



(12) 发明专利申请

(10) 申请公布号 CN 105247504 A

(43) 申请公布日 2016. 01. 13

(21) 申请号 201380076334. 0

(51) Int. Cl.

(22) 申请日 2013. 08. 22

G06F 15/173(2006. 01)

(30) 优先权数据

61/772, 489 2013. 03. 04 US

13/944, 756 2013. 07. 17 US

(85) PCT国际申请进入国家阶段日

2015. 11. 03

(86) PCT国际申请的申请数据

PCT/US2013/056123 2013. 08. 22

(87) PCT国际申请的公布数据

W02014/137382 EN 2014. 09. 12

(71) 申请人 开放花园有限公司

地址 美国加利福尼亚州旧金山市

(72) 发明人 斯坦尼斯拉夫·沙昂奥维

格雷戈里·黑兹尔 米查·贝诺利尔

(74) 专利代理机构 广州华进联合专利商标代理

有限公司 44224

代理人 郑小粤

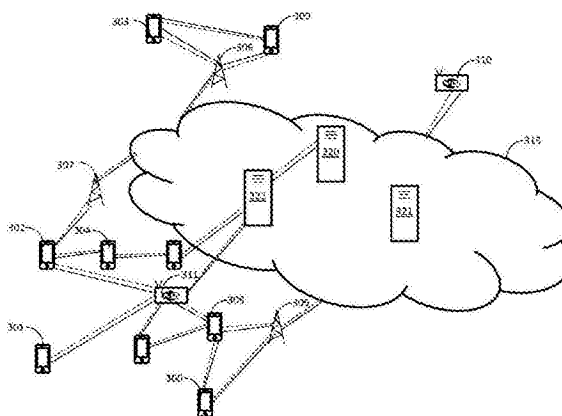
权利要求书3页 说明书10页 附图5页
按照条约第19条修改的权利要求书3页

(54) 发明名称

虚拟通道合并

(57) 摘要

使用多个通道建立因特网连接的方法。一个设备利用其内部可提供的和 / 或从相邻设备可提供的若干通道请求各种网页资源, 并使用来自不同通道的资源对网页进行组装。当一个设备能够使用多个内部通道连接至因特网时, 该设备使用内部试探法、通过这些通道来请求网页资源。云出口服务器可以用来提高安全性能, 并处理可能无法使用多个通道来处理的请求。



1. 一种在处理器可读代码中实现的方法,所述处理器可读代码在被执行时,会使计算设备执行若干步骤,所述步骤包括:

a. 与多个其他设备建立并同时保持多个连接,由此建立多个通信通道;

b. 通过向目标设备发送多个资源请求而获取多个网络资源,其中所述多个资源请求中的至少两个资源请求通过所述多个通信通道中的两个不同的通信通道被发送。

2. 如权利要求 1 所述的方法,其中所述保持多个连接包括保持每个所述通信通道的唯一的 IP 地址。

3. 如权利要求 1 所述的方法,其中所述多个资源请求中的至少一个资源请求在其传输之前被封装。

4. 如权利要求 3 所述的方法,其中被封装的请求被发送至云出口服务器,以由所述云出口服务器进行解封装,并转发至所述目标设备。

5. 如权利要求 1 所述的方法,其中所述多个网络资源请求中的至少一个网络资源请求包括 HTTP 范围请求,指定有待被发送至计算设备的一个文件部分。

6. 如权利要求 1 所述的方法,其中所述计算设备通过所述多个通信通道中的每一个通信通道发送相等数量的请求。

7. 如权利要求 1 所述的方法,其中所述计算设备按每一个资源请求中的字节数对所述资源请求进行衡量,并对通过所述多个通信通道中的每一个通信通道发送的所述字节数进行平衡。

8. 如权利要求 1 所述的方法,其中所述计算设备会保持所述多个通信通道中的每一个通信通道的以往性能的估算结果,并根据对所述以往性能的相应的估算结果来衡量要在所述多个通信通道中的每一个通信通道上发送的字节数。

9. 如权利要求 1 所述的方法,其中所述计算设备会保持所述多个通信通道中的每一个通信通道的以往性能的估算结果,并根据对所述以往性能的相应的估算结果来衡量将在所述多个通信通道中的每一个通信通道上发送的请求数。

10. 如权利要求 1 所述的方法,其中所述计算设备会保持对所述多个通信通道中的每一个通信通道的使用成本的估算结果,并根据对所述使用成本的相应的估算结果来衡量将在所述多个通信通道中的每一个通信通道上发送的请求数。

11. 如权利要求 1 所述的方法,其中所述计算设备还向服务器发送请求,以获取所述多个其他设备中的至少一个设备的连接信息。

12. 如权利要求 1 所述的方法,其中所述多个其他设备中的一个设备包括智能手机,且所述多个其他设备中的一个设备包括无线 WiFi 路由器。

13. 如权利要求 12 所述的方法,其中所述多个资源请求中的至少一个资源请求通过所述智能手机被发送至因特网。

14. 如权利要求 1 所述的方法,其中所述多个通信通道中的一个通信通道通过移动网络建立,且所述多个通信通道中的一个通信通道通过无线 WiFi 传输建立。

15. 如权利要求 1 所述的方法,其中所述多个通信通道中的一个通信通道通过移动网络建立,且所述多个通信通道中的一个通信通道通过蓝牙建立。

16. 如权利要求 12 所述的方法,其中所述多个资源请求中的至少一个资源请求通过所述蓝牙发送至因特网。

17. 一种用于以移动设备实现多通道通信的系统,所述移动设备具有在其内运行的用户应用件,所述系统包括:

驻留在联网服务器上的主应用件,所述主应用件使服务器执行以下方法,所述方法包括:

接收从所述移动设备发送的被封装的请求,并确定所述被封装的请求的原发地址;

对所述被封装的请求进行解封装,以获取被解封装的请求,并从所述被封装的请求确定目标地址,以确定所述被解封装的请求的目标;

使用所述目标地址将所述被解封装的请求发送至所述目标;

从所述目标接收对所述被解封装的请求的响应;以及

使用所述原发地址将所述响应转发至所述移动设备。

18. 如权利要求 17 所述的系统,其中确定所述被封装的请求的原发地址在对所述封装的请求进行解封装之后进行。

19. 如权利要求 18 所述的系统,还包括:

驻留在所述移动设备上的客户端应用件,所述客户端应用件使所述移动设备执行以下方法,所述方法包括:

拦截由用户应用件所生成的请求;

对所述请求进行分析,以确定所述请求是否可被译码,以及确定所述请求是否能由所述客户端应用件处理;

当所述请求不可被译码时,将所述请求封装在封装数据包内,其中被封装的请求包括源地址和目标地址,且所述封装数据包包括所述联网服务器的地址;以及

将所述被封装的请求发送至所述联网服务器。

20. 如权利要求 19 所述的系统,其中当所述请求能被译码时,将所述请求发送至所述目标地址。

21. 如权利要求 19 所述的系统,其中当所述请求能作为幂等型请求被译码时,使用多个通道将所述请求多次发送至所述目标地址。

22. 如权利要求 19 所述的系统,其中当所述请求可被译码时,将所述请求分为多个子请求,并使用所述多个通道将所述多个子请求发送至所述目标地址。

23. 如权利要求 19 所述的系统,其中当所述请求能被译码时,将所述请求发送至所述目标地址,且在接收到对所述请求的应答时,将所述应答的至少一部分保存在高速缓存中。

24. 如权利要求 23 所述的系统,还包括接收来自第二移动设备的转发请求,并确定是否将对所述转发请求的应答保存在所述移动设备的所述高速缓存内,如果确定保存,将对所述转发请求的应答发送至所述第二移动设备,而不转发所述转发请求。

25. 如权利要求 19 所述的系统,其中当所述请求能被译码时,将所述请求通过所述第二移动设备发送至所述目标地址,且在所述第二移动设备接收到对所述请求的应答时,将所述应答的至少一部分保存在所述第二移动设备的高速缓存中。

26. 如权利要求 19 所述的系统,其中当所述请求能被译码时,将所述请求通过所述第二移动设备发送至所述目标地址,且在所述第二移动设备接收到所述请求时,确定是否将对所述请求的应答保存在所述第二移动设备的高速缓存内,如果是,则从所述第二移动设备的高速缓存中取出所述应答,并将所述应答发送至原发移动设备。

27. 如权利要求 19 所述的系统,其中当所述请求能被译码以包含多个子请求时,确定是否将对至少一个所述子请求的应答保存在所述高速缓存内,如果确定保存,则将对至少一个所述子请求的应答提供给所述用户应用件,并将其余子请求发送至所述目标地址。

28. 如权利要求 27 所述的系统,其中所述多个子请求通过多个通信通道被发送,且其中至少一个所述通信通道包括到第二移动设备的点对点连接。

29. 一种在处理器可读代码中实现的方法,当所述处理器可读代码被执行时,会使计算设备执行若干步骤,所述步骤包括:

在开放式系统互连 OSI IP 层上接收通信请求;

分析所述通信请求,以对所述通信请求进行归类,并确定其是否能在更高的开放式系统互连 OSI 层上被处理;

当所述通信请求不能在更高的开放式系统互连 OSI 层上被处理时,通过 IP 层发送所述通信请求;

当所述通信请求能在更高的开放式系统互连 OSI 层上被处理时,通过所述更高的开放式系统互连 OSI 层发送所述通信请求。

30. 如权利要求 29 所述的方法,其中分析所述通信请求包括确定所述通信请求的端口号、所述通信请求的数据包类型和所述通信请求的内容。

31. 如权利要求 29 所述的方法,其中当确定所述通信请求能在所述开放式系统互连 OSI 应用层上被处理时,通过多个通道发送所述通信请求和所述通信请求的复本。

32. 如权利要求 29 所述的方法,其中当所述通信请求被归类为 HTTP 请求时,在所述开放式系统互连 OSI 应用层上通过多个通信通道发送所述通信请求。

33. 如权利要求 29 所述的方法,其中当确定所述通信请求不能在更高的开放式系统互连 OSI 层上被处理时,通过以下方式对所述通信请求进行封装:将目标地址和原发地址纳入被封装的请求中,并在所述通信请求的封装上采用云出口服务器的网络出口地址,允许所述云出口服务器对所述通信请求进行解封装,并使用所述目标地址转发所述通信请求。

