 WCDMA 链路(例如,为每 TCP 流分配的专用信道)的分组丢弃防止计数器算法(PDPC)。

[0010] 此外,国际专利公开 WO-02098153 A1 介绍了一种用于管理缓冲器中数据分组队列的方法。针对分组队列定义了最小阈值等级和最大阈值等级,该方法在阈值超过最大等级时或者当缓冲器队列处于限定的等级之间时,将针对缓冲器接收到的那些数据分组执行拥塞避免过程。并且当缓冲队列小于最小阈值时,不执行拥塞避免过程。

[0011] 但是,这些现有技术的公开的缺点在于,在其上发送缓冲的分组的链路存在利用不足的风险。也就是说,在分组丢弃之后,为了确保充分利用该链路, TCP 管道容量(为充分利用瓶颈接口的带宽时 TCP 流所需要具有的传输中的最小数据量)需要在分组被丢弃时被缓冲。因为该管道容量非常依赖于链路的带宽,因此这种现有技术的解决方案在链路带宽在短时间范围内变动的环境下不是优选的。

发明内容

[0012] 为了改善现有技术,根据本发明的第一方面,提供了一种用于控制通信系统的传输链路中的数据分组队列的方法。在所述队列中存储了要经由无线接口发送的多个接收到的数据分组。该方法包括从队列中丢弃所选择的分组使其不被发送。至少根据分组在队列中存储的时间长度来选择分组。自发生了前一次分组丢弃起算已经经过了至少预定时段的情况下,才执行丢弃所选择的分组。

[0013] 这种方案的优点在于,对于传输链路中改变的传输带宽条件,可以实现良好的链路利用和吞吐量,而同时使在链路中传输的分组能实现低的端对端延迟。这能实现的原因是因为,发现了分组在队列中存储的时间长度在很大程度上与链路中的瞬时数据速率无关,利用被丢弃的分组之间要经过了预定的时段的条件来避免从一个相同发送窗口中丢弃多个分组。

[0014] 实施方式可包括多种操作,包括:计算自分组从所述队列中丢弃而不被发送起算的时段,以及对于队列的分组中的至少一个子集的各个分组,计算分组被存储在队列中的时段,对于所述各个分组,如果分组被存储在队列中的时段大于第一阈值,并且自分组从所述队列中丢弃而不被发送起算的时段大于第二阈值,则丢弃该分组。

[0015] 实施方式还可以包括,对于所述各个分组,如果分组被存储在队列中的时段大于第三阈值,则丢弃该分组。

[0016] 在存储在队列中的分组的数量小于第四阈值的情况下,防止丢弃该分组。

[0017] 实施方式还可包括计算作为无线接口的特性的往返时间,并且分配至少所计算出的往返时间的的时间给所述第一阈值。

[0018] 利用 TCP/IP 环境的术语(并通过实施例进行例示),即,存储了用于经由低层传输链路(包括无线接口)发送至对应的 TCP/IP 接收实体的 TCP/IP 分组的队列,发现了优于现有技术方案的多个优点。通过根据第二阈值丢弃排队的分组,丢弃数据的数量减少,并因此丢弃 TCP 重传的概率也减少,由此来避免 TCP 重传超时(RTO:s)。

[0019] 关于第三阈值的使用(第三阈值为与根据第二阈值设置的任何丢弃条件无关的、可以控制队列中存储的时段超过该第三阈值的分组的丢弃的参数),其可以控制队列中的“陈旧”分组的丢弃。也就是说,通过应用第三阈值,可以为遵循 RTO 输入的 TCP 重传提供短的队列,由此使得经受重复超时的概率低。

[0020] 关于第四阈值的使用(第四阈值为可以在队列短的情况下防止从队列中丢弃分组的参数),优点在于,在低带宽情况下,分组比在更高的带宽的情况下在队列中停留更久。第四阈值的使用防止了丢弃分组,从而避免了空队列的情况,这是有益的,因为空队列会将传输链路的使用等级次优化。

