



(12) 发明专利

(10) 授权公告号 CN 101292475 B

(45) 授权公告日 2012. 05. 02

(21) 申请号 200680039015. 2

(22) 申请日 2006. 10. 05

(30) 优先权数据

11/256, 259 2005. 10. 21 US

(85) PCT申请进入国家阶段日

2008. 04. 18

(86) PCT申请的申请数据

PCT/EP2006/067094 2006. 10. 05

(87) PCT申请的公布数据

W02007/045561 EN 2007. 04. 26

(73) 专利权人 国际商业机器公司

地址 美国纽约

(72) 发明人 S·埃利奥特 C·V·拉扎罗

T·特兰

(74) 专利代理机构 北京市中咨律师事务所

11247

代理人 于静 杨晓光

(51) Int. Cl.

H04L 12/56 (2006. 01)

(56) 对比文件

WO 02052768 A2, 2002. 07. 04, 说明书第 8-13, 21-27, 48 段、图 2, 5, 7A, 7B.

WO 02063891 A1, 2002. 08. 15, 全文.

CN 1588827 A, 2005. 03. 02, 全文.

CN 1529471 A, 2004. 09. 15, 全文.

WO 02052768 A2, 2002. 07. 04, 说明书第 8-13, 21-27, 48 段、图 2, 5, 7A, 7B.

CN 1498461 A, 2004. 05. 19, 全文.

BRAKMO L S ET AL. TCP TEGAS: NEW TECHNIQUES FOR CONGESTION DETECTION AND AVOIDANCE. 《computer communication review》. 1994, 第 24 卷 (第 4 期), 24-35.

审查员 罗芳洁

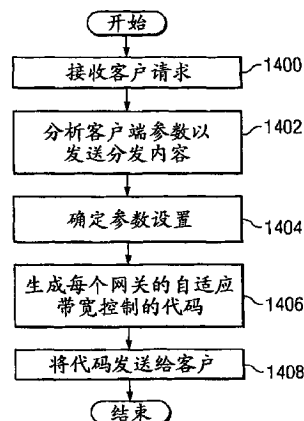
权利要求书 2 页 说明书 15 页 附图 8 页

(54) 发明名称

自适应带宽控制的方法和装置

(57) 摘要

一种计算机实现的方法、装置和计算机可用程序代码,用于在多个网关(304、306、308)处接收来自源(302)的数据以使用选定优先级进行分发。使用所述选定优先级将所述数据从所述多个网关传输到多个接收器(310、320、330)。所述多个网关中的每个网关都具有自适应带宽控制过程和用于控制所述自适应带宽控制过程的相应参数组以便以所述选定优先级发送数据。当为所述多个网关中的不同网关的参数组设置了不同的值时,从每个网关进行的具有所述选定优先级的数据传输对所述不同网关处的具有所述选定优先级的其他业务具有不同的影响。



1. 一种用于自适应地控制数据传输的计算机实现的方法,所述计算机实现的方法包括:

为多个网关中的特定网关配置参数组,其中所述特定网关的参数组不同于所述多个网关中的其他网关的参数组;

在多个网关处接收来自源的数据以使用选定优先级进行分发;以及

使用所述选定优先级将所述数据从所述多个网关传输到多个接收器,其中所述多个网关中的每个网关都具有自适应带宽控制过程和用于控制所述自适应带宽控制过程的相应参数组以便以所述选定优先级发送数据,其中所述自适应带宽控制过程考虑了从所述多个网关中的一个网关到所述分发的接收器的路径中的拥塞,并且其中当为所述多个网关中的不同网关的参数组设置了不同的值时,从每个网关进行的具有所述选定优先级的数据传输使用的带宽不同于所述多个网关中的不同网关进行的具有所述选定优先级的数据传输使用的带宽。

2. 根据权利要求1所述的计算机实现的方法,其中所述参数组包括阈值、 β 参数和最大等待时间。

3. 根据权利要求1所述的计算机实现的方法,其中所述数据包括数据文件、应用更新以及病毒修补程序之一。

4. 一种用于自适应地控制数据传输的计算机实现的装置,所述计算机实现的装置包括:

用于为多个网关中的特定网关配置参数组的模块,其中所述特定网关的参数组不同于所述多个网关中的其他网关的参数组;

用于在多个网关处接收来自源的数据以使用选定优先级进行分发的模块;以及

用于使用所述选定优先级将所述数据从所述多个网关传输到多个接收器的模块,其中所述多个网关中的每个网关都具有自适应带宽控制过程和用于控制所述自适应带宽控制过程的相应参数组以便以所述选定优先级发送数据,其中所述自适应带宽控制过程考虑了从所述多个网关中的一个网关到所述分发的接收器的路径中的拥塞,并且其中当为所述多个网关中的不同网关的参数组设置了不同的值时,从每个网关进行的具有所述选定优先级的数据传输使用的带宽不同于所述多个网关中的不同网关进行的具有所述选定优先级的数据传输使用的带宽。

5. 根据权利要求4所述的计算机实现的装置,其中所述参数组包括阈值、 β 参数和最大等待时间。

6. 根据权利要求4所述的计算机实现的装置,其中所述数据包括数据文件、应用更新以及病毒修补程序之一。

7. 一种用于向客户端提供定制的数据传输的方法,所述方法包括:

从客户端接收对定制的数据传输的请求,其中所述请求包括网络特征和网关标识;

生成定制参数组,其中根据所述网关标识和所述网络特征将每个参数组分配给网关;

将所述定制参数组发送给所述客户端以与计算机可用程序代码一起使用,以便执行以下操作:在多个网关处接收来自源的数据以使用选定优先级进行分发;以及使用所述选定优先级将所述数据从所述多个网关传输到多个接收器,其中所述多个网关中的每个网关都具有自适应带宽控制过程和用于控制所述自适应带宽控制过程的相应参数组以便以所述

选定优先级发送数据,并且其中当为所述多个网关中的不同网关的参数组设置了不同的值时,从每个网关进行的具有所述选定优先级的数据传输使用的带宽不同于所述多个网关中的不同网关进行的具有所述选定优先级的数据传输使用的带宽。

8. 根据权利要求7所述的方法,其中所述发送步骤进一步包括与所述定制参数组一起发送所述计算机可用程序代码。

自适应带宽控制的方法和装置

[0001] 相关申请的交叉引用

[0002] 本发明与以下同日申请的受让给同一受让人的专利申请相关：“Method and Apparatus for Adaptive Bandwidth Control with User Settings”，序号为 11/256261，代理人案号 AUS920050797US1；以及“Method and Apparatus for Adaptive Bandwidth Control with a Bandwidth Guarantee”，序号为 11/256260，代理人案号 AUS920050798US1；以上申请在此引入作为参考。

[0003] 技术领域

[0004] 本发明一般地涉及改进的数据处理系统。具体地说，本发明涉及用于传输数据的计算机实现的方法和装置。更具体地说，本发明涉及自适应地控制用于传输数据的带宽的计算机实现的方法、装置和计算机可用程序代码。

[0005] 背景技术

[0006] 随着因特网和网络的普遍使用，商业通信发生了革命性的变化。网络通常被用于传输数据。许多分布式应用利用大型后台传输来改进服务质量。使用这些类型的后台传输，用户无需等待这些传输完成即可执行其他操作。包括诸如数据备份、预取、企业数据分发、因特网内容分发以及对等存储之类的大量应用和服务都采用后台传输。这些和其他类型的应用增加了网络带宽消耗。这些服务中的某些服务潜在地具有无限的带宽需求，其中使用不断增加的带宽可以逐渐提供更佳的服务。

[0007] 这些类型的应用所具有的一个问题是多数网络只有有限的带宽可用于传输数据。有些应用执行关键性功能，而其他应用则执行非关键性功能。通常，后台传输是非关键性的并且可以使用所有可用的带宽，这使得关键性的网络业务变慢。

[0008] 自适应带宽控制已被用于自动适应网络状况以降低对网络的影响。目前，已采用了不同的自适应带宽控制过程和算法来控制不同应用所使用的带宽量以避免拥塞。当前可用的自适应带宽控制过程考虑了服务器或客户端计算机上的本地接口级别处的网络状况，但是无法考虑数据传输中可能存在的其他状况。

[0009] 因此，具有一种自适应地控制数据传输中的带宽使用的改进的计算机实现的方法、装置和计算机可用程序代码是有利的。

发明内容

[0010] 本发明提供了一种计算机实现的方法、装置和计算机可用程序代码，用于在多个网关处接收来自源的数据以使用选定优先级进行分发。使用所述选定优先级将所述数据从所述多个网关传输到多个接收器。所述多个网关中的每个网关都具有自适应带宽控制过程和用于控制所述自适应带宽控制过程的相应参数组以便以所述选定优先级发送数据。当为所述多个网关中的不同网关的参数组设置了不同的值时，从每个网关进行的具有所述选定优先级的数据传输对所述不同网关处的具有所述选定优先级的其他业务具有不同的影响。

[0011] 所述参数组可以包括阈值、 β 参数和最大等待时间。可以为所述多个网关中的特定网关配置所述参数组，其中所述特定网关的所述参数组可以不同于所述多个网关中的其

他网关的参数组。所述自适应带宽控制过程考虑了从所述多个网关中的一个网关到所述分发的接收器的路径中的拥塞。此外,所述相应参数组对于所述多个网关中的不同网关可以是不同的。在示例性实例中,所述数据包括数据文件、应用更新以及病毒修补程序之一。

附图说明

[0012] 现在仅通过实例的方式参考附图描述本发明的一个或多个实施例,这些附图是:

[0013] 图 1 是其中可实现本发明的各方面的数据处理系统网络的图形表示;

[0014] 图 2 是其中可实现本发明的各方面的数据处理系统的方块图;

[0015] 图 3 是示出根据本发明的示例性实施例的其中可以使用不同的优先级进行分发的实例网络数据处理系统的示意图;

[0016] 图 4 是示出根据本发明的示例性实施例的自适应带宽控制中使用的组件的示意图;

[0017] 图 5 是根据本发明的示例性实施例的传输控制协议 / 网际协议 (TCP/IP) 以及类似协议的示意图;

[0018] 图 6 是示出根据本发明的示例性实施例在提供自适应带宽控制的过程中使用的软件过程和组件的示意图;

[0019] 图 7 是示出根据本发明的示例性实施例的用于使用优先级将分发内容从自适应发送器发送到接收器的网络的示意图;

[0020] 图 8 是示出根据本发明的示例性实施例的网络中带宽使用的示意图;

[0021] 图 9 是根据本发明的示例性实施例的配置参数的用户设置的过程的流程图;

[0022] 图 10 是根据本发明的示例性实施例的发送线程的过程的流程图;

[0023] 图 11 是根据本发明的示例性实施例的分组嗅探线程的过程的流程图;

[0024] 图 12 是根据本发明的示例性实施例的拥塞控制线程的过程的流程图;

[0025] 图 13 是根据本发明的示例性实施例的执行自适应带宽控制的过程的流程图;以及

[0026] 图 14 是根据本发明的示例性实施例的定制将分发内容发送到客户的接收器的过程的流程图。

具体实施方式

[0027] 图 1-2 是其中可实现本发明的实施例的数据处理环境的示意图。应该理解,图 1-2 只是示例性的,并非旨在断言或暗示对其中可实现本发明的各方面或实施例的环境的任何限制。可以在不偏离本发明的精神和范围的情况下对所示环境做出许多修改。