34. 如权利要求 33 所述的方法,还包括对所述通信请求进行加密。

35. 如权利要求 33 所述的方法,其中对所述通信请求进行封装包括将具有源地址和目标地址的所述通信请求的数据包置于具有上述云出口服务器的地址的封装数据包内。

36. 如权利要求 33 所述的方法,其中对所述通信请求进行封装包括将所述通信请求的数据包分断为多个子数据包,并将具有源地址和目标地址的每个所述子数据包置于具有上述云出口服务器的地址的封装数据包内,由此生成多个被封装的请求。

37. 如权利要求 36 所述的方法,其中所述多个被封装的请求通过多个通信通道被发送。

虚拟通道合并

[0001] 相关申请

[0002] 本申请涉及并主张 2013 年 7 月 17 日提交的序列号为 13/944,756 的美国专利申请的优先权,后者涉及并主张 2013 年 3 月 4 日提交的序列号为 61/772,489 的美国临时专利申请的优先权,该申请的披露内容通过参考整体并入本文中。

背景技术

[0003] 1. 领域

[0004] 本披露内容涉及无线连接,具体涉及采用多个通道建立连接。

[0005] 2. 相关技术

[0006] 访问因特网等网络可采用多种有线和无线技术。例如,采用现有技术的智能手机可使用 3G、4G、WiFi 及其他类似的无线技术访问因特网。此外,无线技术可实现两个或更多设备之间的互连。这些技术包括近场通信 (Near Field Communication, NFC)、WiFi 直连 (WiFi Direct)、蓝牙等。

[0007] Tethering(网络共享)是一种需要用户进行很多操作并具备相当多知识的连接规程,这使其基本上停留在“极客功能 (geek feature)”的范畴内,主要是由通晓技术的用户使用。Tethering 最常用于在 WiFi 或其他因特网连接不可用时,将电脑连接至手机,以通过手机网络访问因特网。除了在建立 tethering 时需要用户操作之外,各个运营商和手机制造商还对 tethering 设置了障碍,这就导致了各种绕道式的“创新”,例如对安卓设备进行 root,或者对 iOS 设备进行越狱,并在设备上安装 tethering 应用件。

[0008] 一般而言,当应用件要求访问因特网时,设备将从可用通道(例如 WiFi)中选择一个通道,并在所选通道上执行应用件所要求的所有通信。例如,当智能手机上的浏览器请求一个页面时,用于该页面的所有资源均会被请求,并在诸如 WiFi 的一个通道上被接收,尽管还有其他通道可用,例如 4G。

[0009] 此外,不同的设备可能采用不同的运营商,这使得在一个地点可能有若干设备,每一个设备均采用不同的运营商,由此具有不同的服务水平。

发明概要

[0010] 包括以下本发明的概要,以对本发明的一些方面和特征提供基本的了解。本概要并不是本发明的全面概述,由此其并非意在具体明确本发明的主要或关键要素,或要描述本发明的范围。其唯一目的是以简化的形式给出本发明的一些概念,以引出下文中所给出的更为详细的说明。

[0011] 所披露的各种实施例提供了使用多个通道建立因特网连接的方法。一个设备会利用其内部和/或从相邻设备可提供的若干通道来请求网页的各种资源,并使用抵达各个不同通道的资源对网页进行组装。实施例可实现为在设备上运行的客户端,例如在智能手机或平板电脑等移动设备上运行的 app。作为简写,此客户端在本文中有时可被称为开放花园 (Open Garden) app。开放花园 app 与其他 app 一起在移动设备上运行,并监测在移动设备

上执行的其他 app。当一个 app 试图与外部设备（例如因特网上的服务器）通信时，开放花园将拦截该通信请求，并确定如何以最佳的方式将该请求发送至外部设备。开放花园还可拦截从外部设备进入的通信，并确定是否在内部路由该通信，即路由至哪一个 app 来转发该通信，或者确定其是否需要被转发至另一外部设备。

[0012] 当一个设备能够使用多个内部通道拦截因特网时，该设备使用内部试探法通过使用这些通道来请求网页资源。例如，一个智能手机设备具有移动网络 (cellular network) 无线和 WiFi 无线方式。然而，常规情况下，该智能手机只会使用这些通道中的一个来连接到因特网并请求网页资源。根据所披露的实施例，智能手机会使用这两个通道来请求网页资源，然后使用通过两个通道接收到的资源来组装并显示网页。

[0013] 根据其他实施例，一个设备还可使用其他设备来请求网页资源，由此对多个通道加以利用。例如，一个智能手机设备可能与另一智能手机设备有蓝牙连接。第一个智能手机可以使用其自身内部的通道（例如，移动网和 WiFi）来请求网页资源，但也使用其与第二个智能手机的蓝牙连接、使用第二个智能手机通道来请求其他网页资源。

[0014] 根据一些实施例，移动设备通过让每个通道使用其自己独有的 IP 地址，使用多个通道来请求网页资源。所请求的网页资源被返回至每个发出请求的 IP 地址，所有这些均指向发出请求的设备。发出请求的设备随后使用返回的资源对网页进行组装。另一方面，根据其他实施例，例如对于安全页面 (https:)，目标设备必须将各请求视为来自同一 IP 地址。为实现此目的，一个云出口服务器被连接至因特网。来自所有移动设备通道的所有请求均被指向该云出口服务器。该云出口服务器使用同一 IP 地址、即云出口服务器自己的地址将请求转至合适的主机。由此，从目标主机的角度看，所有请求均来自同一个 IP 地址，即同一个设备。这样，主机将所请求的资源返回给发出请求的 IP 地址，即云出口服务器 IP 地址。随后该云出口服务器将所接收的资源转发给合适的发出请求的各 IP 地址。由此，从移动设备的角度看，采用了多个通道进行请求的发送和资源的接收。

[0015] 所披露的各种实施例可实现对因特网的多路径访问，以提供更高的可靠性和带宽。此外，各种实施例均可无需配置选择：用户将不再需要选择让其设备通过何种方式连接至因特网，因为设备就会简单地同时使用多种方式。此外，设备还会自动查找访问因特网的可用路径。例如，如果一个路径失效，将会选择一个新的路径，并建立新的连接。因此，网络能自我修复和自我建构。每个节点仅以其本地的知识运行，但已连接的设备一起会使用一种概率分布式算法构建一个网络。使用网状网络，当没有直接的因特网连接时，设备将通过其他设备的链来访问因特网。如有必要，这些链将通过连接至其他设备而增长，以连接至因特网。所述实施例使用户能够采用最合适的连接来访问因特网，而不需要配置其设备或进行跳转 (jumping through hoops)。各实施例还使用户能够以尽可能低廉的价格访问因特网。用户无需查看每个可用网络即可找到最快的连接和最强的信号，并可在网络之间无缝转移。各实施例提供了在更多的位置以更快的速度访问更多数据的方式。用户成为了网络的一部分，在他们能够提供可能的最佳接入方式的时间和地点共享连接。这将可实现更高质量的流视频和音频、更具即时性的多玩家游戏和更快的下载。

[0016] 附图简要说明

[0017] 纳入本说明书并作为其组成部分的附图例示了本发明的实施例，并与文字说明一起用以解释和例示本发明的原理。附图意在以简图形式例示出示例实施例的主要特征。附

图并非要描述实际实施例的每一项特征,也并非要给出所示元件的相对尺寸,且并非按照比例绘制。

[0018] 图 1 为示出根据一种实施例的使用多个内部通道请求 web 资源的一种移动设备的示意图。

[0019] 图 2A 为示出根据一种实施例的使用多个外部通道请求 web 资源的一种移动设备的示意图。

[0020] 图 2B 为示出根据一种实施例的使用多个外部通道和一个云出口服务器请求 web 资源的一种移动设备的示意图。

[0021] 图 2C 为示出根据一种实施例的使用多个内部通道和外部通道请求 web 资源的一种移动设备的示意图。

[0022] 图 2D 为示出可由一种实施例执行的一个流程的示例的示意图。

[0023] 图 2E 为示出可由根据一种实施例的云出口服务器执行的一个流程的示例的示意图。

[0024] 图 3 所示为根据本文所披露的各种实施例运行多个连接的环境的示例。

[0025] 图 4 所示为一个网状网络的示例,其中根据本文所披露的各种实施例,若干设备使用多个通道。

[0026] 图 5 所示为根据本文所披露的各种实施例运行多个连接、并包括一个云路由 oracle 服务器的环境的示例。

[0027] 详细说明

[0028] 下文提供了并行使用多个通道建立因特网连接的方法和系统的示例。

[0029] 虚拟通道合并是一种计算机联网技术,它可以在有多个因特网连接可用时,提高计算机、智能手机、平板电脑或其他设备的因特网连接的可靠性、速度和可用性。特定设备自身可能具有其可使用的多种因特网连接,或者是每个设备仅有一个连接,而通过网状网络共同访问因特网并在本地组网,由此设备可实现虚拟通道合并。虚拟通道合并可以得益于任何网络设备,但并不要求任何网络设备(例如交换机或路由器)来改变其行为或规范,因此易于部署。然而,由于它可由网络设备与终端系统(例如计算机、智能手机、平板电脑、智能电视或其他设备)一起使用,其可被用于在更大规模上显著改善因特网连接性。虚拟通道合并还可改进在本地网络上进行设备间通信的现有方式,或者为其提供专用的方式。

