



(12) 发明专利申请

(10) 申请公布号 CN 102498486 A

(43) 申请公布日 2012.06.13

(21) 申请号 201080040489.5

(22) 申请日 2010.09.10

(30) 优先权数据

61/241,370 2009.09.10 US

12/878,705 2010.09.09 US

(85) PCT申请进入国家阶段日

2012.03.12

(86) PCT申请的申请数据

PCT/US2010/048420 2010.09.10

(87) PCT申请的公布数据

W02011/031962 EN 2011.03.17

(71) 申请人 摩托罗拉移动公司

地址 美国伊利诺伊州

(72) 发明人 马克森·R·惠勒

威廉·N·坎普二世

利恩·T·马米特苏卡

克里斯托弗·A·米特拉

斯科特·I·普特曼

赛鲁斯·P·马斯特

史蒂芬·J·塞韦里内克

米兰·S·布拉姆巴特

阿尼什·M·沙阿

保罗·韦恩·汉加斯 胡鑫

萨普纳·索尼 托尼·鲁宾逊

希瑟·M·勒罗伊

克里斯托弗·雷平斯基 魏凯

(74) 专利代理机构 中原信达知识产权代理有限
责任公司 11219

代理人 李宝泉 周亚荣

(51) Int. Cl.

G06F 17/30(2006.01)

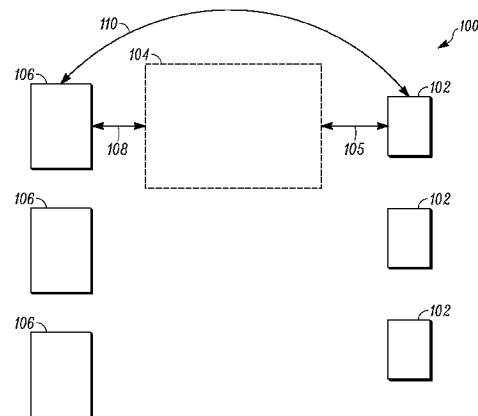
权利要求书 2 页 说明书 46 页 附图 36 页

(54) 发明名称

内容提供商网站交互的系统、服务器和移动
设备及其方法

(57) 摘要

一种聚合服务服务器,该聚合服务服务器被
配置成与用户设备和多个不同的内容提供商进行
通信。处理器被配置成从所述多个不同的内容提
供商获得内容并且进一步被配置成将所获得的内
容推到所述用户设备。



1. 一种聚合服务服务器,所述聚合服务服务器被配置成与用户设备和多个不同的内容提供商进行通信,包括:

处理器,所述处理器被配置成从所述多个不同的内容提供商获得内容并且进一步被配置成将所获得的内容推到所述用户设备。

2. 根据权利要求 1 所述的聚合服务服务器,其中,所述处理器进一步被配置成从所述用户设备接收状态更新并且将所述状态更新发送到所述多个内容提供商中的对应的内容提供商。

3. 根据权利要求 1 所述的聚合服务服务器,进一步包括:

多个插件,每个插件对应于所述多个内容提供商中的相应的内容提供商并且被配置成传送和接收来自所述对应的内容提供商的通信。

4. 根据权利要求 3 所述的聚合服务服务器,进一步包括:

存储器,

其中,所述处理器进一步被配置成从所述插件中的一个插件接收更新,以将优先级指派给所述更新并且将所述更新和所指派的优先级存储在所述存储器中。

5. 根据权利要求 4 所述的聚合服务服务器,其中,所述处理器进一步被配置成基于向所接收到的更新所指派的优先级来对到所述用户设备的推进行调度。

6. 根据权利要求 5 所述的聚合服务服务器,其中,所述处理器进一步被配置成从所述用户设备接收对于数据的请求并且被进一步配置成将存储在所述存储器中的任何更新推到所述用户设备。

7. 根据权利要求 6 所述的聚合服务服务器,其中,所述对于数据的请求是对于来自所有所述内容提供商的更新的单个请求。

8. 根据权利要求 4 所述的聚合服务服务器,其中,每个插件被配置成将所接收到的更新重新格式化成与所述用户设备兼容的格式。

9. 根据权利要求 7 所述的聚合服务服务器,其中,所述处理器进一步被配置成通过以下将所接收到的内容推到与单个账户相关联的多个用户设备:

监视推到所述用户设备中的每一个用户设备的所述内容;以及

基于所述监视将内容推到所述用户设备中的每一个用户设备。

10. 一种使用中间服务器来控制客户端设备与多个内容提供商之间的通信的方法,包括:

由所述中间服务器从所述多个内容提供商中的一个内容提供商接收更新;

由所述中间服务器将优先级指派给所接收到的更新;以及

由所述中间服务器基于所指派的优先级对所接收到的更新到所述客户端设备的推进行调度。

11. 根据权利要求 10 所述的方法,进一步包括:

由所述中间服务器从所述客户端设备接收对于更新的单个请求;以及

响应于所述单个请求,由所述中间服务器将更新从所有所述内容提供商推到所述客户端设备。

12. 根据权利要求 10 所述的方法,进一步包括:

监视已经被推到所述客户端设备的所述更新;以及

基于所述监视,将更新推到所述用户设备。

13. 根据权利要求 10 所述的方法,进一步包括:

将所接收到的更新重新格式化成与所述客户端设备兼容的格式。

14. 根据权利要求 10 所述的方法,进一步包括:

定期性地将更新推到所述客户端设备。

15. 根据权利要求 10 所述的方法,进一步包括:

当更新被指派了高优先级时,将所述更新推到所述客户端设备。

16. 一种中间服务器,所述中间服务器被配置成与用户设备和多个不同的内容提供商进行通信,包括:

存储器;

内容提供商处理器,所述内容提供商处理器被配置成从所述多个不同的内容提供商获得更新并且将所述更新存储在所述存储器中;以及

核心服务处理器,所述核心服务处理器被配置成将存储在存储器中的所述更新推到所述用户设备。

17. 根据权利要求 16 所述的中间服务器,其中,所述内容提供商处理器进一步被配置成将从所述多个不同的内容提供商所获得的更新重新格式化成与所述用户设备兼容的格式。

18. 根据权利要求 16 所述的中间服务器,其中,所述核心服务处理器将优先级指派给所获得的存储在所述存储器中的更新。

19. 根据权利要求 18 所述的中间服务器,其中,所述核心服务处理器基于所指派的优先级对所获得的存储在存储器中的更新到所述用户设备的推进行调度。

20. 根据权利要求 16 所述的中间服务器,其中,在所获得的更新已经被推到所述用户设备之后,所述核心服务处理器从所述存储器中删除所获得的更新。

内容提供商网站交互的系统、服务器和移动设备及其方法

技术领域

[0001] 本发明涉及包括移动设备的通信,并且更具体地涉及这样的移动设备与互联网内容提供商网站之间的通信。

背景技术

[0002] 诸如社交网网站 (SNW)、新闻订阅源、音乐和相片网站的内容提供商网站 (CPW)、以及诸如企业到企业 (b2b) 或企业到消费者 (b2c) 网站的其它类型的网站是交互式网站,其支持诸如新闻、天气、个人和 / 或企业信息、图片、视频以及歌曲的各种形式的数据的下载和 / 或上传 (例如,发帖),并且从而有助于在人和人群之中的人与人之间的连接的创建和维护。由一个用户将数据上传到 CPW 能够允许其他用户访问和 / 或下载所上传的数据。通常,SNW 为无数用户提供了架构,以创建分别标识相应的用户的相应的个人空间或专业空间并且允许上传的数据与相应的空间相关联。

[0003] CPW 能够与正在操作常常经由因特网型网络与 CPW 联系的各种不同类型的设备中的任何一个的用户进行通信。逐渐地,用户采用移动设备来与 CPW 进行交互。随着这样的通信活动增加,有发展着对于改进在进行这样的通信活动中的质量和 / 或用户友好性的不断增加的需求。另外,还发展着对于改进这样的通信活动的效率以改进移动设备的电池性能并且减少所有设备的数据传输的不断增加的需求。

[0004] 因此,如果能够提供将帮助至少部分地解决前述发展需要中的一个或多个、以改进的移动设备和 / 或其它设备和 / 或用于允许移动设备与 CPW 进行通信的改进的方法的形式的改进,其将是有益的。

附图说明

[0005] 图 1 以示意形式示出了包括与多个内容提供商网站进行通信的多个移动设备的示例通信系统,其中通信中的一些经由中间 web 服务器发生;

[0006] 图 2 是示出图 1 的移动设备中的一个的示例组件的框图;

[0007] 图 3 是示出图 1 的中间 web 服务器的示例组件的框图;以及

[0008] 图 4-9 是示出图 1 的中间 web 服务器和移动设备的操作的各种示例步骤的示例性流程图。

[0009] 图 10 是图示中间服务器的操作的示例性流程图;

[0010] 图 11 是图示移动设备的操作的示例性流程图;

[0011] 图 12 是图示中间服务器的操作的示例性流程图;

[0012] 图 13 是图示中间服务器的操作的示例性流程图;

[0013] 图 14 图示了移动设备的示例性操作;

[0014] 图 15 是图示移动设备的操作的示例性流程图;

[0015] 图 16 是图示中间服务器的操作的示例性流程图;

[0016] 图 17 是根据实施例的另一示例性通信系统;

- [0017] 图 18 是根据实施例的又一示例性通信系统；
- [0018] 图 19 图示了客户端设备与中间服务器之间的示例性通信；
- [0019] 图 20 图示了中间服务器的后端的示例性推送服务；
- [0020] 图 21 图示了根据一个实施例的、用于导入联系人的示例性方法；
- [0021] 图 22 图示了根据实施例的示例性序列；
- [0022] 图 23-30 图示了根据实施例的来自客户端设备的示例性屏幕；
- [0023] 图 31 图示了根据实施例的另一示例性通信系统；以及
- [0024] 图 32 图示了根据一个实施例的示例性协议。

具体实施方式

[0025] 参考图 1, 以简化的示意形式示出了示例通信系统 100 的框图。如所示, 在这个实施例中通信系统 100 包括三个移动设备 102, 其中的一个被示出为经由通信链路 105 与服务器进行通信, 所述服务器在本实施例中为 web 服务器 104。移动设备 102 分别表示由个人 (或用户) 或可能地由期望或需要通信能力的其它实体 (例如, 上网本或其它计算机) 操作的通信设备。在一些实施例中, 例如, 移动设备 102 可以为蜂窝电话、诸如个人数字助理的其它无线设备、和 / 或诸如能连接到网络 (通信链路 105) 并且与网络进行通信的膝上型计算机和台式计算机的设备中的任何一个。

[0026] 通信系统 100 此外被示出为包括三个内容提供商网站 (CPW) 106, 其中的一个被示出为经由通信链路 108 与中间 web 服务器 104 进行通信。另外, 通信链路 110 还被提供成支持与 web 服务器 104 进行通信的移动设备 102 中的一个直接与也与该 web 服务器进行通信的 CPW 106 中的那一个进行通信, 而无需 web 服务器 104 的调解。尽管移动设备 102 中的仅一个和 CPW 106 中的一个被示出为与 web 服务器 104 进行通信, 但是将理解的是, 取决于时间或者操作环境, 移动设备 102 和 CPW 106 中的任何一个或全部都能够与 web 服务器进行通信。同样地, 取决于时间或者操作环境, 移动设备 102 中的任何一个都能够经由诸如链路 110 的直接通信链路进入与 CPW 106 中的任何一个的通信。

[0027] 尽管图 1 中示出了三个移动设备 102, 但是在其它实施例中, 仅一个移动设备与 web 服务器 104 进行通信, 或者替代地任何任意数量的移动设备能够与 web 服务器 104 进行通信。同样地, 尽管在图 1 中示出了三个 CPW 106, 但是在其它实施例中, 仅一个 CPW 与 web 服务器 104 进行通信, 或者替代地任何任意数量的 CPW 能够与 web 服务器 104 进行通信。此外, 在其它实施例中, 任何任意数量的移动设备能够经由诸如链路 110 的直接通信链路与任何任意数量的 CPW 进行通信。也就是说, 图 1 旨在表示采用经由 web 服务器接口彼此间接地或者彼此直接进行通信的任何任意数量的移动设备和任何任意数量的 CPW 的各种系统中的任何一个系统。

[0028] 取决于实施例, 通信链路 105、108 以及 110 能够为单个网络或多个网络的一部分, 并且每个链路能够包括一个或多个有线和 / 或无线通信通道, 例如, 陆上通讯线 (例如, 光纤、铜) 布线, 微波通信、无线电信道、无线路径、内联网、互联网和 / 或万维网通信通路 (他们本身能够采用无数的中间硬件和 / 或软件设备, 包括例如无数的路由器等)。此外, 各种通信协议和方法学能够被用来在移动设备 102、web 服务器 104 以及 CPW 106 之间经由通信链路 105、108 以及 110 进行通信, 包括例如传输控制协议 / 网际协议 (TCP/IP)、可扩展消息

存在协议 (XMPP)、文件传输协议 (FTP) 等。在其它实施例中,也能够利用用于帮助多个移动设备 102 与 CPW 106 之间的信号的传输的其它类型的通信链路。尽管在本实施例中,通信链路 / 网络和服务器都被讨论为基于 web 的,但是在其它实施例中,链路 / 网络和服务器能够采取各种非基于 web 的形式。

[0029] 如将在下文中有图 3-16 更详细地讨论的,web 服务器 104 被配置成充当移动设备 102 与 CPW 106 之间的中间物。移动设备 102 与 CPW106 之间的各种类型的通信通过 web 服务器 104、被 web 服务器 104 处理和 / 或监视,所述各种类型的通信包括例如包括文件 (例如,相片、音乐、视频、文本输入等) 的上传和下载、博客发布以及消息传递 (例如,短消息服务 (SMS)、多媒体消息服务 (MMS) 以及即时发消息 (IM)) 的通信。CPW 通常旨在包括支持诸如个人和 / 或企业信息、图片、视频以及歌曲的各种形式的数据的下载和上传 (例如,发布) 并且从而有助于人和人群之中的人与人之间的连接的创建和维持的各种交互式网站。CPW 的示例包括例如 Facebook™、MySpace™、hi5™、LinkedIn™ 以及 Twitter™。为了本发明的目的,CPW 还能够被理解成包括各种其它类型的网站 (例如,企业到企业或企业到消费者网站),尽管不完全地或主要地集中在社交网上,然而也包括社交网类型特征。其它内容提供商网站包括 RSS 的源或其它新闻订阅源、诸如 Picasa™ 或 Photobucket™ 的相片服务、以及诸如 LastFM™ 的音乐服务。

[0030] 简易信息聚合 (RSS) 指的是用来频繁地发布诸如博客条目、新闻头条、音频以及视频的更新的作品 web 订阅源格式。RSS 文档 (其被称为“订阅源”或“频道”) 包括完全的或概略的文本、加上诸如出版日期和原创作者的元数据,并且可以包括相片。

[0031] 参考图 2,提供了图示根据一个实施例的诸如移动设备 102 的移动设备的示例内部组件 200 的框图。如图 2 中所示,组件 200 包括一个或多个无线收发器 202、203、205、处理器 204 (例如,微处理器、微计算机、专用集成电路等)、存储器部分 206、一个或多个输出设备 208、以及一个或多个输入设备 210。在至少一些实施例中,存在包括诸如显示器的一个或多个输出设备 208 和诸如小键盘或触摸传感器的一个或多个输入设备 210 的用户接口。内部组件 200 进一步能够包括组件接口 212 以向辅助组件或附件提供直接连接以用于另外的或增强的功能性。内部组件 200 优选地还包括诸如电池的电源 214,以用于向其它内部组件提供电力同时使移动设备能够是可携带的。内部组件 200 中的全部都能够经由一个或多个内部通信链路 232 (例如,内部总线) 耦合到彼此,并且与彼此进行通信。

[0032] 无线收发器 202 中的每一个都利用无线技术以便进行通信,所述无线技术能够包括:例如 (但不限于) 基于蜂窝的通信技术,诸如模拟通信 (使用 AMPS)、数字通信 (使用 CDMA、TDMA、GSM、iDEN、GPRS、EDGE 等)、以及下一代通信 (使用 UMTS、WCDMA、LTE、IEEE 802.16 等) 或其变体;或对等或者自组织通信技术,诸如 HomeRF (射频)、蓝牙以及 IEEE 802.11 (a、b、g 或 n);或其它无线通信技术,诸如红外技术。在本实施例中,无线收发器 202 包括蜂窝收发器 203 和无线局域网 (WLAN) 收发器 205,但是在其它实施例中,存在这些类型中的无线收发器中的仅一个 (并且可能地这些类型的无线收发器和 / 或其它类型的无线收发器都不存在)。通过使用无线收发器 202,移动设备 102 不仅能经由通信链路 110 与 CPW 106 进行通信并且还能经由通信链路 105 与 web 服务器 104 (并且因此再次间接与 CPW 106) 进行通信。

[0033] 与移动设备 102 的内部组件 200 的其它部分相结合的无线收发器 202 的示例操作

能够采取各种形式并且能够包括例如操作,在该操作中,在接收到无线信号时,内部组件检测通信信号并且收发器 202 对该通信信号进行解调以重新获得由无线信号传送的诸如语音和 / 或数据的传入信息。在从收发器 202 接收到传入信息之后,处理器 204 对该传入信息进行格式化以用于一个或多个输出设备 208。同样地,为了无线信号的传输,处理器 204 对传出的信息进行格式化,其可以或可以不由输入设备 210 来激活,并且将该传出信息传递到无线收发器 202 中的一个或多个以调制成通信信号。无线收发器 202 经由无线和 (也可能有线的) 通信链路将经调制的信号传递到诸如 web 服务器 104 和 CPW 106 中的一个或多个的其它设备 (以及可能地传递到诸如小区塔、接入点、或另一服务器或各种远程设备中的任何一个的其它设备)。

[0034] 取决于实施例,内部组件 200 的输入和输出设备 208、210 能够包括各种视觉、音频和 / 或机械输出。例如,输出设备 208 能够包括:诸如液晶显示器和发光二极管指示器的一个或多个视觉输出设备 216;诸如扬声器、警报器和 / 或蜂鸣器的一个或多个音频输出设备 218;和 / 或诸如振动机构或其它触觉反馈系统的一个或多个机械输出设备 220。视觉输出设备 216 尤其能够包括视频屏幕。同样地,以举例的方式,输入设备 210 能够包括:诸如光学传感器 (例如,相机) 的一个或多个视觉输入设备 222;诸如麦克风的一个或多个音频输入设备 224;以及诸如翻转传感器、键盘、小键盘、选择按钮、导航群集、触摸板、触摸屏、电容传感器、运动传感器、以及开关的一个或多个机械输入设备 226。能够致动输入设备 210 中的一个或多个的动作能够不仅包括按钮或其它致动器的物理按压 / 致动,而且还包括例如打开移动设备、将设备解锁、使设备移动以致动运动、使设备移动以致动位置定位系统以及对设备进行操作。

[0035] 如图 2 中所示,移动设备 102 的内部组件 200 还能够包括各种类型的传感器 228 中的一个或多个。传感器 228 能够包括例如接近传感器 (光检测传感器、超声收发器或红外收发器)、触摸传感器、高度传感器、能够包括例如全球定位系统 (GPS) 接收器、三角测量接收器、加速计、倾斜传感器、陀螺仪的位置电路、或能够标识移动设备 102 的当前位置或用户设备接口 (运载模式) 的任何其它信息收集设备。

[0036] 内部组件 200 的存储器部分 206 能够包括各种形式 (例如,只读存储器、随机存取存储器、静态随机存取存储器、动态随机存取存储器等) 中的任何一个的一个或多个存储器设备,并且能够由处理器 204 用来存储和检索数据。由存储器部分 206 存储的数据能够包括但是不必限于操作系统、应用以及信息数据。每个操作系统都包括控制通信设备的基本功能的可执行代码,所述通信设备的基本功能诸如在内部组件 200 之中包括的各种组件之间的交互,经由无线收发器 202 和 / 或组件接口 212 与外部设备的通信、以及应用和数据向存储器部分 206 的存储和从存储器部分 206 的对应用和数据的检索。每个应用都包括利用操作系统为通信设备提供诸如文件系统服务的更多的特定功能性和在存储器部分 206 中存储的受保护的和无保护的数据的处理的可执行代码。信息数据是能够被用于执行通信设备的功能的操作系统或应用引用和 / 或操纵的非可执行代码或信息。

[0037] 接下来参考图 3,更详细地示出了图 1 的 web 服务器 104 的另外的示例组件。如所示,web 服务器 104 包括存储器部分 302、与该存储器部分进行通信的处理器部分 304、以及用于将通信链路 105、108 与处理器 304 对接的一个或多个输入 / 输出 (I/O) 接口 (未示出)。处理器部分 304 进一步包括后端部分 306 (或社交网处理器) 和前端部分 308。后端

部分 306 经由通信链路 108 与 CPW 106 (以虚线示出) 进行通信, 而前端部分 308 经由通信链路 105 与移动设备 102 (也以虚线示出) 进行通信。

[0038] 如在下文中进一步详细地讨论的, 在至少一些实施例中, 后端部分 306 支持与诸如 CPW 106 的 CPW 的拉式 (pull) 通信。拉式通信能够例如使用对 web 典型的类型的表述性状态转移 (REST) 架构来实现, 并且当这样的后端部分被配置成生成对于要在由 web 服务器 104 确定的时间 / 环境从诸如 CPW 106 的 CPW 提供给后端部分 306 的信息的请求时, 响应于此 CPW 搜索所请求的数据并且将其提供回 web 服务器。同样地, 如在下文中进一步详细地讨论的, 在至少一些实施例中, 前端部分 308 与诸如移动设备 102 的移动设备协力建立推式信道。

[0039] 在至少一些这样的实施例中, 推式信道允许前端部分 308 在由 web 服务器 104 确定的时间 / 环境将来自 web 服务器 104 的通知 (由前端部分所生成) 提供给移动设备 102。该通知能够指示可用于提供给移动设备的信息内容。移动设备 102 进而能够以移动设备认为适当的方式对该通知作出响应。这样的响应通常 (但是未必一直) 构成可用的信息内容中的一些或全部从中间 web 服务器 104 的前端部分提供给移动设备的请求。

[0040] 参考图 4, 提供了示出图 1 和图 3 的 web 服务器 104 具体地与诸如如图 1 中所示的移动设备 102 和 CPW 106 的移动设备和 CPW 进行交互以及在诸如如图 1 中所示的移动设备 102 与 CPW 106 的移动设备与 CPW 之间调解通信时的操作的示例步骤的流程图。在开始步骤 400 处开始由图 4 的流程图表示的处理之后, 在步骤 402 处 web 服务器 104 通过建立诸如与图 1 的移动设备 102 的通信链路 105 的与移动设备的通信链路来开始操作。如将在下文中进一步详细地描述的, 取决于实施例, 与移动设备建立通信链路能够实际上包括与该移动设备建立多个通信链路 (能够并行或在不同的时间存在)。

[0041] 在一些这样的情况下, 多个通信链路是不同的类型的, 例如, 包括推式信道或除了推式信道之外的通信协议。而且, 尽管与移动设备 102 建立通信链路通常包括与基站建立电路转换连接, 并且因此通信设备将标识信息提供给基站, 移动设备由此将它本身标识到电信网络, 但是到 web 服务器 104 的连接还能够经由网际协议 (IP) 连接或移动设备正与其进行通信的基站与负载均衡器 / 防火墙之间的点到点 (P2P) 电信连接, 并且还能够包括将来自 web 服务器的响应信号往后提供给移动设备, 移动设备由此识别到其正与 web 服务器联系。

[0042] 一旦完成步骤 402, 则在步骤 404 处, web 服务器 104 进一步建立与 CPW 的通信链路, 诸如与图 1 所示的 CPW 106 的通信链路 108。在步骤 404 处通信链路的建立能够包括例如提供一个或多个 web 服务调用和 / 或其它技术。在步骤 404 之后, web 服务器 104 维持与 CPW 106 的正在进行的通信, 所述通信可以是 (但是不必是) 定期的通信, 并且 web 服务器 104 在一个或多个时间从 CPW 获得 (拉式) 信息。从 CPW 获得的信息能够包括各种不同类型的信息中的任何一个, 包括例如有关联系人或朋友 (包括联系人列表)、新朋友或更新的联系人、特殊消息、新闻、事件的信息, 和可能地包括文件 (诸如图像文件或文本文件) 或其它形式的数据的其它类型的信息。一旦在步骤 406 处获得信息, 则在步骤 408 处 web 服务器对所获得的信息进行处理。

[0043] 另外参考图 5, 根据一个实施例示出了与图 4 的步骤 406 和步骤 408 相对应的例子步骤。如所示, 步骤 406 (获得步骤) 能够被理解为包括以开始于步骤 500 开始并且进

一步包括三个另外的子步骤 502、504 以及 506 的若干子步骤。更具体地,在子步骤 502 中,web 服务器 104 将拉式信号发送到 CPW 106,并且在子步骤 504 处,在 web 服务器的后端部分 306 处从 CPW 往回接收信息。在后端部分 306 处接收到信息之后,在步骤 506 处,该信息然后从后端部分被推到 web 服务器 104 的前端部分 308。

[0044] 进一步如图 5 中所示,在一个实施例中步骤 408(处理步骤)能够包括在子步骤 518 处结束之前在子步骤 508 处开始的若干子步骤(图 5 示出了与步骤 408 相对应的子步骤,如为与步骤 406 相对应的子步骤的接续)。更具体地,在子步骤 508 处,一旦 web 服务器 104 的前端部分 308 接收到在子步骤 506 处从后端 306 部分所推的信息,则该信息被放入到通用传输队列中。接下来,在子步骤 510 处,信息能够可选地被压缩。此外,在子步骤 512 处,信息能够可选地被转换成不同的格式,例如二进制格式。如由框 509 所另外表示的(以虚线示出),在子步骤 512 处发生的格式转换能够包括对由 CPW 106 提供的特定格式信息的移除,以便对信息的格式进行标准化并且移除站点特定的格式信息,然而不是来源身份,或者另外修改信息的格式以成为提供给移动设备的统一的或通用的格式,无论作为信息的来源的 CPW 格式如何。

[0045] 接下来,在子步骤 514 处,信息基于其是否是高重要性或低重要性被过滤。如由子步骤 511、513、515 以及 517(以虚线示出)所进一步表示的,这个过滤操作能够进一步包括确定。即,如子步骤 511 处所示,web 服务器 104 能够确定信息是否关系到朋友、新朋友、特殊消息、新闻或者事件。如果是这样的话,则在子步骤 513 处,该信息被指派低级别状态。然而,如果信息不落入那些分组中的一个,则该过滤处理继续进行到子步骤 515,在该处 web 服务器确定信息是否关系到状态更新。如果其关系到,则在子步骤 517 处高级别状态被指派给该信息。在本示例实施例中,如果在子步骤 515 处信息被确定为不关系到状态更新,则处理再次返回到子步骤 513。将认识到 web 服务器 104 能够确定该信息是否为用户的状态更新,并且如果它是,则将该信息看作高级别,或者高优先级,并且如果它不是,则将该信息看作低级别,或低优先级。其它类型的信息也可以被看作高优先级,然而期望限制导致通信设备的不断增加的活动的消息的数量。

[0046] 一旦完成了过滤子步骤 514,则处理前进到子步骤 516,其中 web 服务器 104(具体地 web 服务器的前端部分 308)确定可能在步骤 406 处从 CPW 106 获得的信息与更早地从该相同的 CPW 接收到的先前的信息之间存在的一个或多个差异。在本实施例中,最终传回移动设备 102 的仅是这样的差异信息。如已经提到,由图 5 所表示的、并且与图 4 的步骤 408 相对应的子步骤在子步骤 518 处结束。将认识到,步骤 516 能够有利地在步骤 504 与步骤 506 之间在后端部分 306 中发生,在该情况下如果从内容被拉(pull)用于特定订户的先前时间在 CPW 信息中存在改变,则信息将仅在 web 服务器 104 中进一步被处理。这将释放服务器资源以继续从 CPW 拉信息以用于设备 102 的用户或使用中间 web 服务器和 CPW 的其它用户。