[0028] 现在参考附图,图 1 示出了其中可实现本发明的各方面的数据处理系统网络的图形表示。网络数据处理系统 100 是其中可实现本发明的实施例的计算机网络。网络数据处理系统 100 包含网络 102,网络 102 是用于提供网络数据处理系统 100 内相互连接的各种设备与计算机之间的通信链路的介质。网络 102 可以包括诸如有线、无线通信链路或光缆之类的连接。

[0029] 在所示实例中,服务器 104 和服务器 106 连同存储单元 108 一起与网络 102 相连。此外,客户端 110、112 和 114 与网络 102 相连。这些客户端 110、112 和 114 可以例如是个

人计算机或网络计算机。在所示实例中,服务器 104 向客户端 110、112 和 114 提供诸如引导文件、操作系统映像,以及应用之类的数据。在此实例中,客户端 110、112 和 114 是服务器 104 的客户端。网络数据处理系统 100 可以包括其他服务器、客户端以及其他未示出的设备。

[0030] 在所示实例中,网络数据处理系统 100 是因特网,同时网络 102 代表全球范围内使用传输控制协议 / 网际协议 (TCP/IP) 协议集来相互通信的网络和网关的集合。在因特网的核心是主节点或主机之间的高速数据通信线路的主干,它包括数以千计的商业、政府、教育以及其他路由数据和消息的计算机系统。当然,网络数据处理系统 100 也可以被实现为许多不同类型的网络,诸如例如企业内部互联网、局域网 (LAN) 或广域网 (WAN)。图 1 旨在作为一个实例,并非旨在作为对本发明的不同实施例的体系结构限制。

[0031] 现在参考图 2,它是其中可实现本发明的各方面的数据处理系统的方块图。数据处理系统 200 是计算机 (如图 1 中的服务器 104 或客户端 110) 的实例,在所述计算机中可以找到实现本发明的实施例的过程的计算机可用代码或指令。

[0032] 在所示实例中,数据处理系统 200 采用包括北桥及存储器控制器中心 (MCH) 202 和南桥及输入 / 输出 (I/O) 控制器中心 (ICH) 204 的中心架构。处理单元 206、主存储器 208 以及图形处理器 210 与北桥及存储器控制器中心 202 相连。图形处理器 210 可以通过加速图形端口 (AGP) 与北桥及存储器控制器中心 202 相连。

[0033] 在所示实例中,局域网 (LAN) 适配器 212 与南桥及 I/O 控制器中心 (ICH) 204 相连。音频适配器 216、键盘和鼠标适配器 220、调制解调器 222、只读存储器 (ROM) 224、硬盘驱动器 (HDD) 226、光盘驱动器 230、通用串口总线 (USB) 端口以及其他通信端口 232,以及 PCI/PCIe 设备 234 通过总线 238 和总线 240 与南桥及 I/O 控制器中心 (ICH) 204 相连。PCI/PCIe 设备可以例如包括用于笔记本计算机的以太网适配器、插入卡和 PC 卡。PCI 使用卡总线控制器,而 PCIe 则不使用。ROM 224 可以例如是闪速二进制输入 / 输出系统 (BIOS)。

[0034] 硬盘驱动器 226 和光盘驱动器 230 通过总线 240 与南桥及 I/O 控制器中心 204 相连。硬盘驱动器 226 和光盘驱动器 230 可以例如使用集成驱动电子设备 (IDE) 或串行高级技术连接 (SATA) 接口。超级 I/O (SIO) 设备 236 可以与南桥及 I/O 控制器中心 204 相连。操作系统在处理单元 206 上运行并对图 2 中的数据处理系统 200 内的各种组件进行协调和控制。作为客户端,操作系统可以是诸如 **Microsoft® Windows®** XP (Microsoft 和 Windows 是 Microsoft Corporation 在美国和 / 或其他国家 / 地区的商标) 之类的商用操作系统。诸如 Java™ 编程系统之类的面向对象的编程系统可以与操作系统一起运行并从数据处理系统 200 上执行的 Java 程序或应用提供对操作系统的调用 (Java 是 Sun Microsystems, Inc. 在美国和 / 或其他国家 / 地区的商标)。

[0035] 作为服务器,数据处理系统 200 可以例如是运行高级交互执行 (AIX) 操作系统或 LINUX 操作系统的 IBM eServer pSeries 计算机系统 (eServer、pSeries 和 AIX 是国际商业机器公司在美国和 / 或其他国家 / 地区的商标,而 Linux 是 Linus Torvalds 在美国和 / 或其他国家 / 地区的商标)。数据处理系统 200 可以是在处理单元 206 中包含多个处理器的对称多处理器 (SMP) 系统。备选地,可以采用单处理器系统。

[0036] 操作系统、面向对象的编程系统以及应用或程序的指令位于诸如硬盘驱动器 226 之类的存储设备上,并且可以被载入主存储器 208 以便由处理单元 206 执行。本发明的实

施例的过程由处理单元 206 使用计算机可用程序代码执行,所述代码可以位于诸如主存储器 208、只读存储器 224 之类的存储器内或一个或多个外围设备 226 和 230 内。

[0037] 本领域的技术人员将理解,图 1-2 中的硬件可以根据实施方式而所有变化。除了图 1-2 中所示的硬件或代替图 1-2 中所示的硬件,还可以使用诸如闪存、等同的非易失性存储器或光盘驱动器之类的其他内部硬件或外围设备。同样,本发明的过程可应用于多处理器数据处理系统。

[0038] 在某些示例性实例中,数据处理系统 200 可以是个人数字助理 (PDA),所述 PDA 可以配备有闪存以提供非易失性存储器来存储操作系统文件和 / 或由用户生成的数据。

[0039] 总线系统可以包括一条或多条总线,例如图 2 中示出的总线 238 或总线 240。当然,也可以使用提供与结构或架构连接的不同组件或设备之间的数据传输的任何类型的通信结构或架构来实现总线系统。通信单元可以包括一个或多个用于发送和接收数据的设备,例如图 2 中的调制解调器 222 或网络适配器 212。存储器可以例如是主存储器 208、只读存储器 224 或在图 2 中的北桥及存储器控制器中心 202 中找到的高速缓存。图 1-2 中所示的实例和上述实例并非旨在暗示架构限制。例如,除了 PDA 形式以外,数据处理系统 200 还可以是平板电脑、膝上型计算机或电话设备。

[0040] 本发明提供了计算机实现的方法、装置和计算机可用程序代码以便自适应地控制在传输数据时所用的带宽。本发明的各方面认识到当前使用的自适应带宽控制过程无法考虑客户端处的网络状况以外的网络状况。本发明的各方面认识到不同的网络具有不同的特征,所述特征可以影响数据的传输。本发明的各方面提供了不同的机制来考虑这些类型的因素。本发明的各方面都提供了设置参数的功能,以控制如何在网络上发送各个数据分组以用于自适应带宽控制。

[0041] 本发明的各方面认识到当前可用的数据传输系统在后台模式中运行太慢并且完成所需的时间太长。本发明的各方面还认识到存在大量客户希望自适应功能的情况,并且在特定情况下不希望分发时间太长。例如,病毒修补程序或更新的分发对于应用的更新非常重要。因此,本发明的各方面提供了根据每次分发设置优先级的功能。在这些示例性实例中,设置了三种不同的优先级:高、中和低。这些实例中的优先级是自适应优先级,其中诸如高优先级之类的选定优先级改变了自适应带宽控制过程的行为方式。例如,可以先以低优先级发送软件更新,然后以高优先级发送病毒修补程序。因此,可以根据不同分发的重要性为这些分发指定不同的优先级。具有不同优先级的分发为自适应带宽控制过程提供了不同的参数组,以便根据特定的选定优先级改变该过程的行为。

[0042] 此外,还可以根据每个网关来设置不同优先级的配置。在这些实例中,网关是用作一组设备的管道的设备或数据处理系统。例如,网关可以用作局域网或广域网的门户或入口。此外,网关还可以用作到无线网络的连接。网关还被认为管理一组客户端。

[0043] 虽然这些示例性实例面向自适应带宽控制,但是本发明的各方面也可以应用于将大量数据发送到一个或多个目标数据处理系统的任何类型的传输。

[0044] 现在参考图 3,其是示出根据本发明的示例性实施例的其中可以使用不同的优先级进行分发的实例网络数据处理系统的示意图。在此实例中,网络数据处理系统 300 包含源 302,所述源 302 与网关 304、306 和 308 相连。网关 304 通过路由器 318 提供了到客户端 310、312、314 和 316 的连接。网关 306 通过卫星路由器 328 提供了到客户端 320、322、324

和 326 的连接。网关 308 提供了到客户端 330、332、334 和 336 的连接。在这些实例中,这些客户端是通过诸如无线连接、拨号连接、电缆调制解调器或某些其他类似连接系统之类的各种方式与网关 308 连接的膝上型计算机。

[0045] 网络 300 是图 1 中的网络数据处理系统 100 内包含的网络的实例。具体地说,可以使用与图 2 中的数据处理系统 200 类似的数据处理系统来实现不同的客户端和网关。

[0046] 在此实例中,网关 304 管理高安全的局域网服务器。换言之,客户端 310、312、314 和 316 是服务器数据处理系统。网关 306 管理位于不同分店的销售点系统。网关 308 用于管理采用便携式计算机形式的客户端。

[0047] 管理员 338 可以使用不同的设置发送分发内容。在这些实例中,分发是指将任何种类的数据发送到一个或多个端点或接收器。例如,分发内容可以是应用的更新、动态链接库更新、病毒修补程序或定义更新,或者是数据文件。在此示例性实例中,可以使用高、中、低这三种优先级级别中的一种优先级来发送分发内容。当然,取决于特定的实施方式,还可以实现其他数量的优先级级别。为了说明本发明的一个实施例,采用了三种不同的优先级级别。

[0048] 因此,管理员 338 可以将一种分发内容发送到网关 304、306 和 308 以便使用诸如高优先级之类的优先级级别分发到客户端。管理员 338 然后可以使用诸如中优先级之类的不同优先级将后续的分发内容发送到网关 304、306 和 308。以此方式,可以为不同的分发指定不同的优先级以便根据与特定分发关联的重要性来传输到接收器。因此,更重要的分发内容可以更快地到达客户端,尽管此类分发会使用更多的带宽。对于较低优先级的分发,此类分发将使用较少的带宽并且分发所需的时间较长。

[0049] 进而,本发明的各方面为选定优先级提供了根据选定优先级级别在不同的网关处使用不同数量的带宽的能力。换言之,对于选定优先级级别,两个网关可以在网络状况相同的情况下使用不同数量的带宽来传输数据。在这些示例性实例中,通过调整在这些网关上执行的自适应带宽控制过程内的参数以便将分发内容发送到不同的客户端,来根据每个网关调整选定优先级级别的不同影响。

[0050] 例如,网关 304 可以在将高优先级分发内容发送到客户端 310、312、314 和 316 时使用百分之七十的带宽。网关 306 可以在将具有高优先级的相同分发内容传输到客户端 320、322、324 和 326 时使用百分之五十的带宽。在网关 308,在将相同的分发内容发送到客户端 330、332、334 和 336 时,以高优先级发送的相同的分发内容最多只能使用百分之三十的带宽。

[0051] 对网关所访问的特定网络内的带宽使用的实际影响通过调整在这些网关上执行的自适应带宽控制过程内的参数来调整。可以在初始设置网关时预先配置这些网关中的每个网关。此外,可以根据网络中的变化或管理员 338 所确定的其他变化来更改这些参数。这些变化可以通过下文中讨论的用户定义的设置来管理。