[0021] 此外,存储接收到的数据分组可包括存储要通过移动通信系统的上行链路发送的分组。存储接收到的数据分组包括存储要通过移动通信系统的下行链路发送的分组。

[0022] 实施方式包括,其中,所述第一阈值处于 0.25 秒至 0.30 秒的区间中,所述第二阈值处于 0.095 秒至 0.105 秒的区间中,所述第三阈值处于 1.1 秒至 1.3 秒的区间中,所述第四阈值为至少 3 个分组,例如,处于 5 个分组到 6 个分组的区间中。这种实施方式的优点在于,优化了链路利用和吞吐量,同时使在链路中传输的分组能实现低的端对端延迟。

[0023] 本发明的第二方面,提供了一种用于控制连接到通信系统的传输链路的数据分组队列存储器的装置,所述队列存储器被配置为存储要经由通信系统的无线接口发送的多个接收到的数据分组,所述装置被配置为从队列存储器中丢弃选择的分组使所选择的分组不被发送。该配置为使得该装置至少可以根据存储在队列中的分组的时间长度来选择分组,并且还被配置为使得其可以在自发生了前一次分组丢弃起算已经经过了至少预定时段的情况下,才丢弃所选择的分组。

[0024] 根据第三方面、根据第二方面的装置可包括在移动通信终端中。根据第四方面的计算机程序可包括软件指令,当在计算机中执行时,所述指令执行根据第一方面的方法。如上面结合第一方面所讨论的,其它方面提供了相应的效果和优点。

附图说明

[0025] 下面将参照附图来介绍实施方式,在附图中:

[0026] 图 1 示意性地例示了通信系统,

[0027] 图 2 示意性地例示了移动通信终端,

[0028] 图 3 示意性地例示了上行链路的协议堆栈,以及

[0029] 图 4 是排队算法的流程图。

具体实施方式

[0030] 图 1 示意性地例示了可以在其中实现以上发明内容简述的方法、装置和软件的通用移动通信系统 (UMTS) 网络 100。然而,应注意,技术人员可以很容易在需要数据分组排队的其它类似的通信系统中进行实现。

[0031] 在图 1 中,UMTS 网络 100 包括核心网 102、和 UMTS 陆地无线电接入网 (UTRAN) 103。UTRAN 103 包括多个无线电网络控制器 (RNC) 104,各无线电网络控制器 (RNC) 104 被耦合到一组相邻的基站收发机 (BTS) 105。这些 BTS 有时也被称作节点 B(Node B)。各节点 B 105 负责给定的地理小区,而进行控制的 RNC 104 负责在节点 B 105 与核心网 102 之间路由用户数据和信令数据。全部的 RNC 104 彼此耦合。在“Technical Specification TS 25.401 V3.2.0 of the 3rd Generation Partnership Project”中给出了 UTRAN 103 的概述。

[0032] 图 1 还例示了移动设备或用户设备 (UE) 106a-c(它们经由各自的空中接口 111a-c 连接到 UTRAN 103 中的各个节点 B 105a-b)、服务 GPRS 支持节点 (SGSN) 107、以及 GPRS 网关支持节点 (GGSN) 108。由一个节点 B 服务的移动设备(诸如由节点 B 105a 服务的设备 106a 和 106b) 位于所谓的无线电小区中。通过将 GGSN 108 耦合到互联网 109(其中,示意性地,服务器 110 例示了移动设备 106 可以通信的实体),SGSN 107 和 GGSN 108 经由 UTRAN 103 为 UE 106 提供分组交换数据业务。如技术人员所意识到的,图 1 中的网络 100 可在核心网 102 和 UTRAN 103 中包括大量类似的功能单元,并且在网络的一般实现中,移动设备的数量可以很大。

[0033] 图 2 示意性地例示了移动通信设备 206,其对应于图 1 中的任一个通信设备 106。与图 1 介绍的设备 106 相比更详细示出的通信设备 206 包括处理器 210、存储器 211 以及经由输入/输出接口单元 214 连接到处理器 210 和存储器 211 的、形式为麦克风 217、扬声器 216、显示器 218 和键盘 215 的输入/输出单元。通过无线电电路 212 和天线 213 来实现经由空中接口 222 的无线电通信。处理器 210 使用存储在存储器 211 中的软件指令以控制终端 206 的全部功能,包括下面要详细介绍的与数据分组的排队有关的功能。换言之,至少处理器和存储器形成了被配置为如以上发明内容部分简述的并且下面具体介绍的那样控制数据分组队列的装置 201 的一部分。关于这些单元如何工作以执行 UMTS 网络(诸如图 1 的网络 100) 内的普通功能的进一步细节对于技术人员来说是公知的,因此不再进一步讨论。