[0047] 返回图 4,一旦完成了步骤 408,则 web 服务器 104 考虑经处理的信息中的一个或多个部分具有高重要性还是不具有高重要性(例如,具有低重要性,或可能具有中间重要性或某个其它重要性级别)。如果确定经处理的信息具有高重要性,则在步骤 412 处 web 服务器 104 的前端部分 308 经由跨越通信链路 105 建立的推式信道将高重要性经处理的信息发送到移动设备 102。这在由 web 服务器确定的时间立即发生,如经由使用推式信道可能进

行的。如果在步骤 410 处确定经处理的信息不具有高重要性,则经处理的信息的发送能够被延迟直到另一更适当的时间为止以从而减少设备与服务器之间的通信活动,并且因此减少设备上的电池消耗。因此,在步骤 414 处,web 服务器 104 等待适当的时间将经处理的信息发送到移动设备 102。然后,一旦适当的时间已经出现,则在步骤 416 处然后信息由 web 服务器 104 发送到移动设备 102。

[0048] 低重要性经处理的信息由 web 服务器 104 发送到移动设备 102 的适当时间能够基于各种考虑。例如,在一些实施例,这样的适当时间仅仅是定期出现的时间,在该时间处移动设备 102 轮询 web 服务器 104 以获得信息。这样的轮询通常包括将来自移动设备 102 的查询信号重复地发送到 web 服务器 104。在其它情况下,当特定的环境已经出现时适当的时间就出现。例如,当移动设备 102 做出请求如果同时是在相同的时间 web 服务器 104 已经确定特定数量的低重要性经处理的信息已经被存储用于传输到移动设备的情况时,用于发送低重要性经处理的信息的适当时间能够出现。尽管在上述描述中,通过 web 服务器 104 获得信息被描述为包括拉,而通过移动设备从 web 服务器获得低重要性信息被描述为包括轮询,但是应当理解,取决于实施例,拉操作或轮询操作(和定期通信或者异步通信)能够分别由 web 服务器和移动设备中的任何一个使用,以分别从 CPW 106 和 web 服务器获得信息。另外,设想当移动设备 102 未被连接到系统 100 时,服务器能够正从内容提供网站 106 拉信息,作为这样的结果服务器将保留信息直到移动设备重新连接为止,或者当足够的时间过去以致服务器删除该信息时。

[0049] 无论高重要性或低重要性信息是否分别在步骤 412 和步骤 416 处发送到移动设备 102,一旦完成这些步骤,则一系列的另外的步骤由正在与移动设备、CPW、或另外的移动设备/CPW 交互的 web 服务器 104 来执行。更具体地在这一点上,一旦完成了步骤 412 和步骤 416,则在步骤 418-428 处,来自移动设备 102 的信息能够被上传到 web 服务器 104 并且进一步提供给 CPW 106。如图 4 中所示出,在步骤 418 处,这样的交互能够通过 web 服务器 104 从移动设备 102 接收标识信息开始。这样的标识信息的接收不必一直发生,例如,如果这样的标识信息已经在步骤 402 处被接收到。然后,在步骤 420 处,web 服务器 104 另外地从移动设备 102 接收内容信息。内容信息能够包括例如诸如图像文件或文本文件的文件、或移动设备的用户想要已经上传到在 CPW 处存在的用户简档(例如,“背景墙”)的其它数据。

[0050] 接下来,在步骤 422 处,web 服务器 104 从移动设备 102 接收指令 web 服务器将内容信息上传到 CPW 106 的命令。在替代实施例中,这个命令不必由移动设备 102 明确地提供给 web 服务器 104,因为在这样的实施例中,web 服务器假定由移动设备提供的所有内容信息应当进一步被上传到该移动设备所关联的任何 CPW。进一步地,然后在步骤 424 处,web 服务器 104 将从移动设备 102 接收到的标识信息发送到 CPW 106 以便认证该 web 服务器与该 CPW 之间的关系。响应于发送这个标识信息,通常如果认证是符合要求的,则从 CPW 往回接收到令牌,如由步骤 426 所指示的。如关于步骤 418,在所有的实施例中步骤 424 和步骤 426 不必在这个时候被明确地执行,特别地在这样的动作被视为在步骤 402、404 中的通信链路的建立的一部分的情况下。无论认证何时发生,认证处理允许 web 服务器 104 代表移动设备 102 和作为移动设备 102 的代理来与 CPW 106 进行交互。假定适当的认证已经发生,则在步骤 428 处内容信息由 web 服务器 104 发送到 CPW 106。

[0051] 设想当移动设备 102 首次连接到服务器并且在 web 服务器上建立内容提供商网站

时,用于内容提供商网站上的特定用户账户的针对 web 服务器 104 将内容上传到内容提供商网站 106 和从内容提供商网站 106 下载内容所需要的用户 ID 和密码能够由用户加载到 web 服务器 104 中。web 服务器将用户 ID 和密码存储在存储器中,并且只要用户不改变他们,就使用该用户 ID 和密码来访问 CPW,以维持与 CPW 的持续连接,不管移动设备 102 是否被连接。进一步设想如果对于预定的时间段移动设备不从服务器请求信息,或者如果用于将内容下载到设备的用户设备队列超过老化阈值和 / 或存储容量阈值,则能够在频率上减少或者完全暂停从 CPW 106 拉信息。

[0052] 除了先前描述的上传处理之外,在一些环境下,操作移动设备 102 的用户将还期望内容被上传到不止一个 CPW 106。这样的处理能够通过 web 服务器 104 来促进,如由图 4 的步骤 430-438 所指示的,特别在内容信息已经由移动设备 102 提供给 web 服务器的情况下。更具体地如所示,在步骤 430 处,由 web 服务器 104 确定 web 服务器是否已经从移动设备 102 接收到指令 web 服务器将内容信息提供给另一 CPW 的进一步命令。如果已经接收到这样的命令,则在下一个步骤 432 处,web 服务器 104 确定是否已经建立与其它 CPW 的通信链路。如果尚未建立这样的通信链路,则处理前进到步骤 434,其中从移动设备 102 接收到另外的标识信息,并且随后在步骤 436 处在 web 服务器 104 与其它 CPW 106 之间建立通信链路。也就是说,如果尚未建立与其它 CPW 的通信链路,如在步骤 432 处所确定的,则为了建立这样的通信链路,再次必须从移动设备 102 向 web 服务器 104 提供标识信息,允许 web 服务器与该其它 CPW 有关地被认证以便操作为与该其它 CPW 有关的移动设备的代理(例如,与上文中与步骤 424-426 有关地所描述的基本上相同的操作)。

[0053] 一旦步骤 436 处建立通信链路,或者如果在步骤 432 处确定已经建立与其它 CPW 的通信链路,则处理前进到步骤 438,在该处内容信息被上传到其它 CPW。因此,借助于步骤 430-438,在步骤 428 处已经提供给第一 CPW 的内容信息被另外地提供给另一 CPW。将理解的是,尽管图 4 没有示出在重复执行步骤 418-438 中的立即循环,但是步骤能够与无数部分的信息和不止一个另外的 CPW 有关地重复无数次。设想将以统一的格式从移动设备 102 提供内容,并且服务器后端将单独地并且适当地对数据进行格式化以用于内容正被上传到的目标 CPW 中的每一个。

[0054] 进一步关于图 4,一旦完成了步骤 438,或者在步骤 430 处由 web 服务器 104 确定没有接收到命令的情况下,则在步骤 440 处 web 服务器另外继续进行以确定移动设备 102 是否已经从 web 服务器断开连接。即使移动设备 102 已经从 web 服务器 104 断开连接,但是作为一般规则 web 服务器将仍然维持其与 CPW 106 的通信链路,web 服务器先前已经进入对于该 CPW 106 的通信,并且与该 CPW 106 相关地该 web 服务器能充当代表已经被断开连接的移动设备的代理,如由步骤 442 所表示的。因此,即使 web 服务器代表其正在行动的移动设备 102 暂时地离开通信,web 服务器 104 也能够在正在进行的基础上继续与 CPW106 相关地进行操作。因此,web 服务器 104 能够继续操作以从各种 CPW 106 拉信息并且可以随着时间的推移访问和监视这样的信息,使得当先前断开连接的移动设备被重新连接到 web 服务器时,web 服务器能够立即(如果有的话)提供可用的最近的、更新的 CPW 信息。

[0055] 尽管上文的描述,并且尽管图 4 中未示出,但是在某些实施例中移动设备 102 还可能将该 web 服务器停止代表它本身与 CPW 106 中的一个或多个相关地行动的指令传送到 web 服务器 104,在该情况下,web 服务器将这样做。最后,还如图 4 中所示,当步骤 442 已

经完成或倘若在步骤 440 处确定移动设备 102 仍然连接时, web 服务器 104 继续确定是否存在与移动设备 102 和 / 或 CPW 106 中的其它移动设备 102 和 / 或 CPW 106 建立另外的通信链路的需要或期望。根据本流程图, 如果不存在这样的需要或期望, 则处理在步骤 446 处结束, 而如果存在这样的需要或期望, 则处理返回到开始步骤 400。

[0056] 应当理解的是, 尽管如图 4 中所示的特定步骤, 但是取决于实施例各种另外的或不同的步骤能够由 web 服务器 104 来执行, 并且取决于实施例能够对图 4 中所示的特定步骤中的一个或多个进行重新排列, 重复或完整地消除。而且, 根据图 4 的流程图执行的步骤中的一些能够当执行步骤中的其它步骤的同时在正在进行的或连续的基础上重复。例如, 即使当诸如与从移动设备到 web 服务器并且然后到 CPW 中的一个或多个的内容信息的上传有关的由步骤 418-438 表示的那些的其它交互也正在进行时, 与从 CPW 106 接收到的信息的获得和处理和到移动设备 102 的高重要性信息的立即 (或基本上立即) 发送有关的步骤 406-412 能够在正在进行的或连续的基础上重复。另外, 尽管图 4 相当详细地描述了 web 服务器 104 正与多个 CPW 106 连续地或同时地进行通信的可能性, 并且图示了由给定移动设备和这样的一个或多个 CPW 之间的 web 服务器促进的示例交互, 但是应当理解的是, 就允许类似的交互在任何数量的其它移动设备和这样的一个或多个 CPW 之间发生而言, 可以由 web 服务器在相同的时间或基本上相同的时间执行相同的处理。

[0057] 设想后端部分 306 能够包括用于每个 CPW 106 的单独的插件, 包括适合其相应 CPW 的相应 API。插件中的每一个都包括用于其相应 CPW 的 API, 插件由此从网站拉信息并且将该信息重新格式化成移动设备 102 客户端的通用格式。此外, 当由后端部分 306 上传时, 来自移动设备的内容将从移动设备 102 客户端程序的统一格式重新格式化为由与该插件相关联的 CPW 规定的适当格式。以这种方式, 来自用户设备 102 的内容能够以具有统一格式的单个消息被发送, 并且其将如由用户选择一样被路由和由用于作为其目标的相应 CPW 中的每一个的后端部分插件中的每一个格式化。

[0058] 转向图 6, 提供了示出当移动设备 102 与 web 服务器进行交互, 并且借助于这个交互能够与一个或多个 CPW 进行交互时移动设备 102 的操作的示例步骤的另外的流程图。也就是说, 图 6 旨在图示移动设备 102 的操作的示例步骤, 其相对于如上文中的图 4 和图 5 中所图示的由 web 服务器 104 执行的多个步骤是补充的 (或者大部分是补充的)。此外, 如在下文中将被进一步描述的, 图 6 还包括使移动设备 102 能在不用通过 web 服务器 104 的调解的情况下直接或者与 (但是独立于) 通过 web 服务器的调解一起同时地与 CPW 106 中的一个或多个进行交互的步骤。如图 6 中所示, 一旦在开始步骤 600 处开始操作, 则在步骤 602 处移动设备 102 通过与 web 服务器建立通信链路并且经由 web 服务器因此与 CPW 建立通信链路来开始其与 web 服务器 104 的交互。

[0059] 另外参考图 7, 步骤 602 能够被理解为包括如在图 7 中所图示的若干子步骤。如所示, 一旦在子步骤 700 处开始, 则移动设备 102 激活在移动设备上支持的推式信道应用, 如在子步骤 702 处所指示的。然后, 在子步骤 704 处移动设备 102 将标识信息提供给 web 服务器 104。这样的标识信息能够包括例如规定特定移动设备 (例如, 序列号、型号或者产品参考号) 的标识码、与利用该移动设备的用户的身份有关的信息、或诸如登录或密码代码的其它编码信息。接下来, 在子步骤 706 处, 确定在移动设备 102 处是否存在经由 web 服务器与 CPW 106 中的特定一个建立通信链路的期望。如果在这个时候不存在这样的期望, 则

由图 7 表示的处理在子步骤 708 处结束。替代地,如果存在经由 web 服务器 104 与 CPW 106 建立通信链路的期望,如能够由将指示这样的期望的命令提供给移动设备 102 的用户所指示的,则在子步骤 710 处移动设备另外地指令 web 服务器建立这样的通信链路的命令发送到 web 服务器。

[0060] 另外,在子步骤 712 处,移动设备 102 另外地指令 web 标识信息发送给 web 服务器 104,允许该 web 服务器与 CPW 106 建立通信链路,并且充当用于与该 CPW 进行其通信的移动设备的代理。在一些实施例中在子步骤 712 处发送的标识信息能够与子步骤 704 的标识信息相同,在该情况下不需要执行子步骤 712。一旦在子步骤 712 处已经提供标识信息,则在子步骤 714 处在移动设备与 web 服务器之间建立了推式信道链路。一旦完成了子步骤 714,则能够执行在步骤 602 之后的由图 6 表示的处理的剩余步骤(如由框“返回到 A”所指示的)。

[0061] 返回到图 6,一旦在步骤 602 处建立了与 web 服务器 104 的通信链路,则在步骤 604 处移动设备 102 经由推式信道(例如,在子步骤 714 处建立的推式信道)从 web 服务器接收高重要性信息。如已经参考图 4-5 所描述的,在本实施例中以异步的方式,也就是,不是由移动设备确定的时间将该信息从 web 服务器 104 提供给移动设备 102。除了在异步的基础之上接收这样的高重要性信息之外,如由后续步骤 606 所进一步表示的,移动设备 102 能够另外地指令待由 web 服务器下载到移动设备的其它信息有关的一个或多个查询发送到 web 服务器 104。如在上文中参考图 5 所讨论的,尽管高重要性信息能够包括诸如状态更新信息的信息,但是其它信息(例如,低重要性信息)能够包括诸如联系人/朋友信息、新朋友信息、联系人列表、相片或视频、特殊消息、新闻或者事件信息。

[0062] 在步骤 606 处由移动设备 102 提供的查询能够在定期性的基础上或者在由该移动设备确定的其它时间被提供。尽管在本实施例中设想移动设备 102 将确定何时对 web 服务器 104 进行查询,该查询进而确定是否将除了高重要性信息之外的信息从 web 服务器传送到移动设备,但是在其它实施例中这样的查询和/或信息的下载能够在通过 web 服务器与移动设备之间的双方协定所确定的时间、在由 web 服务器独自单独地所确定的时间(例如,当 web 服务器已经确定已经收集到足够量的低重要性信息时)、或在诸如已经对两个设备进行编程的制造商的另一实体或一方所确定的时间发生。无论其是否是来自移动设备 102 的促使通过 web 服务器 104 将信息往回发送到移动设备的查询或者其是否是促使发送这样的信息的其它触发,如在步骤 608 处所指示的,最终还由移动设备从 web 服务器接收到这样的其它信息。尽管步骤 602 能够被认为是图 4 的步骤 402 的补充,但是步骤 604-608 能够被认为是由图 4 的步骤 406-412(并且特别地步骤 414-412)表示的 web 服务器操作的补充。

[0063] 仍参考图 6,在后续步骤 609 处,由移动设备 102 从 web 服务器 104 接收到的信息通过移动设备来显示或以其它方式输出。这样的信息的显示/输出发生的程度将取决于实施例。在至少一些实施例中,信息通过移动设备 102 以标准化的方式来显示/输出,使得 CPW 特定格式信息或特征不被提供为所显示的/输出的信息的一部分。更具体地在一些这样的实施例中,CPW 特定格式信息和特征由 web 服务器 104、或者在一些替代实施例中由移动设备或 web 服务器和移动设备两者的组合来编校。

[0064] 在执行这样的编校中,在不同的 CPW 处发现的类似类型的信息,即使通过不同的 CPW 以不同的方式引用(例如,如在发布站点发现的信息或代替如在背景墙上发现的信

息),也被识别为在概念上类似类型的,并且基于这样的识别,这样的信息能够以常见的方式被显示(可输出)在移动设备上,而无论信息的起源如何。也就是说,考虑到这样的 CPW 特定格式信息或特征的编校,来自不同的 CPW 的相同的概念类型的信息,即使在不同的 CPW 处不同地格式化,仍然以相同或类似、一致的方式被显示在移动设备上,而无论该信息的起源如何,因此有助于用户的对这样的信息的回顾。应当进一步注意到,这样的信息能够不仅包括文本和图像数据而且包括广泛的各种其它数据,包括支持在移动设备上的交互式窗口和数据条目字段的显示的数据,用户能够将然后能够被发送回到 web 服务器的另外的信息或命令键入到交互式窗口和数据条目字段中。

[0065] 接下来,在步骤 610 处,移动设备 102 确定是否存在将在移动设备处当前可用的内容信息上传到 web 服务器和 / 或最终地上传到 CPW106 的需要或期望。例如,基于是否已经由移动设备从用户或其它来源接收到特定类型的信息或者特定的事件是否已经发生或者触发这样的上传事件的时间是否已经过去,该需要或期望能够由移动设备 102 自动地确定。通常,响应于提供给移动设备 102 的用户命令,这样的需要 / 期望将发生。如果在步骤 610 处,确定不存在这样的需要 / 期望,则如所示该处理前进到下文中所讨论的步骤 622。然而,如果在步骤 610 处确定存在这样的需要 / 期望,则在步骤 612 处移动设备 102 将内容信息发送到 web 服务器 104 并且在步骤 614 处移动设备另外地将用于将该内容信息上传到 CPW 106 的命令发送到 web 服务器。步骤 610-614 能够被理解为通常为图 4 的步骤 418-428 的补充,除了在如参考步骤 418 所讨论的从移动设备 102 提供以用于认证目的标识信息能够被理解为已经在图 6 中所示出的步骤 602 处提供(替代地,适合于这个目的的另外的标识信息能够被刚好在步骤 612 之前提供)的范围内除外。

[0066] 一旦完成步骤 614,则在步骤 616 处,移动设备 102 进一步确定是否存在将内容信息上传到除了该信息已经被上传到的第一个 CPW 以外的一个或多个另外的 CPW 的需要 / 期望。再者,该需要或期望能够基于包括除了别的以外由移动设备的用户提供给移动设备的一个或多个指令的各种因素来确定。如果在步骤 616 处确定不存在这样的需要或期望,则该处理再次前进到下文中所讨论的步骤 622。然而,如果在步骤 616 处确定存在这样的需要或期望,则处理前进到步骤 618,其中经由 web 服务器在移动设备与这样的另外的 CPW 之间建立另外的通信链路。步骤 618 能够被视为图 4 的步骤 432-436 的补充并且取决于实施例能够包括子步骤,其中移动设备首先确定与这样的另外的 CPW 的通信链路是否已经存在,并且如果确定这样的通信链路还不存在,则将另外的标识信息发送到 web 服务器以与这样的另外的 CPW 建立通信链路并且允许 web 服务器在这样的通信中充当移动设备的代理。

[0067] 一旦在步骤 618 处与另外的 CPW 106 建立了另外的通信链路,则在步骤 620 处移动设备 102 进一步将用于将内容信息上传到该另外的 CPW 106 的命令发送到 web 服务器 104。步骤 620 的执行能够被理解为对应于图 4 的步骤 430,要进一步理解的是,步骤 618 和步骤 620 的执行的顺序是可逆的,使得那些步骤更接近地对应于图 4 的步骤 430-436 的顺序。另外参考图 6,一旦完成步骤 620,则假定 web 服务器 104 实际上将内容信息上传到另外的 CPW。尽管未示出,但是在一些实施例中,一旦完成这样的上传,则 web 服务器 104 将确认这样的上传已经发生的指示信号发送回到移动设备 102。

[0068] 尽管图 6 的上述步骤以及图 4 的步骤将 web 服务器 104 的使用预想为移动设备 102 与 CPW 之间的中间物,但是 web 服务器不必一直调解这样的通信而是在一些情况下移

动设备直接（也就是说，直接经由不包括任何 web 服务器、或者至少不包括如上文所描述的 web 服务器的一个或多个网络）关于 CPW 中的一个或多个进行交互。在该点上，一旦完成步骤 620（或，在一些情况下，如上文所讨论的步骤 610 和步骤 616），则在步骤 622 处移动设备 102 进一步确定是否存在移动设备与 CPW 106 中的一个或多个直接进行通信的需要或期望。

[0069] 如果在步骤 622 处移动设备 102 确定这不是该情况，则移动设备能够将其操作返回到节点 A，响应于此，该处理在步骤 604 处再次开始并且向前进行。假定这个发生，移动设备 102 因此继续不仅从 web 服务器 104 接收信息而且也继续操作以在重复、持续的基础上将内容信息上传到 web 服务器。然而如果在步骤 622 处移动设备 102 确定存在直接与 CPW 106 进行通信的需要或期望，则移动设备继续进行到步骤 624，在该处移动设备建立这样的直接通信链路。

[0070] 是否存在直接与 CPW 106 进行通信的需要或期望能够基于各种考虑来确定。在一些情况下，移动设备 102 对此自动地确定并且因此自动地继续进行以与 CPW 106 建立直接通信链路。例如，如果用户请求关于特定主题的更多的信息并且从给定 CPW 的该信息的下载经由与 CPW 的直接通信来最好地完成（例如，就数据传输的效率等等而言），则移动设备能够尝试直接连接到 CPW。而且，可能在一些情况下，用户可能希望查看在特定 CPW 处可用的具有与该 CPW 相关联的特定格式的信息，并且可能不希望查看这样的信息的编校的视图，如在信息到移动设备的途中被 web 服务器 104 处理的情况下可能提供的。而且，是否存在直接与 CPW 106 进行通信的需要或期望的确定能够基于明确地请求这样的通信的用户命令的接收来确定。

[0071] 取决于实施例，在步骤 624 处的直接通信链路的建立能够包括各种特定的命令或移动设备的操作，其在一些环境下能够包括从用户接收输入。例如，在一种环境下，用户通过使浏览器应用 / 程序在移动设备上打开和运行并且通过将用于 CPW 的 URL（统一资源定位符）键入到由浏览器提供的输入字段来发起这样的直接通信链路的建立，因此浏览器进入与 CPW 的通信中并且 CPW 进而将网页或其它信息返回到浏览器，移动设备（和用户）由此能够参与与 CPW 的进一步的通信。在其它实施例中，直接通信链路的建立是不包括任何特定用户动作的自动处理。

[0072] 无论如何建立直接通信链路，一旦建立了该链路，则在进一步的步骤 626 处移动设备 102 直接将信息发送到 CPW 106 和 / 或直接从 CPW106 接收信息（再次，没有上文所描述的 web 服务器的调解）。随后，在步骤 628 处，移动设备进一步确定是否存在停止与 web 服务器 104 的现有通信链路的需要 / 期望。如果不存在这样的需要 / 期望，则处理返回到节点 A 并且再次步骤 604 和后续步骤被重复。也就是说，在移动设备与 CPW 之间的直接通信（没有 web 服务器调解）和间接通信（经由 web 服务器）两者能够同时地继续。然而，如果在步骤 628 处确定存在停止基于服务器的通信的需要或期望，则处理前进到步骤 630，在该处与 web 服务器的移动设备通信被中断（其对应于在上文中关于图 4 所讨论的步骤 440）。

[0073] 在本实施例中，如上文所讨论的，web 服务器 104 被配置成将其本身维持在其先前代表移动设备与其进行通信的 CPW 或站点的通信中，即使在与移动设备的通信已经终止之后，其中 web 服务器继续充当移动设备的代理。然而，在其它实施例中，当移动设备终止其与 web 服务器的通信时，web 服务器与 CPW 的通信被切断。在任何情况下，在步骤 630 之后，

在步骤 632 处可能存在代表移动设备与 web 服务器重新建立通信的新的需要或期望。正如在步骤 622 处确定是否与 CPW106 进入直接通信或者在步骤 628 处停止与 web 服务器 104 通信的情况下一样,在步骤 632 处是否存在代表移动设备 102 重新建立与 web 服务器 104 通信的需要或期望能够基于各种考虑中的任何一个,包括例如触发这样的活动的用户命令、电池电源考虑等。如果在步骤 632 处确定应当重新建立基于服务器的通信,则处理返回到开始步骤 600。如果不是,则在结束步骤 634 处结束由图 6 表示的处理。

[0074] 分别转向图 8 和图 9,在进一步的实施例中,由 web 服务器 104 和移动设备 102 执行的操作能够与图 4-7 中所示的那些稍微不同。更具体地,在一些其它实施例中,web 服务器 104 不是执行图 4 中所示的节点 B 与节点 C 之间的步骤 408-416,而是以不同的方式操作,包括图 8 中所示的步骤 800-814。如所示,一旦从节点 B 行进,则 web 服务器 104 不是执行处理步骤 408(和图 5 中所示的对应步骤),而是执行步骤 800、802 以及 804。特别在步骤 800 处,web 服务器 104 确定在步骤 406 中刚刚从 CPW 106 获得 / 拉的信息与在更早时间从该 CPW 先前接收到的信息之间是否已经发生改变。如果在步骤 802 处检测到改变,则在步骤 804 处 web 服务器 104 的前端部分 308 将该改变信息放到改变列表中。在这些步骤与 web 服务器 104 与其联系的多个 CPW 有关地重复地被执行的情况下,与 CPW 中的每一个有关地检测到的改变信息能够全部被置于改变列表中,在该情况下其能够被称为通用改变列表。

[0075] 接下来,在步骤 806 处,web 服务器 104 的前端部分 308 确定经处理的信息是具有高重要性还是不具有高重要性(例如,低重要性)。在执行这个确定中,能够考虑到与在上文中与图 4 的步骤 410 有关地所讨论的相同的考虑,并且因此在图 8 中步骤 806 还被标记为步骤 410。取决于经处理的信息是被确定具有高重要性还是低重要性,该处理然后相应地前进到步骤 808 或步骤 810。在步骤 808 中,一旦已经确定经处理的信息具有高重要性(例如,信息关系到状态更新),则 web 服务器 104 的前端部分 308 经由推式信道将指示已经发生高重要性改变的通知发送到移动设备 102。同样地,在步骤 810 处,一旦已经确定经处理的信息具有低重要性,则 web 服务器 104 的前端部分 308 经由推式信道将指示已经发生低重要性改变的通知发送到移动设备 102。

[0076] 一旦在步骤 808 或步骤 810 中已经发送了通知,则在步骤 812 处 web 服务器 104 的前端部分 308 在稍后的时间能够从移动设备 102 接收用于自身发送该改变信息的请求。能够在如由移动设备 102 确定的任何时间接收该请求。通常,如果改变信息具有高重要性,则移动设备 102 将在步骤 808 处接收到通知之后立即或非常快发送对于该信息的请求。与此相反,如果改变信息具有低重要性,则移动设备经常等待直到对于这样的请求的预定时间(例如,定期性或非定期性轮询时间)已经到达为止。例如,设备可以等待不超过 5 分钟来请求高重要性信息并且在请求之间等待 15-30 分钟来下载低重要性信息。在任何情况下,一旦在步骤 812 处从移动设备 102 接收到对于改变信息的传输的请求,则所请求的改变信息随后由 web 服务器 104 的前端部分 308 发送到移动设备 102。在本示例中,优选的是,这个改变信息不通过推式信道来发送,或者替代地仅高重要性改变信息通过推式信道来发送,以减少移动设备被加电以接收该改变内容的时间量,然而认识到在其它实施例中所有的改变信息都能够经由推式信道来发送。一旦在步骤 814 处发送这个信息,或如果在步骤 812 处未接收到对于该信息的请求(或至少在预定时间段内未接收到)或如果在步骤 802