[0052] 现在参考图 4,它是示出根据本发明的示例性实施例的自适应带宽控制中使用的组件的示意图。在此实例中,自适应发送器 400 将数据发送到接收器 402、404 和 406。在此示例性实例中,自适应发送器 400 可以是诸如图 3 中的网关 304 之类的网关。具体地说,可以使用在图 2 的数据处理系统 200 中找到的硬件将自适应发送器 400 实现为图 1 中的服务器 106。接收器 402、404 和 406 可以是诸如图 3 中的客户端 310、312 和 314 之类的客户

端。还可以使用诸如图 2 中的数据处理系统 200 之类的数据处理系统实现这些接收器。通过将分组 408 发送到诸如路由器 410 之类的路由装置来发送数据。路由器 410 是用于根据在分组 408 内找到的路由数据将分组 408 路由或发送到适当接收器的设备。当路由器 410 被迫处理太多的分组时,它被称为缓冲 (backlogged) 路由器。换言之,缓冲路由器是在其队列中具有最大负荷或最多分组的路由器。发送器与接收器之间可以没有路由器或具有多个路由器。发送器是分组的源,而接收器是确认的源。当这些接收器接收分组 408 时,它们将确认 412 返回到自适应发送器 400。在此实例中,确认是常规 TCP/IP 通信的一部分。

[0053] 在这些示例性实例中,自适应发送器 400 可以实现为诸如图 1 中的服务器 104 之类的数据处理系统中的过程。接收器 402、404、406 是在诸如图 1 中的客户端 108、110 和 112 之类的接收设备上执行的过程。具体地说,这些不同的过程可以在诸如图 2 中的数据处理系统 200 之类的数据处理系统上实现。

[0054] 自适应发送器 400 跟踪发送到每个接收器的分组 408。还跟踪确认 412 的接收并将其用于确定诸如往返时间之类的参数。往返时间是从发送分组到收到确认之间的时间。在此实例中,往返时间基于自适应发送器 400 的角度。

[0055] 此外,此信息用于标识块。块在发送任意分组时开始;将保留块中所有分组的统计数据直到接收器确认了开始块的初始分组。当返回任意分组的确认时,自适应发送器 400 将计算该分组块的统计信息。换言之,块中可以存在一个或多个分组,具体取决于在返回块开头中的任意分组的确认之前发送了多少分组。此外,自适应发送器 400 还标识窗口。窗口是在未收到确认的情况下网络中已发送的分组数。

[0056] 进而,自适应发送器 400 还估计路由器 410 中的分组数。使用当前的往返时间确定此信息以计算预期的未确认分组数与实际的未确认分组数。例如,如果基于当前的往返时间,网络上应存在五个分组并且自适应发送器 400 标识了八个未确认的分组,则自适应发送器 400 可以估计路由器 410 上存在三个分组。

[0057] 自适应发送器 400 通过当尝试在路由器 410 上保留选定数目的分组时增加或减小预期窗口大小来控制速度。较大的窗口传输速度较快,因为相对于其他网络业务,路由器 410 将花费更多的时间来处理自适应分组。以此方式,自适应发送器 400 可以调整路由器 410 的 α 和 β 参数。 α 是一个整数,其指示了在窗口中增加一个分组之前,每个连接在路由器 410 中的最小分组数。 β 是一个整数,其指示了在窗口中减少一个分组之前,每个连接在缓冲路由器中的最大分组数。这些参数允许通过根据 α 和 β 值调整窗口来小幅提高和降低速度。另一参数(即阈值)允许在块中百分之五十的分组满足该阈值的条件时快速降低速度。在此实例中,阈值是从基本往返时间到最大往返时间的时间的可配置百分比。

[0058] 现在参考图 5,它是根据本发明的示例性实施例的传输控制协议/网际协议(TCP/IP)以及类似协议的示意图。TCP/IP 以及类似的协议由通信架构 500 使用。在此实例中,通信架构 500 是一个四层系统。此架构包括应用层 502、传输层 504、网络层 506 以及链路层 508。每个层都负责处理各种通信任务。链路层 508 还被称为数据链路层或网络接口层并且通常包括操作系统中的设备驱动程序以及计算机中的相应网络接口卡。该层处理与所使用的网络介质(例如光缆或以太网电缆)进行物理连接的所有硬件细节。

[0059] 网络层 506 还被称为因特网层并处理网络周围的数据分组的移动。例如,网络层 506 处理通过网络传输的各个数据分组的路由。TCP/IP 套件中的网络层 506 由多种协议组

成,其中包括网际协议(IP)、网际控制消息协议(ICMP)以及网际组管理协议(IGMP)。

[0060] 接下来,传输层 504 提供网络层 506 与应用层 508 之间的接口来帮助在两个主机之间传输数据。传输层 504 涉及例如以下操作:为之下的网络层将来自应用层的数据分为适当大小的块,确认收到的分组以及设置超时以确保另一端确认所发送的分组。在 TCP/IP 协议套件中,存在两种明显不同的传输协议:TCP 和用户数据报协议(UDP)。TCP 提供可靠服务以确保在两个主机之间正确传输数据,其中包括遗失检测和重传服务。

[0061] 相反,UDP 通过只是将称为数据报的数据分组从一个主机发送到另一个主机(而不提供任何保证数据正确传输的机制)来为应用层提供简单得多的服务。当使用 UDP 时,应用层必须执行可靠性功能。

[0062] 应用层 502 处理特定应用的细节。对于几乎所有的实施方式都存在许多通用的 TCP/IP 应用,其中包括用于远程登录的 Telnet;文件传输协议(FTP);用于电子邮件的简单邮件传输协议(SMTP);以及简单网络管理协议(SNMP)。

[0063] 本发明的各方面在应用层 502 实现以便通过允许用户输入用户设置的方式来自适应地控制数据传输。以此方式,用户可以更改不同的设置以控制如何在网络上发送分组以用于自适应带宽控制。通过允许从应用级别输入用户设置,用户可以针对自适应带宽控制过程通常没有考虑的不同网络类型和不同网络状态进行更改。以此方式,可以考虑客户端上的状况之外的状况。例如,本发明的各方面允许用户根据因素的确定来更改设置,所述因素包括到接收器的路径中的跃点数和到接收器的路径中的具有大量业务或拥塞的链路数。

[0064] 尽管在应用层上实现了本发明的示例性实例,但是也可以在其他层上实现其他过程。例如,可以在传输层 504 或网络层 506 内实现本发明的各方面,具体取决于特定的实施方式。

[0065] 现在参考图 6,它是示出根据本发明的示例性实施例在提供自适应带宽控制的过程中使用的软件过程和组件的示意图。在此实例中,自适应发送器 600 是图 4 中的自适应发送器 400 内的过程的更详细示例。在此示例性实例中,自适应发送器 600 包含三个线程。发送线程 602、拥塞控制线程 604 和分组嗅探线程 606 是用于自适应地将数据发送到一个或多个接收器的组件。发送线程 602 用于通过调用套接字来发送数据。套接字是将应用连接到诸如 TCP/IP 堆栈中的 TCP/IP 协议之类的网络协议的软件对象。

[0066] 发送线程 602 将请求 608 发送到拥塞控制线程 604 以询问发送线程 602 可以发送多少数据,拥塞控制线程 604 返回应答 610,告知可以发送多少数据。发送线程 602 使用应答 610 发送要传输的分组 612。这些分组存储在缓冲路由器队列 614 中,直到缓冲路由器将分组路由到它们的目的地。缓冲路由器队列 614 位于诸如图 4 中的路由器 410 之类的缓冲路由器上。当收到分组时,会将确认 616 返回到自适应发送器 600。

[0067] 拥塞控制线程 604 通过执行自适应带宽控制过程来确定要发送的数据量。这些示例中的拥塞控制线程 604 使用自适应带宽控制过程来自适应地确定用于在网络上传输数据分组的传输率以响应网络状态的改变。这些网络状况包括例如除了自适应发送器处理的数据传输以外,还由于各种其他数据传输所导致的网络上的拥塞量。换言之,发送数据的速率将根据网络状况而改变。拥塞控制线程 604 根据用户设置的不同参数来确定不同的加快或减缓速率以维持最小的网络影响。

[0068] 使用下面的图 7 示出了网络状况的一个实例。在 FTP 服务器 752 将数据发送到 FTP 客户端 754 之前,路由器 746 的状况为闲置。这种情况允许自适应发送器 700 即使在优先级很低时也能以非常高的速度发送到接收器或端点。但是,FTP 服务器 752 一开始将数据发送到 FTP 客户端 754,路由器 746 就会变得拥塞。然后,自适应带宽控制过程将根据其优先级做出反应。在低优先级,自适应带宽控制过程将降低到几乎不发送任何数据的速度,并且将对 FTP 分发具有最小的影响。在高优先级,该过程将以非常快的速度发送并导致 FTP 分发的速度降低。

[0069] 下面是自适应分发如何对网络状况做出反应的另一实例。一家银行安装了单个路由器来管理银行内 10 个系统的网络连接。银行通过此路由器连接到中央站点(其管理全国的 300 家银行)。在凌晨六点(银行开门之前),银行没有一个人,并且网络(具体地说,银行的单个路由器)处于闲置状态。此时,从中央站点以低优先级发送非常大的自适应分发。由于所述分发是银行路由器上的唯一业务,因此该分发开始加速达到使用百分之百的路由器带宽。这种情况持续到早上八点,此时客户开始使用银行中还必须与所述单个路由器共享网络的 ATM 以及贷款处理系统。所述自适应分发将立即发现有其他业务通过缓慢的链路(在这种情况下指银行的路由器)。由于自适应分发处于低优先级,因此它将变慢以使用非常小百分比的路由器带宽,从而对 ATM 和贷款处理业务具有最小影响。在下午五点,所述自适应分发仍在进行。当银行关门时,贷款处理业务减少。所述自适应分发将发觉网络负荷减小并在有更多带宽可用时继续加速。

[0070] 返回参考图 6,此过程使用队列 618 中的信息。具体地说,分组嗅探线程 606 从网络中捕获分组和确认并将分组信息 620 和确认信息 622 放入队列 618。分组嗅探线程 606 过滤分组和确认以将拥塞控制线程 604 所需的相应数据放入队列 618。备选地,可以将所有分组和确认放入队列 618 以便由拥塞控制线程 604 进行处理。队列 618 中的信息由拥塞控制线程 604 用于标识参数,例如从发送分组到收到该分组的确认为止的往返时间。可以由拥塞控制线程 604 通过队列 618 中的信息标识和维护的其他参数包括块、窗口以及缓冲路由器中的分组。

[0071] 进而,本发明的各方面包括可用于设置用户定义参数 626 的用户接口 624。用户接口 624 由参数线程 628 提供。通常,将在系统上首次建立自适应过程时设置参数。可以通过用户接口 624 和参数线程 628 修改这些参数。用户定义参数 626 存储在诸如盘之类的非易失性存储器中。稍后可以在启动自适应带宽过程时读取这些参数。在此示例性实例中,这些参数包括一组参数,其中不同的值用于不同的优先级。在示例性实例中,根据分发的优先级级别改变的参数组包括阈值、 β 参数和最大等待时间。尽管用户可以为不同的优先级级别输入不同的值,但是这些参数可以被直接设置以便由自适应发送器 600 使用。用户接口 624 允许用户选择或输入不同的参数以便由拥塞控制线程 604 在执行自适应带宽控制过程时使用。

[0072] 在这些示例性实例中,本发明的各方面允许用户为五个参数输入或选择值。这五个参数包括阈值、往返时间最大变化、 β 参数、往返时间平滑以及最大等待时间。往返时间最大变化是用于限制当前测量的往返时间相对于先前计算的往返时间的变化量的参数。所述过程跟踪当前的往返时间、最大往返时间以及基本往返时间。