[0030] 在现有技术中,每一个被启用的设备均使用其自有的资源访问因特网。当因特网连接失效时,设备将不再能访问因特网,直至其重新建立连接或者发现并建立了另一个连接。这些设备采用单一的通道来与因特网进行连接和通信。例如,一个可使用 4G 的智能手机将使用 4G 连接与因特网进行连接和通信。然而,如果该智能手机发现了一个 WiFi 连接并与之连接,它将使用 WiFi 连接而非 4G 连接来与因特网进行连接和通信。

[0031] 也就是说,只要 WiFi 连接可用,该设备就将使用该通道进行所有因特网通信。然而,该特定通道可能较慢,或者在内部或外部可用的其他通道可能提供更好的保真度。此外,多个通道上的并行通信可以提高因特网通信的速度和保真度。

[0032] 在本语境中,网状网络是通过设备之间的直接点对点(ad hoc)链路建立的网络,可用于在包含网状网络的设备之间进行本地通信。在一些示例中,网状网络可以由因特网上的服务器进行协调。根据所披露的实施例,虚拟通道合并可以利用网状组网作为在设备

间建立附加连接的方式之一,并为因特网连接和通信提供附加通道。为此,网状网络可以使用任何物理介质来建立端到端或局部的连接,从诸如以太网的有线连接,到无线连接,例如处于接入点或点对点模式的 Wi-Fi、Wi-Fi 直连、蓝牙、ZygBee、NFC、3G 技术或各种 4G 技术,4G 技术例如是 LTE 和 WiMAX 等。对于虚拟通道合并,会考虑底层技术的具体属性,但任何底层联网技术均可被采用。应注意,按照定义,网状网络的任何组合也是网状网络,即使其偶尔被断开。

[0033] 图 1 示出了一个环境,其中诸如智能手机或平板电脑等通信设备 100 能够与耦合至因特网 115 的服务器 120 进行通信,其通信使用的可以是采用无线连接 112 的 WiFi 110 接入点,也可以是采用诸如 3G、4G、LTE 及其他类似协议的移动网络 105。例如,移动网络 105 可以是一个无线通信提供商的标准无线电话系统,而 WiFi 100 接入点 110 可以是一个通过诸如电缆或陆基电话线连接至因特网的无线路由器。一般而言,此类通信采用可用连接 107 或 112 中的任何一个来完成。对于多数移动设备,如果 WiFi 接入 112 可用,则与服务器 120 的所有通信均将通过 WiFi 连接进行。例如,当一个智能手机检测到一个 WiFi 路由器的存在时,如果智能手机可识别该 WiFi 路由器,它可以自动与之连接,如果它无法识别,它可以请用户确定是否进行连接。另一方面,如果 WiFi 连接 112 不可用,则移动连接 107 将会被用于所有通信,即语音和数据。这种安排的理由之一是,一个因特网连接需要一个 IP 地址,以使服务器 120 “知晓”其在与谁通信。由此,TCP-IP 标准限制设备 100 仅通过单一连接,即移动连接 107 或 WiFi 连接 112,与因特网通信。这种局限将通过下文所述方式加以缓解。

[0034] 在图 1 的实施例中,设备 100 的处理器执行特定的指令(例如,客户端 app),使其能够使用移动通道 107 和 WiFi 通道 112 这两种方式与因特网和服务器 120 进行通信。例如,当设备 100 上的一个浏览器尝试从服务器 120 下载网页时,此进程将需要反复发送针对构成该网页的各种资源(例如,文本、图像、Java 脚本及其他资源)的请求。然而,处理器并不是像在现有技术中一样使用单个通道发送所有请求,而是采用了一种分配机制,使用两个通道 107 和 112 发送各种请求。在请求到达服务器 120 时,服务器将获得来自两个不同 IP 地址的请求;然而,对服务器 120 而言,这无关紧要:服务器仅需将所请求的每个资源发送至特定的请求地址即可。在资源到达设备 100 时,它们会被高速缓存,并被组装在一起以形成所请求的网页。也就是说,设备 100 的处理器知道,到达两个通道的所有资源均涉及所发送的请求,并一起构成所请求的网页。

[0035] 关于选择在哪个通道上发送哪个请求的问题,可以采用各种算法或试探法,如编号 101 所例示。例如,简单的方法就是在两个通道之间交替,使每一请求均在与前一请求不同的通道上发送。另一个示例是要考虑到每个通道的运行速度,根据运行速度发送请求。在一个示例中,对图像和视频等重资源的请求通过较快的通道发送,而文本等轻资源则通过较慢的通道发送。根据另一个示例,会考虑到每个通道的服务成本,例如,重资源将通过较廉价的通道发出请求,而轻资源则通过较昂贵的通道发出请求。

[0036] 在图 1 的示例中,设备 100 是一个终端系统,其会请求和接收来自其他设备的资源,但它不会向其他任何设备发送资源,即,它不充当其他任何设备的中介或因特网接入点。这并非该实施例的要求,只是会使其更容易解释。在下文的多数说明中,此示例将被采用,以使解释更为简单和清晰。然而,发出请求的设备并不需要都是终端系统。而是,该设备可以为自己请求 web 资源,也可以为了向网状网络中的其他设备中继而请求 web 资源。

[0037] 在图 1 的系统中, 移动设备 100 使用其内部通道直接与因特网通信。然而, 有时可能无法或不希望使用移动设备内部通道进行通信。例如, 在海外旅行期间, 使用移动设备 100 直接与因特网通信可能会过分昂贵, 或者不属于用户的服务提供商协议的范围。此问题可通过采用图 2A 所示示例加以缓解。

[0038] 在图 2A 中, 移动设备 200 请求并接收来自服务器 220 的 web 资源, 而与因特网没有直接连接。例如, 设备 200 可能是在外国旅行的用户的智能手机, 用户可能无法或不愿意连接当地的无线提供商网络。因此, 在本例中, 移动设备 200 采用端到端连接, 以使用诸如蓝牙、NFC 及类似协议与设备 201 和 202 形成一个网状网络。为了示例的目的, 示出了两个蓝牙连接, 一个将设备 200 连接至设备 201, 一个将设备 200 连接至设备 202。设备 200 使用端到端连接来指示设备 201 和 202 中的处理器请求 web 资源, 每个设备均使用其自己的 IP 地址, 并通过端到端网络将所接收的资源转发给发起请求的设备 200。

[0039] 在图 2A 的示意图中, 实线表示物理层连接, 而虚线表示数据通信。可以看出, 设备 200 与设备 201 和 202 之间既有物理层又有数据通信。在图 2A 的具体示例中, 两个连接均被识别为蓝牙连接, 但也可采用其他端到端协议。反之, 在图 2A 的具体示例中, 设备 201 使用 4G 移动网络与因特网通信, 而设备 202 则通过 WiFi 连接与因特网通信。

[0040] 如图 1 示例所示, 设备 200 可采用各种惯例或试探法来确定哪个请求要通过哪一设备发送。例如, 设备 200 可测量网状网络中每一设备的响应时间, 由此得出其各自的带宽, 并在分配请求时使用这一信息。此外, 网状网络中的每一设备还可通报其带宽成本, 发出请求的设备可能使用该信息来确定 web 资源请求的分配。例如, 设备 200 可能会保存有一个列出与之通信的每个端的成本和连接速度的表, 并使用此表确定对每个原发请求使用哪个端。

[0041] 图 1 和图 2A 的实施例假定服务器 120 或 220 能够对来自不同 web 地址的不同的 web 资源请求提供服务。然而, 在某些情况下这无法实现。例如, 加密的请求, 例如 https 或诸如 Skype® 等即时通信请求必须使用单一的 IP 地址进行处理。图 2B 的示例示出了在服务器 220 必须使用单一 IP 地址对请求提供服务的情况下, 一个设备如何能够在多个通道上发送 web 资源请求。

[0042] 在图 2B 的示例中, 设备 200 与设备 201 和 202 形成一个网状网络, 并使用那些设备从因特网 (例如服务器 220) 获取资源。然而, 在这一具体示例中, 无法使用两个不同的 IP 地址来对请求提供服务。相应地, 在本例中, 采用云出口服务器 222 来接收请求, 并使用单一的 IP 地址将请求中继至目标设备, 例如服务器 220。当目标设备使用云出口服务器 222 的单一 IP 地址将所请求的资源返回至云出口服务器 222 时, 云出口服务器 222 会通过到达的请求所来自的 IP 地址或通过其他通道将响应发送至原发设备 200。

[0043] 在一个示例中, 设备 200 的处理器会执行用以操作 web 请求的指令。在本例中, 这通过运行应用件的设备 200 来实现, 该应用件在本文中称为开放花园 (Open Garden)。在开放花园可理解用于请求资源的协议并能够通过多个路径发送请求的所有情况下, 设备 200 将使用可用通道发送请求, 如图 1 和图 2A 中的示例所示。另一方面, 当开放花园不理解协议或理解协议但无法将请求分入多个通道时, 它会对请求进行封装, 并将其发送至云出口服务器。设备 200 可通过其选择的任何通道发送请求, 因为从中继设备 (例如设备 201 和 202 中的任何一个) 起, 请求仅需被发送至云出口服务器 222 的 IP 地址, 而这些设备仅需将

请求中继至该 IP 地址。

[0044] 可由开放花园应用件执行的一个进程流程的示例在图 2D 中示出。在 281 步, 进程例行地检查新的请求。在 282 步, 当收到一个请求时, 进程会尝试对请求进行译码, 如果成功, 进程将行进至 282 步, 以选择用以发送请求的通道。当选择合适的通道后, 请求在 284 步被接收。反之, 如果请求不能被译码, 进程将行进至 285 步, 在此请求会被封装, 在 286 步, 云出口服务器 222 的地址被应用于请求。随后进程行进至 283 步, 以选择通道, 被封装的请求在 284 步被发送。