处在从 CPW 106 接收到的信息中未检测到改变,则该处理返回到图 4 的节点 C(并且因此返回到步骤 418)。应当认识到,如果没有内容应当需要用于上传到内容提供商网站,则当其将继续从内容提供商网站 106 拉内容时 web 服务器 104 将返回到步骤 406,与内容是否正被上传到移动设备 102 客户端无关。

[0077] 尽管在本示例中,在步骤 808 和 812 处经由推式信道以相同的方式来提供改变信息的通知,而不管该改变信息是具有高重要性还是具有低重要性,但是这个不必一直是这种情况。在其它实施例中,例如,有关高重要性改变的通知能够比有关低重要性改变的通知更迅速地、或以某种其它方式被发送。另外,虽然在图 8 的本示例中在步骤 814 处改变信息的发送发生在与在步骤 808、810 处的通知的发送不同的时间,但是这个不必一直是这种情况。例如,在一个其它实施例中,在高重要性改变信息的内容为小的(例如,小于 100 个字符的文本消息)的情况下,该内容能够与高重要性改变的通知一起(或者甚至充当高重要性改变的通知)被提供。从上述描述中,还应当显而易见的是,在至少一些实施例中,就与 CPW 106 和移动设备 102 的不同部分的相应的通信而言,后端部分的操作能够大体上或完全地与前端部分的操作无关。取决于实施例,各种不同类型的通信,例如包括拉或轮询的那些、或定期性或异步的通信能够被任一端部分采用,而不管另一端部分的操作。因此,后端部分 306 能够连续地从 CPW 106 拉内容并且将改变发送到前端部分 308,而不管前端部分正在做什么。前端部分 308 能够同样地推到移动设备 102 并且等待请求下载改变内容,或使服务器和移动设备同步,而不用关心在任何特定时刻后端部分 306 正在做什么。

[0078] 至于图 9,在其中所提供的流程图示出了在一些其它实施例中,如何不是执行图 6 中所示的节点 A 与节点 D 之间的步骤 604-609,而是移动设备 102 以包括步骤 900-914 的不同的方式操作。图 9 中所示的由移动设备 102 执行的步骤 900-914 是特别地相对于图 8 中所示的由 web 服务器 104 执行的步骤 800-814 的补充。如图 9 中所示,一旦从节点 A 继续,则移动设备 102 不是执行图 6 的接收步骤 604,而是能够从 web 服务器 104 接收在从 CPW 106 最近并且在更早时间提供的信息中已经检测到一个或多个改变的通知(在步骤 808、810 中的一个或两者处发送的)。如果在步骤 900 处接收到通知,则在步骤 902 处移动设备 102 确定该通知是指示该改变具有高重要性还是具有低重要性。

[0079] 如果在步骤 902 处确定该改变具有高重要性,则在步骤 904 处移动设备 102 确定是否应当立即从 web 服务器 104 获得该高重要性改变信息。尽管在一些实施例中一直是应当尽可能快地获得高重要性改变信息的情况,但是在其它实施例中,由于各种原因移动设备仍然能够确定将优选地推迟尝试从 web 服务器获得该信息(例如,因为移动设备电量低)。假定在步骤 904 处移动设备 102 确定应当立即获得改变信息,则处理前进到步骤 906,在该处移动设备立即将用于请求将高重要性改变信息立刻提供给移动设备的请求信号发送到 web 服务器。作为响应,在步骤 908 处,移动设备 102 最终从 web 服务器接收到所请求的改变信息(或该信息中的至少一些,如由 web 服务器 104 所确定的)。在这点上,步骤 908 的执行补充了图 8 的步骤 814 的执行。

[0080] 如果替代地在步骤 902 处由移动设备确定该通知指示改变信息具有低重要性,或如果在步骤 904 处移动设备确定不应当(或需要)立即获得改变信息,则处理前进到步骤 910。在步骤 910 处,移动设备 102 进一步确定用于轮询 web 服务器 104 以获得改变信息的适当时间是否已经出现。这样的适当时间可以是定期性出现的时间,或在其它实施例中,能

够由移动设备 102 基于各种其它考虑（例如，自从另一事件已经流逝的预定量的时间，或如接收到指令移动设备从 web 服务器 104 获得内容信息的用户命令）来确定。

[0081] 如果在步骤 910 处用于轮询 web 服务器的适当时间仍然未出现，则处理能够重复该步骤直到这样的时间出现（或能够前进到处理的另一步骤和 / 或可能地在不同的时间返回到步骤 910）为止。然而，如果在步骤 910 处适当时间已经出现，则处理前进到步骤 912，在该处轮询 / 请求信号由移动设备 102 发送到 web 服务器 104。一旦发送了该信号，则该处理返回到移动设备 102 接收所请求的改变信息的步骤 908。进一步如图 9 中所示，一旦完成步骤 908，则移动设备 102 继续进行执行步骤 913，其中所接收到的信息由移动设备 102 来显示或以别的方式输出以支持移动设备的用户对信息的回顾。如所示，步骤 913 能够相同于或类似于图 6 的步骤 609。

[0082] 虽然在步骤 908 处由 web 服务器 104 发送的改变信息常常是移动设备 102 的用户最感兴趣的，但是这个改变信息常常不包括在由 web 服务器对该信息进行处理之前在 CPW 106 处原始可用的各种内容（以及格式化）信息。也就是说，虽然由 web 服务器 104 提供的信息能够包括诸如事件、最近的状态信息、来自其它人的评论等的各种内容，并且虽然移动设备 102 也能够理所当然将某个标准信息显示为其用户界面的一部分（例如，用户的名称、用户与其联系的 CPW 等），但是由于 web 服务器 104 的调解而造成相当量的内容和 / 或其它信息能够被排除在外。因此，一旦在步骤 913 处显示了改变信息，则用户可以决定不仅获得改变信息而且获得其它内容（或甚至格式化）信息将是所希望的。如果用户希望获得这样的其它信息，则在后续步骤 914 处移动设备进一步确定是否已经接收到用于获得不是在步骤 908 处从 web 服务器 104 接收到的其它信息的用户命令。例如，当用户选择由移动设备显示的图标时，能够接收到这样的命令，其在步骤 913 处可以被显示为改变信息的一部分。

[0083] 如果在步骤 914 处确定接收到这样的命令，则在步骤 916 处移动设备 102 与 CPW 106 建立直接通信链路。建立直接通信链路的这个操作能够相同于或类似于与上文所讨论的步骤 624 相关联的操作，并且能够包括被设计成既建立通信链路又发出用户所期望的其它信息的标准的基于 web 的客户端 - 服务器通信（例如，包括统一资源定位符 (URL) 的输入 / 传输和 / 或与 CPW 106 的 web 页面进行对接）。因此，一旦在步骤 916 处建立直接通信链路之后，则在步骤 918 处从 CPW 106 接收用户所期望的其它信息。一旦完成步骤 918，以及在确定在步骤 914 处没有接收到用户命令的情况下或在步骤 900 处接收到来自 web 服务器 104 的通知的情况下，则该处理返回到节点 D 并且继续图 6 的步骤 610。

[0084] 在本发明的另一替代实施例中，后端部分 306 包括多个插件或处理器，其中的每一个都与相应的 CPW 106 相关联。每个插件都包括用于其关联 CPW 106 的应用编程接口 (API)。每个插件都使用超文本传输协议 (HTTP) 以持续地从其相应内容提供商网站 106 拉信息。

[0085] 当后端部分 306 插件检测到改变时，改变被加载到队列中并且前端部分 308 将通知推到移动设备 102。后端部分 306 中的所有插件将继续向队列加载根据包括例如信息源的 ID（源内容提供商网站标识）、用户设备的账户 ID、内容类型、优先级以及信息的通用格式所格式化的信息。例如，对于状态，格式可以为：类型（状态、情绪、STATUS_AND_MOOD）、动作（清除状态或更新状态）、提供商、聚合服务账号 id、外部 id、如果更新是用于朋友的朋友 id、状态文本、发布日期和时间。web 服务器 104 通过将由所有的插件拉的内容组合到用于

每个相应设备（或用户账号）的通用改变列表中来为每个用户设备（或用户账号）构建统一的订阅源。内容被随着时间的推移而构建，并且每个条目能够被加时间戳。

[0086] 以下算法能够用于检测在服务器同步期间的改变，其中服务器同步被理解成包括 web 服务器 104 与 CPW 106 的同步（通过比较，客户端同步能够被理解成包括诸如移动设备 102 的客户端与 web 服务器的同步）。web 服务器 104 程序维持用于每个账号的三个号码： cla 、 $w1$ 以及 $w2$ 。 cla 是改变列表锚， $w1$ 是改变列表窗口的开始时间（采样），而 $w2$ 是改变列表窗口的结束时间（采样）。服务器 104 存储落入窗口 $[w1, w2]$ 内部的改变列表的部分。在服务器同步期间（即，后端部分从 CPW 拉）找到的所有改变用等于当前 $w2$ （即，在 $w2$ 递增 1 之前）的同步锚被加戳。一旦窗口大小达到或超过最大窗口大小 mw ，则该程序就暂停服务器同步（内容提供商网站大小同步）。一旦暂停，则当接收到新的客户端轮询时服务器将恢复服务器同步。其它变量是作为客户端锚的 ca ，OFF 是指示没有同步活动的标记。 cla 、 $w1$ 以及 $w2$ 的值根据以下状态转变规则来更新：

事情	状态转变
初始化	$cla = 0, w1 = 0, w2 = 0, off = 0$
服务器同步	如果 $w2 - w1 \geq mw$ ，则 $w2 = w2 + 1$ ， $off = 1$
客户端同步， $w1 \leq ca \leq w2$	$w1 = ca, off = 0$
客户端同步， $ca < w1$ 并且 ($ca < w1$ 或 $ca > w2$)	$cla = w2 + 1, w1 = cla, w2 = cla, off = 0$ (也可以叫“改变列表复位”)

[0088] 当客户端轮询以获得改变时，如果客户端锚 ca 落入 $[w1, w2]$ 内部，则部分同步将工作并且服务器发送回落入 $[ca, w2]$ 的改变（并且删除比 ca 老的改变）。在对同步作出结论后， ca 将被更新。如果当客户端轮询以获得改变时，客户端锚落在 $[w1, w2]$ 的外部，则在 web 服务器 104 与移动设备 102 中的客户端程序之间将发生新的完全同步。

[0089] 设想当窗口大小达到 mw 时服务器同步（后端插件拉用于特定设备 102 的内容）能够针对特定移动设备 102 账户被暂停，在此情况下在缺少客户端轮询的情况下稍有丢失的推（到设备的通知）可能造成设备的服务中断。设想如果自上一个 $w2$ 以来存在未决的改变，则对于发送推式可能是有利的，只要自 $w1$ 以来存在未决的改变，则能够发送推式。

[0090] 进一步设想在本文中所描述的中间 web 服务器 104 能够有利地与于 2009 年 5 月 21 日提交的、题为 A MOBILE COMPUTING DEVICE AND METHOD WITH ENHANCED POLING MANAGEMENT 的美国临时申请 61/180,301 中所描述的设备轮询管理器一起使用。

[0091] 现将对作为上传内容的示例的还被称为图片的照片上传进行描述。通过将照片缓存在中间 web 服务器 104 存储器 302 处，能够采用中间 web 服务器 104 来优化将照片从用户设备 102 上传到多个社交网系统 106 的处理。示例性流程可以为如下：

[0092] 1. 在步骤 1002 处，中间 web 服务器前端 308 向后端部分 306 指示用户设备 102 上传了照片（图 10）；

[0093] 2. 在步骤 1004 处，web 服务器前端或后端部分将照片 URL 和系统范围的唯一照片

ID 给予新照片；

[0094] 3. 在步骤 1006 处, 照片 ID 被下载到设备 102, 响应于此, 设备客户端程序将照片 ID 与照片名称相关联；

[0095] 4. 在步骤 1008 处, 后端通过 HTTP 将文件下载到诸如 /tmp/uniquephotoid.tmp 的位置；

[0096] 5. 在步骤 1010 处, 与目标内容提供商网站中的每一个相关联的相应后端部分 306 插件为每个内容提供商网站提交 work.uploadPhoto 以上传这个照片文件；

[0097] 6. 在步骤 1012 处, 后端部分将照片分享的成功或失败的报告往回提供给前端部分；

[0098] 7. 在步骤 1014 处, 前端能够可选地向用户设备 102 通知成功或失败；

[0099] 8. 在步骤 1016 处, 在预定时间段已经过去之后, 该照片被删除。

[0100] 在操作中, 来自用户设备 102 的照片从用户设备 102 上传到网络的前端部分 308, 如由步骤 1102(图 11) 处所指示的。前端部分 308 或后端部分 306 将照片缓存在中间 web 服务器 302 中持续预定的时间段以允许在步骤 1206(图 12) 中将同一照片提交到不同系统的网站而不要求该照片由用户设备 102 再次上传。在预定的时间段之后, 该照片将被擦除。预定的时间段可以是任何时间段, 并且根据存储器约束和使用的频率来选择。时间段可以例如为 24 小时, 该时间段能够在相片被上传到存储器 302 的时间开始, 从而一旦图片被上传, 则设置了该时间段, 或者该时间段可以一旦照片上传到内容提供商 106 就开始, 从而每当图片被上传到新的内容提供商时该时间段就被延长。

[0101] 对于一个示例性实施例而言, 在步骤 1102(图 11) 处, 相片作为动作与指定的内容提供商网站 106 的标识一起从移动设备 102 被上传到中间服务器前端 308 并且存储在网络服务器的临时存储器中。在步骤 1004(图 10) 处, 前端 308 将该照片转发到后端 306 中的插件, 该插件能够例如专用于由移动设备指定的内容提供商网站。在步骤 1006 处, 网络服务器前端部分 308 还将包括与所保存的相片相关联的照片标识 (ID) 的消息往回发送到用户设备 102。照片 ID 标识位置或指向在 web 服务器存储器 302 处存储照片的位置的指示器。在步骤 1104 中移动设备接收照片 ID 并且如在步骤 1106 中所指示的将照片 ID 与照片的名称相关联 (映射)。随后, 在步骤 1108 中, 如果移动设备 102 经由用户接口选择将同一照片发送到不同的社交网系统, 则移动设备将照片 ID 而不是实际的图片发送到网络服务器。作为响应, 中间服务器 104 将检索照片并且将其转发到专用于另一内容提供商网站 106 的另一插件, 如步骤 1204(图 12) 中所指示的。在步骤 1110 中, 设想一旦从存储器 302 中移除照片, 则更新将被发送到用户设备 102 以使用户设备移除照片名称和照片 ID 的关联, 使得设备将上传该相片。如果另一方面不再存储照片, 并且服务器接收到用于上传与照片 ID 相关联的照片的请求, 则前端部分将错误消息发送到订户设备, 响应于此, 该订户设备将被邀请再次上传该照片。

[0102] 对于其它实施例而言, web 服务器后端部分 306 将确定从用户设备 102 上传的相片是否在目标社交网系统的必要限制 (例如, 尺寸和大小) 内。设想当从存储器 302 中移除图片时这个能够由与每个内容提供商网站相关联的插件来处理, 因为每个插件都能够存储对于相片的内容提供商网站的限制。如果满足限制, 则后端 306 能够将照片一直发送到目标内容提供商网站。否则, 照片将根据内容提供商网站的要求来调整大小。为了调整相

片大小和 / 或将照片缩放到目标大小,确定了调整大小因子。能够被用来确定调整大小因子 X 的特别有利的算法为如下:

$$[0103] \quad x/100 = ((t-f)/(kc))^{(0.5)}$$

[0104] 其中

[0105] x 是调整大小百分比

[0106] t 是以字节为单位的目标大小,并且可以例如大约 1 兆字节或更小,并能够有利地小于 200,000 字节,并且在一个实现中是 100,000 字节。

[0107] f 是用于文件大小的小容差因子

[0108] k 是常量因子,并且可以小于 1,以及有利地能够小于 0.5,而且在一个实现中被选择为 0.23。

[0109] c 是以字节为单位的原始文件的大小。

[0110] 通过将相片存储在 web 服务器 104 中,通过准许移动设备在不同的时间将媒体发送到不同的社交网络服务器同时通过设备通过其进行通信的局域网或广域网将该媒体仅仅上传一次,服务器有助于减少设备的电力消耗和通信网络上的带宽负担。此外,web 服务器能够采用每个内容提供商网站所期望的格式的媒体,并且设备不必知道或适应这些需求以成功地上传该媒体。

[0111] 还设想,能够在步骤 1302(图 13)中经由中间 web 服务器 104 将照片下载到设备 102。例如,对于 RSS 新闻订阅源而言,来自 RSS 内容源的照片由具有新闻订阅源概要的后端新闻订阅源来拉。当在步骤 1304 中后端 306 检测到这样的新闻信息是新的,或者换句话说自由后端部分从这个 CPW 拉先前的 RSS 新闻订阅源以来已经发生改变时,则服务器 104 的后端部分 306 将针对客户端设备将在步骤 1306 中适当地格式化的订阅源传送到前端部分。前端部分 308 将向客户端设备 102 生成低优先级推式通知并且设备 102 的队列将装载有概要和照片。当客户端设备此后将轮询请求发送到前端部分以获得内容时,在步骤 1308 中前端将传送包括包含已格式化的照片和概要的该新闻订阅源的队列的内容。移动设备 102 上的客户端程序将会将概要和关联的照片显示在移动设备 102 显示器 216 上。通过示例的方式,如果输入 210 包括在显示器上的触摸传感器(通常被称为触摸屏),则用户能够在概要和照片处触摸该屏,并且用户接口将通过链路 110 直接连接到与新闻订阅源概要 / 照片相关联的内容提供商网站并且将有关该新闻订阅源的另外的信息加载在显示器 216 上以供用户查看。后端部分 306 因此检测并且格式化用于该设备的新照片和概要,并且前端部分 308 向设备通知内容是可用的并且响应于来自设备的轮询请求而将新闻订阅源下载到移动设备 102。

[0112] 进一步设想,移动设备 102 中的客户端程序将存储有关用户具有服务器帐户的每个内容提供商网站的内容类型和特性的一些定义。在步骤 1502 中,设备的用户接口将根据用户在服务器上设定哪些账号而变化。例如,假定用户在他们的 web 服务器 104 帐号上进入 Facebook™ 和 Twitter™。当用户与用户接口交互以构建待上传到内容提供商网站的消息 1402(图 14)时,用户接口显示器呈现用于消息将被发送的目标内容提供商网站的“Facebook”、“Twitter”或“所有”的选择(未示出)。在步骤 1504 中,取决于做出了哪个选择,用于消息的参数可以是不同的(例如,字符的数量)。如果用户选择所有,则长度 1404 将是两个内容提供商网站限制中的较短的。进一步设想能够提供长度计数 1404 和告警

1406。在步骤 1506 中当用户键入文本时,在达到限制之前准许的剩余字符被显示在字符限制 1404 中。在步骤 1514 中,在某个阈值处,诸如 30 个字符,将显示告警 1406。在步骤 1516 中,当超过限制时,剩余字符将转向负计数,或者用户将被阻止输入另外的字符。在用户改变目的地内容提供商源的情况下,限制将适当地改变。例如,如果在创建消息之后 Twitter™ 网站被添加为目的地,则限制将减小。如果 Twitter™ 网站作为目的地被移除,则限制增加。

[0113] 移动设备 102 生成具有取决于用户设备在中间 web 服务器上设定的一个或多个内容提供商网站的操作参数 1404 的用户接口显示 216。对于消息,通用消息输入域 1402 被呈现在显示器 216 上以供用户输入文本,大小上限基于由被选择为消息文本的目的地的一个或多个社交网网站所准许的最小的最大消息大小。限制能够保留在客户端移动设备上。当消息大小变得在限制的预定量内时,移动设备客户端程序能够生成告警 1406。如果在步骤 1508 中一个或多个社交网网站改变,则在步骤 1510 中限制改变。来自用户接口输入的内容填充了消息输入区域 1402,并且当达到限制时能够生成告警。客户端程序用一个或多个目的地社交网 web 网站的身份来传送在步骤 1602(图 16)处由服务器前端接收到的消息。后端部分在步骤 1604 中针对一个或多个目的地网站对消息进行格式化并且在步骤 1606 中以社交网网站所期望的格式上传该消息。

[0114] 从以上描述中,应当明显的是,采用诸如上文所讨论的那些的许多不同的操作步骤的各种方法均由本发明包括。此外,除上文所描述的具体实施例以外本发明还旨在包括各种替代实施例,包括采用具有除了或代替上文所描述的那些以外的其它操作步骤的方法的实施例,以及采用具有以除了或代替上文所讨论的步骤的特定顺序或组合以外的各种顺序或组合的步骤的方法的实施例。进一步应当明显的是,根据上文所描述的实施例中的一个或多个的系统在就促进由用户操作的移动设备与社交网网站之间的交互而言的在若干方面中能够提供增强的功能性。取决于实施例,能够增强用户与社交网网站之间的通信的质量、如移动设备用户所体验的社交网网站和关联的事务处理的用户友好性、和 / 或移动设备与这样的网站之间的通信的效率中的任何一个或多个。

[0115] 图 17 图示了另一示例性通信系统 1700。图中的通信系统 1700 包括中间 web 服务器 1704,在本文中还被称为 web 服务器、媒介 web 服务器、聚合服务器、或聚合服务。web 服务器 1704 可以包括内容提供商处理器 1716、核心服务处理器 1718 以及存储器 1714,如下文进一步详细地描述的。

[0116] web 服务器 1704 在一个或多个用户设备 1702 和 1703 与一个或多个内容提供商网站 1706-1708 之间交换信息,所述一个或多个内容提供商网站 1706-1708 还可以被称为内容提供商、社交网网站、新闻源或新闻订阅源。用户设备 1702 和 1703 可是在本文中所述的讨论的用户设备中的任何一个。响应于来自用户设备的轮询,中间 web 服务器 1704 从内容提供商网站 1706-1708 拉信息并且使信息对用户设备 1702-1703 可用。用户设备 1702 和 1703 还能够经由中间 web 服务器 1704 将内容推到内容提供商网站。此外,用户设备 1702、1703 能够通过互联网 1705 经由直接连接 1720 直接访问内容提供商网站 (CPW) 1706-1708,所述互联网 1705 还可以被称为万维网或简单地称为 web。

[0117] 用户设备 1702 和 1703、中间 web 服务器 1704、以及内容提供商网站 1706 经由互联网 1705 连接,并且由第一互联网网络 1705' 和第二互联网网络 1705" 来图示。根据对于每个设备 1702 和 1703 的设置、根据选择的与每个特定内容提供商相关联的应用程序接

口 (API), 中间 web 服务器 1704 经由互联网网络 1705” 访问内容提供商网站使得可访问的内容, 并且以容易处理的格式使内容对用户设备可用。中间 web 服务器 1704 还接收通过互联网 1705 或经由诸如蜂窝网络的另一通信网络传送的、源自用户设备 1702 和 1703 的内容, 并且以用于相应内容提供商网站的适当格式将该信息提供给适当的内容提供商 1706-1708。

[0118] 图 18 图示了又一示例性通信系统 1800, 其中, 用户设备 1802 经由互联网网络 1805 连接到中间 web 服务器 1804。图 18 中被图示为移动设备的用户设备 1802 还可以经由 P2P 载波连接 1822 连接到中间 web 服务器 1804。中间 web 服务器 1804 经由互联网 1805 连接到内容提供商网站 1806-1808, 并且经由 P2P 连接 1821 连接到运营商网络 1820。运营商网络 1820 可以包括可以具有到后端插件的连接的地址簿。

[0119] 中间 web 服务器 1804 包括内容提供商处理器 1816 以从内容提供商网站 1806-1808 拉信息, 处理信息以标识新内容, 并且如果标识了新信息, 则生成发送到核心服务处理器 1818 的通知。该信息可以被本地地存储在 web 服务器 1804 内的存储器 1814 中。核心服务处理器 1818 向用户设备 1802、1803 通知新信息是可用的并且存储用户设备信息以用于与用户设备同步。核心服务 1818 对来自用户设备 1802 和 1803 的轮询做出响应以将内容提供给相应用户设备。

[0120] 由用户设备 1802 或 1803 传送的信息还可以通过 web 服务器 1804 发布到内容提供商网站 1806-1808。从用户设备 1802 或 1803 提供的内容由核心服务处理器 1818 来接收, 由内容提供商处理器 1816 来格式化以与内容提供商网站 1806-1808 中的一个或全部兼容, 并且发送到适当的内容提供商 1806-1808。

[0121] 内容提供商网站 1806-1808 的示例包括社交网络网站 (SNW), 诸如 Facebook™、Twitter™ 以及 Myspace™。其它内容提供商网站可以是照片网站, 诸如 Photobucket™、或者新闻源, 包括提供简易信息聚合 (RSS) 订阅源的任何源或新闻内容的任何其它源。上述示例不被认为是详尽的, 而是能够为用户设备提供内容的源的类型的示例。源可以包括用于移动设备的特殊内容, 或者用于个人计算机的内容。

[0122] 还可以被称为后端、后端部分、或社交网络处理器 (SNP) 的中间 web 服务器的内容提供商处理器 1816 包括用于每个内容提供商的相应插件 1824。每个插件 1824 都可以具有相应的处理器和 / 或存储定义的程序或 API, 以用于从其相应的内容提供商拉内容并且用适当的格式将信息从设备上传到相应的内容提供商。

[0123] 核心服务处理器 1818 还被称为前端。核心服务处理器 1818 包括 web 支持门户应用服务器 1826、核心 web 服务应用服务器 1828、以及推式服务器 1830。核心服务处理器 1818 和内容提供商处理器 1816 连接到存储器 1814, 所述存储器 1814 充当临时存储器或缓存, 其能够例如使用分割成数据库的大存储器系统来实现。

[0124] 如图 18 中所图示, 移动设备 1802 能够经由诸如有线以太网或无线 802. x 连接的局域网或经由蜂窝网络连接到基站 1832。运营商能够通过基站 1832 与负载均衡器和防火墙 SSL 1834 之间的 P2P 连接 1822 连接到中间 web 服务器。运营商基站 1832 还能够通过互联网 1805 连接到中间 web 服务器 1804。P2P 连接 1821 还可能存在于插件 1824 与运营商网络 1820 之间。

[0125] 推式服务器 1830 可以例如通过 TCP 连接连接到用户设备 1802。核心 web 服务服

务器 1828 和 web 支持门户 1826 可以经由 HTTP 连接来连接。插件 1824 还可以经由 HTTP 连接与内容提供商网站 1806-1808 进行通信。

[0126] 如上文所讨论的,相应的插件 1824 支持诸如 Facebook™、Twitter™、MySpace™ 的社交网网站和诸如 RSS、ATOM、Podcasts 等等的的内容订阅源。此外,插件 1824 可以被提供用于发消息门户,诸如 Yahoo™ 邮件、Google™ 邮件、以及 Microsoft™ 邮件。然而,电子邮件服务仍然可以由移动设备 1802 和 1803 直接访问,并且来自邮件服务的联系人列表经由前端服务进行备份。数据聚合(从内容提供商到设备的信息)和通知 API(从设备到内容提供商)可以支持各种活动,诸如状态、通知(新的邮件、订阅源改变)、朋友订阅源、以及朋友/联系人。web 服务器 1804 还支持推式信道(以向设备提供通知)、空中软件供应、对于设备的中间 web 服务器的设定和配置(例如,用户的账号、偏好等)以及 web 服务。中间 web 服务器 1804 特征是用设备安全性、基于微件的 web 访问、设备备份和恢复、以及运营商支持工具。