[0073] 拥塞控制线程在每个套接字的基础上并且按顺序跟踪往返时间。如果某个分组的

往返时间为 10,而第二个分组的往返时间为 20,则第二个分组的往返时间受 1.5 倍往返时间最大变化值的限制。即使实际的往返时间为 20,也将往返时间值记录为 10 乘以 1.5,即 15($10 \times 1.5 = 15$)。然后,之后的分组的往返时间最多为 1.5 乘以 15(1.5×15),15 是为第二个分组记录的往返时间。

[0074] 网络通常具有一定的随机性。此参数允许自适应带宽控制过程忽略过大或过小的往返时间,但是仍允许较大的往返时间,前提是这些较大的往返时间的出现频率足够高。此参数允许用户配置容许多少变化。

[0075] 这些实例中的往返时间平滑参数是一个整数,其用于指示往返时间测量值应由指数衰减平滑的程度。通过将先前测量值进行平均来执行指数衰减。当自适应带宽控制过程过度补偿往返时间中的波动而行为不规律时,此参数可以有所帮助。平滑通常在考虑最大变化参数之前发生。

[0076] 最大等待时间是指指示在放弃并重置之前连接为了发送将等待的最大往返时间的倍数的参数。有时确认会在网络上丢失,从而导致发送器等待很长一段时间才能发送其他数据。这是一个重置值,用于在发送数据之前经过了太长时间的情况下重置自适应带宽控制过程。对于阈值参数,当分组的往返时间在套接字上发送时,时间往往落入选定的范围内。当网络处于 闲置时尤其如此。

[0077] 阈值是从最小往返时间到最大往返时间的路程百分比。例如,百分之二十的阈值表示阈值处于从基本往返时间到最大往返时间的路程的百分之二十。根据网络类型,往返时间具有或多或少的变化。不同的变化通常需要不同的阈值。例如,运行正常的局域网的往返时间变化非常小,从而允许较低的阈值。广域网的往返时间变化较大。对于此类网络,需要较高的阈值。用户可以配置此特定参数以考虑在发送数据时可能遇到的不同类型的网络。对于阈值参数,当分组的往返时间在套接字上发送时,时间往往落入选定的范围内。当网络闲置时尤其如此。

[0078] 如果分组的往返时间比阈值慢,则认为该分组很慢。如果认为块中百分之五十的分组很慢,则可以将窗口缩小一半,从而显著降低分发的速度。因此,允许用户配置该值使得可以考虑不同的网络类型和状况。

[0079] 本发明的各方面通过修改示例性实例中的三个参数来更改优先级。这些实例中针对不同优先级修改的参数包括阈值、 β 参数和最大等待时间。

[0080] 现在参考图 7,它是示出根据本发明的示例性实施例的使用优先级将分发内容从自适应发送器发送到接收器的网络的示意图。在此实例中,自适应发送器 700 通过网络 702 将分发内容发送到客户端 704、706、708、710、712、714、716、718、720、722、724、726、728、730、732、734、736、738、740 和 742。750、748、746 和 744 是路由器。746 是缓冲路由器。这些客户端与路由器 744 相连,路由器 744 又与路由器 746 相连。路由器 746 与路由器 748 相连。自适应发送器 700 通过服务器 750 与网络 702 相连。在此实例中,网络 702 包含二十个跃点。除了分发内容之外,文件传输协议 (FTP) 服务器 752 还可以将数据发送到 FTP 客户端 754。在此实例中,FTP 服务器 752 将数据发送到 FTP 客户端 754,同时将分发内容从自适应发送器 700 发送到客户端。示例性实例导致 FTP 服务器产生导致路由器 746 处的链路拥塞的高业务需求。通过设置优先级,自适应发送器 700 可以在将分发内容发送到这些客户端时使用不同的带宽量。

[0081] 现在参考图 8,它是示出根据本发明的示例性实施例的网络中带宽使用的示意图。图 8 中的图示出了在诸如图 7 中示出的网络之类的网络传输数据时使用的网络带宽百分比。在此实例中,线 802 示出了用于不同类型分发的带宽量。在部分 804 中,仅示出了从图 7 中的 FTP 服务器 752 到 FTP 客户端 754 的 FTP 传输。在此分发中使用了百分之百的带宽。在为从图 7 中的自适应发送器 700 到客户端的分发设置了低优先级的情况下,图 7 中的路由器 744 分配大约百分之八十的带宽以用于部分 806 中示出的 FTP 传输。对于中等优先级,部分 808 中示出了使用大约百分之五十的带宽以用于 FTP 传输。如可以看到的,随着优先级增加,会将较少的带宽分配给 FTP 传输,而将较多的带宽分配给自适应发送器执行的分发。在部分 810 中,将高优先级给予到客户端的分发。如可以看到的,FTP 传输降低到大约百分之三十带宽的水平。在部分 812 中,不使用自适应带宽控制过程。因此,在此特定部分中不使用任何所述的拥塞控制。

[0082] 现在参考图 9,它是根据本发明的示例性实施例的配置参数的用户设置的过程的流程图。图 9 中示出的过程可以在图 6 中的参数线程 628 中实现。此过程用于允许用户定义或更改在自适应带宽控制中使用的参数。在这些实例中,所述参数包括阈值、往返时间最大变化、往返时间平滑、最大等待时间和 β 参数。具体地说,图 9 中示出的过程可用于设置不同网关的参数以便在诸如高、中、低之类的不同优先级级别中使用。在这些实例中,高优先级可以设置 β 值等于 7,阈值等于 99 以及最大等待时间等于 20。中优先级可以设置 β 值等于 5,阈值等于 40 以及最大等待时间等于 30。对于低优先级,设置 β 值等于 3,阈值等于 25 以及最大等待时间等于 40。这些特定设置是用于不同优先级级别的设置。可以通过所提供的用户接口来设置这些设置。

[0083] 进而,在这些示例性实例中,本发明的各方面还考虑了从来自网关的任何路由器或服务器到端点的影响。通过在自适应带宽控制过程中使用往返时间来提供此能力。

[0084] 通过呈现用户设置开始过程(步骤 900)。可以在诸如图 6 中的用户接口 624 之类的用户接口中呈现这些设置。然后过程接收用户输入(步骤 902)。判定用户输入是否更改了参数中的设置(步骤 904)。如果用户输入更改了设置,则使用新的设置替换原有设置(步骤 906)。之后,过程返回步骤 900 以将这些设置呈现给用户。

[0085] 再次参考步骤 904,如果用户输入未改变设置,则判定用户输入是否结束更改用户设置的过程(步骤 908)。如果用户输入未结束此过程,则所述过程将返回步骤 900。否则,将保存用户设置(步骤 910)且之后过程终止。在这些实例中,将这些设置保存为图 6 中的用户定义的参数 626。

[0086] 接下来参考图 10,它是根据本发明的示例性实施例的发送线程的过程的流程图。图 10 中示出的过程可以在诸如图 6 中的发送线程 602 之类的发送线程中实现。

[0087] 通过发送有关可以将多少数据发送到拥塞控制线程的请求来开始此过程(步骤 1000)。然后过程接收应答(步骤 1002)。此应答包含作为拥塞控制线程所执行的自适应带宽控制过程的结果而可以发送的数据量。响应于接收到应答,过程发送对套接字的调用以便仅发送可接受的数据量(步骤 1004)。随后,判定是否存在更多要发送的数据(步骤 1006)。如果有更多的数据,则过程返回步骤 1000。否则,过程终止。

[0088] 接下来参考图 11,它是根据本发明的示例性实施例的分组嗅探线程的过程的流程图。图 11 中示出的过程可以在图 6 的分组嗅探线程 606 中实现。

[0089] 通过判定是否检测到分组来开始此过程（步骤 1100）。如果检测到分组，则过程从分组提取分组标识符和时间戳（步骤 1102）。然后过程将数据存储到队列中（步骤 1104）。该队列可以由拥塞控制线程访问以便可以在确定往返时间和执行自适应带宽控制过程中使用所述数据。

[0090] 再次参考步骤 1100，如果未检测到分组，则判定是否检测到确认（步骤 1106）。如果未检测到确认，则过程返回步骤 1100。如果在步骤 1106 中检测到确认，则提取确认的分组标识符和时间戳（步骤 1108）。如上所述，过程然后继续到步骤 1104。

[0091] 现在参考图 12，它是根据本发明的示例性实施例的拥塞控制线程的过程的流程图，在此实例中，图 12 中示出的过程可以在图 6 的拥塞控制线程 604 中实现。

[0092] 通过接收来自发送线程的请求开始此过程（步骤 1200）。在这些实例中，拥塞控制线程所使用的自适应控制过程位于网关中。每个网关都针对于优先级的不同参数具有自己的值。当发送线程开始发送分发内容时，发送线程将这些值与当前分发优先级一起传递给拥塞控制线程。拥塞控制线程将使用与当前分发优先级对应的值。可以在首次请求发送分发内容时在步骤 1200 中接收此信息。此请求将请求确定可以发送多少数据。之后，将从队列中取出数据（步骤 1202）。此数据包括到达时间和分组标识符。然后，从队列中的数据确定往返时间（步骤 1204）。然后根据分发的优先级获取参数（步骤 1206）。这些参数来自图 6 中的用户定义参数 626。具体地说，用户定义参数用于与选定优先级关联的特定参数组。在这些示例性实例中，针对不同优先级的具有不同设置的参数包括 β 参数、阈值和最大等待时间。然后，所述过程执行自适应带宽控制过程（步骤 1208）。此过程可以例如是拥塞控制线程中包含的步骤。备选地，线程可以在步骤 1208 中调用函数或外部过程。然后过程获取结果（步骤 1210），返回包含可以发送的数据量的应答（步骤 1212），随后过程终止。

[0093] 现在参考图 13，它是根据本发明的示例性实施例的执行自适应带宽控制的过程的流程图。图 13 中的过程是图 12 中的步骤 1208 的更详细描述。

[0094] 通过确定最小往返时间和最大往返时间来开始此过程（步骤 1300）。随后，确定窗口（步骤 1302）。然后过程确定每个确认的高于和低于阈值的分组数（步骤 1304）。接下来，过程判定块中是否有百分之五十的分组高于阈值（步骤 1306）。如果块中百分之五十的分组不高于阈值，将计算预期吞吐量（步骤 1308）。在步骤 1308 中，将按照如下方式计算预期吞吐量：

$$[0095] \quad E \leftarrow \frac{W}{\min RTT}$$

[0096] E 是预期吞吐量，W 是窗口，minRTT 是从通过队列获取的往返时间中所见的最小往返时间。随后，过程确定实际的吞吐量（步骤 1310）。该实际吞吐量使用以下公式来确定：

$$[0097] \quad A \leftarrow \frac{W}{\text{observedRTT}}$$

[0098] A 是实际吞吐量，W 是窗口，observedRTT 是使用发送分组和收到确认之间的时间差来测量的值。结果，预期吞吐量基于最小往返时间，因为预期在闲置网络上发送分组所用的时间始终为最小往返时间。实际吞吐量基于当前的网络状态，其中当前往返时间大于最小往返时间。在这些实例中，最小往返时间与基本往返时间相同。然后过程计算在缓冲路

由器上的分组数（步骤 1312）。使用以下公式估算实际的分组数：

[0099] $\text{Diff} \leftarrow (E-A) \cdot \text{minRTT}$

[0100] Diff 是缓冲路由器上的分组数, E 是预期的吞吐量, A 是实际吞吐量, minRTT 是所见的最小往返时间。

[0101] 接下来, 判定缓冲路由器上的分组数是否小于 α 值（步骤 1314）。如上所述, α 是一个整数, 其指示在窗口中增加一个分组之前, 每个连接在缓冲路由器中应存在的最小分组数。如果缓冲路由器上的分组数小于 α 值, 则过程将窗口大小增加 1（步骤 1322）。此后, 过程指示可以发送窗口大小的数据（步骤 1318），然后过程终止。