[0045] 当云出口服务器 222 接收到请求时, 它会对请求进行解封装, 并确定原发设备和目标设备的地址。如果收到的数据包仅为一个部分请求, 云出口会对从所有通道接收的所有部分请求进行组装。一旦云出口获得整个请求后, 它会将其发送至目标目的地服务器, 例如服务器 220。目的地设备接收的请求以云出口服务器地址作为其原发地址。由此, 目标设备会将应答发送至云出口服务器 222。当云出口服务器 222 接收到应答时, 它会通过中间请求设备 (例如设备 201 和 202) 或其所选择的其他任何合适的通道将应答中继至原发设备 200。也就是说, 由于云出口服务器 222 知道该请求源自何处, 它可以使用任何可用通道发送应答。

[0046] 可由云出口服务器 222 执行的一个进程的示例在图 2E 中示出。在一个请求在 291 步被接收后, 进程行进至 292 步, 以检查所接收的请求是部分还是完整的请求。如果它是完整的请求, 则在 293 步, 目标服务器 220 的地址会被应用于该请求, 随后该请求在 294 步被发送。另一方面, 如果该请求仅为一个部分请求, 则在 295 步, 对该请求的其余部分进行接收, 在 296 步对整个请求进行组装。随后进程转回 293 步, 以应用目标的地址, 并在 294 步发送请求。

[0047] 由此, 例如, 如果设备 200 通过设备 201、使用云出口 222 的 IP 地址作为其目的地发送请求 1, 并通过设备 202、同样使用云出口 222 的 IP 地址作为其目的地发送请求 2, 设备 201 和 202 将使用云出口服务器 222 的 IP 地址中继请求。当云出口服务器 222 接收到请求 1 和 2 时, 其会对它们进行解封装, 并发现原发设备为设备 200, 目标设备为服务器 220。因此, 它采用其自己的 IP 地址作为请求发出方, 将请求中继至服务器 220。当目标服务器 220 接收到以云出口 222 的 IP 地址作为发出方的请求 1 和 2 时, 它会通过将应答 1 和应答 2 发送至云出口服务器 222 的 IP 地址来满足请求 1 和请求 2。当云出口服务器 222 接收到应答 1 和应答 2 时, 由于它知道其请求源自于设备 200, 它可以使用任何可用通道将响应中继至设备 200, 并不一定通过设备 201 和 202。

[0048] 如上文所述, 设备 200 可使用其自己的多个通道访问因特网。同样, 设备 200 还可使用多个已连接设备的通道访问因特网。设备 200 可同时利用这两种方法, 即使用内部通道和所连接的设备。图 2C 示出了一个示例, 其中, 设备 200 采用其内部通道、经移动网络 105 和 WiFi 设备 110 连接至因特网, 还使用以网状网络连接的两个设备 201 和 202 连接至因特网。此外, 设备 201 和 202 中的每一个还可使用多个通道连接至因特网, 设备 200 同样也可利用这一点。例如, 图 2C 示出, 设备 202 通过一个移动网络并通过相同的 WiFi 接入点 110 (尽管它可能已经同样轻易地使用了另一个 WiFi 接入点) 连接至因特网。

[0049] 图 3 示出了根据本文所披露的各种实施例运行多个连接的一个环境的示例。图 3 仅示出了整个环境中的一个非常小的部分, 其中可能包括相互连接为多个网状网络的许多

设备。例如,设备 300、301 和 302 与其他设备连接至一个网状网络,而设备 303 和 309 则在另一个网状网络中互联。设备 300 使用其移动收发器与移动塔 305 通信,同时使用第二收发器(例如蓝牙收发器)与移动设备 308 通信。由于移动设备 308 通过 WiFi 设备 311 被连接至因特网,设备 300 可使用此连接与因特网服务器、例如服务器 320 和 321 进行通信。此外,云出口服务器 322 可以服务于通常无法使用多个连接处理的流量,例如,加密或 VOIP 流量。由此,设备 300 采用一个通道、经移动网络与因特网直接通信,而采用第二通道、通过移动设备 308 连接至一个 WiFi 设备。另一方面,设备 302 使用 3 个通道:一个移动通道、一个 WiFi 通道和一个由其连接至设备 304 的通道。

[0050] 如上文所述,虚拟通道合并可通过若干技术提高设备之间的网络连接和设备间的因特网连接的速度、可靠性和可用性。在可能的时候,虚拟通道合并可使用网状联网,以连接相互访问的所有设备或一起访问因特网的所有设备。需要注意的是,到目前为止,所述的虚拟通道合并对于没有因特网连接的环境或与有无因特网连接无关的环境也有利。例如,图 4 示出了一个显示多个设备 400-409 相互连接的环境,其中至少有一些设备、例如设备 401-409 会使用多个通道。由此,此网状网络中的每个设备均可使用一个或多个路径与另一设备通信。这可能在诸如社交网络设置、设备间直接聊天和文本传输以及其他可能的应用中发挥作用。当然,当网络中的一个设备能够访问因特网时,它可以立即充当其他所有互联的设备访问因特网的网关。

[0051] 在本文所述的虚拟通道合并的方法中,可以对通信流量进行分析,以了解其某些属性。也就是说,尽管常规情况下处理的是一特定 OSI(开放式系统互连)层上的流量,而无视在更高或更低的 OSI 层上发生的任何情况,本发明的实施例会对流量进行分析,并确定使用哪个 OSI 层。具体而言,本方法会查看通信的属性,并在该层上对其进行处理。例如,本方法可检测 web(HTTP) 请求、DNS 请求、BitTorrent 流量和 HTTPS 请求,并以对特定请求最高效的方式处理每个请求。有些流量可能无法被译码,其可被保留为未归类,但对尽可能多的流量进行归类通常会有好处。原因在于,对更高 OSI 层上的请求进行服务通常会产生更好的性能。

[0052] 以下是对流量的分析和归类如何利用虚拟通道合并来改善通信的一些示例。第一个示例是在确定请求为幂等型之时。例如,当检测到通常为幂等型的 HTTP 请求(例如 GET 请求)时,系统可以尝试重试,或者甚至在多个路径上发送冗余的查询。在这种情况下,请求可被复制,并在不同通道或不同路径上同时发送。同样,如果在一个特定时间内未收到响应,即使该请求可能尚未失败,也可在相同或不同的通道上再次发送该请求。由于请求为幂等型,对于接收到多次请求的服务器来说并无关系。另一方面,如果接收到两个响应,它们可保证相同,以使所接收到的后一个响应可被丢弃。同样,为确保这种实现方式的可靠性不低于仅使用单一通道(称为默认通道)运行的移动设备的可靠性,本方法将尝试在其他通道之外还使用默认通道发送第一请求。

[0053] 在另一个示例中,如果确定请求为诸如 web 或 DNS 请求等,一个合适的应答可能已经驻留在原发或网状联网设备的高速缓存中。使用高速缓存可在网络使用量和加载速度方面实现中等到大幅度的降低。对于 DNS 请求,系统可以提供附加的高速缓存层,并且以类似的智能方式将其作为单元进行路由,即使它们并不在同一个 IP 数据包内到达。

[0054] 为了提供一个具体示例,如果两个用户相互距离较近,且他们的移动设备运行根

据本文所述一种实施例的一个客户端应用件,则每个设备均可至少使用其自己的移动网络连接、其自己的 WiFi 连接和使用网状网络、彼此的移动和 WiFi 连接进行通信。在这种情况下,如果两个用户均决定登录诸如脸书 (Facebook) 等,则实际上无需两个设备均下载脸书样式表,因为它总是相同的 - 只有内容对不同用户有差异。由此,当第一设备下载脸书样式表时,它可以将其存储在高速缓存中,当第二设备请求脸书样式表时,可以将其从另一设备的高速缓存中发送至第二设备,而不是将请求实际发送至脸书服务器。

[0055] 在以上给出的脸书的示例中,流量的减少量将会有适度的增加。然而,当一个网状网络中的许多用户之间存在某种局部相关性时,可能形成大得多的增加效果。例如,一个会议中的许多用户都希望查看同一份演示幻灯片。并不用每个用户都下载该演示文稿,而是只有一个或几个设备可以下载该演示文稿,并使用高速缓存向网状网络中的其他设备提交演示文稿,由此使网络流量大幅度降低。

[0056] 应注意, DNS 请求几乎总是幂等型。因此,如果应答没有在高速缓存中,系统可以使用上文所述的处理幂等型请求的方法处理 DNS 请求。同样,由于 DNS 请求较小,发送冗余 DNS 请求的开销也很低,但好处可能在于其可使运行更为稳定,由此使收益相当高。

[0057] 不透明和被加密的流量通常由系统在 IP 层上处理,对于 HTTPS 流量,则在 TCP 层上处理。流量通常在 IP 层上被注入系统中,但与仅单纯在 IP 层上发送此类流量的现有技术不同的是,本方法会分析流量,以查看其使用一个不同的层是否有利。甚至将 TCP 层作为字节流而非 IP 数据包流处理都会在实践中形成显著的性能提升,因为网状网络可能经常在非拥塞数据包丢失率相对较高的介质上运行,由此,如果被作为 IP 数据包流进行处理, TCP 连接的性能可能会受网状网络上数据包丢失的限制。

[0058] 一般而言,系统可在若干层上处理流量:IP 层,在此数据包被接收和转发;TCP 层,在此字节流被接收和转发;应用层,在此应用请求被接收和转发。在本语境下应注意,当可以在可能的最高层(例如,应用层)上处理请求时,所披露的实施例的优点可得以最大化。例如,将物理层连接加倍将不会使接收给定请求的响应的速度更高。另一方面,HTTP 或 DNS 请求的加倍可以提高获得响应的速度和可靠性。因此,即使在一个请求在 IP 层被注入时,也会对其进行分析,以查看其是否能在一个更高的层上被处理,例如,其是否为 HTTP 或 DNS 请求。

[0059] 可以将优先权赋予在流量中占到显著部分或在用户的应用方面所花时间中占到显著部分的应用件。高价值的应用件也被添加到专门识别的应用件组合中。幂等型请求的数量不可忽略的应用件对于虚拟通道合并的识别特别有吸引力;当前最重要的示例有 HTTP、DNS、BitTorrent 和 HTTPS。