[0034] 图 3 示意性地例示了通信网(诸如图 1 的网络 100) 中的上行链路 300。图 3 示出了用移动设备(诸如图 1 的设备 106) 中的协议堆栈 306 以及网络(诸如图 1 中的网络 100) 中的协议堆栈 330 表示的上行链路 300。移动设备中的一个或几个应用 321 与驻留在服务器(诸如图 1 中的服务器 110) 中的一个或几个应用 331 进行通信。应用 321 发送使用任何合适的应用协议(包括诸如 HTTP、SMTP、SIP、FTP 等) 封装在 TCP 层 322 的 TCP 段中的以及 IP 层 323 的 IP 分组中的数据。然后,这些数据分组经由网络层 324 和数据链路层 325 被进一步封装,并通过物理无线电接口 311 被发送至网络中的接收堆栈 330,在接收堆栈 330 中,按照相反的顺序,在数据链路层 335、网络层 334、IP 层 333、TCP 层 332 中提取数据,最后的应用 331 中被提取为应用协议数据而终止。

[0035] 可以在 UMTS 网络(诸如图 1 的网络 100) 中实现图 3 中的示例堆栈。在这种情况下

下,网络层 324、334 和数据链路层 325、335 分别是无线电资源控制 (RRC) 层和介质访问控制 (MAC) 层。

[0036] 如上面所指示的,当前正在对 UMTS 网络(诸如图 1 的网络 100)进行升级以提供甚至更高的数据速率,并降低发送延迟。这可以通过进一步增强 HSPA(e-HSPA)来实现,而长期演进(LTE)将允许使用更宽的频段。这些技术共同之处是存在由小区中的全部移动终端(即,图 1 的示例性网络 100 中的设备 106a 和 106b)共享的高速无线链路。由根据网络专用算法来工作的、优选地位于 RNC 104 处的、执行调度功能的调度器来从网络 100 控制在该共享信道上的传输。该调度器发送信道接入授权给小区中的移动终端以控制允许谁使用该共享信道。该接入授权处理信令非常快,并且接入授权可以每秒几次在终端之间进行交换。调度器算法、激活移动终端的数量以及小区中当前的无线电资源情况对于移动终端是未知的。这导致从移动终端观察的无线链路可以具有大速率的变化,并且在最差的情况下可以每秒几次从几 Mbps 变化到几百 Kbps。

[0037] 还是参照图 2,图 2 更具体地例示了与图 1 中的移动终端 106 中的任一个对应的移动终端 206,处理器 210 通过将存储器 211 的一部分用作队列存储器来执行数据分组的排队。就图 3 例示的协议堆栈而言,可以在 TCP 层 322 与物理链路 311 之间的任何层处执行分组的排队。例如,可以实施排队使得在将 IP 分组进一步向上发送给 RRC 层 324 之前,在 IP 层 323 处对 IP 分组进行排队。

[0038] 现在参照图 4,将以在 UMTS 系统的上行链路上处理 TCP 分组的算法来介绍这种排队。通过流程图例示了该算法,技术人员通过提供适当配置的软件指令给移动终端(诸如图 2 的终端 206)的存储器电路可以实现该流程图。

[0039] 本算法定义了 4 个参数。第一阈值参数(表示为 minAgeThreshold)定义了允许分组在队列中停留的时间。如果外发分组超过了该时间,则该分组被丢弃。为了避免多次连续的丢弃,第二阈值参数被定义为丢弃分离参数(表示为 minInterDropTime),其防止本算法在经过特定时间之前进行第二次丢弃。但是,在带宽改变的情况下,当带宽被减少过多时(即,大的缩减转换(down-switch)),缓冲器有可能大大地超长(over-dimensioned)。因此,独立于 minInterDropTime 的第三阈值参数(表示为 maxAgeThreshold)被定义为在上述情况下允许突发丢弃。最后,第四阈值参数(表示为 lowerDropThreshold)防止该算法在该缓冲器包括少于一定数量的分组的情况下丢弃分组。这使得缓冲器始终包含足够数量的分组来触发 TCP 快速重发。