[0102] 再次参考步骤 1314, 如果缓冲路由器上的分组数不小于 α 值, 则判定分组数是否大于 β 值（步骤 1316）。如果分组数大于 β 值, 则将窗口大小减少 1（步骤 1324）。然后, 过程按照上文所述继续到步骤 1318。否则, 过程继续到步骤 1318 而不更改窗口大小。

[0103] 再次参考步骤 1306, 如果块中有百分之五十的分组高于阈值, 则将窗口缩小一半（步骤 1320）。过程按照上文所述继续到步骤 1318。

[0104] 上面图 13 中所示的过程基于对当前可用的带宽控制算法的修改, 所述算法如 Venkataramani 等人的“TCP Nice: A Mechanism for Background Transfers” (ACM SIGOPS Operating System Review, Vol. 36, Issue SI Winter 2002, pp. 1-15, 其在此引入作为参考) 中所述的 Nice 算法。

[0105] 本发明的各方面确定了多个影响图 13 中所述的自适应带宽控制过程的实现的参数。下面列出了这些参数及其描述：

[0106] NICE_ALPHA- 指示在窗口增大之前每个连接在缓冲路由器中的最小分组数的整数。默认值为 1。NICE_BAES_DISCARD- 指示每个连接要丢弃的初始基本 RTT 测量数的整数。原因是当首次开始分发并且网络尚未过度饱和时, 最初执行的基本 RTT 测量可能过低。默认值为 3。

[0107] NICE_BAES_SCALE- 指定应用于由 nice 维护的全域最小 RTT (v-baseRTT) 的缩放因子的浮点数。如果网络偶尔允许不规则的短 RTT, 则将此参数设置为诸如 1.1 之类的较小正数将会有所帮助。默认值为 0.0。

[0108] NICE_BELOW_ONE- 提供 v_cwnd_below_one 的下限的整数, 它是对于小窗口, 连接可以闲置的最大 RTT 数。默认值为 48。

[0109] NICE_BETA- 指示在窗口缩小之前每个连接在缓冲路由器中的最大分组数的整数。这是 β 参数的默认值。默认值基于优先级。

[0110] NICE_CLAMP- 当设置箝位时, 将窗口 (snd_cwnd) 的大小限制为多于网络中当前存在的分组数的数量不超过四。默认值已设置。

[0111] NICE_COND- 当设置时, 发送线程等待由确认线程发信号通知的状况, 而不是等待任意长的时间。默认值未设置。

[0112] NICE_COND_RTX_Q- 当设置时, 拥塞线程将每个出站分组的估算添加或更新到用于计算 RTT 的 rtx_q。拥塞线程的优点是它对出站时间的估算很准确, 但是可能丢弃分组。默认值已设置。

[0113] NICE_DYNAMIC_MSS- 当设置时, 自适应带宽控制过程将从 MSS 的低值开始, 然后在每次嗅探到具有更大 MSS 的出站分组时增大该值。通过这种方式, MSS 应能够快速逼近用

于连接的 MSS。默认值已设置。

[0114] NICE_FAST- 指定每个自适应带宽控制过程套接字将要处于快速启动阶段的毫秒数的整数。快速启动阶段越长, maxRTT 的估算就越准确。但是, 将此参数设置为较大的值会导致自适应带宽控制过程被实际禁用所述长度的时间。默认值为 5000 = 5 秒。

[0115] NICE_FAST_RETURN- 当设置时, 快速启动阶段在第一次 send() 失败时退出, 并显示错误 EWOULDBLOCK。这可最大程度上缩短快速启动模式 花费的时间, 其不影响网关, 因为填充套接字的出站队列将非常快。默认值已设置。

[0116] NICE_INTERFACE- 将由 nice 所使用的接口 (网卡标识符)。此时, 自适应带宽控制过程无法动态地确定正确的接口。因此, 如果接口不是第一个活动的接口, 则需要手动进行设置。通常应设置为诸如 “eth1” 之类的值。默认值未设置。

[0117] NICE_MAX_MULT- 指定作为 baseRTT 倍数的 maxRTT 的最小值的浮点数。当设置时, 它应有助于避免 maxRTT 的值过低以及所伴随的低吞吐量。默认值为 0.0。

[0118] NICE_MIN_MSS- 指示要使用的最小 MSS 的整数。为了增加效率, 考虑 NICE_DEFAULT_MSS 而忽略操作系统所提供的低值。默认值为 1000。

[0119] NICE_MIN_PACKET- 指定一次应发送的最小数据量的整数。其与 NICE_MIN_SEND 不同, 因为如果 ok_to_send 小于指定的值, 它会被设置为 0。这是为了避免出现拥塞控制线程持续告知发送线程发送诸如 1 字节之类的非常小的值的情况。通过将该值设置为 10, 拥塞控制线程将一直等待, 直到可以发送至少 10 个字节 (在计算指定 10 之前将一直返回 0)。默认值为 0。

[0120] NICE_MIN_SEND- 指定 ok_to_send 的最小值的整数, 其表示拥塞控制线程将始终告知发送线程至少发送此数量的数据。通过将此参数设置为 10, 则即使自适应计算指定 3, 拥塞控制线程也会返回 10。默认值为 0。

[0121] NICE_NANO_FIXED- 指定内部回调函数 select_delay() 在被使用时暂停的微秒数的整数。

[0122] 当回调设置为 select_delay() 之外的值时, NICE_NANO_FIXED 没有任何影响。当未设置时, 根据需要多长时间才能有足够的空间发送上一个块的吞吐量 (由上一个块中最快的分组的 RTT 和块大小确定) 所指示的分组来动态地计算延迟。当设置为 1234 时, 不调用 nanosleep() : 未设置 - 动态地计算延迟, 0- 使用值 0 调用 nanosleep(), 在某些系统上可以延迟 10ms, 1234 不调用 nanosleep(), 其他 - 使用指定的微秒数调用 nanosleep(), 默认值为 0。

[0123] NICE_PCAP_TIMEOUT- 指示在返回捕获的分组列表之前操作系统应等待的时长的整数。该变量对 Linux 系统没有影响, 因为 Linux 系统只等待直至一个分组可用而不考虑花费的时间。在诸如 Solaris 之类的该变量具有影响的系统上, 需要在当超时设置得很低时获取准确的 RTT 与当网络闲置时浪费 CPU 时间之间进行权衡。默认值为 10ms (Solaris 上支持的最低的值)。

[0124] NICE_QUEUE_LIMIT- 指定由嗅探线程写入并由拥塞线程读取的分组队列的最大长度的整数。队列越长, nice 的响应性越弱, 因为拥塞线程正在处理由分组通过队列所需的时间而导致延迟的信息。默认值为 10。

[0125] NICE_RTT_MAX_CHANGE- 指示允许基本 RTT 和最大 RTT 相对于先前值变化的最大量

的整数。当设置时,自适应带宽控制过程应更能容忍虚假的 RTT 极值。默认值为 1.5。

[0126] NICE_RTT_MIN_STDS- 指示高于基本 RTT 的标准差的最小数的整数。RTT 必须被认为高于基本 RTT。默认值为 0。

[0127] NICE_RTT_SMOOTHING- 指示 RTT (往返时间) 测量值应按指数衰减平滑的程度的整数,指数衰减通过将先前测量值进行平均来完成。当 nice 过度补偿 RTT 中的每个波动而行为不规律时,此参数会有用。默认值为 50。

[0128] NICE_RTT_STD_SMOOTH- 与 NICE_RTT_SMOOTHING 类似的整数,但是用于标准差。标准差基于最近的 RTT 测量值的加权平均。默认值为 0。

[0129] NICE_SEND_RTX_Q- 当设置时,发送线程将每个出站分组的估算添加或更新到用于计算 RTT 的 rtx_q。发送线程的优点是它不会丢弃分组,但是对出站时间的估算可能不准确。默认值已设置。

[0130] NICE_THROUGHPUT_AVG- 指示要将多少最近分组包括在吞吐量计算中的整数。较大的值将导致吞吐量的计算更准确,但代价是响应性减弱。默认值为 20。

[0131] NICE_THROUGHPUT_START- 指示必须在开始计算吞吐量之前发送的最小分组数的整数。这是为了考虑最初为空的慢链路队列,所以在套接字前面部分的吞吐量不是套接字典型的吞吐量并应被忽略。默认值为 100。

[0132] NICE_WAIT_BASE- 指示在放弃并重置 snd_nxt 和 snd_una 之前连接为了发送将等待的 v_baseRTT (全局最小 RTT) 的倍数的整数。该值加上计算出的超时就是 NICE_WAIT_MAX。默认值为 0。

[0133] NICE_WAIT_MAX- 指示在放弃并重置之前连接为了发送将等待的 v_maxRTT (全局最大 RTT) 的倍数的整数。默认值基于优先级。

[0134] 现在参考图 14,它是根据本发明的示例性实施例的定制将分发内容发送到客户的接收器的过程的流程图。图 14 中所示的过程是用于为希望在其网络内安装定制分发系统的客户提供解决方案的过程。

[0135] 通过接收客户请求开始此过程 (步骤 1400)。该请求包括生成解决方案所需的信息。例如,客户网络内的网关的标识包含在请求中。此外,还可以包括通过网关访问的不同客户端的特征。另外,客户端发送不同类型的分发内容的要求也包括在示例性实例中。响应于接收到该请求,将分析客户端参数以便将分发内容发送到端点 (步骤 1402)。

[0136] 在此分析中,将确定用于不同优先级的参数的设置 (步骤 1404)。在这些示例性实例中,优先级包括阈值、 β 参数和最大等待时间。确定的参数可以用于各种优先级级别。例如,所示的实例使用三种优先级:高、中和低。取决于客户请求,可以使用不同数量的优先级级别,例如两种或四种级别。针对特定优先级级别生成的参数可以随不同的网关而有所不同,具体取决于客户端的请求和网关所在的网络的特征。

[0137] 此后,生成每个网关的自适应带宽控制的代码 (步骤 1406)。此代码被封装以便可以安装在特定网关处。此代码包括自适应带宽控制过程和该特定网关的参数。此代码还包括在网关上安装和设置自适应带宽控制过程所需的可执行文件。然后,将代码发送给客户 (步骤 1408) 以便分发。以此方式,客户可以从响应请求的企业或其他供应商接收定制的分发系统。

[0138] 因此,本发明的各方面提供了用于调整数据传输速率的计算机实现的方法、装置

和计算机可用程序代码。本发明的各方面在应用级别上接收用户输入以选择参数值。这些用户定义的参数在自适应带宽控制过程中使用 以确定可以通过网络发送到目的地的分组数。本发明的各方面允许用户根据不同的网络状况和特征来定义参数。以此方式,本发明的各方面允许根据变化的网络类型和状况来自适应地发送数据。

[0139] 本发明可以采取完全硬件实施例、完全软件实施例或同时包含硬件和软件元素的实施例的形式。在优选实施例中,本发明以软件实现,所述软件包括但不限于固件、驻留软件、微代码等。

[0140] 此外,本发明可以采取可从计算机可用或计算机可读介质访问的计算机程序产品的形式,所述计算机可用或计算机可读介质提供了可以被计算机或任何指令执行系统使用或与计算机或任何指令执行系统结合的程序代码。出于在此说明的目的,计算机可用或计算机可读介质可以是任何能够包含、存储、传送、传播或传输由指令执行系统、装置或设备使用或与所述指令执行系统、装置或设备结合的程序的有形装置。