[0060] 系统可使用各种参数来识别和检测所发送请求的类型。例如,系统可能会查看端口号、数据包类型(例如, TCP、UDP)和内容。一个具体的示例可以是,如果请求指定端口 80,且内容以 GET_ABC 开始,则其表示一个 HTTP 请求,可作为 HTTP 请求被处理;或者如果其端口为 53 且其为 UDP 数据包,其具有 DNS 数据包的布局,则它可作为 DNS 数据包被处理。

[0061] 采用虚拟通道合并的方法满足请求的方式可以是,在一些情况下,使用一个不同的网络接口对流量进行路由,使其路由与不采用该接口时不同,或者在一些情况下,将请求经网状网络路由到具有其自己的因特网接入方式的一个不同的设备。当使用虚拟通道合并时,最佳的情况时,系统具有能在每个设备上启用的尽可能多的网络接口。例如,一个计算

机可以启用一个有线以太网连接、Wi-Fi 和一个 4G LTE 适配器；一个智能手机可以启用其 4G 接口，加入一个 Wi-Fi 网络，并使用 Wi-Fi 直连和蓝牙来加入网状网络。

[0062] 在实现本文所披露的任何方法时，由于每个设备均可采用多个通道进行通信，提供一些方法或试探法、以启用通道或路由选择是有利的方式。可以采用多种路由选择引擎，以优化各种所需的设计考虑因素。例如，当速度为首要要求时，可以充分地利用所有可用因特网出口的方法效果最好，例如，等字节数、等请求数、字节数与以往性能成正比、以及请求数与以往性能成正比等方式。等请求数方式最为简单，它将尽力向所有可用的因特网出口发送数量基本相等的请求。这可通过多种方式实现，例如，使用轮询调度为当前请求选取一个随机出口，或者选取随机出口、并对所形成的附加不平衡情况进行跟踪和校正（由于大数定律的原因，从长远来看倍增式不平衡是不可能的）。等字节数方式是对等请求数方法的一种改良，其对请求数按字节数进行衡量。这样就可参与的通道中实现更为均匀的字节分布。在字节数与以往性能成正比的方法中，系统将基于自然的使用或基于综合测试流量，保存有对一个通道的以往性能的估算，并按以往性能的估算，对沿此通道传送的字节数进行衡量。请求数与以往性能成正比是一种类似的技术，只是系统所跟踪的是请求数而非字节数。

[0063] 当保存的可靠性为首要要求时，队列溢出路由选择件器就有很好的效果。队列溢出方法将在每个具有不经其他设备的直接因特网连接的设备内保持有一个虚拟队列。在队列溢出规则之下，这些设备默认会使用直接因特网连接发送流量，发送方式与没有虚拟通道合并时一样。只有当请求的虚拟队列达到一个特定的阈值（可提前或根据对此设备性能的测量进行设置）时，设备才会开始将一些请求路由至网状网络中的其他设备，以使它们的因特网连接也被使用。队列溢出可提供一种非常保守的系统，其将可靠性和可用性置于比速度更重要的地位。与队列溢出配合良好的一种重试策略是，当在限值以下的虚拟队列中的时隙变为可用时，对于经过其他设备仍然未完成的请求，在直接连接上进行重试发送。

[0064] 当对一个典型网页进行加载时，通常会有许多对象（web 资源）被请求。虚拟通道合并的方法利用此方式的措施是，分离出要在不同路径上发送的请求。然而，有时可能会对单一的非常大的对象进行请求，例如在软件更新下载中，或者在使用 HTTP 流观看视频时。在此情况下，一个单个对象仍然可以使用多个连接通过使用 HTTP 系列请求而获得，其可使用多个请求，即每个请求均构成原请求的一个子请求，每一个请求均仅请求文件的一个部分。由此，文件被以分部分方式请求，而非一次请求整个文件。每个子请求，即每个部分均可使用任何可用通道进行请求。应注意，大多数视频流服务必须支持范围请求（range requests），以使用户能够在视频中进行跳转和搜寻；由此，系列请求就是一种完全正常的请求形式，使用户能够在 YouTube、Netflix、Vimeo 及所有由 Akamai 服务的网站上获得良好的观看和工作效果。

[0065] 在图 1-4 所示的实施例中，虚拟通道合并方法作为一个分布式系统加以实现，其中每个设备均独立决定采用何种选择方案，就采用哪些通道作出自己的决定，并建立自己的连接。然而，根据一种实施例，一个云路由 oracle 服务器是该系统的一个可选部分，其可用于将网状网络路由协议的一个任何部分卸载至云中的一个连接良好且准备良好的服务器上，该服务器通常位于因特网上的某个地方。图 5 的示例示出了一个环境，其中云路由 oracle 服务器 524 对来自系统中各种设备的关于路由的任何查询提供服务。这些查询可以

被设备直接发送或经本地网络上的其他设备进行发送。一个样例查询可能包括对关于如何到达一个给定设备的信息的请求,关于各种连接的成本、各种可用服务的带宽等方面的查询。云路由 oracle 服务器 524 会保留一些在查询中提供的背景信息,对其进行高速缓存或存储。然后 oracle 服务器 524 会帮助设备查找其他设备,选择路由,权衡路由和因特网出口特性。

[0066] 云路由 oracle 服务器 524 可构成云出口服务器 522 的组成部分,也可以是一个单独或独立的服务器。云出口服务器 522 可用于提供安全、隐私和速度的改进,而通道合并有利于那些甚至是完全不透明并一直未被系统归类的流量。云出口服务器 524 也可以被应用于已归类的流量,在那里,其已被了解的属性可用于提高速度。例如,HTTP 流量在 JavaScript 中的注释可以被剥离,变量名可以缩短,图像的尺寸可通过以下方式减小:丢弃元数据,转换为一个更为高效的图像格式,抛弃在图像中被编码但又不太可能被人眼所见的一些信息,对视频流量进行更好的再编码。

[0067] 在配合不透明流量使用时,采用云出口服务器 524 的系统的运行方式如下。当在一移动设备的用户应用中发起一个请求时,驻留在该移动设备中的客户端将尝试对该请求进行译码或归类。如果该请求可以被译码且客户端可以处理该请求而无需云出口服务器 524 的支持,客户端会处理该请求。否则,如果该请求不可被译码,该请求的数据包会被原发移动设备的客户端封装,并通过一个或多个路径进行发送。封装用数据包以云出口服务器 524 的地址作为目的地,而被封装的数据包则以目标服务器地址作为目的地,以移动设备的地址作为原发地址。

[0068] 可以采用纠错技术来确保被封装的数据包到达云出口服务器 524。可以使用许多种纠错码,例如总体而言的 Reed Soloman 码,但在实践中,以下简单的技术可能也很有效:当有超过两条路径可用时,使用其中一条路径发送在其他路径上所发送的数据包的异或值。如果数据包中有一个未能到达,则可使用已到达的该数据包的异或值对其进行重建。在这种方案下,两个数据包丢失时仍需要重新发送,但低开销与低重传概率的组合使这一机制具备吸引力。

[0069] 当被封装的数据包到达云出口服务器 524 时,它们会被解封装,以暴露出目标地址和原发地址。此时被解封装的数据包会以目标地址作为目的地、以云出口服务器 524 地址作为原发地,以此被引向目标服务器。随后应答被目标服务器引向云出口服务器 524。在应答到达云出口服务器 524 时,使用任何可用通道将其发送至原发设备,即不一定与由云出口服务器 524 接收的请求的通道相同。

[0070] 应该理解的是,本文所述的进程和技术并不在本质上与任何特定装置相关联,并可采用任何合适的组件组合来实现。此外,根据本文所述的讲述内容,可以采用各种类型的通用型设备。对本发明的说明参照了特定的示例,这些示例在所有方面均应仅被视为示意性,而非限定性。本领域技术人员应理解,可以有多种不同的组合适用于实践本发明。

[0071] 此外,考虑本文所披露的本发明的说明书和实践方式,本发明的其他实现方式对于本领域技术人员将是显而易见的。所述实施例的各个方面和/或组件均可单独或以任何组合方式实现。说明和示例仅应被视为示例,本发明真正的范围和精神应由以下权利要求给出。