[0040] 算法起始于初始化步骤 401,其中全部所需的参数(即,四个阈值参数以及内部计数器等)都被初始化。下面将结合测试运行的结果来讨论这些参数的特定值。

[0041] 对于每个到来的数据分组,存储到达时间,即,存储到队列的时间。这通过步骤 403 和 405 的循环来例示。与构成针对要在上行链路上发送的、排队的外发数据分组的控制循环 407-419 的步骤无关地继续该时间记录循环。

[0042] 如块 407 所例示的,(并非一定是串行地也可以是并行地)考虑每个外发分组。得到队列的当前长度,并且在比较步骤 409 中,将其与第四阈值参数(lowerDropThreshold)进行比较,如果发现该队列的当前长度不大于 lowerDropThreshold,则在发送步骤 411 中发送该分组,否则,本算法获得表示当前时间的值并继续,在比较步骤 413 中将该值与所考虑的分组被存入队列的时间进行比较。如果当前时间与存入队列的时间之间的差不大于第

一阈值参数 (minAgeThreshold), 则在发送步骤 411 中发送该分组。

[0043] 在发送步骤 411 中从队列中发送了分组之后, 处理下一个排队的分组 (如步骤 411 与步骤 407 之间的返回箭头所例示的)。

[0044] 另一方面, 如果在步骤 413 中确定为当前时间与存入队列的时间之间的差大于第一阈值参数, 则算法前进到步骤 415, 其中, 将当前时间与发生最近一次分组丢弃的时间进行比较。如果当前时间与发生最近一次分组丢弃的时间之间的差不大于第二阈值参数 (minInterDropTime), 则算法进入比较步骤 417。

[0045] 另一方面, 如果在步骤 415 中确定为当前时间与发生最近一次分组丢弃的时间之间的差大于第二阈值参数, 则算法前进到丢弃步骤 419, 其中, 将所考虑的分组从队列中丢弃, 并且利用当前时间的值来更新保持代表发生最近一次分组丢弃的时间的值的参数。

[0046] 在比较步骤 417 中, 将表示当前时间的值与所考虑的分组被存入队列的时间进行比较, 如果当前时间与存入队列的时间之间的差不大于第三阈值参数 (maxAgeThreshold), 则在发送步骤 411 中发送分组。

[0047] 否则, 如果当前时间与存入队列的时间之间的差大于第三阈值参数 (maxAgeThreshold), 则在丢弃步骤 419 中丢弃分组。

[0048] 在丢弃步骤 419 中从队列中丢弃了分组之后, 处理下一个排队的分组 (如步骤 419 与步骤 407 之间的返回箭头所例示的)。

[0049] 该算法还可以用伪码来如下表示:

[0050] 对于各外发分组

[0051] 如果 (长度 \leq lowerDropThreshold)

[0052] 发送分组

[0053] 否则

[0054] 如果 ((延迟 $>$ minAgeThreshold 并且

[0055] 当前时间 - 上次进行丢弃的时间 $>$ minInterDropTime))

[0056] 或者 (delay $>$ maxAgeThreshold))

[0057] 丢弃分组

[0058] previousDropTime = 当前时间

[0059] 否则

[0060] 发送分组

[0061] 通过测试获得并分析这些参数的最优值。已发现用于固定带宽场景的 minAgeThreshold 的最优值为 0.25 秒 - 0.3 秒。但是, 对于低带宽等级, 这些值导致链路利用不足。因此, 优选的将 lowerDropThreshold 设置为至少 3 个分组, 例如在 5-6 个分组的区间中, 以防止本算法耗尽队列而减少吞吐量。

[0062] 因为不同的 TCP 版本对于突发丢弃的反应不同, 因此, 无法确定用于 minInterDropTime 的通用最优值。利用 TCP SACK 进行了测试, 并且测试表明, 需要进行的在多流的场景中是保持吞吐量还是优化性能的决定是最关键的。测试表明, 将 minInterDropTime 设置为 0.7 秒的值使得重传没有丢弃, 提高了吞吐量。然而, 对于多流的场景, 其有可能从非主导的流中出现丢弃。因而, 0.7 秒的值将导致极大地超长的缓冲器, 导致延迟增加以及非主导的流挨饿 (starvation)。为了避免这种情况, 应当按照 100ms 的级