[0141] 所述介质可以是电、磁、光、电磁、红外线或半导体系统(或装置或设备)或传播介质。计算机可读介质的实例包括半导体或固态存储器、磁带、可移动计算机盘、随机存取存储器(RAM)、只读存储器(ROM)、硬磁盘和光盘。光盘的当前实例包括光盘-只读存储器(CD-ROM)、光盘-读/写(CR-R/W)和DVD。

[0142] 适于存储和/或执行程序代码的数据处理系统将包括至少一个通过系统总线直接或间接连接到存储器元件的处理器。所述存储器元件可以包括在程序代码的实际执行期间采用的本地存储器、大容量存储装置以及提供至少某些程序代码的临时存储以减少必须在执行期间从大容量存储装置检索代码的次數的高速缓冲存储器。

[0143] 输入/输出或I/O设备(包括但不限于键盘、显示器、指点设备等)可以直接或通过中间I/O控制器与系统相连。

[0144] 网络适配器接口也可以被连接到系统,以使所述数据处理系统能够通过中间专用或公共网络变得与其他数据处理系统或远程打印机或存储设备相连。调制解调器、电缆调制解调器以及以太网卡只是少数几种当前可用的网络适配器类型。

[0145] 出于示例和说明目的给出了对本发明的描述,并且所述描述并非旨在是穷举的或是将本发明限于所公开的形式。对于本领域的技术人员来说,许多修改和变化都将是显而易见的。实施例的选择和描述是为了最佳地解释本发明的原理、实际应用,并且当适合于所构想的特定使用时,使得本领域的其他技术人员能够理解本发明的具有各种修改的各种实施例。