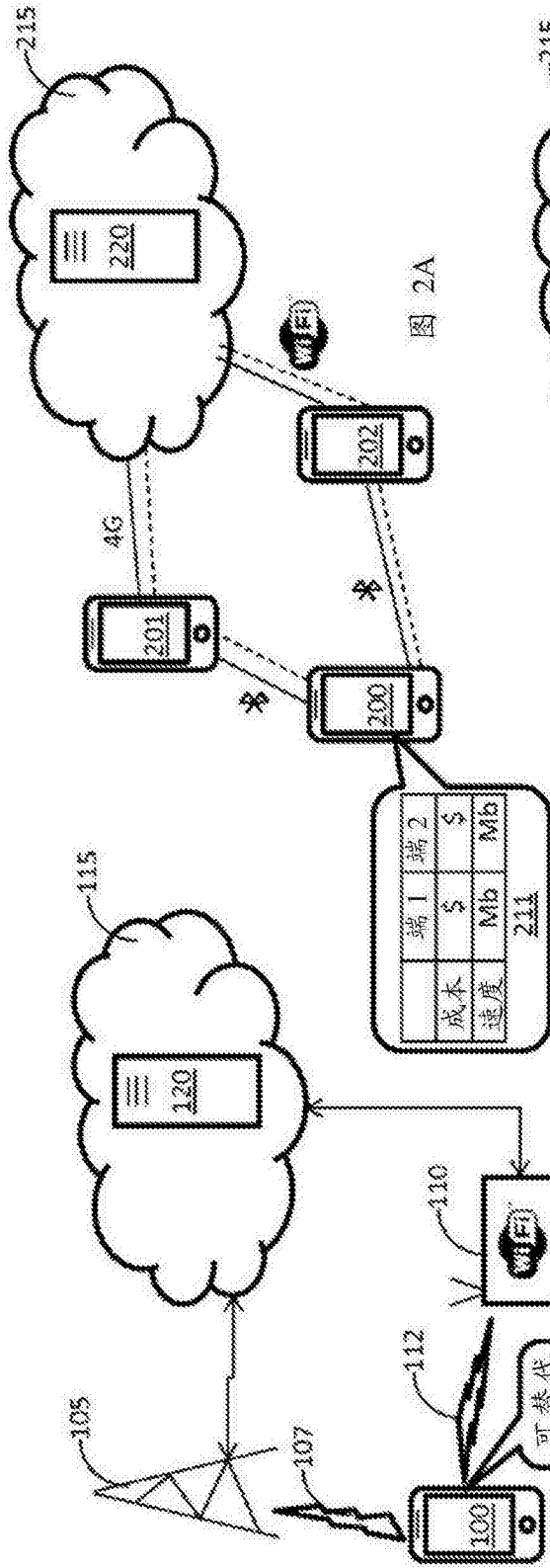


图 1

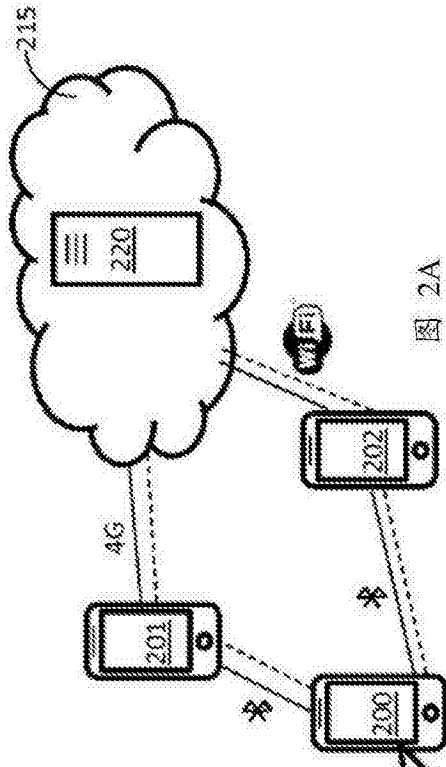


图 2A

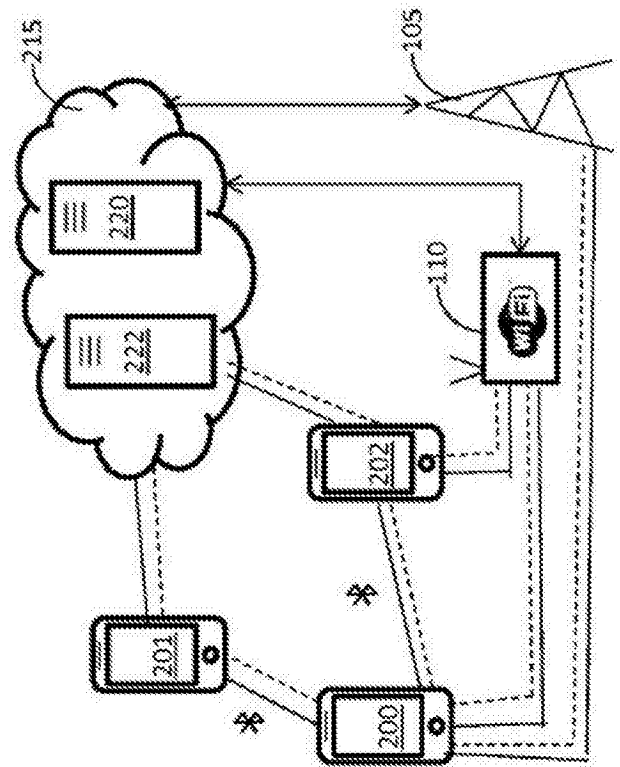


图 2C

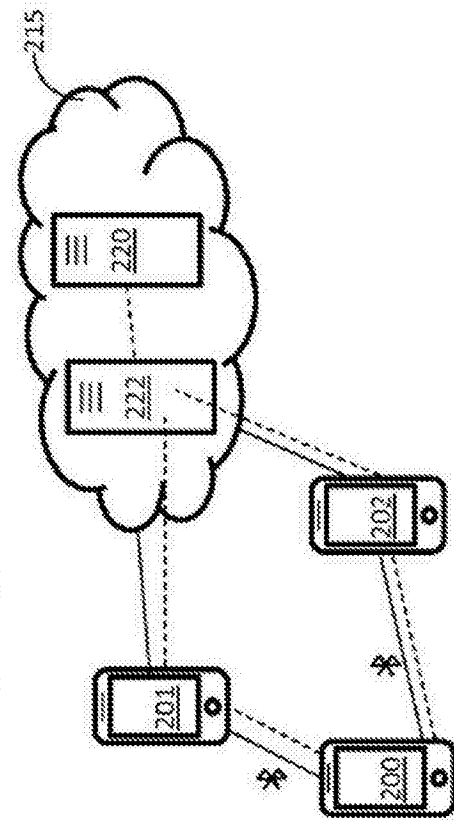


图 2B

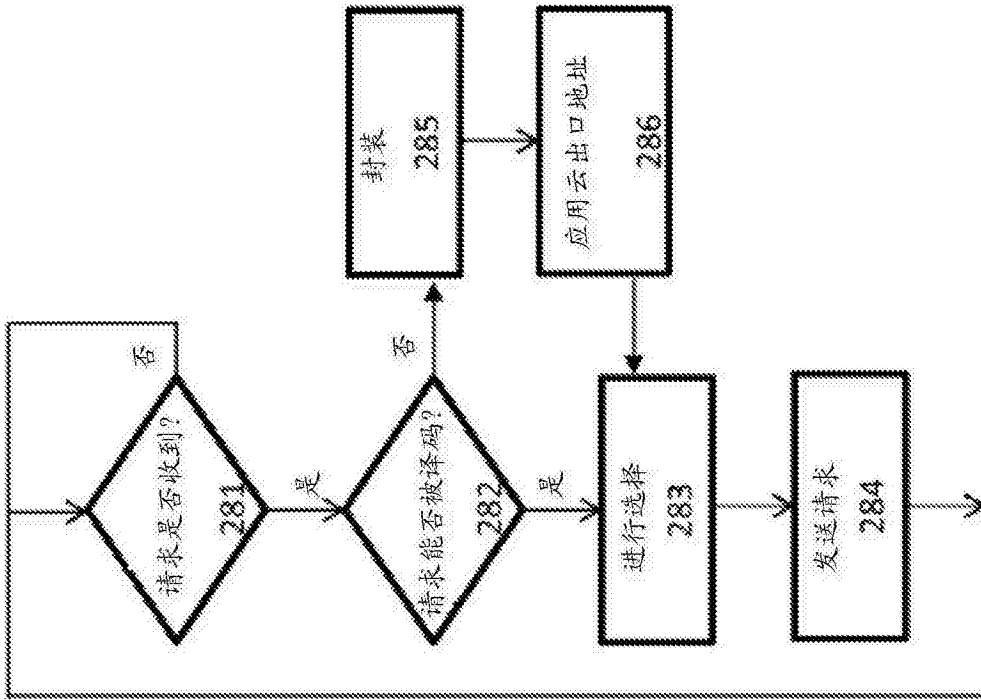


图 2D