别来设置该值。该测试还表明,将 maxAgeThreshold 设置为 1.2 秒的值减少了大的带宽缩减转换的延迟。利用这些设置,针对 UMTS EUL 范围中的全部带宽等级实现了端到端延迟的减少而没有减少吞吐量。

[0063] 针对不同的固定带宽等级,对上述算法与消极的前端丢弃缓冲器进行了比较。这些测试表明,上述算法针对较低的带宽等级,将平均端到端延迟提高了高达 10 倍。可以看出,要在吞吐量以及端到端延迟方面实现与上面描述的算法相同的性能,该消极缓冲方案必须针对全部带宽等级以及互联网延迟进行配置。另一方面,上述算法被配置为其优选的参数设置而不管互联网延迟和带宽等级如何,从而实现了几乎不要求配置的优点。

[0064] 在带宽变化的场景中进行了其它的测试,其中,对上述算法与消极的前端丢弃缓冲器之间进行了比较。从前端丢弃的方法显示出了很大的端到端延迟,而上述算法始终保持延迟为小于 1 秒。对于具有较小变动的场景,与从前端丢弃的方法相比,上述算法的吞吐量减少小得多(最差情况 15%的级别),但是平均端到端延迟改善了至少两倍。此外,这些测试表明,上述算法在 TCP 段往返时间(RTT)的分布方面有较强的优点。已发现实际 RTT 几乎为 1 秒,而从前端丢弃的方法示出 RTT 高达 15 秒。在这些测试中清楚地看到了吞吐量与端到端延迟之间的平衡(trade-off)。

[0065] 在进一步的测试中,针对具有多个流的场景(例如,移动终端中的几个应用需要在上行链路上发送数据的情况)进行了调查。该结果示出了,与消极方法相比,上述算法在多个流的场景中明显胜出。上述算法成功地防止了在消极缓冲器管理情况下可以看到的、将非主导流死锁(lock-out)的行为。此外,从消极前端丢弃方法中发现了与 TCP 基础机制初始慢启动相矛盾的情况。在许多情况下,TCP SYN 段超过了初始重传计时器,并且由于缓冲器溢出而导致队列超时。这导致这样一个事实,初始的慢启动阶段被省略,并且 TCP 流起始于拥塞避免阶段。已发现上述算法能避免这种现象。

[0066] 应注意,虽然上述实施方式的描述主要涉及移动通信系统中的上行链路,但是技术人员能容易地将这些教导应用于通信系统中的下行链路。也就是说,所介绍的实施方式同样地适用于当前 WCDMA 系统的增强上行链路(EUL)和高速下行分组接入(HSDPA)以及长期演进(LTE)系统中的上行链路和下行链路。

[0067] 因此,总之,本文介绍了对通信系统中的数据分组缓冲器的控制。示出了对传输链路中数据分组队列的控制。在该队列中,存储了要通过无线接口发送的多个接收到的数据分组。从该队列中将选择的分组丢弃,使所选择的分组不被发送。至少根据分组在队列中存储的时间长度来选择要丢弃的分组。在前一次分组丢弃发生起已经经过了至少预定时段的情况下,才执行所选择的分组的丢弃。这种方案的优点在于,对于传输链路中传输带宽条件的改变,可以实现良好的链路使用和吞吐量,同时使在链路中传输的分组能实现低的端对端延迟。