[0127] 中间 web 服务器 1804 可以向设备 1802 或 1803 提供账户管理、推以及数据服务。账户管理服务可以提供设定和设备设定、支持服务标识、用于用户的服务器设置以及其它账户/用户配置。推式服务可以提供用于向设备通知其具有新内容或可用数据的服务,因此支持向用户及时呈现诸如状态、新闻以及朋友更新的动态数据,而无需用户设备必须直接轮询内容提供商。数据服务可以包括对于设备上传、检索以及用于各种类型的数据的同步服务。例如,订阅源将从内容提供商网站检索到的信息提供给设备,上传将来自设备的内容提供给内容提供商网站,并且同步使得能够实现设备 1802 或 1803 与服务器上缓存的同步。

[0128] 连接到中间 web 服务器 1804 的设备 1802 或 1803 首次可以连接到设备设定服务以进行初始设备设定,诸如设置键入电子邮件账户以与聚合服务相关联、设定内容提供商网站以由设备访问、键入密码和用户标识,以及建立诸如语言的偏好。一旦完成了设备设定,则为客户端设备和推式服务建立了持续的 TCP/IP 连接。当中间 web 服务器 1804 检测到已经发生影响用于特定设备的数据的改变时,信号经由推式服务发送到设备。在这个时候由设备决定判断是否直接连接到相关的数据服务并且检索任何新的或修改的数据。例如,设备可以轮询服务器以获得该信息,并且经由用户界面(例如,显示器)将该信息呈现给用户。该用户能够查看该信息,并且决定是否直接访问该源以获得另外的信息。

[0129] 由插件 1824 从内容提供商网站 1806-1808 拉的信息在中间 web 服务器 1804 中处理并且与先前的信息进行比较以标识改变。

[0130] 可从内容提供商 1806-1808 得到的信息可以具有众多类型。聚合服务 1804 所支持的那些内容类型被监视以获得改变。当检测到事情时,条目被添加到改变列表以提供给用户设备。改变列表包含从初始化或从最后的改变列表锚开始已经发生的所有事情。事情包括发布,包括但不限于状态或评论,新闻订阅源、社交网络联系人的更新。

[0131] 更具体地,对于每个内容提供商 1806-1808,定义集合被提供用于聚合服务支持的内容类型的子集。与内容提供商相关联的插件 1824 将从内容提供商网站 1806-1808 拉所支持的内容。然后可以再检查内容以获得改变。当改变发生时,该改变作为改变列表的一部分被传送到前端 1818。

[0132] 在操作中,服务器 1804 和设备 1802 每个都存储改变锚。服务器 1804 将连续不断

地从内容提供商网站 1806-1808 拉内容以保持与内容提供商网站 1806-1808 同步。每当后端部分 1816 拉内容,其检查改变。如果检测到改变,则其被添加到可以被存储在例如存储器 1814 中的改变列表。设备 1802 同步到服务器 1804 以保持最新的,并且服务器 1804 和设备 1802 使用锚来确定要交换多少信息。

[0133] 以下术语与 web 服务器 1804 的后端部分改变同步相关联:

[0134] ■ 服务器同步 - 服务器从数据的原版同步 (例如,后端从诸如 Facebook™、Twitter™ 等的内容提供方网站的同步)

[0135] ■ 客户端同步 - 移动设备 102 中的客户端程序与前端同步 (核心服务服务器)

[0136] ■ CLA- 改变列表锚 - 采样 (时间或改变) 在该处如在服务器中记录的发生最后的完全或部分同步 (在聚合服务服务器程序处记录的)

[0137] ■ 改变列表版本 - 改变列表可以被给予版本号以帮助同步跟踪

[0138] ■ W1- 存储的改变列表窗口的较低的采样 (开始)

[0139] ■ W2- 存储的改变列表窗口的较高的采样 (结束)

[0140] ■ MW- 最大窗口大小 (能够被指定为时间跨度,改变数量,或时间跨度和改变数量的组合)

[0141] ■ OFF- 指示同步是否被暂停的标志

[0142] ■ CA- 客户端锚 - 采样 (时间或改变) 在该处如在设备中记录的发生最后的完全或部分同步 (在聚合服务客户端程序处记录的)。

[0143] 当在服务器同步期间检测到改变时能够使用以下同步传输队列算法,并且确保新信息被提供给设备使得设备被更新。web 服务器 1804 程序针对每个账户维持三个数值: CLA、W1 以及 W2。服务器 1804 程序存储落入在窗口 [W1,W2] 内部的改变列表的部分。在服务器同步期间发现的所有改变用等于当前 W2 (即,在 W2 递增 1 之前) 的同步锚来加戳。一旦窗口大小到达或者超过最大窗口大小 mw,则程序就暂停服务器同步 (内容提供方 web 大小同步)。一旦暂停,则当接收到新的客户端轮询时,服务器 1804 将重新开始服务器同步。CLA、W1 以及 W2 的值根据以下状态转换规则来更新:

事情	状态转换
初始化	CLA = 0, W1 = 0, W2 = 0, OFF = 0
[0144] 服务器同步	如果 $W2 - W1 \geq MW$, 则 $W2 = W2 + 1$, off = 1
客户端同步, $W1 \leq CA \leq W2$	$W1 = CA$, OFF = 0
客户端同步, CLA < W1 并且 (CA < W1 或 CA > W2)	CLA = W2 + 1, W1 = CLA, W2 = CLA, OFF = 0 (也可以叫“改变列表复位”)

[0145] 当客户端 1802 或 1803 轮询以获得改变时,如果客户端锚 CA 落入 [W1, W2] 内部,则部分同步将工作并且服务器往回发送落入 [Ca, W2] 内部的改变 (并且删除老于 CA 的改变)。否则,服务器和客户端开始完全同步。当且仅当客户端锚 CA 小于 W1 或大于 W2 (即,CA 在窗口的外部) 时客户端才需要完全同步。在这种情况下,客户端锚 CA 是“无效的”。当且仅当改变列表锚 CAL 小于 W1 (即,服务器不具有当前改变列表的完整历史) 时服务器 1804 复位其改变列表。服务器 1804 将窗口 [0, W2] 发送给客户端,其中“0”告诉客户端其

应当进行完全同步。

[0146] 尽管账户要与单个设备相关联,但是在聚合服务中用户可能想要不止一个设备同步到同一用户账户。在缺少改变列表复位的情况下,多个设备和单个设备一样工作。在改变列表复位的情况下,可能在多个设备之间发生将引起改变列表在服务器上不断地复位的竞态条件。例如,设备 1 轮询以获得改变,发送无效的 CA。服务器 1804 不具有当前改变列表的完整历史,因此其复位其改变列表。设备 1 再次轮询以获得改变,使 W1 往上移动。然后设备 2 轮询以获得改变,并且由于由设备 1 的轮询引起的改变列表复位,所以其 CA 肯定是无效的。然后服务器 1804 被迫再次复位其改变列表。设备 2 然后可以再次轮询以获得改变,使 W1 往上移动。如果设备 1 然后轮询以获得改变,并且由于改变列表复位所以其 CA 肯定是无效的并且服务器 1804 被迫再次复位其改变列表。

[0147] 为了避免这个竞态条件,服务器 1804 能够为每个设备保持一个确认的客户端锚 (“ACA”),并且当最小的 ACA 大于当前的 W1 时仅将 W1 往上移动。替代地,服务器可以为 W1 创建缓冲区,即服务器将不一直将 W1 往上移动到 CA,而是移动到 CA 或 $W2 - MW/2$ 中的最小值。

[0148] 在与内容提供商 (例如, Facebook™) 进行同步的背景下,由于内容提供商处理器 1816 (即, SNP 处理器或后端部分) 不具有数据的原版的事实的原因 (即,来自 Facebook™ 或其它内容提供商网站 1808-1808 中的一个的数据未被全部拷贝),并且服务器 1804 没有本地地存储改变的完整历史,所以服务器 1804 将需要时不时地 (例如,当设备过期时) 复位 (即,从头开始重建) 改变列表。因此,在不同的时间构建的改变列表的背景下,相同的时间戳 / 同步锚能够意指不同的情况。

[0149] 考虑图 22 中所图示的事情 2200 的以下序列。在图 22 中, aX 意指“添加项目 X”, mX 意指“修改项目 X”,以及 dX 意指“删除项目 X”。

[0150] 如图 22 中所见,服务器以仅具有在存储器 1814 中本地地存储的 (m2, a2) 的改变列表 CL1 (2210) 开始,并且客户端具有 (a1, a2)。客户端发送同步锚 t (2212),其为过期的因此服务器擦除 CL1,并且重建改变列表,具有 (a2, a3) 的 CL2 (2220) 本地地存储在存储器 114 中。客户端然后带着其同步锚 t (2222) 回来并且服务器往回发送 (a3)。客户端现在具有 (a1, a2, a3),并且失去 (d1)。应当已经发生的是服务器往回发送了 (a2, a3) 并且通知设备进行完全同步。如果这个发送失败,则当客户端再带着 t 回来时,服务器可能强制完全同步或仅往回发送 (a3)。

[0151] 为了避免这个问题,版本号能够与改变列表一起保存,而不是仅保存个别改变。也就是说,每当改变列表在服务器上复位时,该改变列表的版本就递增 1。然后当从客户端接收到老的版本号时,服务器将明确地知道客户端需要完全同步。在另一实施例中,web 服务器 1804 可以为每个设备存储客户端锚,并且当所存储的客户端锚与从设备接收到的同步锚 (例如,在上文示例中的 t) 失去同步时强制完全同步。对于任何给定的改变列表 CL 和任何给定的改变 C,当且仅当 C 来自其版本低于 CL 的版本的改变列表时才存在锚 (C) < 版本 (CL),然后服务器不必在服务器与客户端之间传送额外的版本号——服务器仅需要将当前改变列表的版本存储在服务器上,并且将传入的客户端同步锚与这个版本号进行比较以确定客户端是否是版本过期的 (并且因此需要完全同步)。服务器能够通过从相同序列号拾取同步锚和改变列表来实现这个。

[0152] 这种方法的一个结果是服务器不再能够将外部的时间戳（例如，伴随 Facebook™ 消息发生的那些）用作同步锚，而是指派其自己的同步锚。

[0153] 中间 web 服务器 1804 可以以无论哪一种的语言将其通过后端从内容提供商网站 1806-1808 所拉的内容提供给移动设备 1802-1803。例如，如果客户的 Facebook™ 偏好是法语，则 Facebook™ 将发送法语的内容，并且该内容将以法语通过中间 web 服务器 1804 的聚合服务。这与用户设备和社交网络提供商的直接关系一致，其中，账户持有者不能够在他们的设备上改变语言偏好，而是在社交网络提供商网站上选择语言。

[0154] 当由后端 1816 在由 web 服务器 1804 支持的“内容类型”中检测到改变时（即，聚合服务），该后端 1816 将通知前端部分 1818 将通知发送到用户设备 1802 或 1803。

[0155] web 服务器 1804 还可以插入 (sink) 新闻订阅源。新闻源可以通过移动设备服务提供商（例如，蜂窝运营商）或用户设备来选择。中间 web 服务器 1804 能够为其中利用了用户设备的市场和语言中的每一个提供本地内容。在一个实施例中，中间 web 服务器 1804 不可以翻译内容的语言，但是仅可以通过后端部分 1816 从新闻服务内容提供商所拉的内容。

[0156] web 服务器 1804 还可以插入管理订阅源。管理订阅源消息能够通过系统管理者来创建，他们被局部化并且他们以用于设备的正确语言被发送到用户设备。

[0157] web 服务器 1804 还可以插入联系人，诸如朋友列表。如果联系人不经常被更新，则聚合服务的前端 1818 可以偶尔使他们同步，例如，不是一天一次以上或两次。支持异步改变通知的提供商对于这个处理是个例外并且这样的改变被立即同步。

[0158] 当中间 web 服务器 1804 后端 1816 检测到在社交网络网站中已经更新联系人时，其通过推式信道将低优先级信号发送到在设备 1802 或 1803 中存储的聚合服务客户端程序。一旦接收到这个信号，则聚合服务客户端将通过在服务器 1804 的前端中调用聚合服务的同步 web 服务来发起双向同步操作。

[0159] 聚合服务客户端还可以允许用户手动地发起同步。例如，设备 1802 可以包括使用户能够请求与 web 服务前端同步的菜单。在聚合服务客户端程序上的联系人的改变可以以“缓慢的”方式与 web 服务器同步。在将改变发送到服务器前端 1818 之前，客户端程序将在可配置的时间段中对改变进行批处理。

[0160] 如上文所讨论的，朋友订阅源是关于社交网网站提供商的网络中的联系人 / 朋友的信息。当聚合服务检测到新的订阅源元素时，其通过推式信道将低优先级信号发送到用户设备中的聚合服务客户端程序。一旦接收到这个信号，则客户端将使用适当的 web 服务轮询交换从聚合服务中取得订阅源。订阅源可以包括来自若干内容提供商 1806-1808 的元素。

[0161] 状态通常由用户针对多数内容提供商手动地设置，但是聚合服务客户端程序也能够支持即时发消息 (IM) 风格呈现。状态被监视用于订户（自身状态）并且用于所有的关联朋友（朋友状态）。自身状态还可以使用适当的 web 服务通过聚合服务客户端程序在提供商中设置。当聚合服务 web 服务器后端检测到状态的变化（自身的或朋友的）时，其通过推式信道将高优先级信号发送到聚合服务客户端程序。这个推式信号能够在其净荷中有利地包含状态值。

[0162] 支持状态的社交网提供商将在提供商设置中具有入口。每个提供商都将包括定义

如下的状态设置：提供商到提供商 ID 的映射；状态文本的最大长度；如果支持情绪，则情绪列表和 / 或可用反应的列表。在一个实施例中，用户或提供商可以键入无效的或空字符串以清除状态。

[0163] 如果用户正在设置情绪，则用户可以提供该情绪的标识、对该情绪的描述，选择描绘该情绪的预定义图片或为用户选择的图片提供 url。

[0164] web 服务器 1804 后端 1816 插件 1824 能够直接从社交网网站获得状态和情绪更新，并且向核心 web 服务 1828 通知任何改变。状态和情绪更新可以包含以下信息：类型（状态、情绪、STATUS_AND_MOOD）、动作（清除状态或更新状态）、内容提供商（例如，内容提供商 1806-1808 中的一个）、聚合服务账户 id、外部 id、如果更新是针对朋友则朋友 id、状态文本和 / 或发布日期。如果更新是情绪或 STATUS_AND_MOOD 更新，则更新可以包括如上文所讨论的 id、描述、图片名称和 / 或图片 url。id、描述、图片名称以及图片 url 被包括以支持定制情绪。Id 常常是数字，并且描述、图片名称以及图片 url 是文本。

[0165] 一些社交网提供商允许用户按照状态行事。Facebook™ 例如允许用户对状态进行评论。Twitter™ 允许用户指示喜爱的状态以及答复状态。状态反应提供这个能力并且能够与朋友动作非常类似而且与朋友订阅源和朋友订阅源反应非常类似。

[0166] 例如，使用聚合服务器 1804 的一个好处是，通过进行改变列表的定期性的主动复位来有利地消除将改变或更新中的任何一个永久地存储在服务器上的需要。结果，客户端设备将必须执行与聚合服务器 1804 的更多的完全同步。

[0167] 因为当窗口大小达到 MW 时服务器同步被暂停，所以在缺少客户端轮询的情况下很少丢失的推式可以造成设备的服务停歇期。能够采用某些措施来使这种情况不太可能发生。例如，代替仅当自上一个 W2 以来存在未决的改变时才发送推式，只要自 W1 以来（或在多个设备情况下自每个设备的 ACA 以来）存在未决的改变就能够发送推式。这个方式确实导致正被发送的冗余推式和因此客户端的冗余轮询，但是如果 W1（或在多个设备情况下每个设备的 ACA）被包括在推式消息中则能够最小化后者的不利。

[0168] 核心 web 服务应用服务器 1828 包括核心服务应用，该核心服务应用的主要功能将从后端 1816（社交网处理器（SNP））接收数据并且对其进行封装以传递到客户端设备。这些应用包括朋友订阅源、社交网朋友、发消息 / 邮件以及新闻。它们可以具有共有的若干属性：数据流可以从服务器到客户端的单方向的；没有客户端改变可以被往回交付给核心 web 服务应用服务器 1828 并且数据不被永久地维持在核心 web 服务应用服务器 1828 中。

[0169] 前端 1818 能够提供同步传输队列、用于通过同步信道将数据向下发送到客户端设备 1802 或 1803 的过渡机构。核心 web 服务应用服务器 1828 中的应用能够向传输队列注册它们的同步应用 ID，所述传输队列进而向由作为应用的代理的前端 1818 控制的同步引擎注册。传输队列支持一个或多个条目到队列中的入队。每个条目都能够指定对队列不透明的多个字段，包括一点数据；提供商的名称；项目是否构成添加、修改或删除以及用于条目的唯一 ID。同步传输队列不关心指派给这些字段中的任何一个的值；其仅仅以其被排队的顺序对数据进行排序。应用在入队期间不获取任何锁。队列允许应用连续不断地将数据添加到队列中。

[0170] 前端 1818 还能够通过用于每个应用的同步锚来提供对客户端状态的跟踪。这个同步锚不可以暴露给应用。前端 1818 可以进一步当其已经验证了客户端已经接收到它时

提供从队列中移除数据。当客户端进来为了其下一次更新请求时,能够容易地确定这个验证;前端 1818 能够假定客户端设备已经接收和处理了具有在先的同步锚的任何条目。

[0171] 在期满时段之后前端 1818 能够清除应用数据。每个应用都能够指定任何数量的限额大小和持续时间对(例如,在一天以后允许不超过 1000 个条目,在一周之后允许不超过 100 个条目,以及不允许老于一个月的条目)。核心 web 服务应用服务器 1828 可以具有例行地检查队列并且弄清用于任何客户端设备的队列是否不符合的保管服务。保管服务能够通过删除其已经发送给客户端的记录来满足限额。否则,保管服务将为过度限额设备清除队列中的所有的该应用的数据。

[0172] 当队列不能够履行客户端的更新请求时,前端 1818 能够调用应用提供的回叫。如果设备已经被擦(例如,擦除)或者如果用户已经脱机足够长使保管服务已经清除了队列,则这种情况可能发生。在这种情况下,应用被通知队列需要复位,并且期望应用从头开始对数据进行重新排队。客户端被返回指示它的锚不再有效的错误代码,并且期望的是客户端将往回复位到锚 0。队列进入复位状态直到应用指示其已经重新住入队列为止。直到该时间为止,队列针对任何客户端更新请求而返回零个条目。

[0173] 前端 1818 还可以允许应用强制设备的队列的复位。如果应用检测到后端部分已经经历灾难性的故障并且不再能够向应用提供差别的改变,则这可能是必要的。

[0174] 在核心 web 服务应用服务器 1828 中的应用能够向队列提供回叫,所述回叫能够将一系列队列条目转化成用于客户端的有效的改变列表。由于队列是数据不可知论的,所以在其能够被返还到同步引擎以用于传输到设备之前,应用可以对该队列数据进行解释。队列可以是数据库支持的。每逢客户端做出更新请求或队列被复位时,获取锁,但是这并不阻止应用继续将数据扔进队列中。

[0175] 聚合服务 web 服务器 1804 暴露两个同步有关的 web 服务调用。

[0176] 第一个调用是称作更新的 POST 方法。这个调用从服务器中检索客户端程序能够使用的改变列表以合并最新的服务器改变中。客户端上的状态信息经由用于每个同步应用的同步锚来维持,其是被初始设置为零的长值。当客户端请求给定的同步应用 ID 的服务器改变时,客户端作为参数提供其当前的锚。服务器返回格式化的改变列表以及最近的服务器锚。在客户端成功地合并服务器改变之后,其必须将其锚更新成服务器的值。客户端能够指定页面大小参数以限制改变列表的大小;在这种情况下,“is_more”标志被设置成指示客户端需要取得更多的更新。更新可以是允许客户端同时地为多个同步应用 id 取得服务器更新的成批调用,这能够有助于减少往返的数量。

[0177] 第二调用是称作交付的 POST 方法,其将客户端的改变列表发送到服务器 1804。客户端 1802 在请求中再次规定其最后的锚。注意,如果客户端 1802 没有被更新成最新的服务器版本,则服务器 1804 将(通过特定的错误响应)拒绝交付请求。在这种情况下,客户端 1802 应当在重试之前调用更新并且合并服务器的最新改变。因此,客户端通常应当在交付其改变之前调用更新。

[0178] 客户端程序检测并且解决任何冲突。交付必须是无冲突的,并且因此冲突可能仅当客户端取得服务器更新并且尝试将其合并到其本地数据存储器中时才发生,其可以包含竞争客户端改变。一般而言,存在客户端必须处理的三个种类的冲突:

[0179] ●服务器 - 模型 - 客户端 - 模型。自上一次成功的同步以来,客户端和服务器两

者都已经对特定数据项目做出了改变。

[0180] ●服务器 - 模型 - 客户端 - 删除。服务器正在试图修改自上一次成功的同步以来客户端已经删除的数据项目。

[0181] ●服务器 - 删除 - 客户端 - 模型。服务器正在试图删除自上一次成功的同步以来客户端已经修改的项目。

[0182] 注意,同步 web 服务调用是关于在客户端与服务器应用之间交换的同步数据的格式是完全地不可知论的。服务器与客户端改变列表是端点经由同步框架交换的不透明点。在这个意义上,同步 web 服务调用主要地是两个端点之间的锚定的传输机制。

[0183] 聚合服务 web 服务器 1804 还可以具有聚合服务同步协议处理机、同步有限状态机构,这在图 19 中被描述。一旦聚合服务同步任务通过同步引擎控制器从队列中取出,则控制器调用处理机的 `executeSync()` 方法,其往下进行在开始处开始的流程图。尽管 `executeSync` 调用块直到完成为止,但是对于要完成的整个任务,存在超时(通常 2 分钟,然而这是可配置的)。

[0184] 图 19 是图示示例性同步操作的流程图 1900。最简单的场景是遵循图 19 中的流程图的结果,从开始 1901 笔直往下。如果同步被强制(即,作为服务器推式通知或另一应用的显式力量的结果)或如果与处理机相关联的同步适配器具有本地改变,则同步继续进行(步骤 1902)。如果处理机不具有改变,则该处理结束。(步骤 1903)。如果存在改变,或者同步被强制,则处理机获得客户端的锚(步骤 1904),并且调用同步 WS 处理机以得到有差别的服务器改变。(步骤 1905)。如果请求是成功的(步骤 1906),则处理机现在具有一点(字节阵列)以经由 `updateClient()` 调用合并到适配器中(步骤 1907)。如果合并成功(步骤 1908),则客户端适配器必须将其锚更新成服务器的锚,并且这个值被发送到缓存(步骤 1909)。处理机可以被配置成在页面中检索更新;10 的页面大小指示客户端仅想要具有相对于客户端的接下来的 10 个锚值记录。如果使用了页面,则处理机将继续取得并且合并更新,直到被明确地告知在服务器上不存在进一步的数据。(步骤 1910)。接下来,处理机确定是否存在任何本地客户端改变(步骤 1911)。如果不存在客户端改变,则处理结束(步骤 1912)。如果存在客户端改变,则处理机检索改变(步骤 1913)并且将该改变交付给服务器。(步骤 1914)。处理机然后将确认该交付是成功的(步骤 1915)。如果该交付是成功的,则服务器将往下发送其新的锚。(步骤 1916)当确认该交付时,锚被给予适配器;适配器必须假定该交付失败直到其接收到这个确认为止。最终,新的锚被中继到缓存(步骤 1916)。

[0185] 流程图 1900 的侧枝图示了如何处理若干失败情况。服务器可以拒绝在更新请求中的客户端锚。如果锚是负的,或者如果客户端提交了大于服务器锚的锚(具有下面的一个例外的情况),则这种情况将发生。在这种情况下,客户端被告知清除其数据并且从头开始。(步骤 1917)。服务器可以推测客户端处于损坏的状态并且要求客户端请求(步骤 1917)。这与上文中的基本上是相同的情况。服务器可能请求恢复(步骤 1918),其指示客户端需要将所有其数据向上发送到服务器(完全同步),使得服务器能够从客户端数据恢复。这可能在灾难恢复的过程中或者当客户端被迁移到新的云(从一个服务器 104 移动到不具有其数据的另一服务器 104)时发生。当客户端锚为非零时,如果服务器锚为零,则服务器还能够检测到对于恢复的需要。

[0186] 该流程图还图示了冲突处理过程（步骤 1919）。可以例如在客户端程序上完成冲突处理。服务器可能告诉客户端程序，当后者试图交付设备改变时其为过期的。为了简单起见，当客户端被更新成最新的服务器数据时服务器才允许交付。因此，从不存在服务器侧冲突。

[0187] 聚合服务客户端电话（例如，图 18 的移动设备 1802）的基础特征中的一个是用来自用户以及及时且有效的方式签约的各种社交网络的新数据来对其进行更新的能力。为了做这个，已经确定，聚合服务包括将通知发送到设备的机制，使得其能够知道何时从图 18 的聚合服务核心服务应用服务器 1818 检索新数据。

[0188] 为了提供其具有的这样的服务，期望通过推式信道将通知发送到用户设备 1802 或 1803。推式信道可以一直在使用作为发消息协议的基础的 XMPP 的 TCP/IP 连接上。能够被用作发消息的基础的系统的一个示例是 Jabber XCP 服务器 (www.jabber.com)。

[0189] 图 20 图示了用于示例性推式信道的服务器侧推式架构 2000，其可以通过图 18 中的推式服务器 1830 来实现。该推式架构 2000 包括推式服务通知 API 2002、推式服务请求处理机 2004、推式服务调度器 2006、推式服务 XMPP 消息生成器 2008 以及推式服务认证插件 2010 至 Jabber XCP 2012。

[0190] 推式服务通知 API 2002 由后端服务（例如，图 18 的推式服务器 1830）用来将通知和数据发送到特定的账户设备。在一个实施例中，API 可以被定义为如下：

[0191]

```
public class SignalData {  
    public enum Action {  
        ADD, REPLACE, DELETE  
    };  
};
```

构建新的 `SignalData` 对象

* `@param key` - 针对当查找这个数据项时能够匹配的关键词

* `@param data` - 持有待发送到客户端的不透明数据的字符串

* `@param maxDelay` - 在发送这个数据之前要等待的最大秒数

* `@param expiresAfter` - 自删除这个数据的当前时间以来的秒数

* `@param Action` - 呈现这个特定数据项的动作

* `Action.ADD` - 添加数据项而不管关键词如何

* `Action.REPLACE` - 如果存在与这个关键词匹配的另一数据，则替换它。否则，添加该数据项。

* `Action.DELETE` - 找到与当前关键词匹配的数据项并且删除它。在这个关键词和数据中可以是空的。

[0192]


```
        */
        public SignalData(String key, String data, int maxDelay, int
expiresAfter, Action action);

        /*
        * Simple getters
        */
        public String getKey();
        public String getData();
        public int getMaxDelay();
        public int getExpiresAfter();
        public Action getAction();
    }
    public interface Signaling {
        /**
        * 准备排队等候将一系列数据发送到与特定账户相关联的
设备。

        * @param appId    -从其发送这个数据的应用的 id
        * @param accountId - 要向其发送这个数据的设备的账户
的 id
        * @param deviceId - 要向其发送这个数据的设备的 id
        * @param SignalData - 要发送到设备的一系列数据项。呈
现每条数据的动作被嵌入每条数据
        */

        public void sendPushData(String appId, String accountId,
String deviceId, SignalData[] data);
    }
}
```

[0193] sendPushData 方法构造 web 服务调用, 其将调用推式服务请求处理机 (PSRH) 2004。PSRH 2004 是暴露可以被表示为 /blur-services-1.0/ws/push/signaldata

的 RESTful 资源的 web 服务调用的服务器。在一个实施例中, 推式服务请求处理机 2004 在每个 POST 上执行以下操作:

[0194] ●对推式数据进行串并转换;

[0195] ●检查以弄清是否已经存在发布用于这个特定账户 id- 设备 id 的数据;

[0196] ●如果已经存在发布的数据, 则基于与每个数据项一起发送的动作来添加 / 替换或删除该数据;

[0197] ●如果不存在数据, 则为这个账户 id- 设备 id 创建新的队列并且将向其添加该数据。

[0198] ●必要时, 终止队列上的任何数据;

[0199] ●将数据往回保存到数据储存器;

[0200] ●对于被添加的每个数据项, 为该数据项生成新的序列 id 并且存储它; 以及

[0201] ●如果 API 调用在其曾经开始这个服务之前失败, 则将不记录丢失的消息。一个解决方案是 API 它本身将这个序列 id 存储在数据库中使得将捕获其这样的错失。

[0202] 数据库中的序列 id 的存储是必要的, 因为 API 被绑定到无国籍的服务器, 如果所存储的调度时间大于上述值, 则通过数据项中的每一个并且得到最小的最大延迟值以及将其添加到当前时间, 然后将调度请求发送到调度器。

[0203] 当推式服务请求处理机 2004 接收到触发调度更新的数据项时, 其将请求发送到推式服务调度器 2006 以调度待发送的消息。推式服务调度器 2006 是经由 RESTful url 暴露单个资源的服务器。例如, RESTful 主体可以是 [{" blurAcctId": "...", " providers": [" facebook", " myspace", ...] }, ...]。优选地, 每云将存在这些调度器中的仅一个, 但是可以存在更多。推式服务调度器 2006 优选地是专用服务器, 然而, 在其它实施例中推式服务调度器 2006 可以是另一服务器的一部分。推式服务调度器 2006 将所有的其调度保持在内部存储器 (未示出) 中。推式服务调度器 2006 能够使用 HTTP 1.1 持久连接以及二进制协议以确保最佳的吞吐量。在一个实施例中, 当进行新的调度请求时, 推式服务调度器 2006 可以对调度进行串并转换 (调度仅包含 account_id、device_id 以及调度时间) 并且将调度放在调度队列上。

[0204] 在后台处理线程中, 调度队列等待并且在适当的时刻对推式服务 xmpp 消息生成器 2008 进行 web 服务调用以生成 XMPP 消息。

[0205] 响应于来自推式服务调度器 2006 的请求, 推式服务 XMPP 消息生成器 2008 创建 XMPP 消息。在一个实施例中, 对于每个请求, 推式服务 XMPP 消息生成器 2008 都可以从请求中得到 account_id 和 device_id, 在数据储存器中找到与 account_id 和 device_id 相关联的所有数据, 并且对数据进行检索以及将它从数据储存器中删除。推式服务 XMPP 消息生成器 2008 可以将数据封装到 XMPP 消息元素中并且将该 XMPP 消息发送到 XMPP 服务器 2012。

[0206] 用于客户端设备 (例如, 图 18 的移动设备 1802) 的账户设定处理能够进行对 XMPP 服务器 2012 的调用以创建 XMPP 账户。替代地, 为了解脱这个额外的调用并且帮助认证, 组件能够用于 XMPP 服务器 2012, XMPP 服务器 2012 将利用聚合服务认证架构的 (例如, 对关于特定账户的信息进行验证和检索的核心服务处理器 1818)。

[0207] 在设备设定期间, 用户能够创建能够有利地与特定设备相关联的新账户 (聚合服务账户)。与这个动作相关联的数据将以账户服务决定存储它的任何方式被存储在账户数

数据库中。XMPP 服务器 2012 可以具有充当附连到 XMPP 服务器的分组过滤器的组件。当该组件接收到认证请求时,其从该请求中得到适当的认证令牌。该组件然后使用那些令牌来进行对帐户应用的调用以认证用户。该组件然后将指示用户认证成功或失败的信号发送到 XMPP 服务器 2012。

[0208] 辅助组件还可以被附连到还能够充当分组过滤器的 XMPP 服务器 2012。这个辅助组件将查找存在分组而非认证分组。由于所有的 XMPP 客户端发出了存在分组以便于接收消息,所以成功地附连到 web 服务器(例如,图 18 中的 web 服务器 1804)的任何 XMPP 客户端都将发送 XMPP 存在分组。这个辅助组件能够检测存在分组并且让推式服务请求处理机 2004 知道何时连接或不连接客户端。这使推式服务请求处理机 2004 能够告诉推式服务调度器 2006 是否需要处理对于特定客户端的调度。

[0209] 推式信道是向设备 1802 及时提供通知的尽力服务。推式信道可能不确保朝信道下发送的所有数据到达设备 1802。实际上,要求是即使当推式信道未被激活时设备也能够操作。很少有支持这个要求的使用情况。首先,有时候会存在由于零星覆盖的原因设备将不能够激活推式信道。在这样的实例中,手动地使数据同步而不用激活推式信道是最理想的。而且,由于推式信道一直在连接,所以如果电话进入漫游模式则服务可能想要关闭推式信道。因此,推式服务的主要责任是及时地提供通知并且提供检测推未达到设备的手段。

[0210] 推式服务将通知作为不透明的、应用特定的、离散数据发送到设备 1802。用从零开始并且在达到例如 2^{31} 的阈值之后往回翻转到零的序列 id 来标记每条数据。在一个实施例中,序列 id 可以被如下生成。核心服务服务器 1828 检测推是必要的并且准备待推到设备 1802 的数据。核心服务服务器 1818 然后用所准备的数据调用能够被推式服务器 1830 控制的推式 API。推式 API 然后检索 account_id/device_id。推式 API 使用 account_id/device_id 来从持久存储器中查找下一个序列 id。推式 API 然后用该序列 id 来标记数据并且将数据发送到推式请求处理机。推式 API 然后将成功消息返回给核心服务,例如核心服务服务器 1828。核心服务服务器 1828 接收该成功消息并且认为该消息待发送。

[0211] 如果推式 API 未能检索到序列 id,则推式 API 抛出异议并且从不发送推式消息。当这种情况发生时,推式 API 能够存储事情的历史并且当序列机构返回到工作次序时,其然后能够基于所存储的历史对该序列进行复位。

[0212] 如果推式 API 不能够将推式发送到推式请求处理机,则推式 API 可以抛出不同的异议。当这种情况发生时,因为序列 id 已经在持久性存储器中递增,所以下一个推式将接收下一个序列 id 并且如果下一个推式碰巧使其达到设备 1802,则设备 1802 现将发生故障。替代地,如果在推式请求处理机将请求发送到调度器之前推式请求处理机崩溃,则推式消息从不被发送,序列中的间隙可能出现。在这些情况下,设备 1802 可以请求完全同步。

[0213] 聚合服务器 1804 可以向客户端设备 1802 提供统一订阅源。当用户设备 1802 请求从中间 web 服务器 1804 发送多个订阅源时,能够采用统一订阅源来提供所有的订阅源数据。统一订阅源包含各种订阅源类型中的一个或多个订阅源。统一订阅源的目的是允许客户端用单个请求得到所有其订阅源,而不是进行单独的请求以得到朋友订阅源数据、新闻订阅源数据以及其它数据。订阅源可以是例如可以包括关于可从社交网网站得到的用户的朋友的更新的朋友订阅源,所述社交网网站诸如 Facebook™、MySpace™、Twitter™、LinkedIn™。订阅源还可以是新闻订阅源,该新闻订阅源包括例如可从可以来自各种源的

RSS 订阅源、Yahoo™ 新闻、Digg™ 等中得到的新闻信息。替代地，订阅源可以是包括来自诸如 Yahoo™ 天气的源的天气信息的天气订阅源；或可以包括服务状态信息的管理订阅源。

[0214] 如上文所讨论，能够通过聚合服务器 1804 从各种源收集订阅源，诸如从用于 Facebook™ 的相应插件 1824 或新闻订阅源。订阅源能够按用户发布到 web 服务服务器 1828。订阅源特定 web 服务将具有通用订阅源格式的订阅源发布到订阅源工厂。当订阅源工厂接收到订阅源时，其将低优先级通知发送到推式管理器，如上文所讨论的。订阅源在订阅源工厂中进行积累并且在某个时间或事情之后，推式管理器将推式发送给客户端。例如，客户端 1802 能够进行单个请求以得到所有其订阅源。

[0215] 在一个实施例中，聚合服务器 1804 能够采用两个协议来提供统一订阅源特征。第一协议定义订阅源特定 web 服务（诸如朋友订阅源 web 服务、新闻订阅源 web 服务等）将如何与订阅源工厂 web 服务进行交互。当订阅源特定 web 服务具有要发送到订阅源工厂 web 服务的订阅源数据时，其使用诸如 /ws/feedmill/0/feedGrain/accountID 的相关请求 URI 进行 HTTP POST 方法请求。在请求主体中所包括的将是已经具有通用订阅源格式的订阅源数据以及订阅源类型。服务器将用 HTTP 状态代码来作出响应。

[0216] 第二协议定义订阅源工厂将如何与客户端 1802 进行交互。当客户端 1802 从订阅源工厂请求它的订阅源时，其进行包含用于被客户端处理的每个订阅源类型的序列号的简单 web 服务调用。服务器响应将包含所有的订阅源数据。

[0217] 关于“订阅源工厂”术语，对“统一订阅源”进行了描述。发送到订阅源工厂 web 服务的订阅源数据将使用通用订阅源格式来发送。这个使得能够以一致的方式来处理来自各种源的任意数据。下文是服务器能够期望接收到的公用和非公用数据。

[0218] 公用数据能够包括提供商、账户 id、主体类型、类别、ID、链接、概要、发布时间标题、动作和 / 或地理标签。订阅源的提供商在朋友订阅源、新闻订阅源、天气订阅源以及管理订阅源中被找到。账户 ID 可以是聚合服务账户 ID。朋友订阅源还能够包括账户 ID。朋友订阅源可以包括主体和主体类型，新闻订阅源和天气订阅源包括 atomContent。管理订阅源还能够包括主体和主体类型。一个或多个类目，如由聚合服务所定义的那样。朋友订阅源和管理订阅源包括类型和子类型。订阅源能够包括一个或多个链接。朋友订阅源能够包括一个或多个 URL，streamURL 或到媒体项的链接。新闻订阅源和天气订阅源能够包括 atomLink。管理订阅源还能够包括 URL。新闻订阅源和天气订阅源能够包括 atomSummary。管理订阅源能够包括概要。订阅源可以包括发布每个项目的时间。订阅源还能够包括标题和用于该订阅源的标题类型或该订阅源中的数据。新闻订阅源和天气订阅源能够包括 atomTitle。订阅源还能够包括能够被执行的一个或多个动作和地理标识元数据。

[0219] 订阅源还可以包括非公用的数据类型，诸如 externalID，用户的提供商侧用户 ID 在朋友订阅源中被发现，contactID、聚合服务 ID 可以在朋友订阅源和 externalContactID 中被发现，联系人的内容提供商网站 ID 在朋友订阅源中被发现。

[0220] 在下文中使用以下术语来讨论客户端 1802 与聚合服务器 1804 之间的示例性交互：

[0221] ● 订阅源类型 - 订阅源的特定类型（即新闻、朋友、管理、天气...）；

[0222] ● 订阅源颗粒 - 在给定订阅源类型中的特定条目。作为示例，对于新闻订阅源，订

阅源颗粒可以是独立的新闻项目。在每种类型的订阅源内包含的实际数据取决于订阅源的类型,然而他们将全部具有相同的格式(见留置权)。每个订阅源颗粒被看作对订阅源工厂的不透明的字符串;

[0223] ● 订阅源小球 - 特定类型的订阅源颗粒集;

[0224] ● 订阅源袋 - 从服务器发送到客户端的数据,由一个或多个小球构成;

[0225] ● 订阅源槽 - 在处理散发从服务器发送的订阅源小球的客户端上的组件;

[0226] ● 订阅源工厂 - 在处理从多个源收集订阅源小球并且将它们封装在订阅源袋中以发送到客户端的服务器上的组件;以及

[0227] ● 订阅源仓 - 在存储订阅源颗粒直到由设备请求为止的组件。基本上其将 DB 表隐藏在下面。

[0228] 希望消耗给定的订阅源类型的在客户端 1802 上的应用将首先使用向订阅源槽注册。这个注册将简单地包括指示给定的应用在消耗中对哪一个订阅源感兴趣和方法,诸如 Android™ 意图,其应当被用来通知应用何时呈现新的数据。可以准许给定应用注册用于许多订阅源类型。当设备 1802 是 Android™ 设备时,当订阅源小球在客户端上被接收时,Android 意图将被发送到所有注册的应用,包含在这个意图中包括的数据的订阅源类型、从服务器发送的实际的订阅源小球(所注册的应用通常具有解析这个数据的方式)、这个订阅源小球的序列号、以及在订阅源槽与应用(在注册时间时设置的)之间共享的应用秘密,使得应用能够认证这个来自订阅源槽的意图。

[0229] 订阅源槽还将提供应用针对给定的订阅源类型取消注册它们自身的方式。订阅源槽还能够提供应用能够向其通知它们已经完成处理的给定订阅源类型的最后的序列号的机制。这个信息将在与订阅源工厂通信中使用以指示客户端具有仅剩的订阅源小球,因此订阅源工厂能够确定其需要将哪个数据发送给客户端。如果多个应用被注册用于给定的订阅源类型,则最小序列号将被保持并且发送给订阅源工厂。

[0230] 响应于由推式管理器告知数据正在服务器上等待它,订阅源槽从订阅源工厂中取得数据。潜在地,还将存在使用户能够手动地发起订阅源的取得的机制,如果它们认为其必要的话。将立即准许来自订阅源槽的仅一个取得请求,并且在当一个正在处理过程中时另一个被请求的情况下,第二个将失败(具有适当的错误代码)。

[0231] 能够存在订阅源槽将开始取得它的订阅源的一个 web 服务调用。例如,可以具有以下 HTTP GET 形式(减去需要发送的任何认证/登录数据),

[0232] /ws/feedmill/0/feedMe/accountID? news = X&admin = Y&friend = Z 其中 accountID 是用户的账户 ID 并且 X、Y 以及 Z 是客户端已经分别针对订阅源类型新闻、管理员以及朋友处理过的最后的序列号。仅仅设备为之设置(经由一些外部的设定机构)的订阅源类型将被指定在以上线中。可以向用户指示处理来自这个调用的错误情况(用户不允许 (userNotAllowed)、用户未找到 (userNotFound)、系统忙 (systemBusy) 等)。在系统是忙的情况下,服务器能够使用已经被配置的(潜在地经由管理/配置订阅源)一些间隔来退回。

[0233] 由于实际的订阅源小球的处理在应用处理中被完成,并且潜在地可能是耗时的,所以由于序列号可能不具有更新的机会,所以订阅源槽取得请求可能结果是接收相同的数据。这个发生应当不是频发的但是可能在完成推式管理器订阅源槽取得之后立即由用户发

起的订阅源槽取得引发,如果许多推被快速连续从服务器发送,指示存在对客户端可用的订阅源数据,则在部分 / 完全接收到订阅源袋之后或在准许多个应用注册用于相同的订阅源类型中之后,设备立即崩溃,流氓应用从不更新它的序列号(或更新成假值)。

[0234] 由于序列号适用于可以包含许多订阅源颗粒的整个订阅源小球,所以注册过的应用将必须告诉订阅源槽它们已经处理了整个订阅源小球,即使实际上它们仅仅关心订阅源颗粒的子集。本质上给定的订阅源小球的处理是全部或没有,不提供精细的颗粒度。没能用序列号更新订阅源槽将导致准许多个应用注册用于相同的订阅源类型。

[0235] 如上文所提到的,客户端请求采取包含由客户端处理的用于每个订阅源类型的序列号的简单 web 服务调用的形式。

[0236] 对客户端请求的服务器响应包含将在下文中描述的实际订阅源袋。

[0237] 订阅源袋头部 (FBH) 可以具有若干元素,例如,版本、内容类型(例如,数据或错误)或所包含的订阅源小球数量(仅用于数据情况)或在错误类型情况下的错误代码。每个订阅源小球都能够包括包含订阅源类型、订阅源小球的长度以及订阅源小球的序列号的头部 (FPH)。在一个实施例中,订阅源袋可以看起来如下:(FBH) (FPH) (订阅源小球数据) (FPH) (订阅源小球数据) (FPH) (订阅源小球数据)... 例如,错误情况可以简单地包含 (FBH)。

[0238] 订阅源工厂将必须处理从以下两个不同的位置传入的请求,(1) 要求它的订阅源的设备,如上文所讨论的,和 (2) 为给定用户提供特定的订阅源类型订阅源颗粒的应用处理器。应用处理器能够经由 HTTPPOST 调用向订阅源工厂提供订阅源颗粒,其在一个实施例中可以看起来如下: /ws/feedmill/0/feedGrain/accountID,其中实际的订阅源颗粒(具有公用格式)和订阅源类型将被包括在请求主体中。订阅源工厂用适当的 HTTP 状态代码来做出响应。在每个成功的订阅源颗粒添加之后,低优先级通知被发送到推式管理器,指示订阅源数据对客户端是可用的。每个用户将具有订阅源仓,其将包含等待传递到设备的所有订阅源颗粒的订阅源颗粒。该表能够有利地包括订阅源类型的列、序列号(无符号整数(单调增加值))以及订阅源准许数据(字符串数组(或许最大大小))。

[0239] 该表中的每行都可以对应于经由以上 HTTP POST 调用添加的订阅源颗粒。为了限制用户的订阅源表中的条目的数量,每订阅源类型能够存在可配置的最大数量的行。当这个限制被命中时所发生的将落入到 2 个类别中:

[0240] 1. 允许老化 - 在这种情况下,当处理器尝试为已经处于限制的订阅源类型添加订阅源颗粒时,最老的订阅源颗粒被移除。

[0241] 2. 不允许老化 - 在这种情况下,添加实际上失败并且错误被往回发送(经由 HTTP 状态代码)到应用处理器。

[0242] 在存储了订阅源颗粒之后,订阅源工厂用来响应于订阅源槽对其订阅源(信息)的请求(轮询)而提供信息。订阅源工厂得到对于特定用户的订阅源的请求,查询用于所有订阅源的用户订阅源仓,考虑该请求指定的序列号来构建订阅源袋并且将其发送到设备以及(基于请求序列号)从用户的订阅源仓中移除老的订阅源颗粒。

[0243] 由于每个订阅源类型订阅源颗粒都是用户的订阅源表中的行,所以存在需要实际上添加订阅源颗粒的多个数据库调用。在一个实施例中,例如,订阅源工厂可以得到当前由这个订阅源类型使用的行的数量。如果行的数量大于行的最大数量并且如果允许老化,则

移除最老的条目。如果不允许老化,则订阅源工厂可以返回错误。订阅源工厂然后可以将订阅源颗粒添加到表。

[0244] 在订阅源表上可以不要求锁定。这是因为每个行都是独立的并且在给定的时间点得到所有的订阅源的处理不应当与新的行的添加发生冲突,因为这两个操作不影响相同的集合。

[0245] 批处理机制存在可以被用来将相同的订阅源颗粒添加到多个用户(即,告诉所有 Twitter™ 用户站点当机了)。为了适应批处理机制,能够采用将在一个实施例中可以采取“/ws/feedmill/0/feedGrainForMany”的形式的另一 HTTP POST 服务调用,其中实际的订阅源颗粒以及用户列表将被包括在请求主体中。用户列表的精确格式将取决于主用户表与订阅源工厂之间的交互。

[0246] 聚合服务客户端应用,例如在客户端设备 1802 上装载的应用,能够使用具有不同的安全性级别的单独的内容提供商来访问联系人数据。可以记住一些重要的目标来构建聚合服务的联系人设计:

[0247] ● 设备上的联系人存储应当符合客户端设备 1802 的标准(例如,当使用 Android™ 设备时的 Android™ 标准)。

[0248] ● 联系人数据应当与网络服务联系人数据保持同步。

[0249] ● 每个联系人应当被作为单个个体呈现给用户,因为那是用户怎样想到联系人。

[0250] 将直接从电子邮件服务器或社交网络网站 1806-1808 中获得许多联系人。这些网络中的一些具有禁止通过第三方应用访问它们的联系人数据的服务条款。为此,来自社交网络的联系人(在下文中被称为朋友)将被保留在客户端 1802 中的单独的数据库中。与呈现合并的个体的目标相组合的这个限制提出了技术挑战,因为在 UI 中正被合并的联系人记录必须从单独的数据库收集。在下文中这个文档使用这些术语:

[0251] ● 联系人 - 指的是在标准 Android 数据库中存储的人。

[0252] ● 朋友 - 指的是在私有社交网数据库中存储的人。

[0253] ● 所有联系人 - 指的是联系人和朋友的联合。

[0254] ● 合并的联系人 - 指的是联系人和朋友的交集。

[0255] 在设备上,联系人被存储在由所有应用共享的数据库中。聚合服务扩展标准联系人以添加另外的数据和能力。聚合服务客户端联系人将是标准联系人的超集;第三方应用将能够使用标准 API 来访问联系人。

[0256] 除了扩展的联系人数据库以外,聚合服务联系人针对社交网朋友引入了并行数据库。这个数据库具有另外的安全性以阻止通过第三方应用的访问。在大多数方式中,尽管存在很少的差别,但是数据库的架构和性能对联系人而言是一样的;朋友中不支持组。如果用户将朋友添加到组,则聚合服务自动地用该朋友的数据的副本来创建联系人,并且将该联系人添加到组。而且,为了支持跨越联系人和朋友合并的身份,朋友数据库包含对联系人身份的引用。

[0257] 用户可以配置设备 1802 以使用来自各种提供商 1806-1808 的信息。这个信息可以包括电子邮件、联系人、照片以及社交网事情。公用联系人提供商的示例为 Google™、Yahoo™ 以及 Facebook™。例如,用户还可以例如从激活同步导入数据以得到电子邮件联系人,或者经由订户身份模块(SIM)卡或电缆连接从他们的先前的电话中导入数据。通过访

问设定,当最初对设备进行配置时或在其后的任何时间可以添加提供商。此外,运营商可以提供商可以将联系人从运营商 1820 直接提供给插件 1824,从而能够将联系人经由服务器前端部分下载到设备。

[0258] 图 21 图示了根据一个实施例的用于导入联系人的示例性方法。从内容提供商 2110 导入的每个联系人都作为单独的联系人记录来存储。指代同一个体的联系人在用户界面 (“UI”) 中被示出为一个人,但是记录在数据库中仍然保持分开。

[0259] 聚合服务将支持与核心服务处理器 2130 进行通信的多个同步适配器 2220-2128。这包括用于 gmail™ 联系人的 Google 的默认同步适配器、地址簿适配器、以及针对聚合服务编写的适配器,包括交换激活同步 (EAS)、社交网联系人以及同步到聚合服务的联系人。这些同步适配器中的每一个都必须将他们同步到设备上的联系人标识为它们自己的,使得联系人的改变仅被往回同步到原始源。对于针对聚合服务编写的适配器,联系人数据库中的新的列被用来标识该同步源。没有用于这个列的值的联系人被假定属于 Google™ 或另一第三方。

[0260] 联系人源中的一个将被标识为向其添加在设备上创建的联系人的默认地址簿。替代地,默认的联系人源可以是聚合服务的联系人。用户可以能够选择哪一个地址簿是默认的,并且哪一个地址簿应当用于特定的联系人。

[0261] 虽然来自不同的提供商的联系人被存储为单独的联系人记录,但是不管它们实际上是否指的是同一个体,UI 一直尝试将联系人作为个体呈现给用户。

[0262] 新的“身份”表可以作为客户端联系人在同一数据库中创建。在该表中存在用于每个个体的一个行。新的列能够被添加到“人们”表,指示联系人记录属于哪一个个体。身份表可以不与聚合服务器 1804 同步。当添加了联系人时,其被动态地创建在客户端设备 1802 上。例外是用户可以手动地合并或取消合并联系人记录。对身份的这些手动改变应当被保留,并且因此与聚合服务器 1804 同步。

[0263] 当联系人被添加在客户端设备 1802 上时,他们与其它联系人记录匹配以确定他们是否指的是同一个体。如果不存在匹配,则新的行被添加到身份表。在人们表中的身份列被更新以指向身份行。

[0264] 联系人权限内的新内容 URL 可能可用来访问身份而非人们。许多聚合服务代码将使用这些 URL 来访问联系人。

[0265] 身份表包括用于存储来自人们的最近的社交网状态更新的列。这个数据能够被显示在出现在整个产品中的“身份徽章”微件上,包括各种联系人屏、主屏回路卡以及消息寻址。

[0266] 联系人的所有用户界面视图或编辑实际上将在聚合服务认为是同一人的联系人记录集合上操作。该联系人记录集合取自身份表。

[0267] 任何合并策略都包含生成两种类型的错误的危险:过度合并(即,将单独的个体呈现为一个人)和欠合并(即,将一个个体呈现为单独的人)。设想聚合服务服务器将在过度合并方面犯错,为此:

[0268] ● 解决一个类别的错误比解决两者更容易。

[0269] ● 单个个体将被存储在多个提供商中比两个个体将合理地看起来像同一人更有可能。

- [0270] ●如果聚合服务过度合并,则用户容易标识实际上属于不同的人的数据。
- [0271] ●如果聚合服务欠合并,则其将对用户来说看起来像是“缺少”的数据。这对用户来说将是令人苦恼的,并且将难以找到应当已经被合并的其它联系人记录。
- [0272] 用户能够决定何时聚合服务具有过度合并或欠合并的联系人。这个从联系人细节活动的“源”部分来完成。
- [0273] 当以下中的一个为真时,通过聚合服务来合并联系人:记录正好具有相同的名称;记录共享电话号码并且记录中的一个将该号码标识为移动电话或工作电话;记录共享电子邮件地址,除非记录中的两者都将其标识为主电子邮件并且记录双方都已经被标识为电话的所有者。
- [0274] 如果他们从相同的源被同步,则不可以合并记录(例如,将不合并具有相同的名称的两个 Facebook™ 朋友)。注意当在客户端设备 1802 上创建联系人时,联系人的副本被创建用于每个可写的源。因此,在电话上用相同名称创建的两个联系人将可能合并在一起,因为第一个的聚合服务版本将与第二个的版本合并。
- [0275] 在联系人应用中,用户能够创建组或从服务提供商导入它们。用户将使用这些组来过滤他查看的联系人集合而且与他们(电子邮件/短信息(sms))进行通信。
- [0276] 因为组成员关系表包含对联系人的表的引用,所以社交网朋友在技术上不可以是组的成员。当用户将朋友添加到组时,聚合服务通过创建虚设的联系人记录来规避这个问题。没有将联系人的数据(电话或地址)添加到虚设的记录。
- [0277] 人能够属于不止一个组。当创建新的组并且指派成员时,用户具有当联系这个组时哪一个电子邮件或号码用于该成员的选项。例如,可以使用相应的主电话或电子邮件来联系该组。这可能是联系该组的默认方法。替代地,可以使用不同的电子邮件和/或电话或所有的电子邮件和/或电话号码来联系该组。
- [0278] 当修改组时,用户能够改变组的成员。此外,然而,当联系该组时用户还可以改变将被用于每个成员的联系方法。例如,在“查看联系人”屏中可以过滤组,其中用户能够例如经由下拉来按组过滤联系人的集合。
- [0279] 用户能够给组发电子邮件或短消息。在一个实施例中,用户从“查看联系人”屏经由菜单选项来做这个。在电子邮件应用中,用户可以将组选择为电子邮件的接受者。当挑选该组时,用户能够暂时地覆写将用于每个成员的电子邮件地址。如果用户不覆写,则将使用用于每个成员的由用户指定的默认电子邮件。然而更可能地,因为大多数用户将不为组中的每个成员指定明确的电子邮件地址,所以将使用主电子邮件。
- [0280] 当将电子邮件发送给组时(或也许当选择组以向其发送电子邮件时),UI 将指示该组中的每个成员是否都拥有电子邮件并且将能够接收它。
- [0281] 表可以被添加到联系人数据库,以使用每种通信模式跟踪拥有者已经与每个联系人进行的最近的通信。在联系人应用中,列表视图的最近选项卡(tab)提及使用任何通信模式的最近的通信。联系人细节视图的历史选项卡列举了有关所有的联系人通信方法的最近活动。