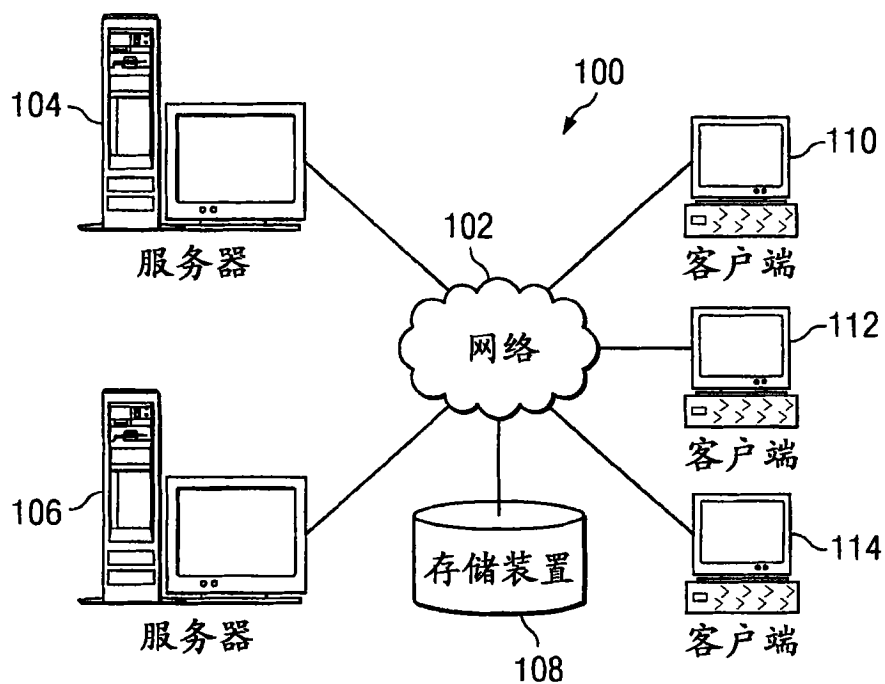


图 1

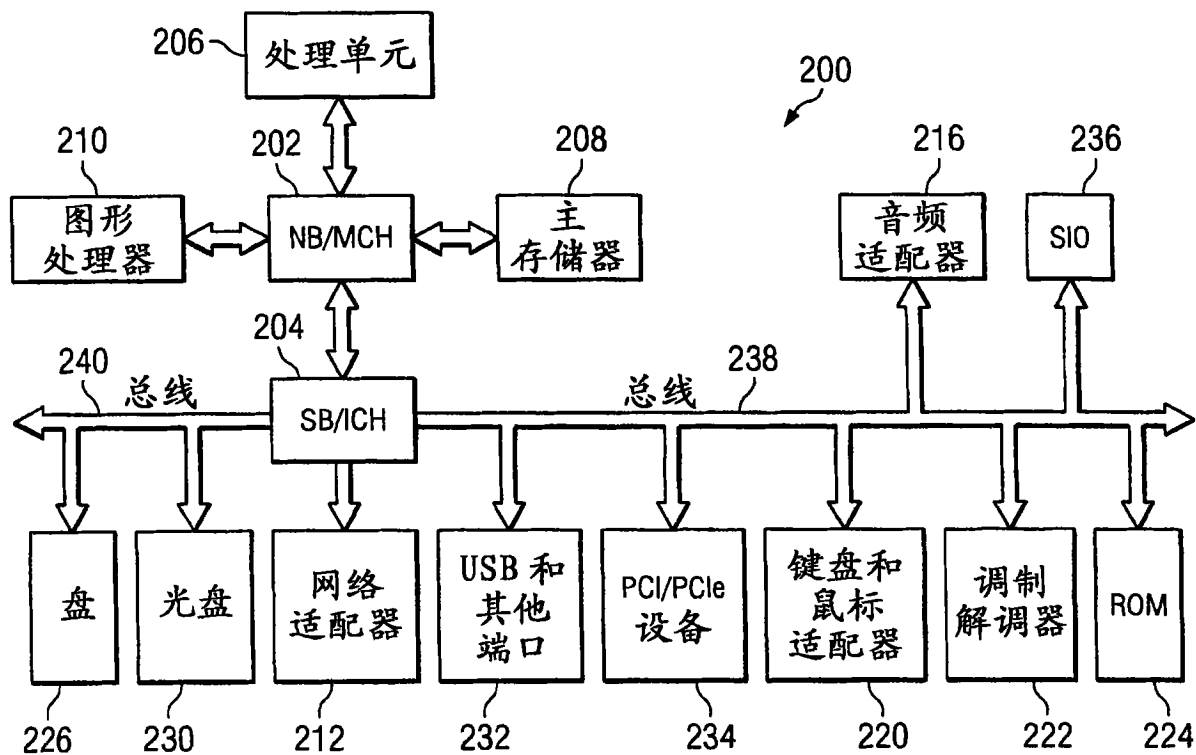


图 2

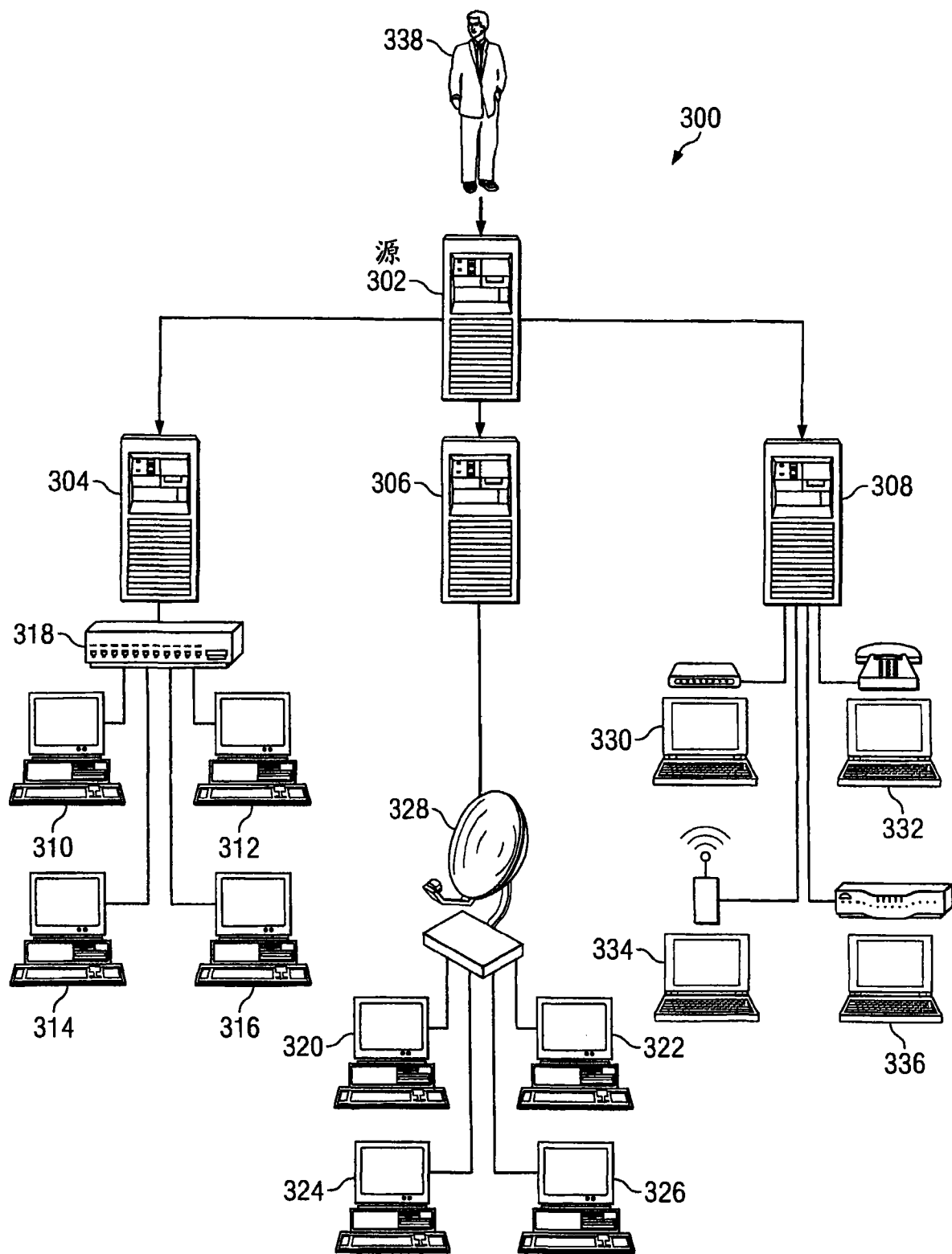


图 3

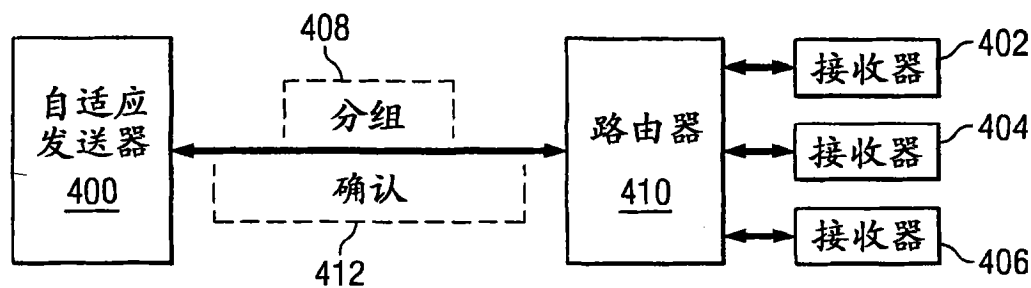


图 4

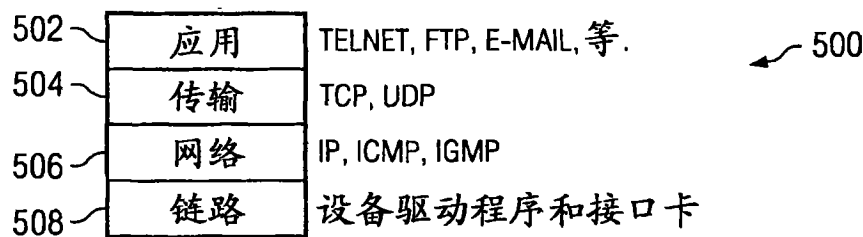


图 5

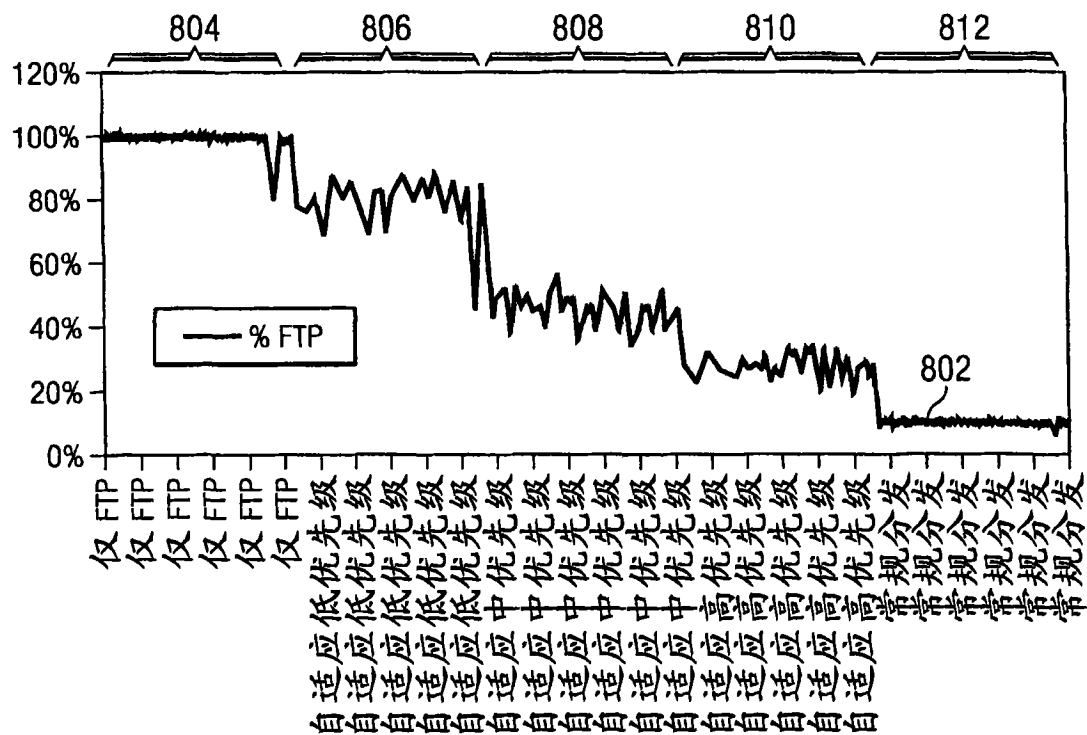


图 8

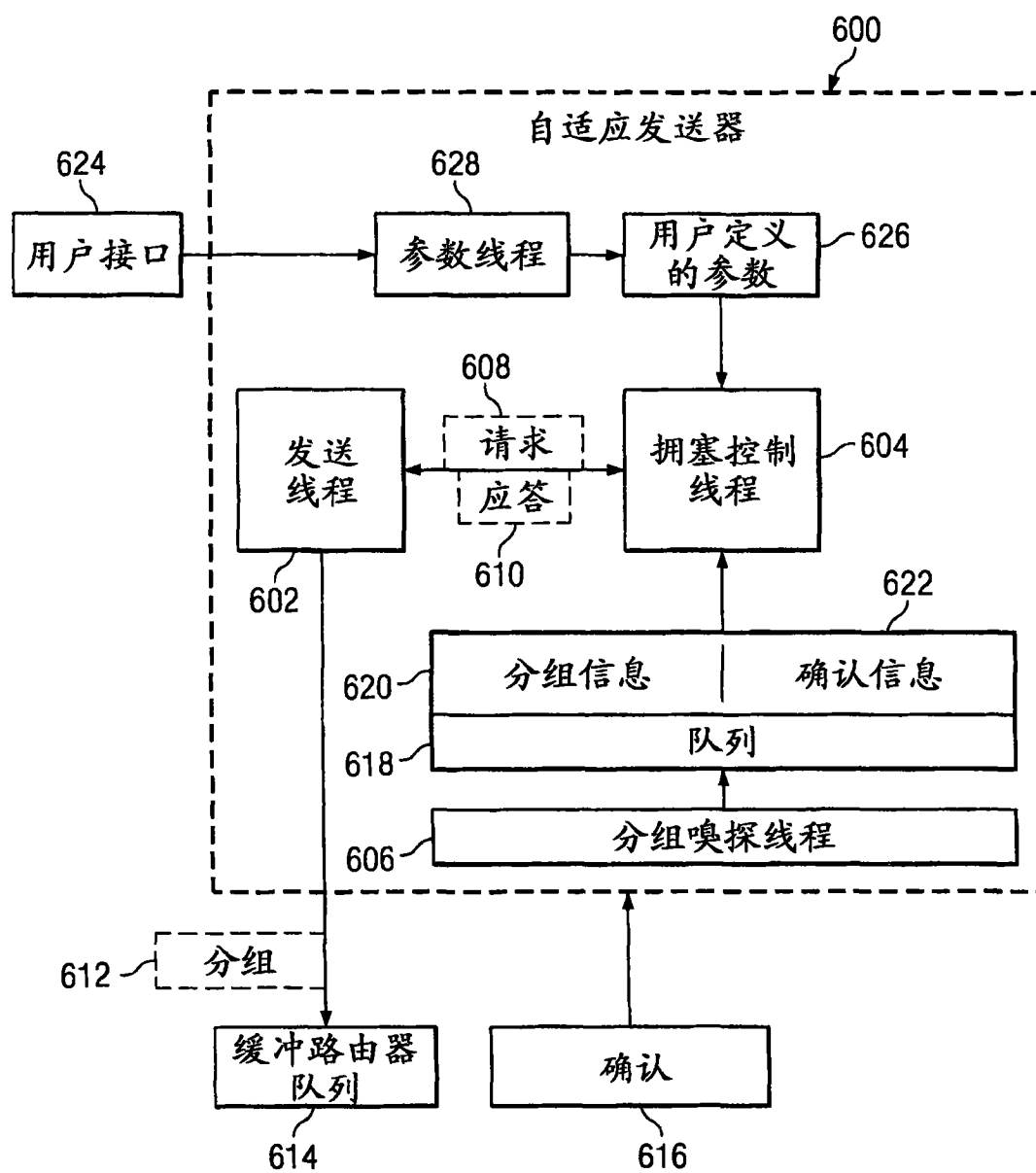


图 6