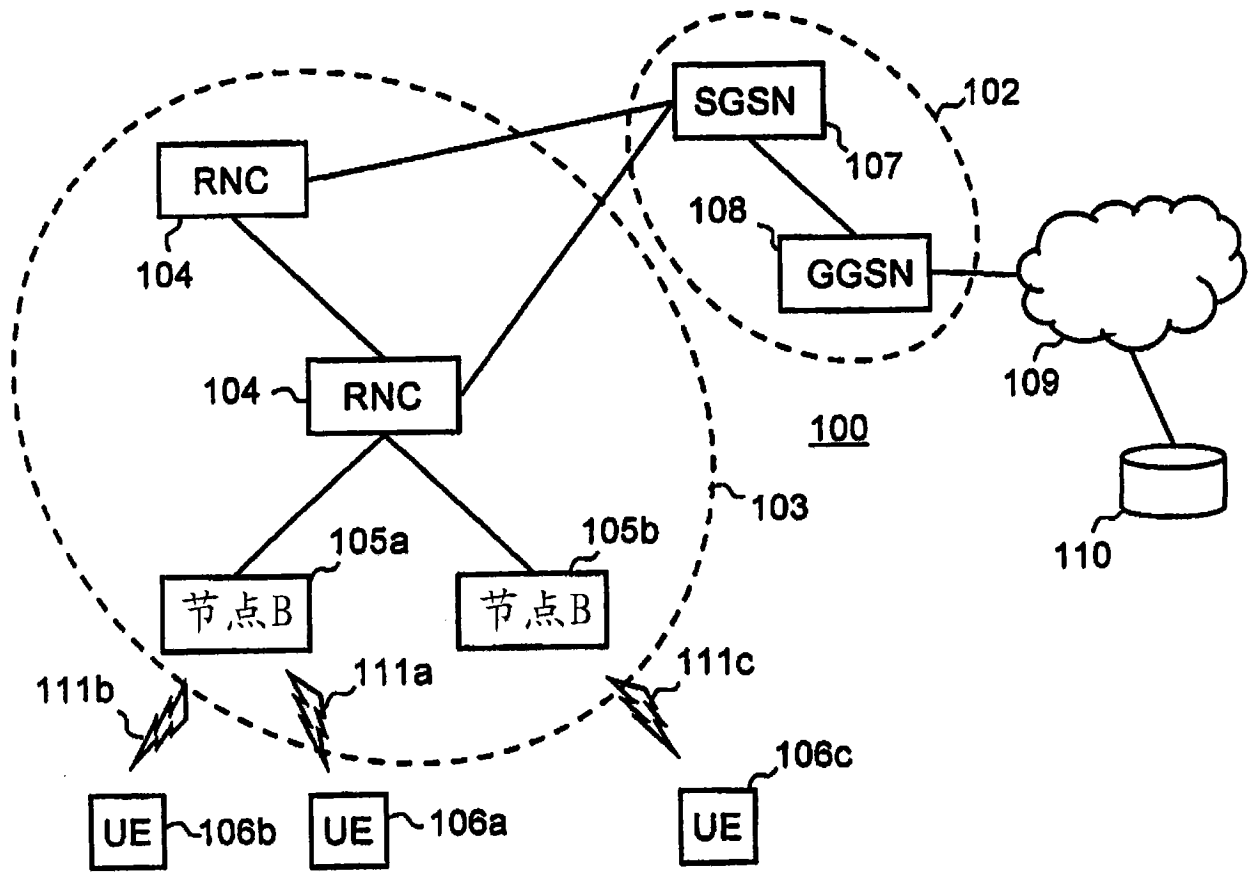


图 1