列	呼叫日志	朋友订阅源	SMS	MMS	电子邮件	社交消息
	CallLog .Calls	FriendFeed .Items	Telephony .SMS	Telephony .MMS	Email .Message	SNMessaging .SN*Columns
recent_kind	R_K_PHONE	R_K_SOCIAL	R_K_SMS	R_K_MMS	R_K_EMAIL	R_K_SNMSG
recent_type	类型	0	SMS_TYPE_PHONE or EMAIL	SMS_TYPE_PHONE or EMAIL	0	0
person	搜索	FRIEND_ID	PERSON_ID	搜索	搜索	FRIEND_ID
recent_friend	搜索	真	搜索	搜索	搜索	真
account_id	搜索	0	搜索	搜索	搜索	ACCOUNT_ID
item_id	ID	ID	ID	ID	ID	ID
icon_uri	空	提供商	空	空	空	从账户找到提供商
event_time	日期	PUB_DATE	日期	日期	DATE_SENT	DATE_SENT
对象	空	标题	对象	对象	对象	对象
主体	号码	内容	主体	空	预览	主体
标题	号码	空	地址	来自	来自	空

[0283] 当在设备 1802 (或聚合服务服务 1804) 上创建联系人时,其按照默认被添加到所有可写的提供商组。用户可以选择不包括这些组中的一个,因此不将新的联系人同步到该服务。不提供只读提供商组。这个部分被称作“提供商”。术语“只读联系人”意指属于只读提供商组的联系人记录。这样的记录可能从未被修改,包括他们的组成员关系。其它联系人被称为“可写的”。

[0284] 其它联系人组还被提供给用户,但是这些不按照默认被选择。这个部分能够被称作“组”。新的字段被添加到所有可写的联系人记录。仅属于只读联系人的字段是不可编辑的。在至少一个可写的联系人中的找到的字段是可编辑的。每当编辑字段时,该字段数据就被写入到所有可写的联系人记录,即使该记录先前不包含该字段。实质上,这慢慢地在用户的提供商之中传播联系人信息。如果属于只读联系人和可写的联系人两者的字段被修改,则该字段的旧值和新值这两者现在都将被显示。旧值将不再是可编辑的。

[0285] 可以不删除属于只读联系人的字段。属于可写的联系人的字段当被删除时消失。可以删除属于只读联系人和可写的联系人两者的字段,但是仅将其从可写的联系人中移除。该字段因此将仍然出现在详情中,但是将不是只读的。

[0286] 当用户删除联系人时,确认对话框询问是否还应当从联系人所属于的所有的可写

提供商中删除联系人。如果回答为是,则所有可写的联系人记录被删除。如果联系人还属于只读提供商,则对话框警告不可以从该提供商中删除该联系人。

[0287] 正如在导入处理期间,添加到提供商网站的联系人生成新的联系人记录。在提供商上进行的改变被反映在聚合服务中的对应联系人记录中。当联系人在提供商上被删除时,其具有从该提供商组移除对应联系人记录的效果。其不从聚合服务的联系人中删除该联系人。从一个提供商导入的联系人可以被添加到聚合服务的 UI 中的其它提供商。这个具有将联系人从一个提供商拷贝到另一个的效果 - 不是通过提供商它们本身容易完成的特征。从提供商组移除联系人将该联系人从提供商的地址簿中删除。如果用户决定不再与提供商同步,则他们被询问是否移除从该提供商导入的所有联系人记录。

[0288] 在设备 1802 上,推式服务将接收推式数据通知。其将查看每个数据项的序列号并且检查以弄清是否有任何一个已经失序。如果其检测到这样的场景,则其将广播指示这样的事物的发生的意图。可能要求该意图接收者决定他们想做什么。即使在失序情形的情况下,推式服务也将不丢弃任何数据。其将继续发射用于其检测到的所有数据项的意图。

[0289] 图 23-30 中图示了用户界面的示例性部分。社交网状态驻留在主屏 2300 上。其允许用户查看电话上或社交网站点上他们的最近更新的状态 2310,以及提供用于将状态更新推到社交网站点的手段。其为社交网体验的核心部分。状态更新被包括在多个社交网服务中,包括 Facebook™、MySpace™ 以及 Twitter™(其中它被称为 tweet)。用户使用状态更新来将他们正在做的、他们正在去哪里、或者任何其它“微博客”想法传送给他们的朋友。

[0290] 用户有能力从用户设备 1802 或 1803 的主屏(图 23)更新他们的社交网状态。为了更新他们自己的状态,用户点击当前状态 2310 并且将呈现使用户能够键入改变的文本字段。

[0291] 主屏状态 2310 将显示在已经被链接到用户的聚合服务账户的社交网网站(例如,所图示的屏上的 Facebook)上更新的最近的状态。如果用户界面没有在主屏 2300 上显示状态,则空间可以由指示选定该区域将允许用户键入状态的文本来填充。状态 2310 将被推到用户所属的一个或多个用户选择的社交网络。例如,一旦选择键入状态 2310,用户界面将向用户呈现下拉菜单,从这里能够选择将发送动作的内容提供商网站。用户能够选择内容提供商网站中的一个或多个。

[0292] 朋友的最近状态 2320 还可以被显示在主屏内,在示例性头条新闻“事件”下面。图 24 图示了如果用户点击图 23 中的事件标题 2320 时的示例性事件屏 2400。如图 24 中所见,事件屏 2400 包括来自用户正在跟随的账户的若干状态更新。图 25 图示了在用户选择了图 24 中图示的状态更新中的一个之后的示例性屏 2500。如图 25 中所见,用户被呈现现有将评论添加到状态更新的选项。图 26 图示了在用户已经选择将评论添加到图 25 中图示的状态更新之后的示例性屏 2600。如图 26 中所见,区域 2610 被提供以供用户键入评论。屏 2600 还可以提供了用于发送评论的发送按钮 2620 和用于放弃 / 取消该评论的放弃按钮 2630。在一个实施例中,如果用户选择如图 25 中图示的朋友的名称 2510,则用于所选择的朋友的联系人信息如果可用就可以被提出在屏上。图 29 图示了在选择了朋友名称之后的朋友联系人信息。如图 29 中所见,用户可以查看在各种内容提供服务器 1806-1808 上的用户移动号码和工作号码、多个电子邮件地址以及朋友账户。如果用户选择图 29 上的向下箭头 2910,则在内容提供服务器 1806-1808 中的一个上的朋友当前状态 3010 可以被提出在移动设备

1802 上,如图 30 中所图示。

[0293] 如上文所讨论的,设备 1802 的用户可以直接连接到 web 内容提供商 1806-1808 中的一个。图 27 图示了在用户已经选择图 25 中所图示的朋友之后的示例性屏 2700。如图 27 中所见,在选择了图 25 中的朋友“Tim”之后,客户端设备 1802 被直接带到在这个示例性实施例中为 Tim 的 Facebook™ 页面的朋友账户。图 28 图示了另一屏 2800,其中已经发生到 web 内容提供商中的一个的直接连接。如图 23 中所见,屏 2300 可以包括接近相称的框的服务中的每一个的视觉指示(徽标)2340 以指示该状态是用于哪个服务。因此,如果用户选择徽标 2340,则用户可以被直接带到对应的服务,如图 28 中所见。

[0294] 状态更新应当以及时的方式到达设备上。例如,他们应当在正由设备设置的时间的 90+% 的不到 15 分钟内到达设备屏上。换句话说,用户在设备 1802 用户界面上设置他们的状态,它在内容提供商网站 1806 上被更新,并且在不到 15 分钟内正被聚合服务更新之后返回到设备 1802。设想在用户在设备 1802 上活动时状态将优选地在不到 5 分钟内被更新。

[0295] 从网站 1806-1808(例如,Facebook™、MySpace™) 设置的清除的状态可以不被反映在主屏上。主屏 2300 将仅反映来自每个服务的最近设置的状态。MySpace™ 具有现有的功能性,如果用户将情绪和状态二者设置为空,则状态被自动地设置成“是在你的扩展网络中”。由于这是 MySpace™ 使他们的 API 工作并且 MySpace 用户所习惯于它的方式,所以当从服务在线或从设备设置空状态和情绪时,这个功能性将继续出现。

[0296] 对于 MySpace™ 状态更新(最新的更新是来自 MySpace™ 网站或来自设备的最近的更新是仅 MySpace™ 更新)，“情绪”被支持并且同时被显示为状态。主页用户界面(UI) 2300 还优选地包括最后状态更新 2330 的日期和时间。

[0297] 如果用户尚未将社交网账户链接到他们的聚合服务账户,则社交网络状态区域将不出现。设想主屏芯片将重组,垂直地集中以占用由于 SN 状态的缺席所留下的空间的部分。

[0298] 如上文所讨论的,用户可以选择或点击状态 2310 以更新状态,将用户带到状态更新用户界面。UI 包括在用户已经将他们的 web 服务器 1804 聚合服务账户链接到的所有社交网络上的所有的他们的当前状态的列表。还可能存在接近于相称的框的服务中的每一个的视觉指示(徽标)2340 以指示该状态是用于哪个服务。用于键入新的状态的 UI 空间可以用“是...”来预填充,其能够通过退格来擦除。

[0299] 当诸如 MySpace™ 的一个服务是选择的唯一服务时,状态和情绪二者都在 MySpace™ 上用用户的当前状态和情绪来预填充。这支持用户仅更新状态或情绪而没有意外地清除任何一个。每个服务都能够通过改变每个个体的社交网站点的状态文本来(一次一个地)单独地更新。当仅更新 MySpace™ 时,UI 还支持情绪的选择。选择情绪允许用户键入文本以跳到情绪列表中的特定点(不是过滤器)并且允许用户滚动列表。

[0300] 用户可能具有选择“所有”以立刻更新所有服务的选项。更新状态并且选择“所有”服务用新的状态更新来替换所有下面的服务。状态被推到用户已经将他们的 web 服务器 104 聚合服务账户链接到的所有社交网络。当更新“所有”服务包括 MySpace 时,情绪不是为 MySpace 更新的选项并且情绪被设置成没有情绪。

[0301] 如果用户仅具有链接到支持状态的他们的 web 服务器 1804 聚合服务账户的一个

服务,则不存在下拉菜单并且没有用于“所有”的选项。唯一的服务链路被简单地列举为目的地服务。在一个实施例中,在主屏上的状态是最近更新状态,不管它是来自移动设备还是来自社交网服务。在主屏上的状态应当具有标识该状态更新源自哪里(例如,移动设备、Facebook、Twitter 等)的接近于它的某个图标 2340。在一个实施例中,用户能够点击屏 2300 中的内容提供商图标 2340 并且跳转以直接访问内容提供商。

[0302] 发送空白状态能够从所选择的特定服务,或者如果选择的所有服务(注意, Twitter 不具有清除状态的概念,因此先前的状态将保持原封不动)中清除状态。在电话上的其它活动期间将存储应用的最后的屏/状态,并且一旦完成在 SN 状态更新实用程序中中断了用户的活动的任何活动(例如在接收到了电话呼叫以后),则用户将返回到这个最后的屏。

[0303] 在事件应用中支持的一个反应是以 @Reply 对 Twitter 项目作出响应。这将导致新的“me”状态被发送到核心服务,并且将被反映在主屏上、以及状态的列表的 Twitter 线中(如果用户具有支持的 Twitter 账户)。

[0304] 当状态被发送到 web 服务器 1804 聚合服务服务时,状态可以以“发射并且忘记”方法被发送,意味着将不向用户发送该动作进入社交网服务(尽管作为每 UE,但是示出了反应已经被从设备发送出去的欢迎(toast)确认)的确认。如果他们的字符不超过所允许的最大字符,则将仅能够发布状态更新。所允许的状态更新的长度取决于正向其上传状态的服务。例如,Facebook™ 当前允许 160 个字符,MySpace™ 当前允许 160 个字符,并且 Twitter™ 当前允许 140 个字符。如果选择了“所有的服务”,则最小的公分母将是限制器(例如,如果包括 Twitter™,则最大值为 140)。

[0305] 当允许 30 个字符或更少时,可以倒计时地显示计数器。当超过字符的最大数量时,计数器能够显示负数。如果用户切换发布模式(例如,从 Twitter 改变到 Facebook™),则能够相应地对计数器进行调整(或者如果用户现在具有还剩超过 30 个字符要键入,则消失)。

[0306] 当计数为负的时如果用户选择“发布”,则出现指示发布不能够进行并且解释哪些服务具有超过的最大值的对话。

[0307] web 服务器 1804 聚合服务用户界面提供 Facebook™ 页面的有限的图像。例如,在事件 2320 中其可以示出列表中的朋友图片和名称。Facebook™ 图片可以直接从 Facebook™ 下载,并且作为联系人保留在设备上。当设备同步到服务器时,其将被保存在服务器上。点击屏的该区域示出了名称、他们的主图片、他们的最后状态以及关于该状态的任何评论。你能点击人的名称以打开用于该人的背景墙的到 Facebook™ 移动装置的直接访问连接。替代地,你能够点击 Facebook™ 图标以跳到 Facebook™ 直接访问(不通过聚合服务)。

[0308] 对于新闻订阅源,诸如上文中所讨论的 RSS 订阅源,简短的评论和图片可以被推到设备。图片将与简短的概要一起被显示。用户可以点击文章以从内容提供商得到另外的信息。这个处理能够通过识别文件输入字段来自动进行——他们是特殊的 HTML 类型文件——并且提示用户选择要包括的图片。然而,这个能够通过配置进行更新所需要的步骤来进一步增强。对于每个社交网站,服务器能够包括指定用户在该站点上能够执行的动作的脚本,和如何通过标准 HTTP 连接将那些动作发送到该站点;浏览器然后在菜单中将这些动作呈现给用户,提示所要求的输入,并且将全体发出到该站点。

[0309] 核心 web 服务将通过同步框架来支持社交网络发消息和通过 SNMAIL 应用（处理器代理）对消息的动作。

[0310] 图 31 图示了示例性聚合服务系统 3100 中的示例性发消息层。将诸如状态更新、照片上传、消息以及评论的动作从设备 3102 发送到 web 服务器前端部分 3118。动作是独立于目的地内容提供商网站的具有通用格式的输入，而不是被包括的目标的身份。前端对该消息进行传送，并且后端处理器 3116 在专用于内容提供商网站的插件中处理该动作。例如，如果动作是将消息发布到 Facebook™ 和 Twitter™，则设备的通用消息在 Facebook 插件中和 Twitter 插件中被格式化。插件中的每一个都将坚持将其相应的适当地格式化的内容上传到 Facebook™ 和 Twitter™ 的内容提供商网站。

[0311] 还如图 31 中所图示的，消息和他们的属性以及动作状态码在由后端 3116 拉之后经由单向同步被发送到设备，并且发送到前端 3118。更新消息和动作状态的服务调用将经由同步单向同步调用来完成。

[0312] 后端 3116 他们本身支持去往和来自社交网络的通信。前端 3118 和设备 3102 保持同步引擎，通过该同步引擎诸如联系人的某些信息在设备与服务器 104 之间同步。

[0313] 服务器 3104 能够经由账户设定服务将提供商设置发送到设备 3102。这个仅是单向的（从服务到设备），并且不要求同步来进行这个工作。在设备上创建了用于消息的类别，其能够包含由每个提供商支持的所允许的消息属性和能够对数据执行的所允许的动作。

[0314] web 服务器 3104 聚合服务电话应当维持包含提供商 ID 和所有字段的消息的数据库，以支持消息加同步元数据。对于从服务接收数据，设备必须处理消息属性的改变列表和来自服务的动作状态。这个包括消息的所有的属性。例如，改变列表可以包含：

	同步消息改变列表
	<< 同步消息改变列表项目 1 >>
[0315]	<<同步消息改变列表项目 2 >>
	<<同步消息改变列表项目 3 >>
	...
	<<同步消息改变列表项目 n >>
	同步消息改变列表项目
[0316]	改变类型 = <改变 添加 移除>
	内容 = <消息属性>

消息属性	
[0317]	toAddressList: String list* fromAddress: String body providerID: Long mesgGUID: Long messageState: enum/Integer(Read, Unread, etc.) [subject: String] [ThreadID: Long] [attachmentUrlList: String List] [folderName: String] [flag: Integer] [importance: Integer]

[0318] 在括号中的项目是为可选的那些。例如，不是所有的提供商都支持对象。具有‘*’的数据类型是可选列表，其中 1 或无可以在列表中（例如，Facebook™ 支持多个地址，而其它的仅支持一个）。mesgGUID 是在设备与服务之间共享的全局 ID。提供商与服务之间的 ID 可以是不同的 ID。假定所有的提供商都支持按 GUID 引用消息。

[0319] 在这个方法中，以下是假定：1. 动作经由 WS 调用被传递给服务。在下文中定义了动作格式；2. 服务器返回立即确定，表明动作被接受以供处理，或者如果其不能够排队等待这个动作则返回错误；3. 服务器（在处理器中）异步地处理这个请求，并且当完成时，用推式管理器将推式发出到设备；以及 4. 设备，使用单向同步，客户端用其最后的服务器锚来提交 GET 请求，并且这导致包含如由同步协议所定义的消息和动作状态的响应主体（例如，到 /ws/sync/1/update/{account_id}/{device_id}/{app_id}/{last_server_anchor}? devkey = android&format = json 的 HTTP GET）。对于 SNMail 同步 app_id 字段将是 3。

[0320] 设备识别到 PUSH 不是可靠的并且必须超时，以及如果其期望推式响应，则再次重试该动作。如果推式信道被有意地断开连接（例如在漫游情况下），则设备必须以轮询间隔执行动作和消息同步。在一个实施例中，存在 2 个超时条件：1. 发送 WS 调用的动作可能超时；和 2. 可能从未接收到推式。在任一情况下，客户端应用必须处理这些超时。

[0321] 状态表可以包含对客户端有用的信息。在一个实施例中，1 的状态码指示处理成功地完成，0 的状态码指示数据被接收并且动作在进行中，并且 -1 的状态码可以指示错误发生。动作 ID 可以由服务和设备来共享。这可能允许服务报告返回关于动作的状态（同步回到设备，其中单向同步到状态表）。经由同步使往回接收到的数据同步是通过经典同步的单向同步，不存在同步框架所需要的改变。

[0322] 图 32 图示了根据一个实施例的示例性协议。能够利用类似朋友订阅源的协议将消息动作发送到 SNMail 应用。服务器可能需要动作的自变量，其是服务器正在作用于的消息 ID。例如，snmail 动作 WS 调用具有以下这种格式：

[0323] /ws/snmail/\$version/\$res/\$accountid/\$deviceid? <standardparams> (版本

1 SNMAIL 动作 WS API, \$res 是作为用于动作的“a “的资源)

[0324] 示例性消息属性是：

	名称	值
	主体	具有主体数据的 utf8 字符串
	文件夹名称	具有文件夹名称的 utf8 字符串
	对象	具有对象数据的 utf8 字符串
	读取状态	布尔型(false=未读, true=已读)
	优先级	整数
[0325]	标志	整数
	AttachURL	utf8 字符串
	ThreadID	长整型
	MessageID	长整型
	ToMemberIDs	目的地列表: ["id1", "id2", "id3", ..., "id<n>"]
	ToSmtPAddresses	目的地电子邮件地址列表: ["address1@mail.com, ..."]

[0326] 动作能够以动作头部开始。动作头部可以包含以下：

[0327]

名称	描述
actionID	标识这个动作的长 GUID
actionType	要执行的动作的类型 (见下文中的动作)
providerName	提供商的标识符

[0328] 动作可以是以下中的一个或多个：

动作类型	描述	参数
sendMsg	为给定提供商发送新消息	<消息属性>
replyMsg	回答现有的消息	<消息属性>
deleteMsg	删除消息	消息 ID, 文件夹名称 (仅仅 MySpace)
deleteThread	删除线程	线程 ID
forwardMsg	将消息转发给联系人	<消息属性>

[0329]

	markReadMsg	将读取状态标记为消息已读	消息 ID, 文件夹名称 (仅仅 MySpace)
	markUnReadMsg	将读取状态标记为消息未读	消息 ID, 文件夹名称 (仅仅 MySpace)
	moveMsg	将消息移动到给定的文件夹	消息 ID, 文件夹名称
	flagMsg	标志消息以用于跟踪	消息 ID, 标志
	setPriMsg	设置消息优先级/重要性	消息 ID, 优先级
	replyThread	回答现有的线程	<消息属性>
	forwardThread	转发线程	<消息属性>
[0330]	markReadThread	将读取状态标记为线程已读	线程 ID
	markUnReadThread	将读取状态标记为线程未读	线程 ID
	moveThread	将线程移动到给定的文件夹	线程 ID, 文件夹名称
	flagThread	标志线程以用于跟踪	线程 ID, 标志
	setPriThread	设置线程优先级/重要性	线程 ID, 优先级
	createFolder	创建消息的存储文件夹	文件夹名称
	syncMessages	强制提供商得到消息 (按需同步)	锚, [limit (integer max # to go get)]

[0331] 动作和参数能够作为 JSONArray 被发送, 并且 liltts 能够作为 JSON 阵列被发送。动作结果能够由服务器存储在状态表中, 其随后由客户端来同步。为了代码级实施和效率目的, 动作类型将作为 ENUM 而不是将由客户端和服务端来共享其值的名称被发送。syncMessages 动作具有作为参数的锚, 因为同步动作在后台中被完成。锚将告诉处理设备是否可能已经通过发送锚 0 擦掉其消息。

[0332] snmail 应用能够具有通知 API, 以当要求同步时向设备实现推式。这可以是例如 URI :/ws/snmail/0/n。

[0333] 特别预期的是, 本发明不限于本文中所包含的实施例和说明, 但是包括那些实施例的修改形式, 包括实施例的部分和落入以下权利要求的范围内的不同实施例的元素的组合。