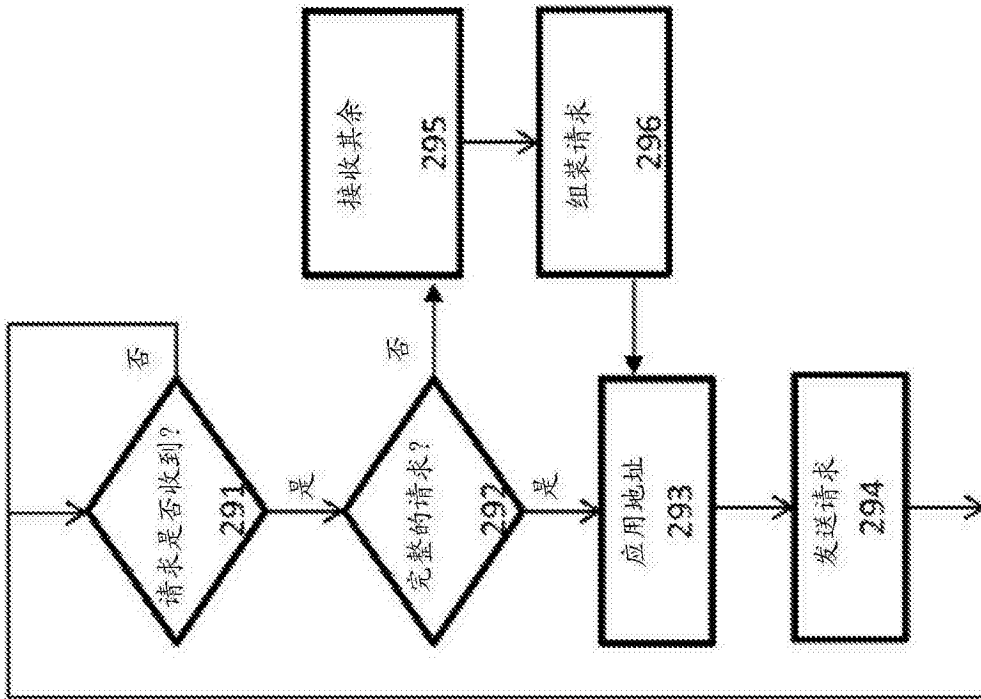


图 2E

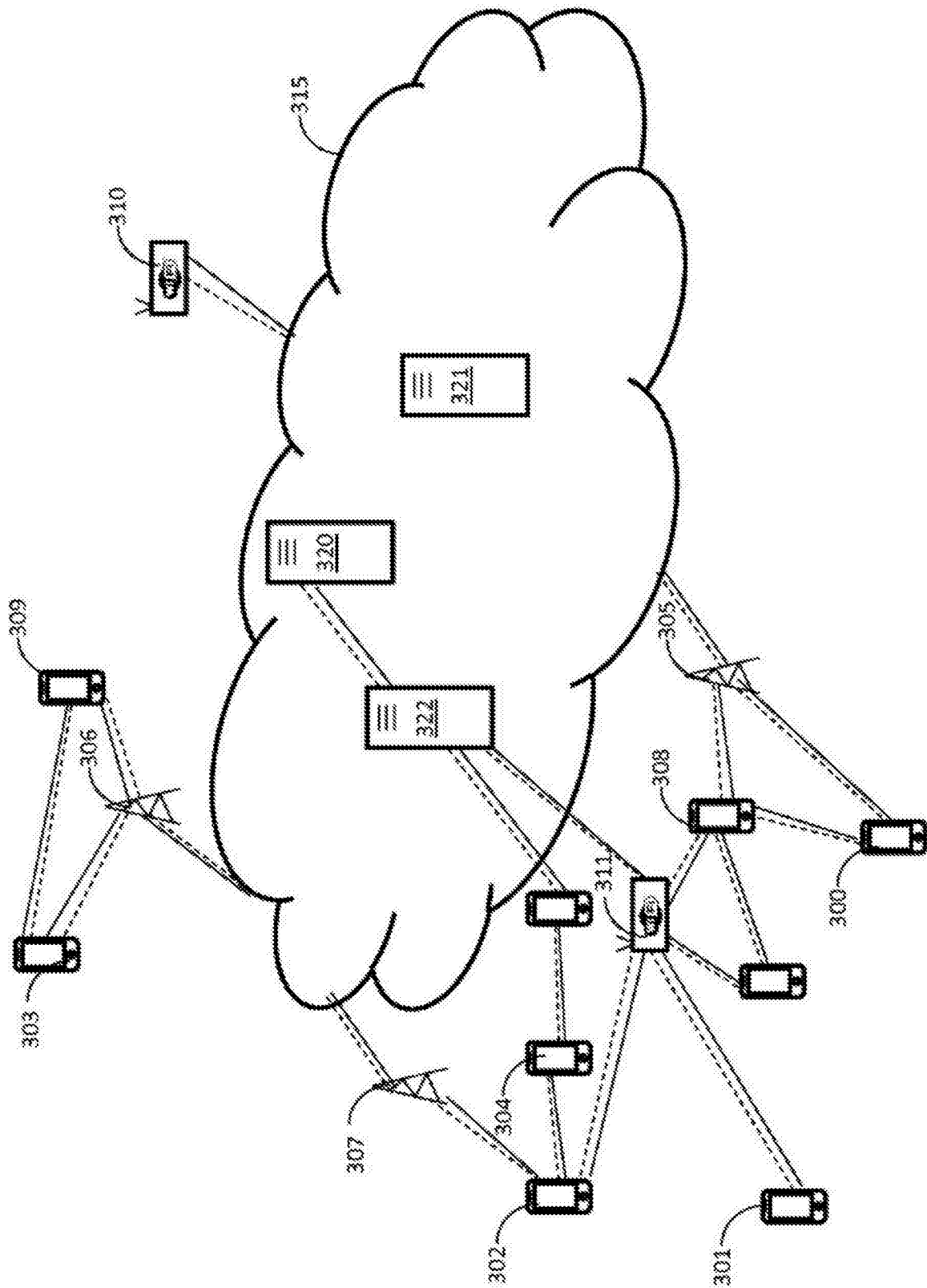


图 3

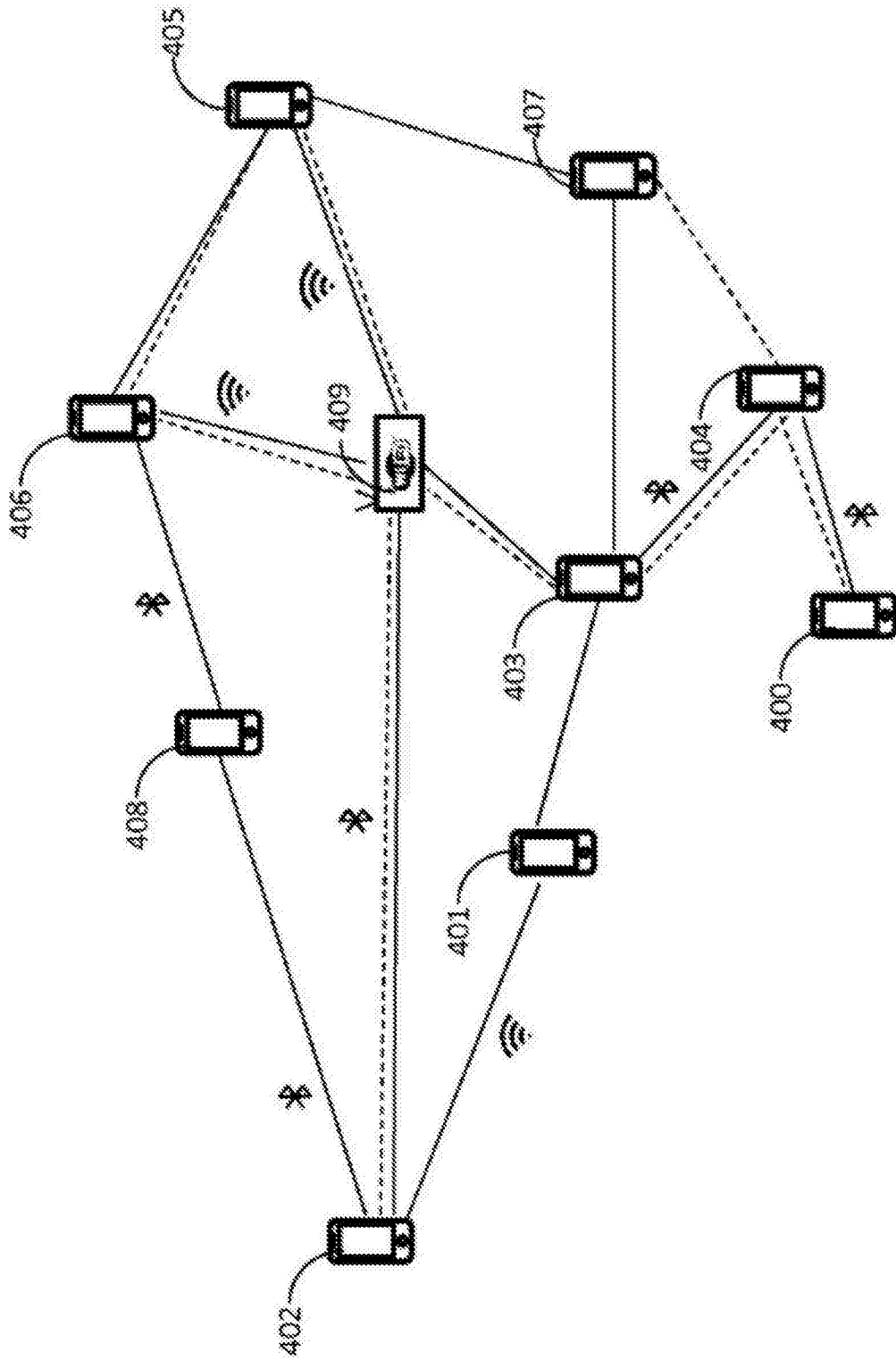


图 4

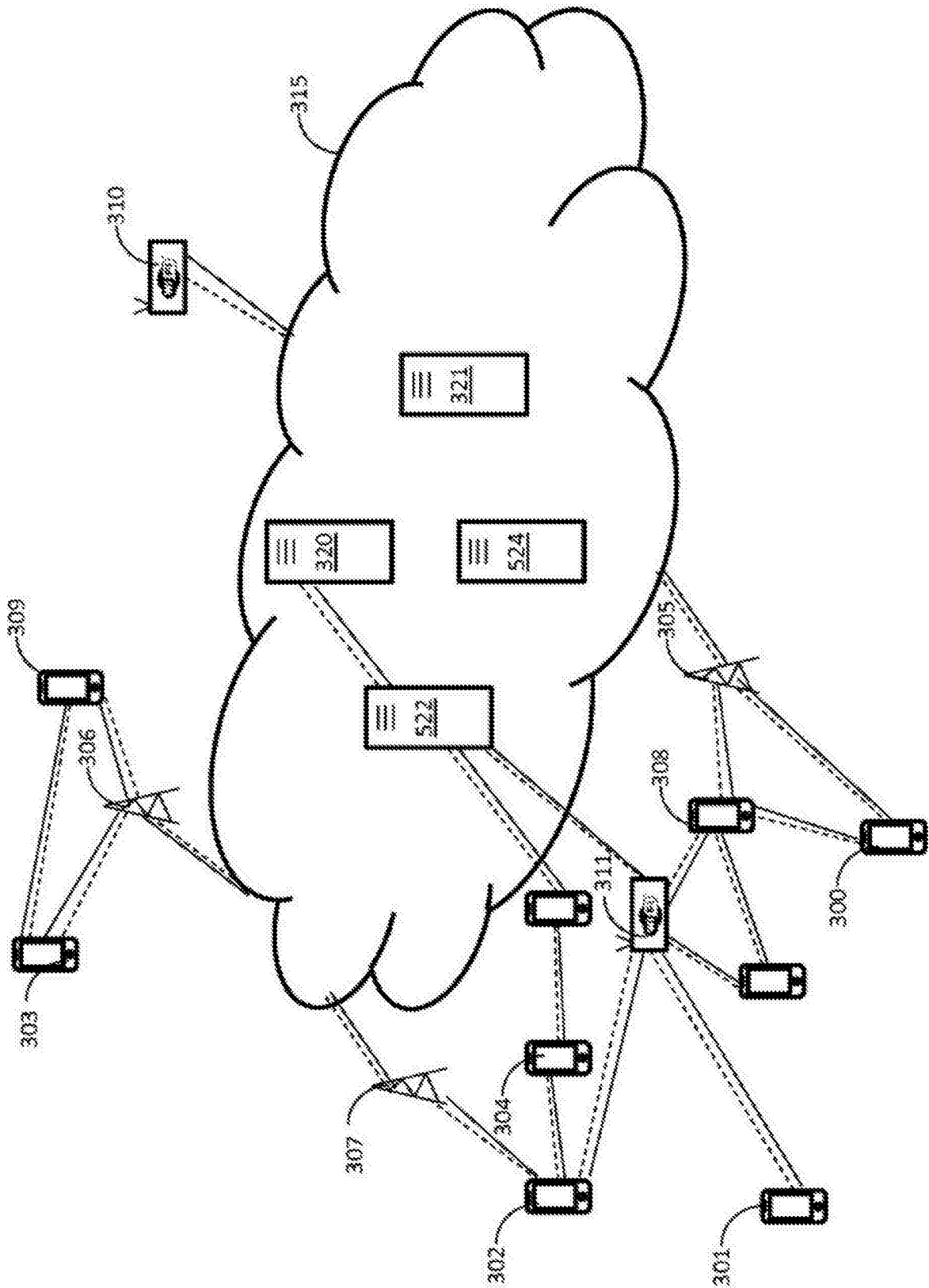


图 5

1. 一种在处理器可读代码中实现的方法,所述处理器可读代码在被执行时,会使移动设备执行若干步骤,所述步骤包括:

a. 由所述移动设备与多个其他设备建立并同时保持多个连接,由此建立多个通信通道;

b. 通过向目标设备发送多个资源请求而获取多个网络资源,其中所述多个资源请求中的至少两个资源请求通过所述多个通信通道中的两个不同的通信通道被发送。

2. 如权利要求 1 所述的方法,其中所述保持多个连接包括保持每个所述通信通道的唯一的 IP 地址。

3. 如权利要求 1 所述的方法,其中所述多个资源请求中的至少一个资源请求在其传输之前被封装。

4. 如权利要求 3 所述的方法,其中被封装的请求被发送至云出口服务器,以由所述云出口服务器进行解封,并转发至所述目标设备。

5. 如权利要求 1 所述的方法,其中所述多个网络资源请求中的至少一个网络资源请求包括 HTTP 范围请求,指定有待被发送至计算设备的一个文件部分。

6. 如权利要求 1 所述的方法,其中所述计算设备通过所述多个通信通道中的每一个通信通道发送相等数量的请求。

7. 如权利要求 1 所述的方法,其中所述计算设备按每一个资源请求中的字节数对所述资源请求进行衡量,并对通过所述多个通信通道中的每一个通信通道发送的所述字节数进行平衡。

8. 如权利要求 1 所述的方法,其中所述计算设备会保持所述多个通信通道中的每一个通信通道的以往性能的估算结果,并根据对所述以往性能的相应的估算结果来衡量要在所述多个通信通道中的每一个通信通道上发送的字节数。

9. 如权利要求 1 所述的方法,其中所述计算设备会保持所述多个通信通道中的每一个通信通道的以往性能的估算结果,并根据对所述以往性能的相应的估算结果来衡量将在所述多个通信通道中的每一个通信通道上发送的请求数。

10. 如权利要求 1 所述的方法,其中所述计算设备会保持对所述多个通信通道中的每一个通信通道的使用成本的估算结果,并根据对所述使用成本的相应的估算结果来衡量将在所述多个通信通道中的每一个通信通道上发送的请求数。

11. 如权利要求 1 所述的方法,其中所述计算设备还向服务器发送请求,以获取所述多个其他设备中的至少一个设备的连接信息。

12. 如权利要求 1 所述的方法,其中所述多个其他设备中的一个设备包括智能手机,且所述多个其他设备中的一个设备包括无线 WiFi 路由器。

13. 如权利要求 12 所述的方法,其中所述多个资源请求中的至少一个资源请求通过所述智能手机被发送至因特网。

14. 如权利要求 1 所述的方法,其中所述多个通信通道中的一个通信通道通过移动网络建立,且所述多个通信通道中的一个通信通道通过无线 WiFi 传输建立。

15. 如权利要求 1 所述的方法,其中所述多个通信通道中的一个通信通道通过移动网络建立,且所述多个通信通道中的一个通信通道通过蓝牙建立。

16. 如权利要求 12 所述的方法,其中所述多个资源请求中的至少一个资源请求通过所

述蓝牙发送至因特网。

17. 一种用于以移动设备实现多通道通信的系统,所述移动设备具有在其内运行的用户应用件,所述系统包括:

驻留在联网服务器上的主应用件,所述主应用件使服务器执行以下方法,所述方法包括:

接收从所述移动设备发送的被封装的请求,并确定所述被封装的请求的原发地址;

对所述被封装的请求进行解封装,以获取被解封装的请求,并从所述被封装的请求确定目标地址,以确定所述被解封装的请求的目标;

使用所述目标地址将所述被解封装的请求发送至所述目标;

从所述目标接收对所述被解封装的请求的响应;以及

使用所述原发地址将所述响应转发至所述移动设备。

18. 如权利要求 17 所述的系统,其中确定所述被封装的请求的原发地址在对所述封装的请求进行解封装之后进行。

19. 如权利要求 18 所述的系统,还包括:

驻留在所述移动设备上的客户端应用件,所述客户端应用件使所述移动设备执行以下方法,所述方法包括:

拦截由用户应用件所生成的请求;

对所述请求进行分析,以确定所述请求是否可被译码,以及确定所述请求是否能由所述客户端应用件处理;

当所述请求不可被译码时,将所述请求封装在封装数据包内,其中被封装的请求包括源地址和目标地址,且所述封装数据包包括所述联网服务器的地址;以及

将所述被封装的请求发送至所述联网服务器。

20. 如权利要求 19 所述的系统,其中当所述请求能被译码时,将所述请求发送至所述目标地址。

21. 如权利要求 19 所述的系统,其中当所述请求能作为幂等型请求被译码时,使用多个通道将所述请求多次发送至所述目标地址。

22. 如权利要求 19 所述的系统,其中当所述请求可被译码时,将所述请求分为多个子请求,并使用所述多个通道将所述多个子请求发送至所述目标地址。

23. 如权利要求 19 所述的系统,其中当所述请求能被译码时,将所述请求发送至所述目标地址,且在接收到对所述请求的应答时,将所述应答的至少一部分保存在高速缓存中。

24. 如权利要求 23 所述的系统,还包括接收来自第二移动设备的转发请求,并确定是否将对所述转发请求的应答保存在所述移动设备的所述高速缓存内,如果确定保存,将对所述转发请求的应答发送至所述第二移动设备,而不转发所述转发请求。

25. 如权利要求 19 所述的系统,其中当所述请求能被译码时,将所述请求通过所述第二移动设备发送至所述目标地址,且在所述第二移动设备接收到对所述请求的应答时,将所述应答的至少一部分保存在所述第二移动设备的高速缓存中。

26. 如权利要求 19 所述的系统,其中当所述请求能被译码时,将所述请求通过所述第二移动设备发送至所述目标地址,且在所述第二移动设备接收到所述请求时,确定是否将对所述请求的应答保存在所述第二移动设备的高速缓存内,如果是,则从所述第二移动设

备的高速缓存中取出所述应答,并将所述应答发送至原发移动设备。

27. 如权利要求 19 所述的系统,其中当所述请求能被译码以包含多个子请求时,确定是否将对至少一个所述子请求的应答保存在所述高速缓存内,如果确定保存,则将对至少一个所述子请求的应答提供给所述用户应用件,并将其余子请求发送至所述目标地址。

28. 如权利要求 27 所述的系统,其中所述多个子请求通过多个通信通道被发送,且其中至少一个所述通信通道包括到第二移动设备的点对点连接。

29. 一种在处理器可读代码中实现的方法,当所述处理器可读代码被执行时,会使计算设备执行若干步骤,所述步骤包括:

在开放式系统互连 OSI IP 层上接收通信请求;

分析所述通信请求,以对所述通信请求进行归类,并确定其是否能在更高的开放式系统互连 OSI 层上被处理;

当所述通信请求不能在更高的开放式系统互连 OSI 层上被处理时,通过 IP 层发送所述通信请求;

当所述通信请求能在更高的开放式系统互连 OSI 层上被处理时,通过所述更高的开放式系统互连 OSI 层发送所述通信请求。

30. 如权利要求 29 所述的方法,其中分析所述通信请求包括确定所述通信请求的端口号、所述通信请求的数据包类型和所述通信请求的内容。

31. 如权利要求 29 所述的方法,其中当确定所述通信请求能在所述开放式系统互连 OSI 应用层上被处理时,通过多个通道发送所述通信请求和所述通信请求的复本。

32. 如权利要求 29 所述的方法,其中当所述通信请求被归类为 HTTP 请求时,在所述开放式系统互连 OSI 应用层上通过多个通信通道发送所述通信请求。

33. 如权利要求 29 所述的方法,其中当确定所述通信请求不能在更高的开放式系统互连 OSI 层上被处理时,通过以下方式对所述通信请求进行封装:将目标地址和原发地址纳入被封装的请求中,并在所述通信请求的封装上采用云出口服务器的网络出口地址,允许所述云出口服务器对所述通信请求进行解封,并使用所述目标地址转发所述通信请求。

34. 如权利要求 33 所述的方法,还包括对所述通信请求进行加密。

35. 如权利要求 33 所述的方法,其中对所述通信请求进行封装包括将具有源地址和目标地址的所述通信请求的数据包置于具有上述云出口服务器的地址的封装数据包内。

36. 如权利要求 33 所述的方法,其中对所述通信请求进行封装包括将所述通信请求的数据包分断为多个子数据包,并将具有源地址和目标地址的每个所述子数据包置于具有上述云出口服务器的地址的封装数据包内,由此生成多个被封装的请求。

37. 如权利要求 36 所述的方法,其中所述多个被封装的请求通过多个通信通道被发送。