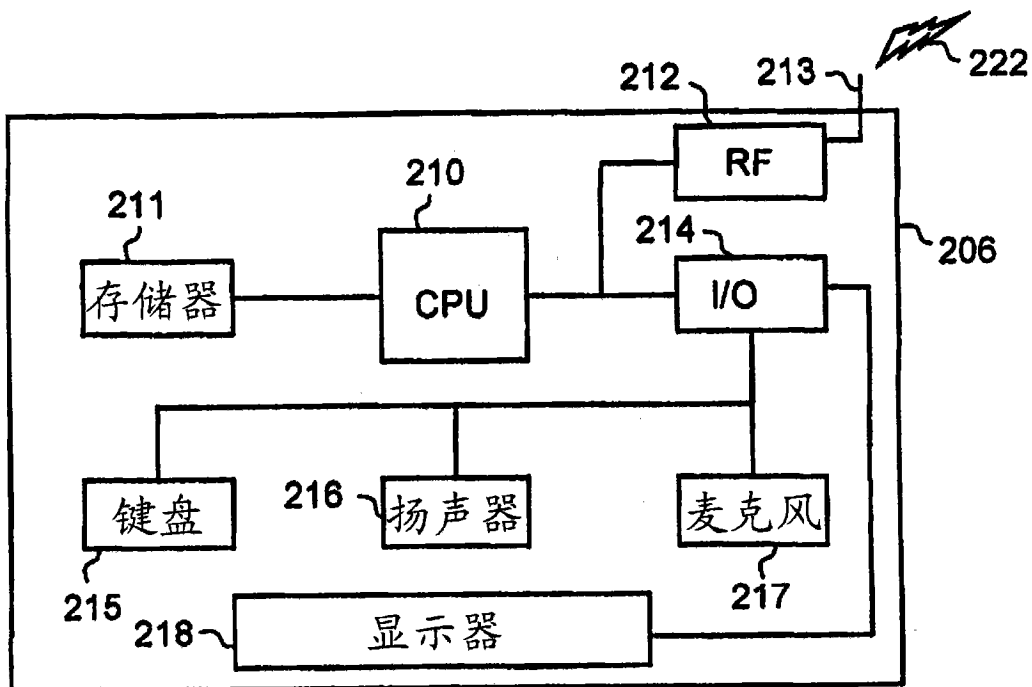


图 2

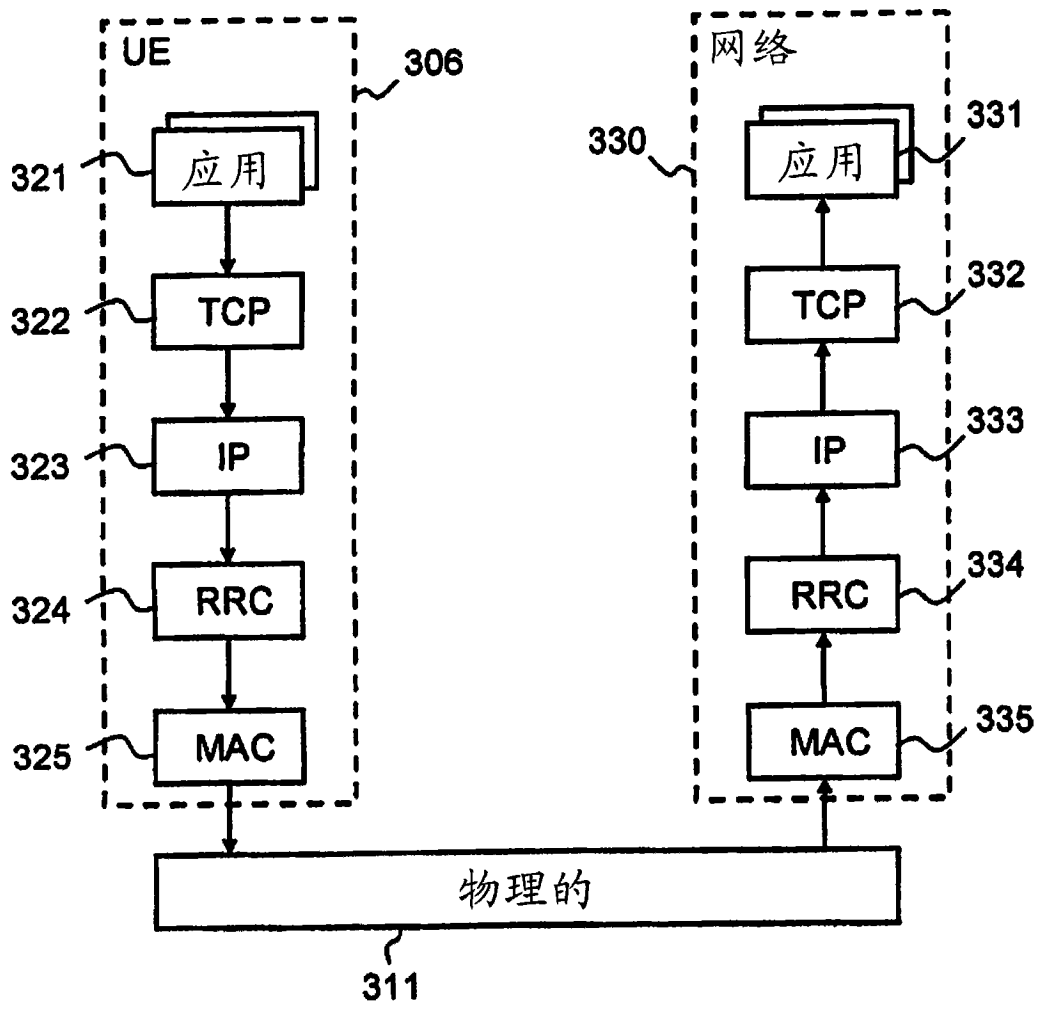