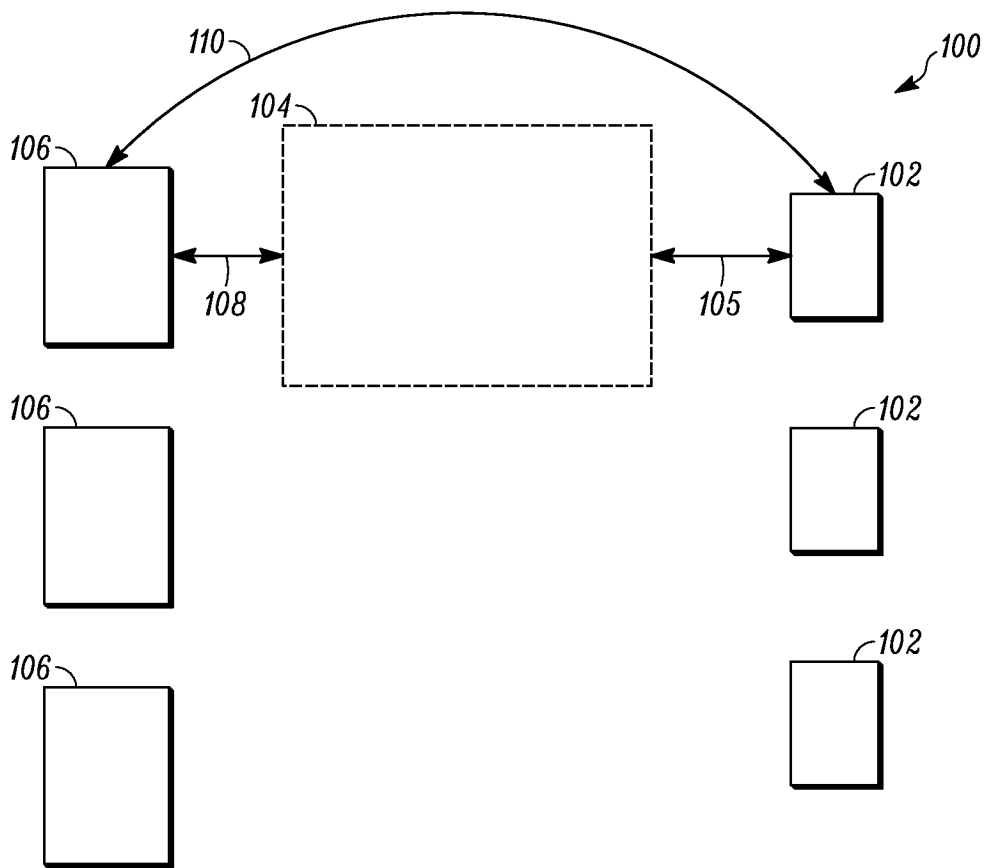


图 1

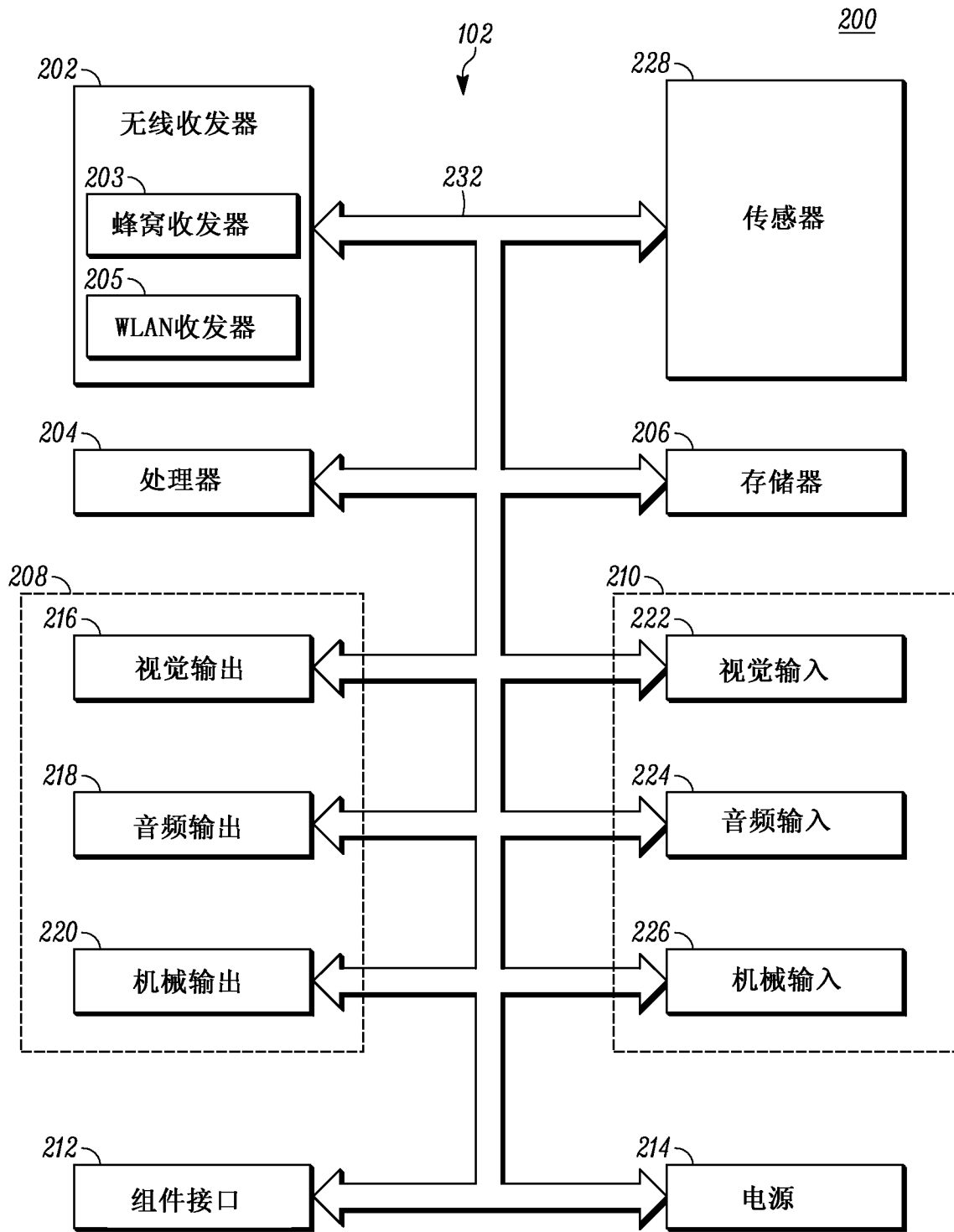


图 2

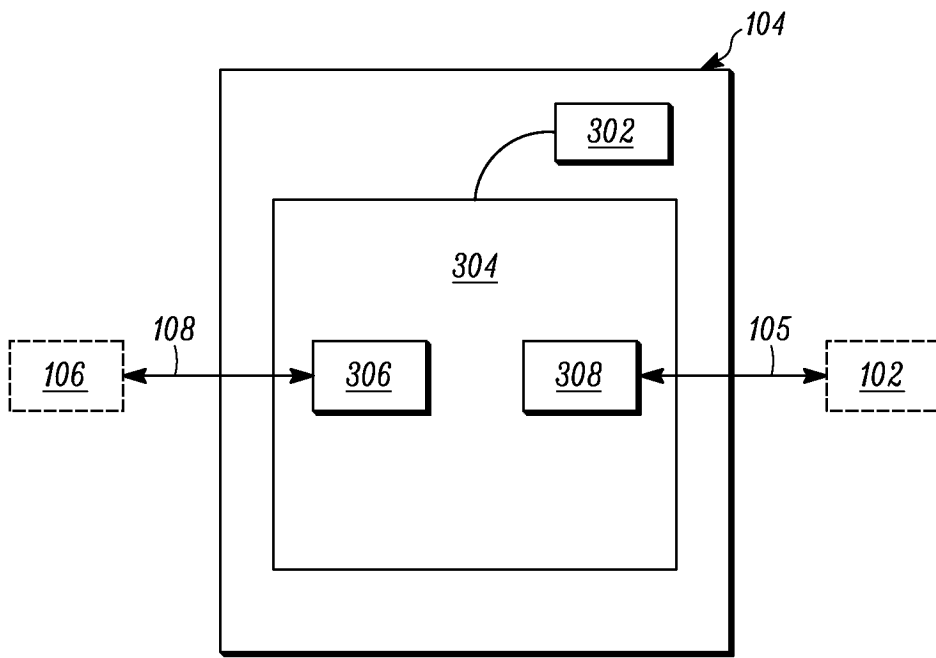


图 3

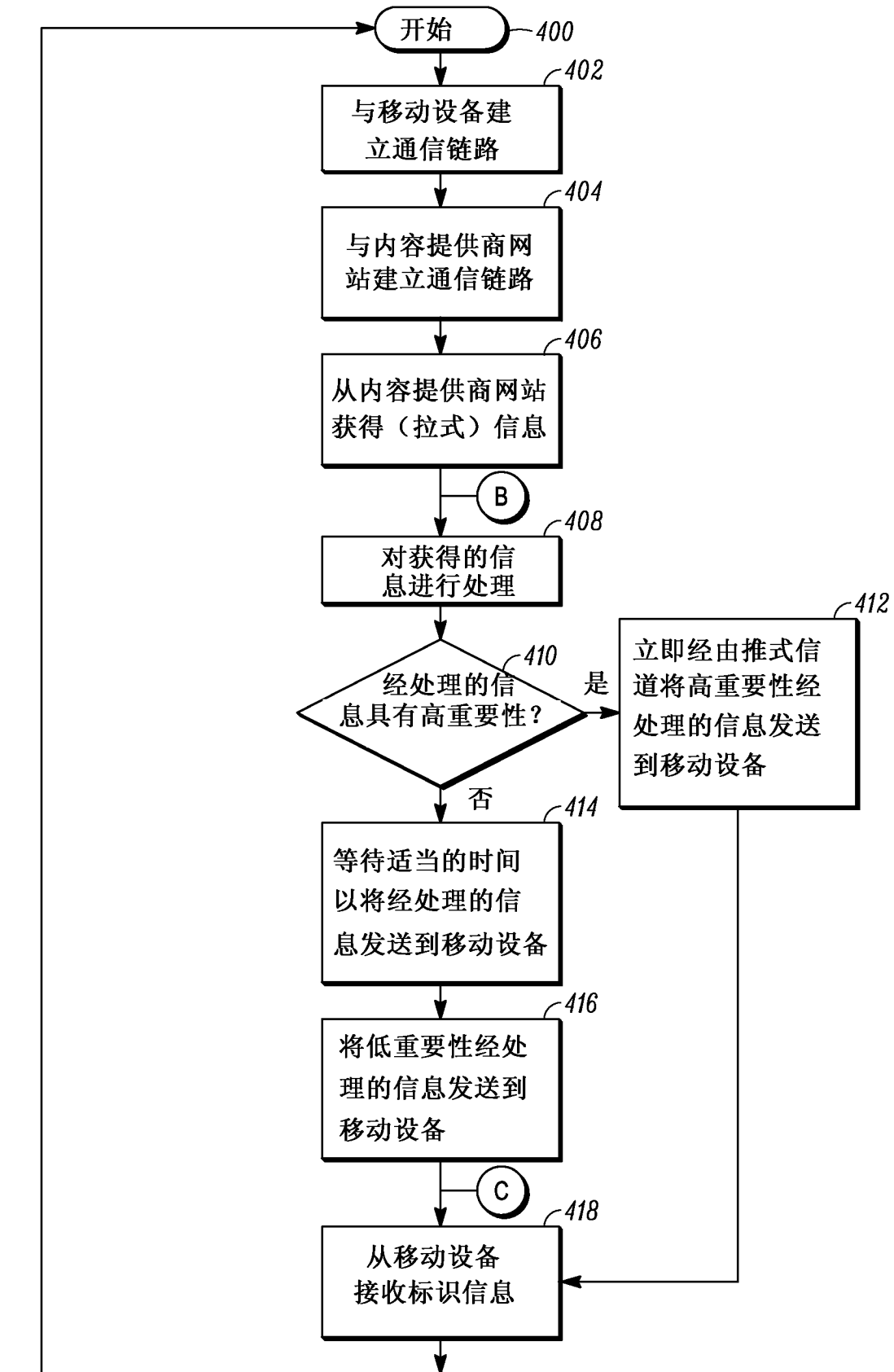


图 4A

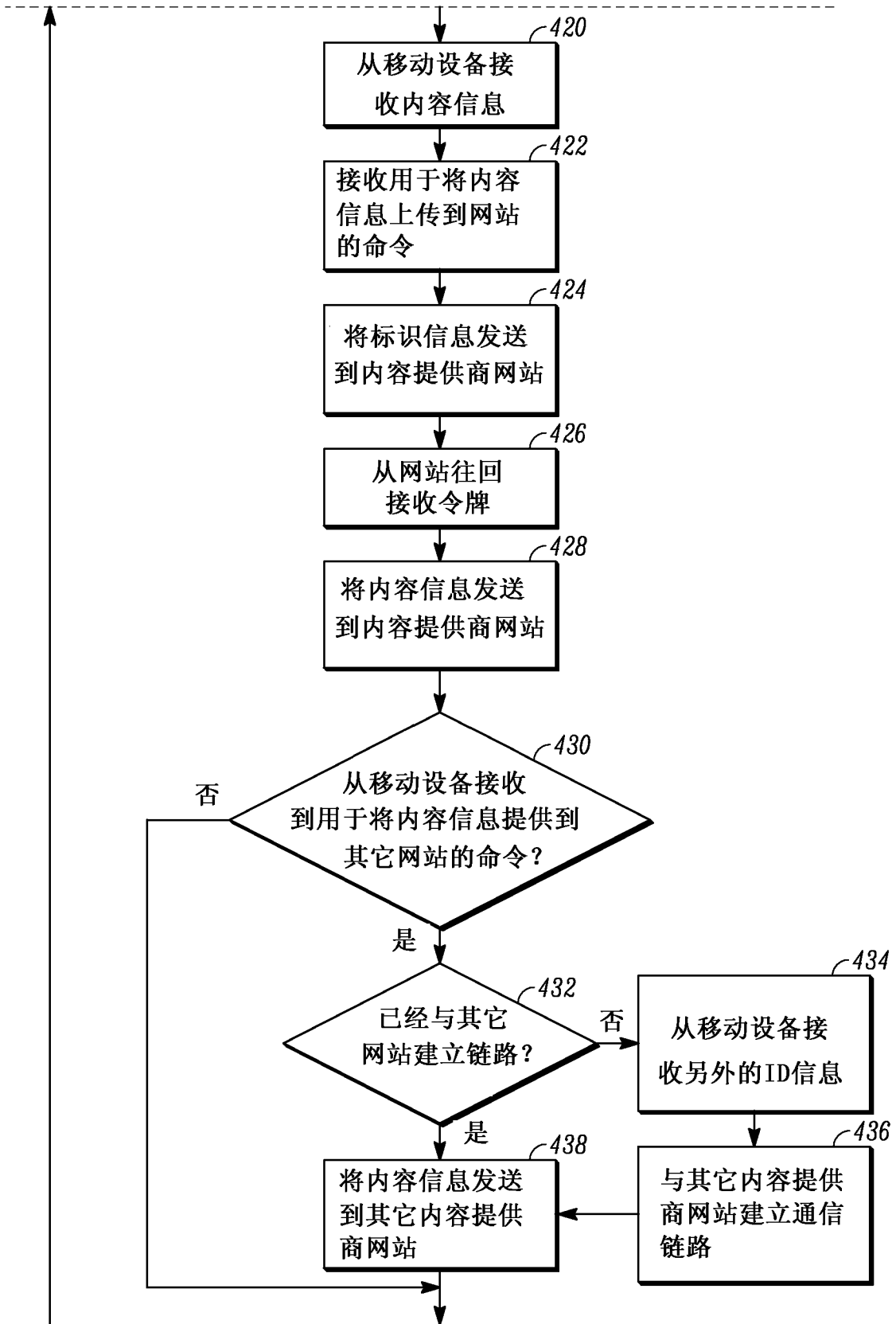


图 4B

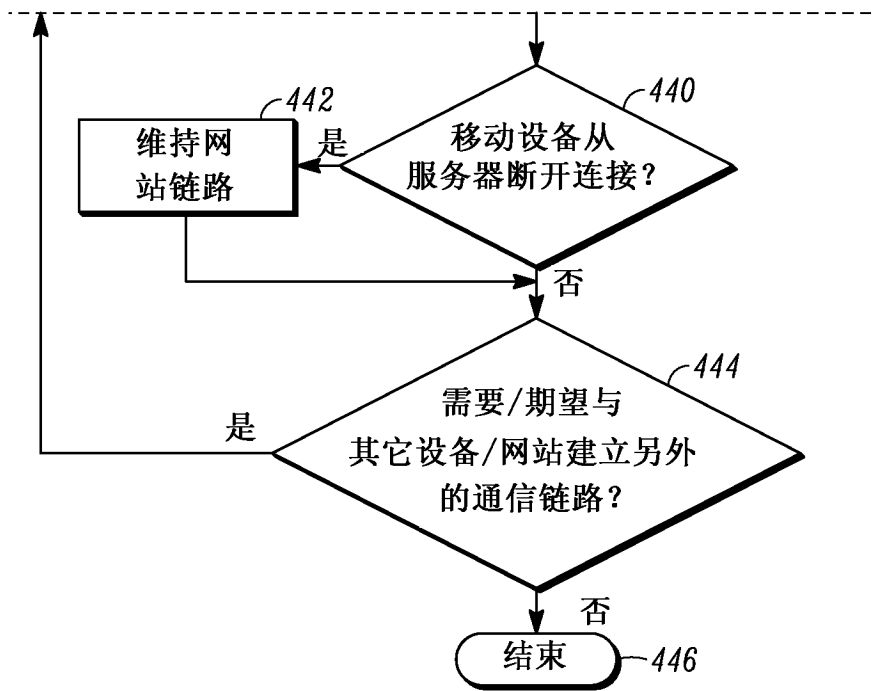


图 4C

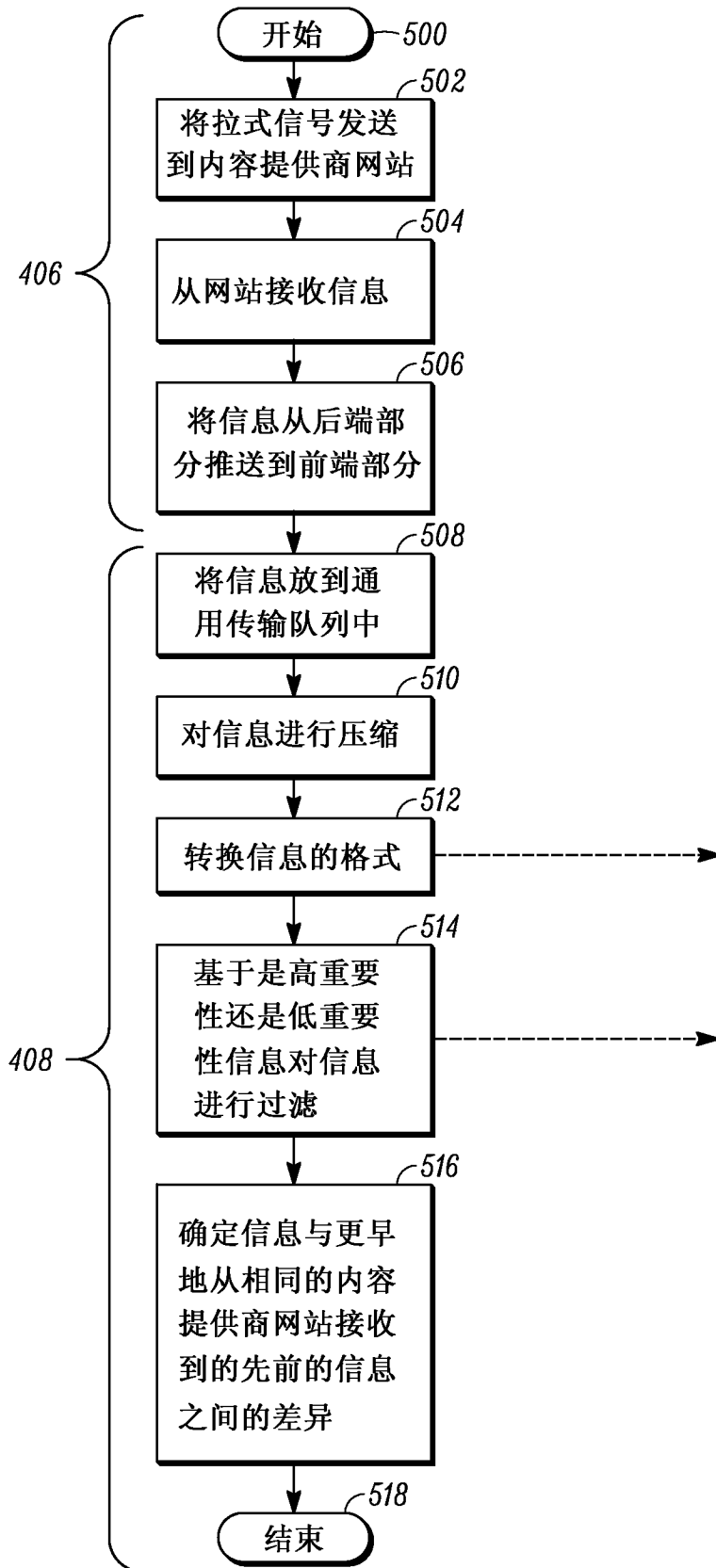


图 5A

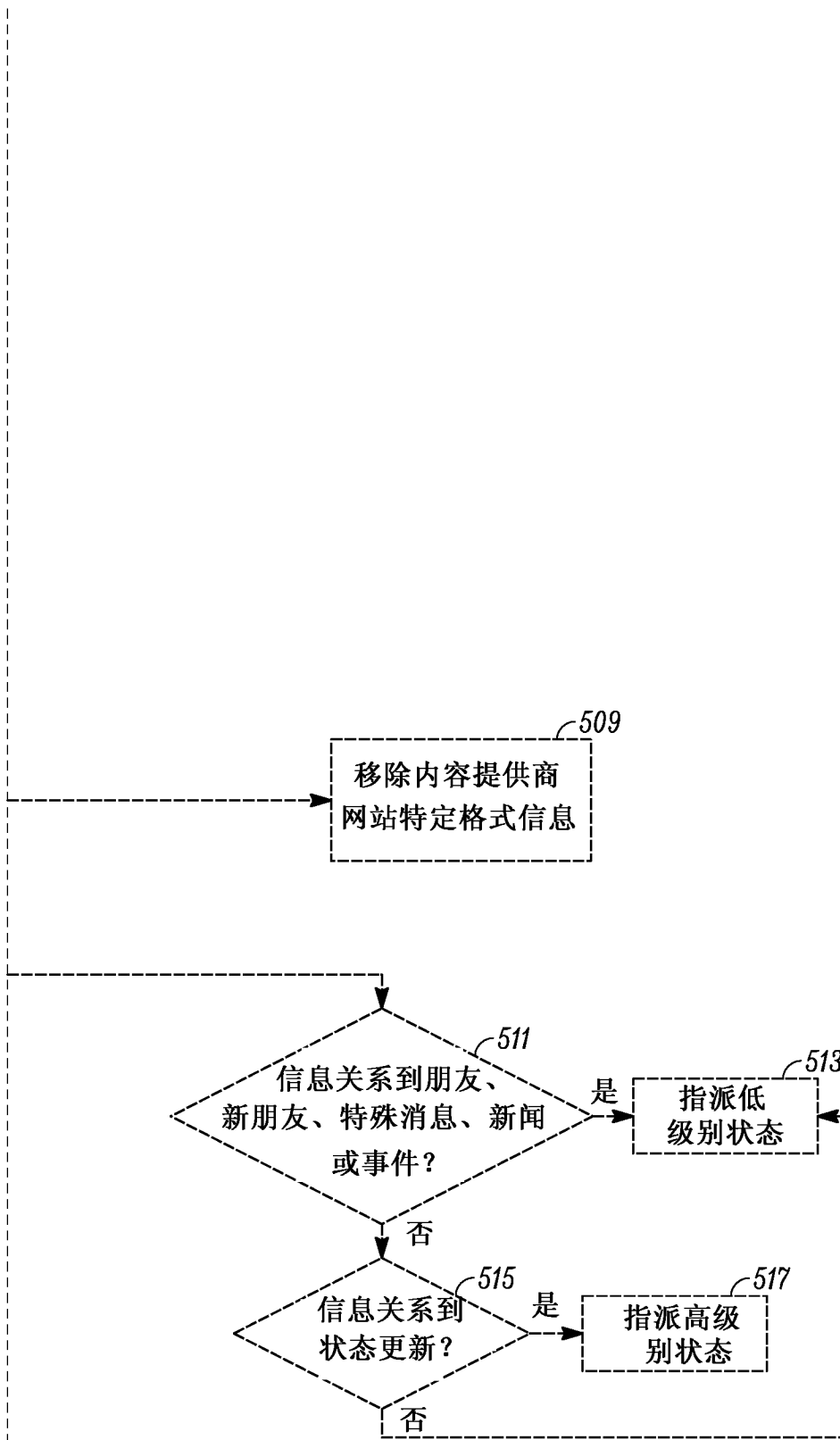


图 5B

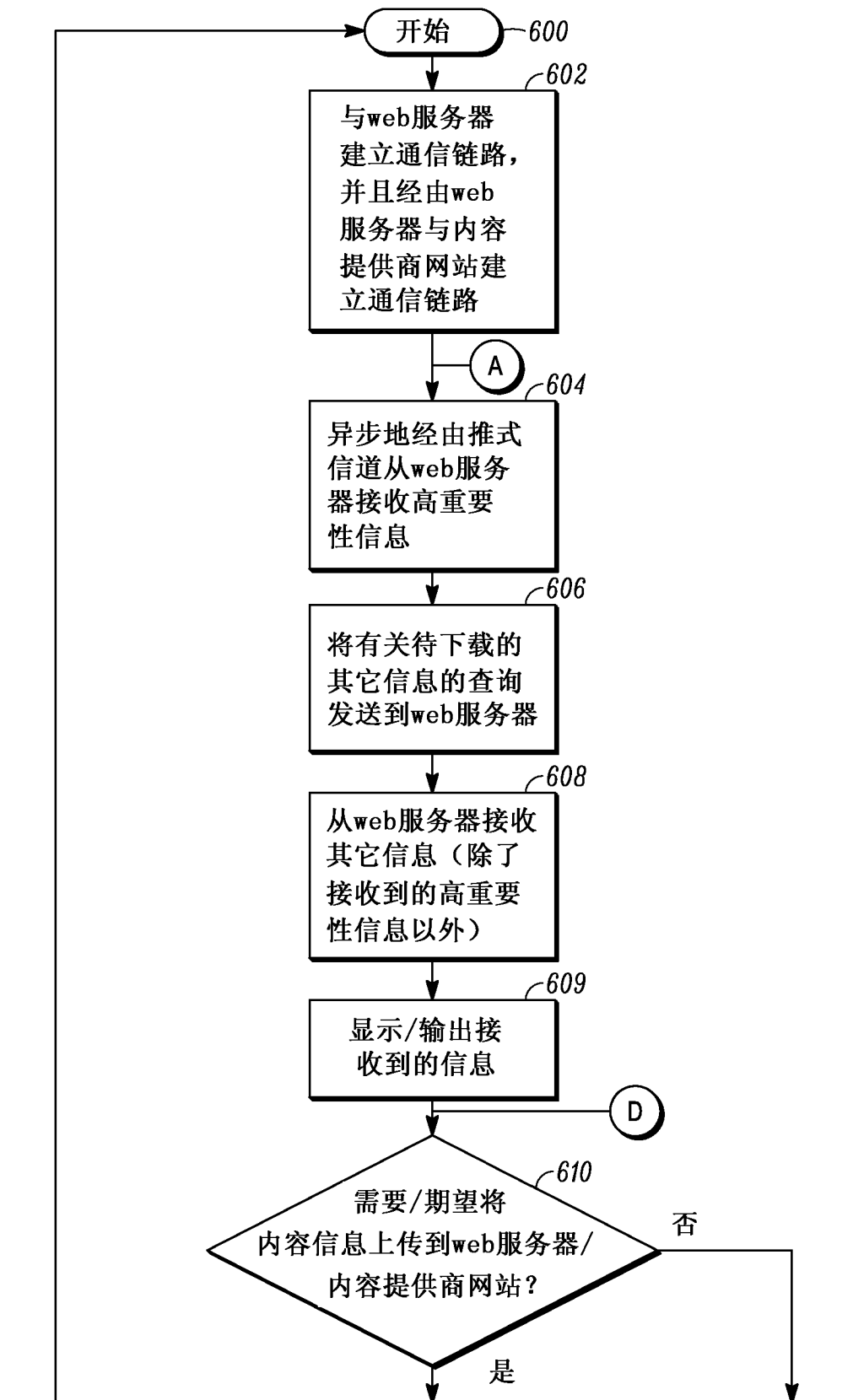


图 6A

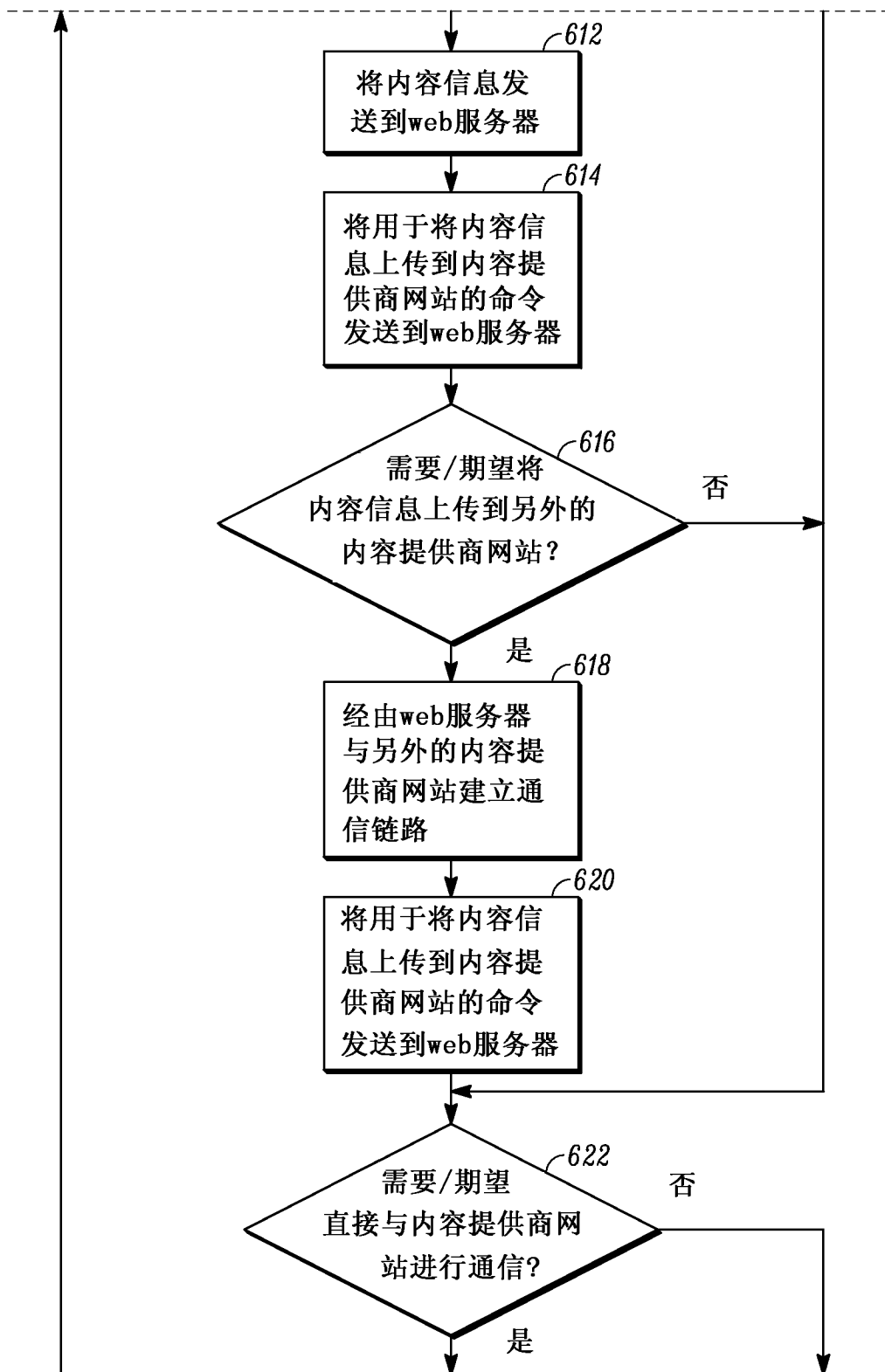


图 6B

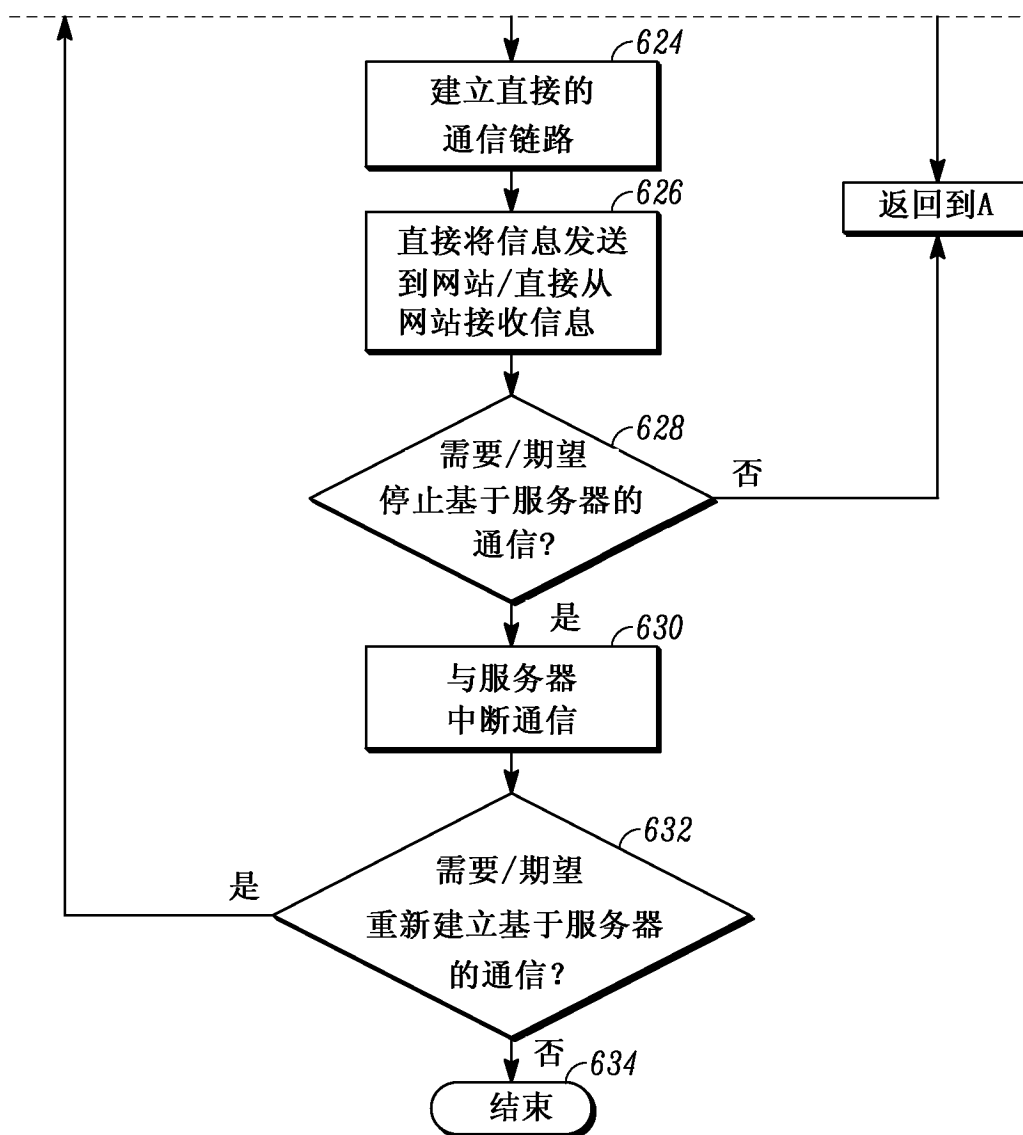


图 6C