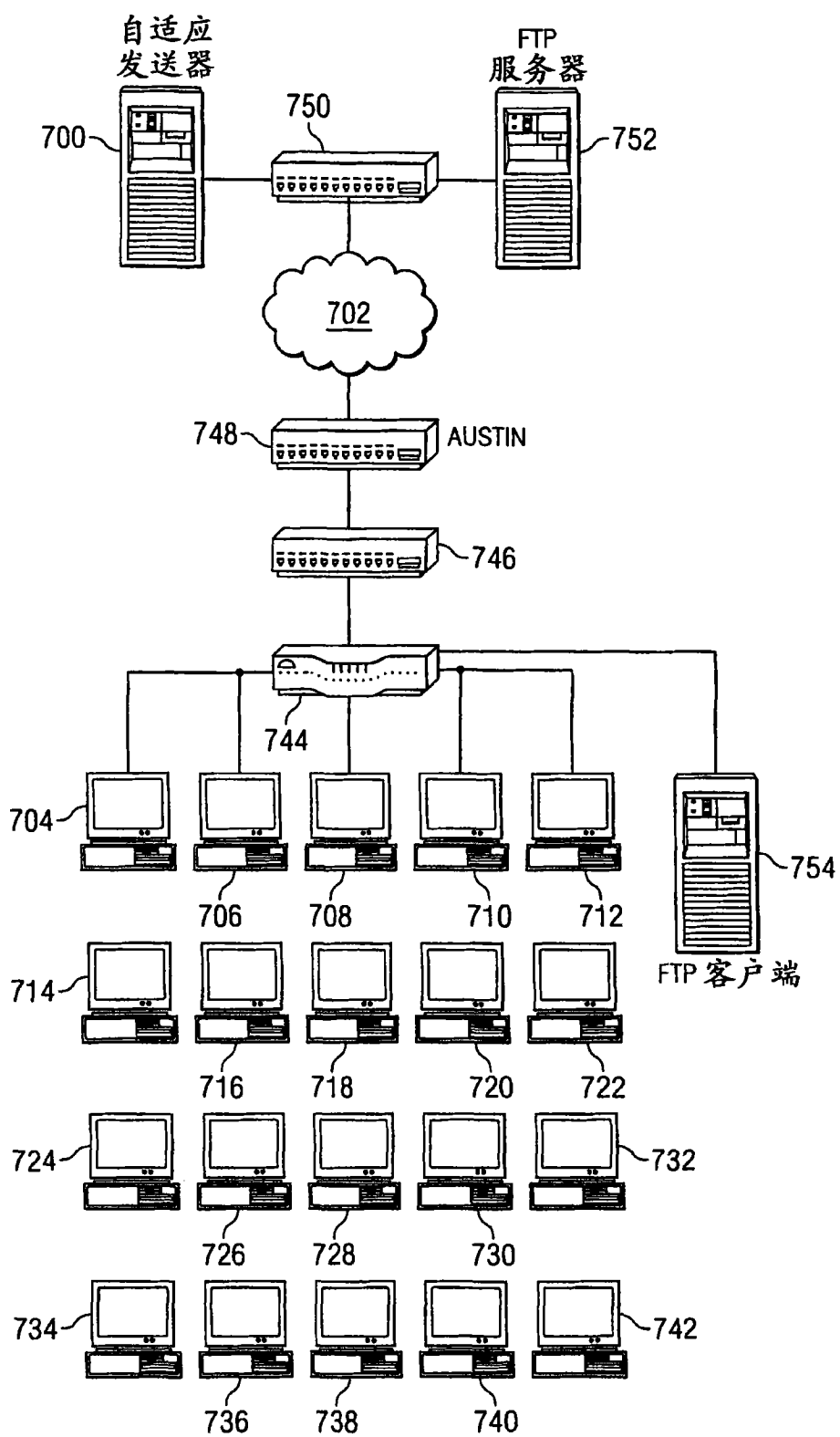


图 7

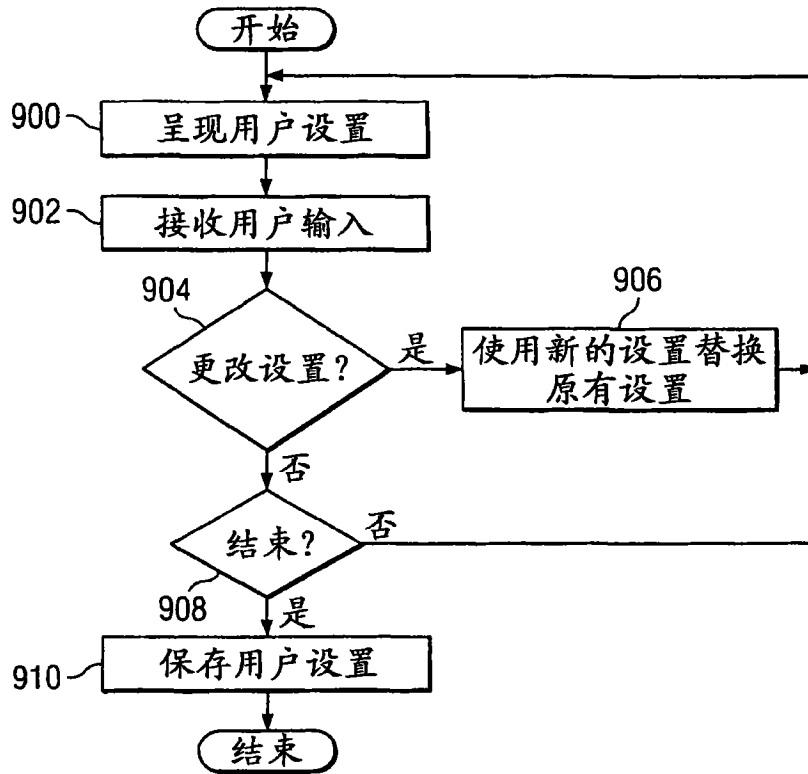


图 9

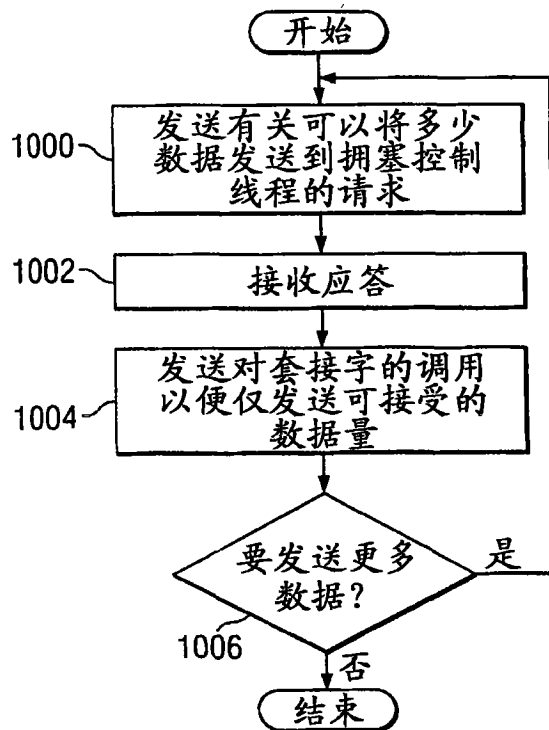


图 10

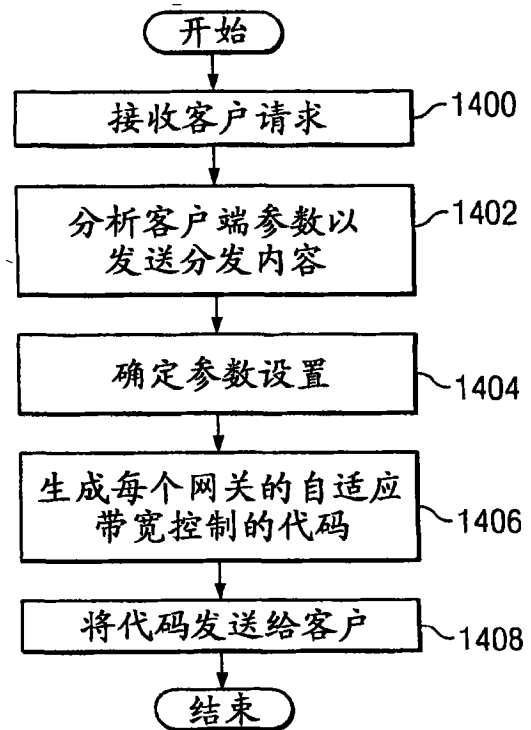


图 14

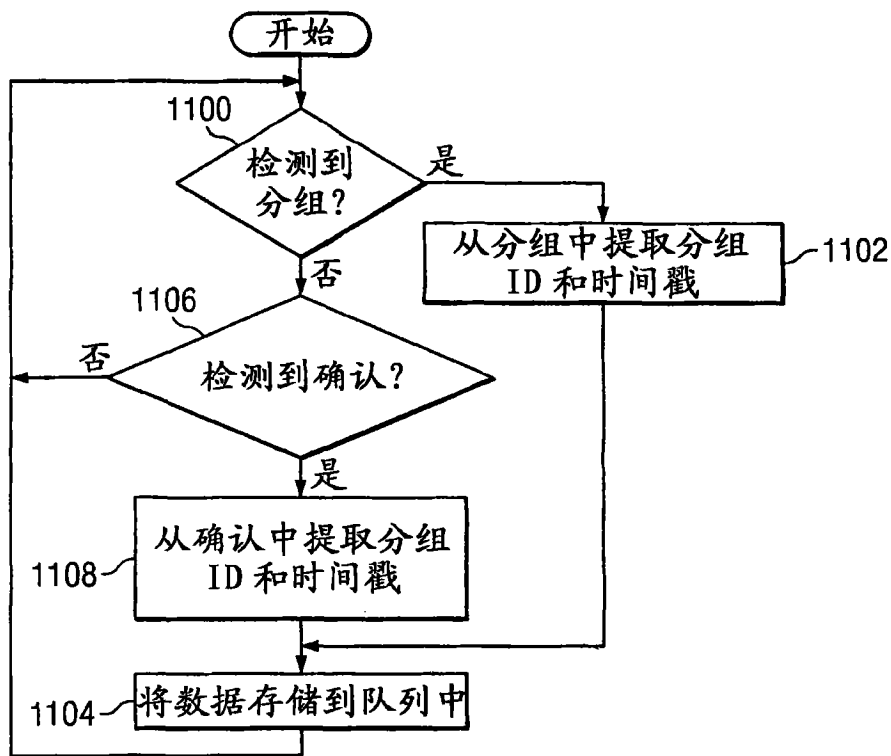


图 11

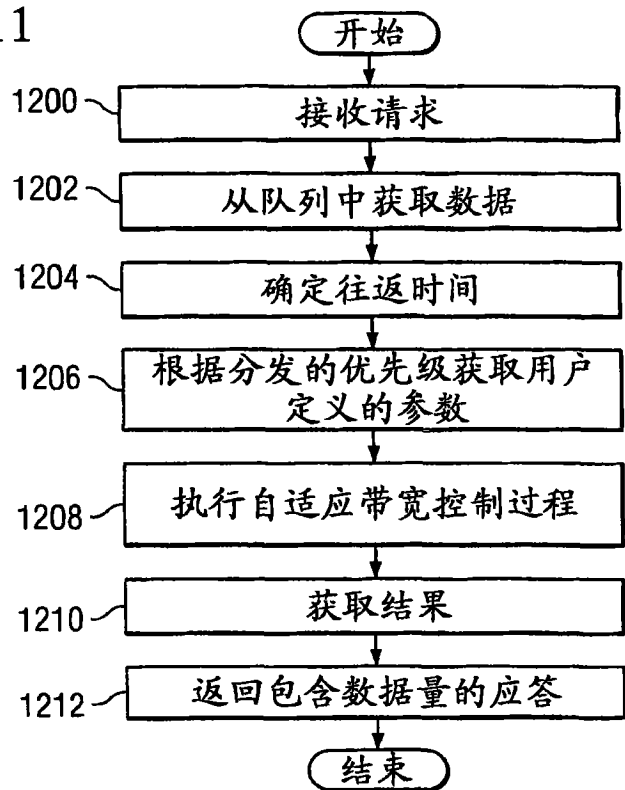


图 12

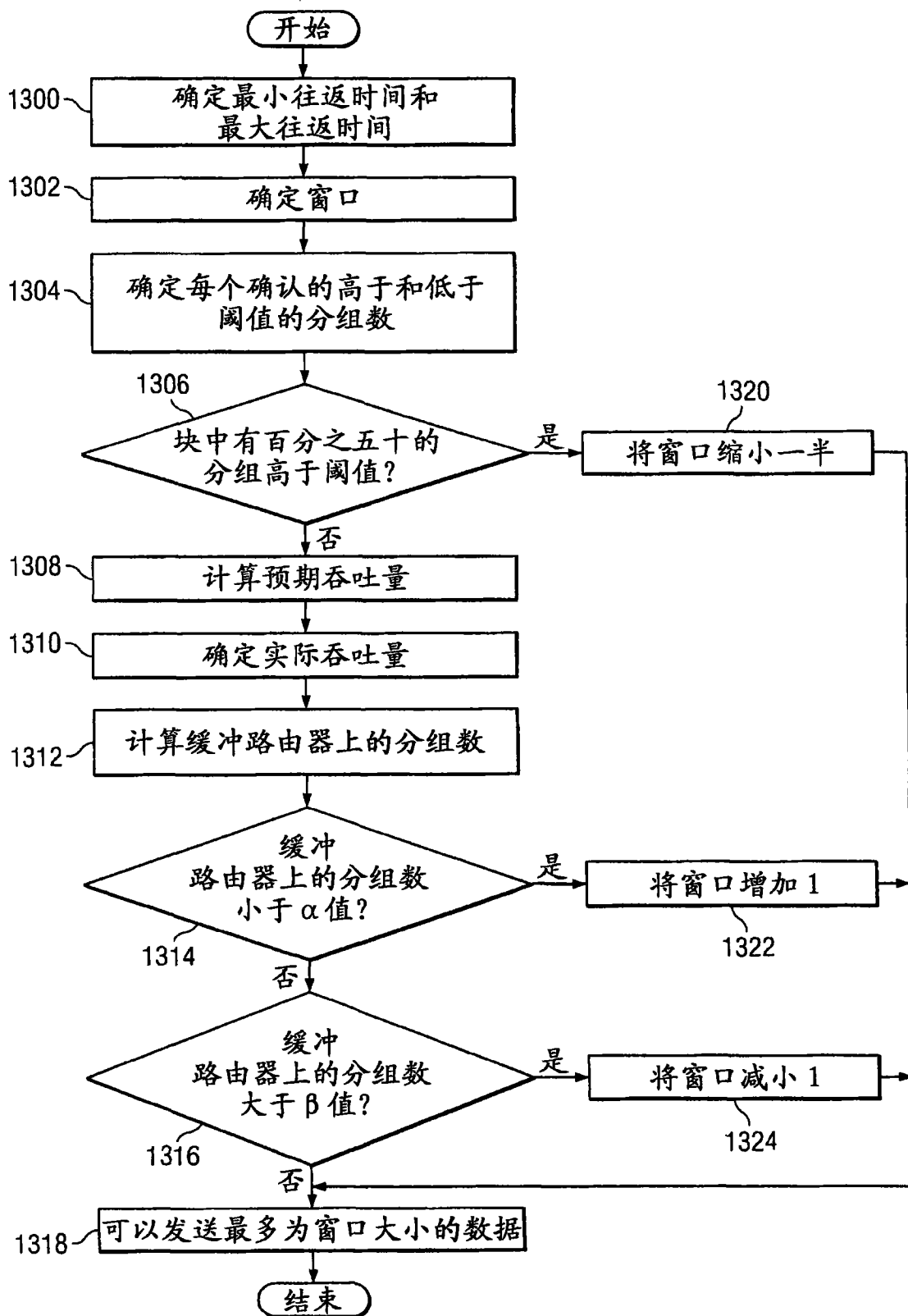


图 13