图 3

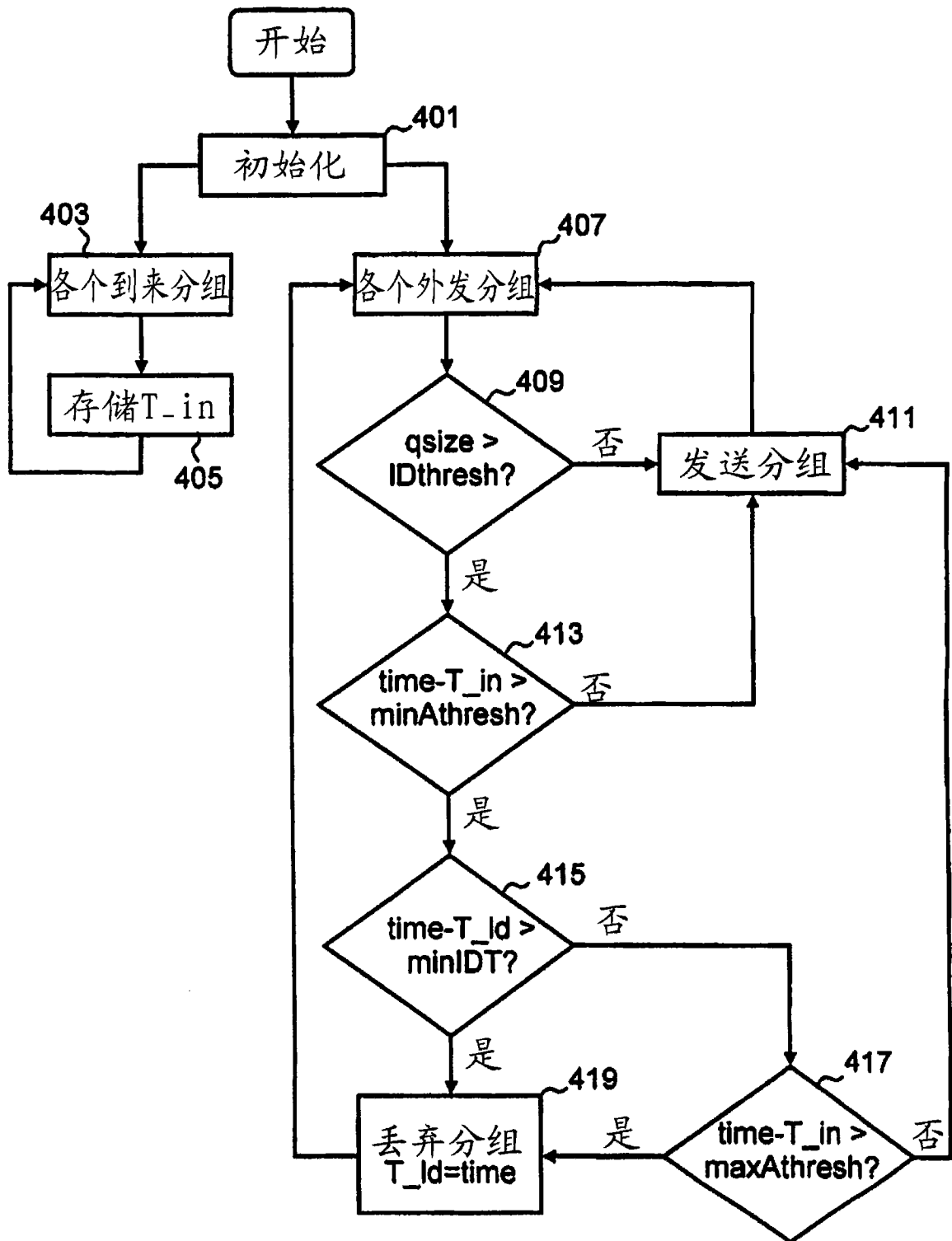


图 4