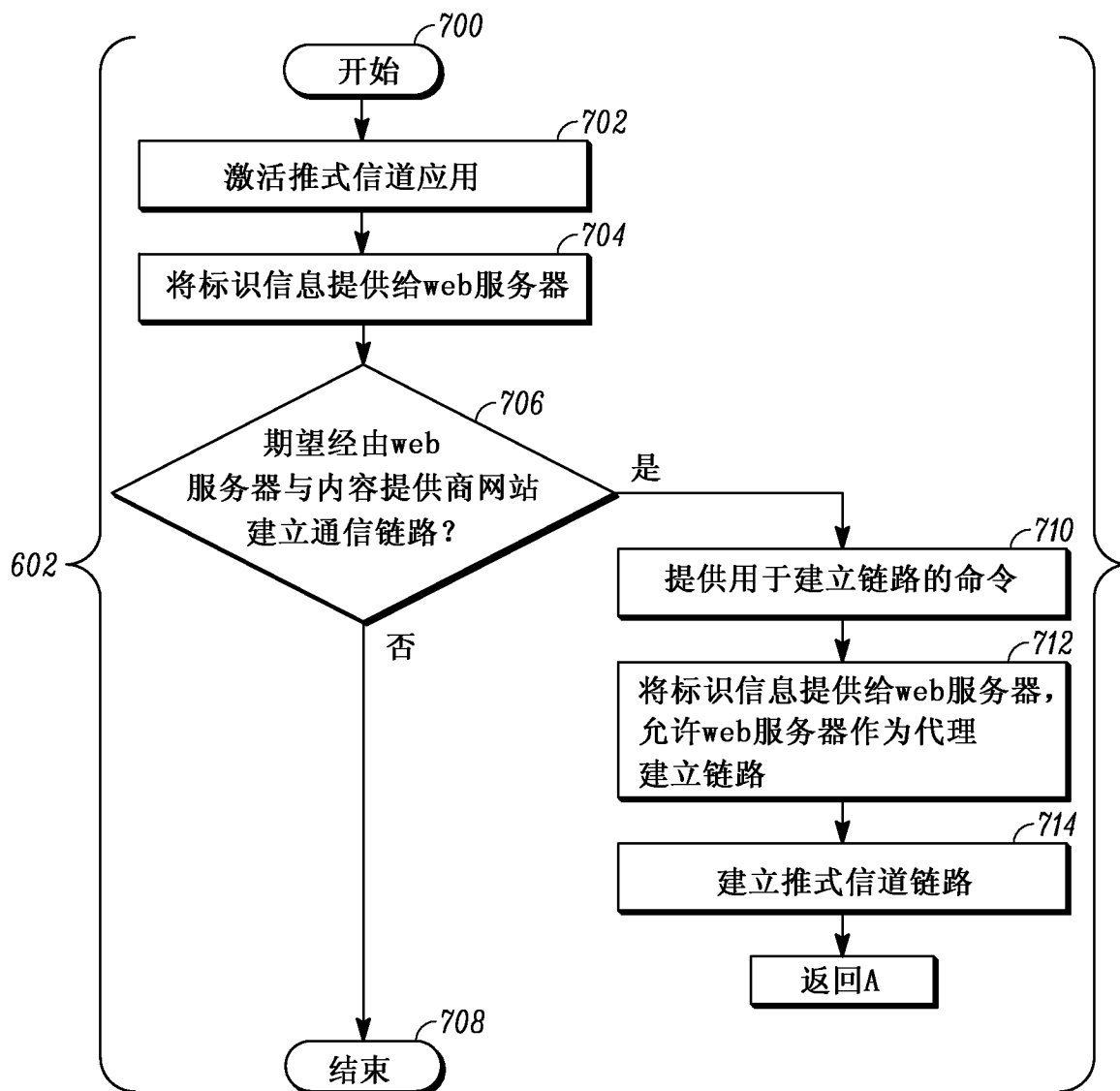


图 7

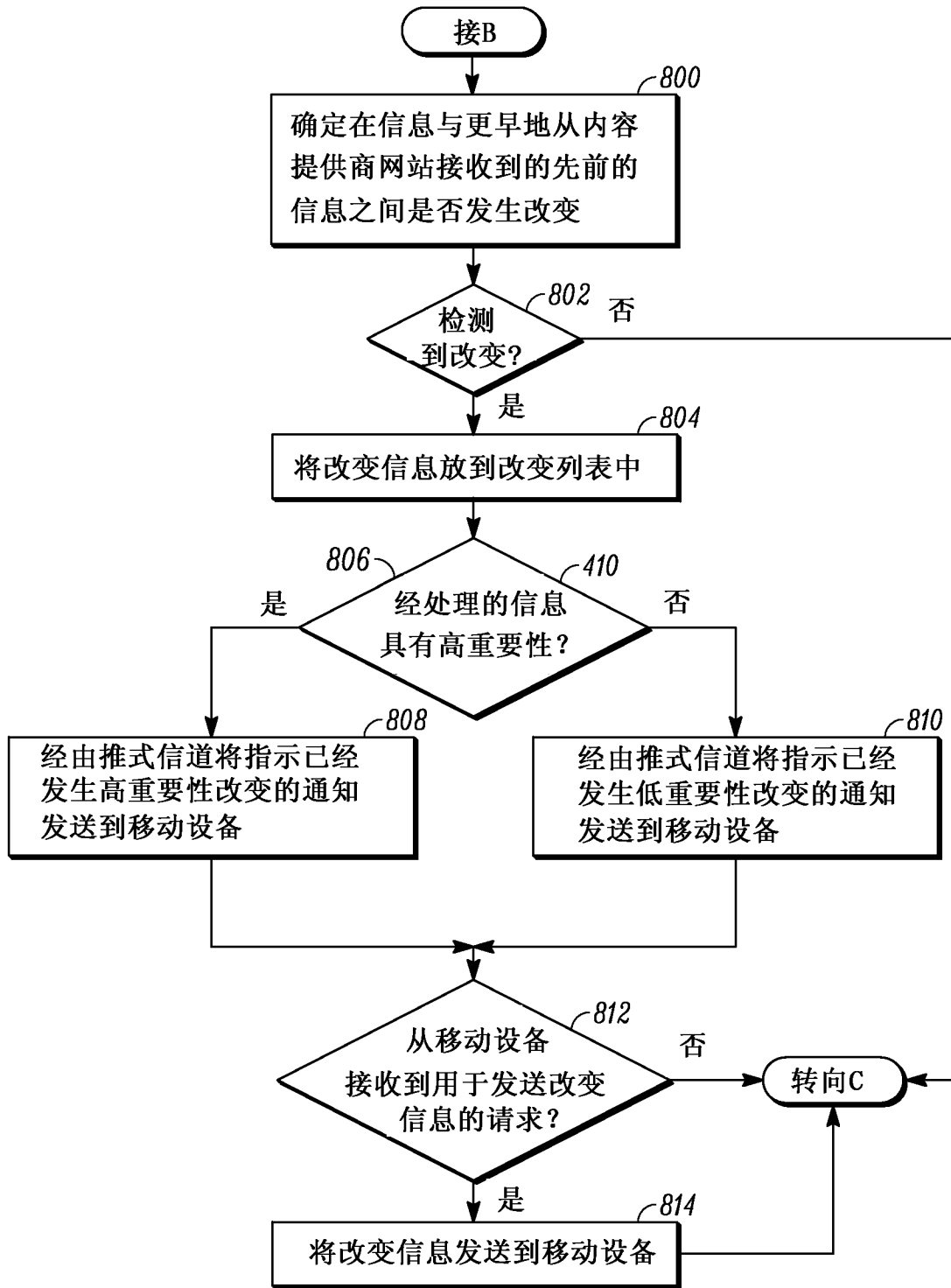


图 8

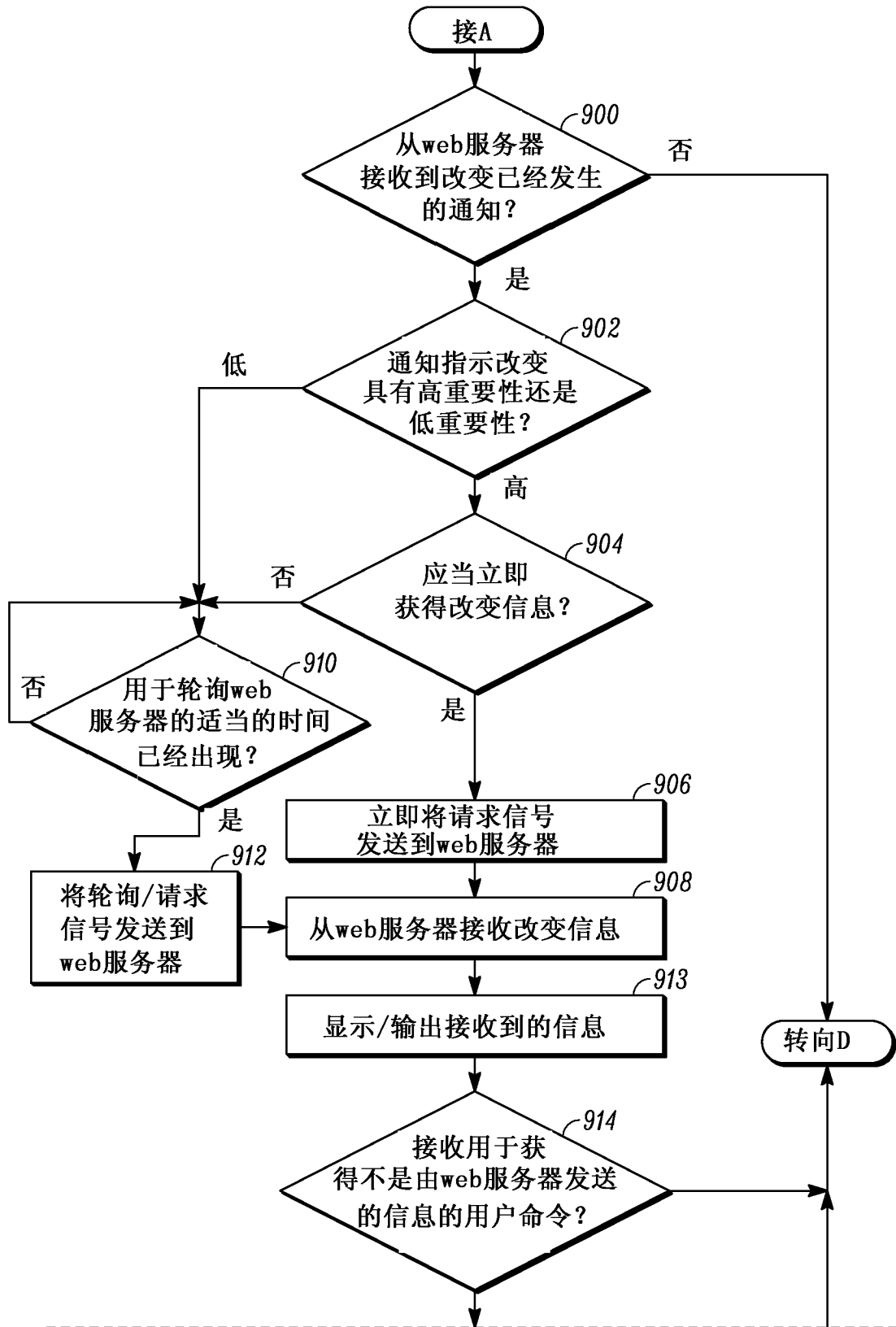


图 9A

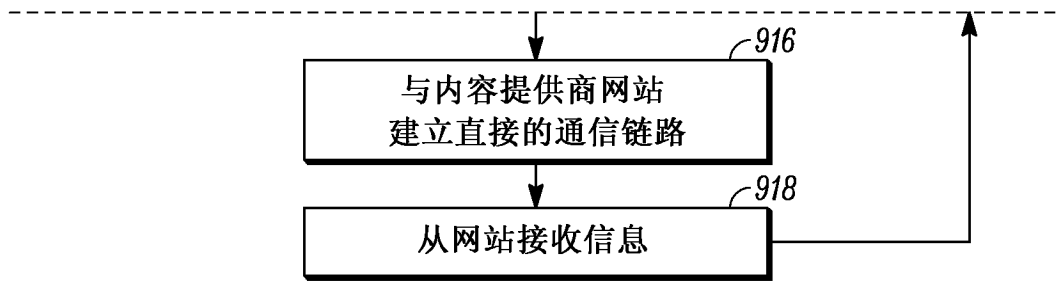


图 9B

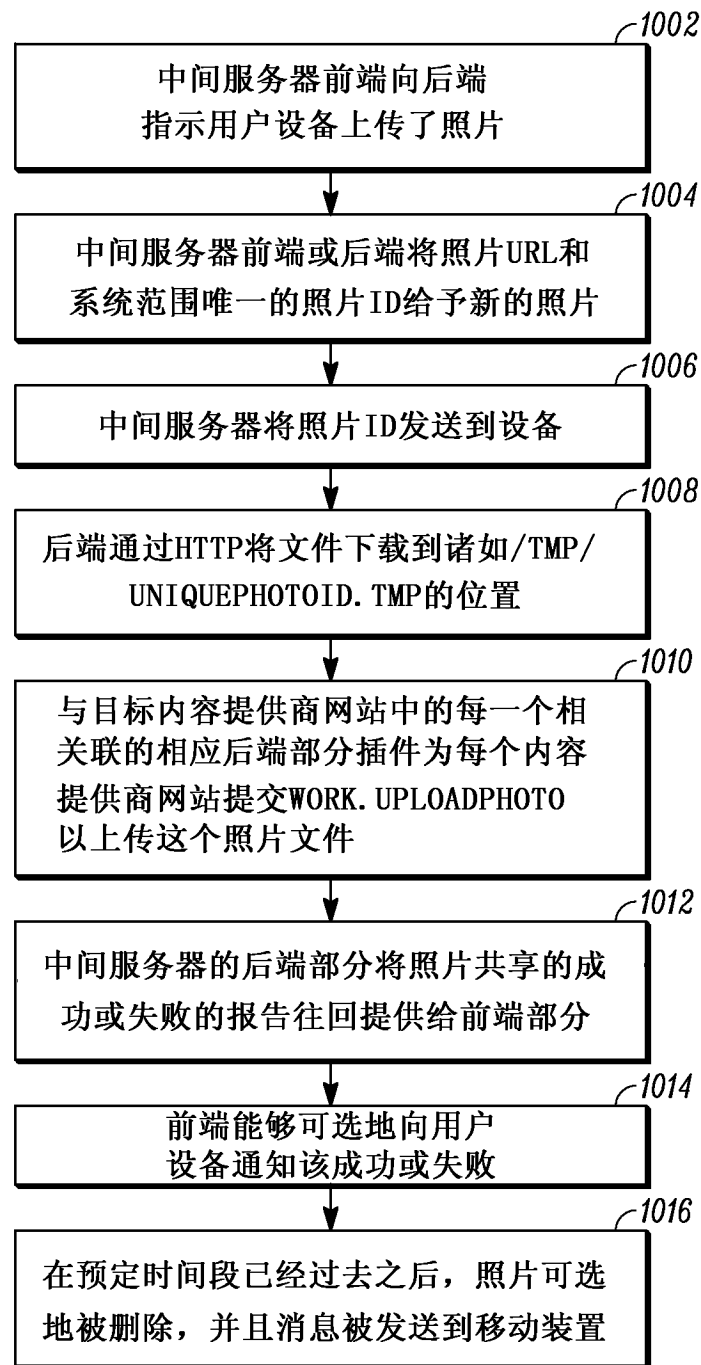


图 10

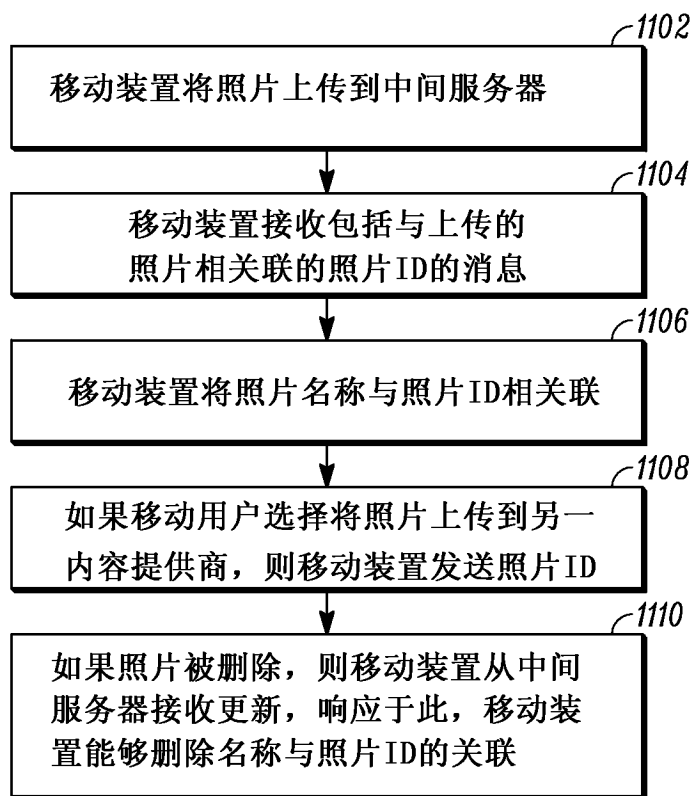


图 11

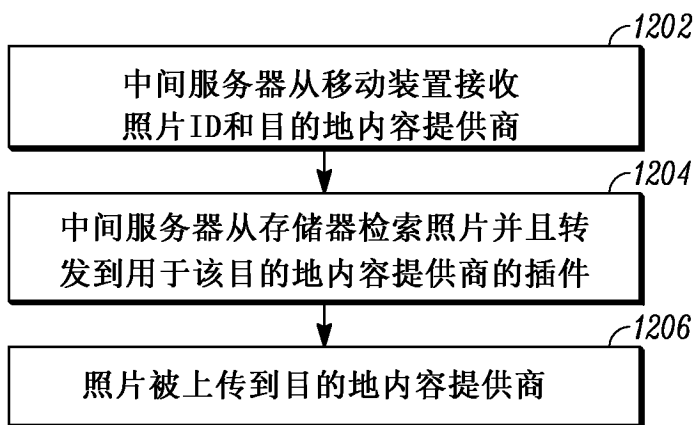


图 12

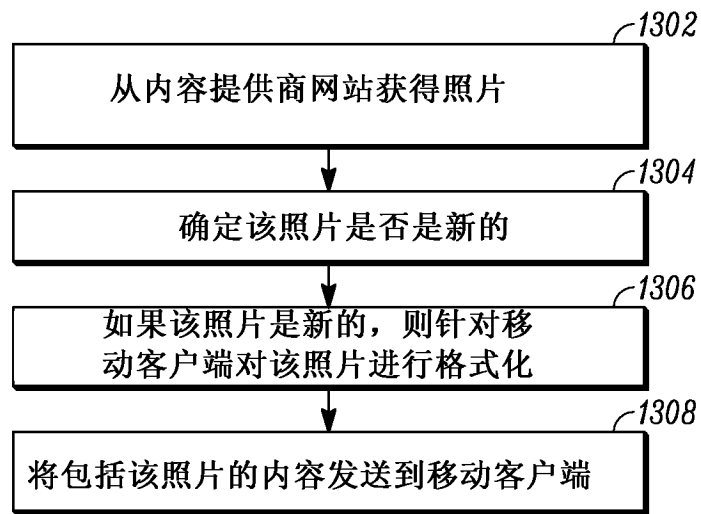


图 13

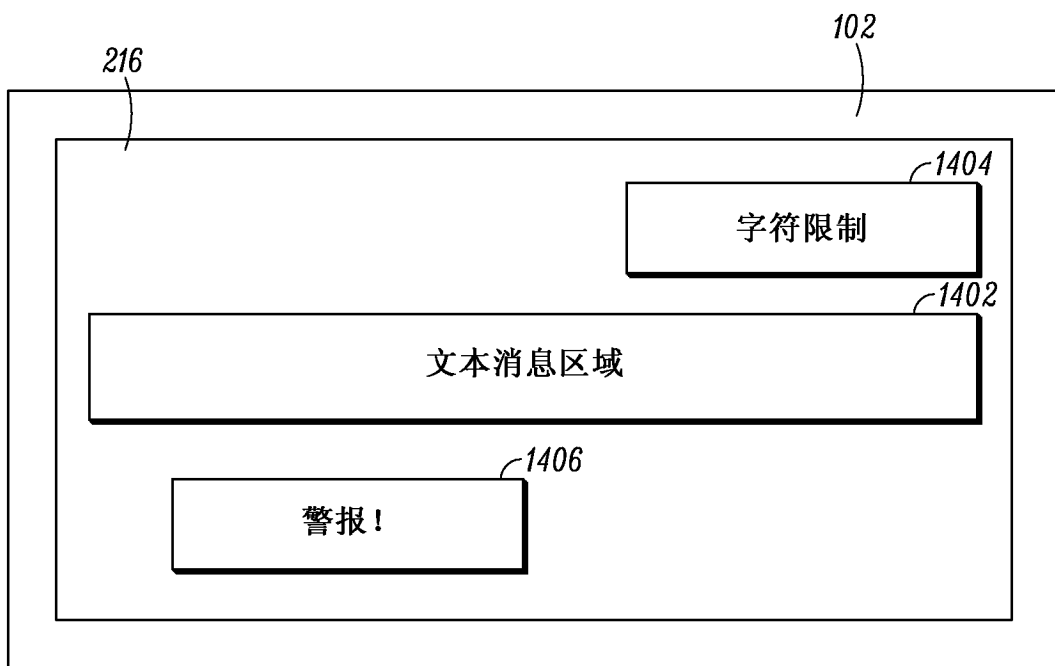


图 14

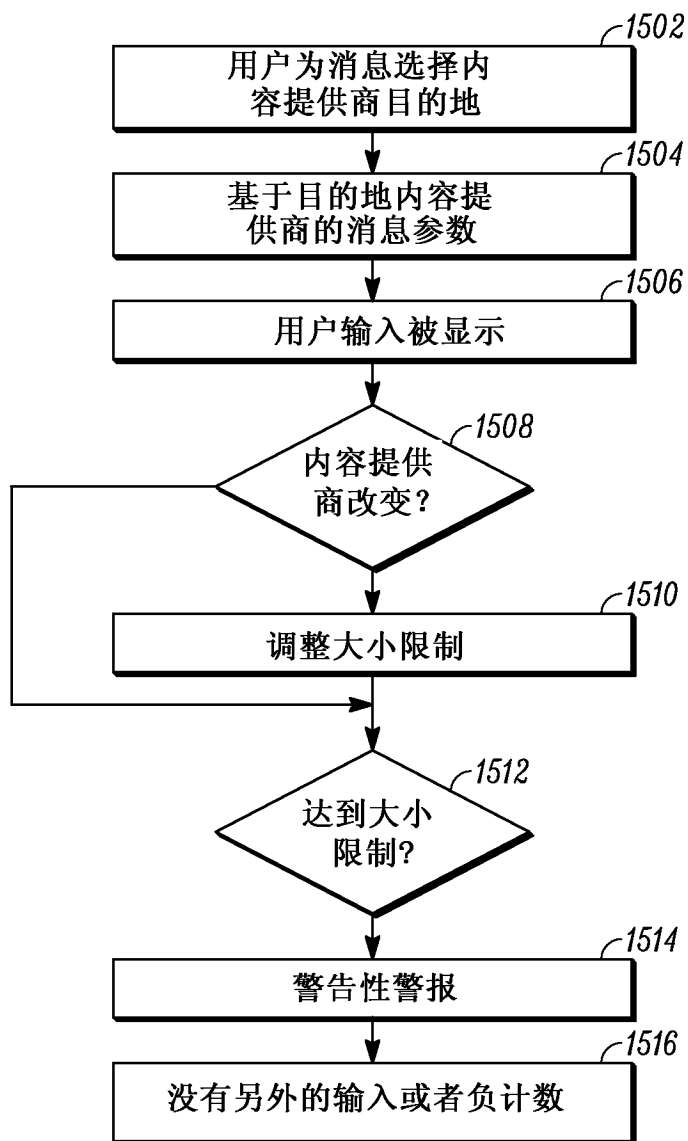


图 15

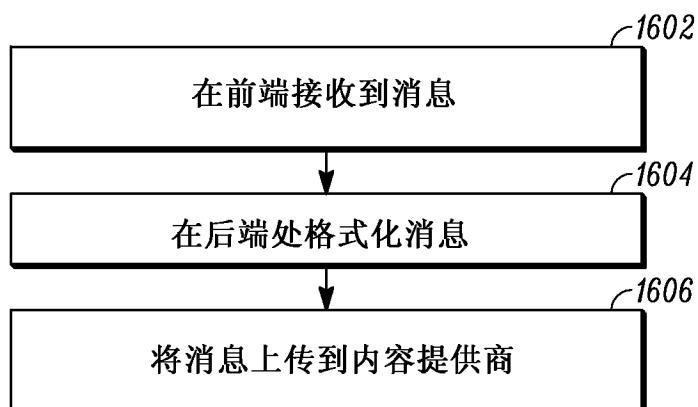


图 16

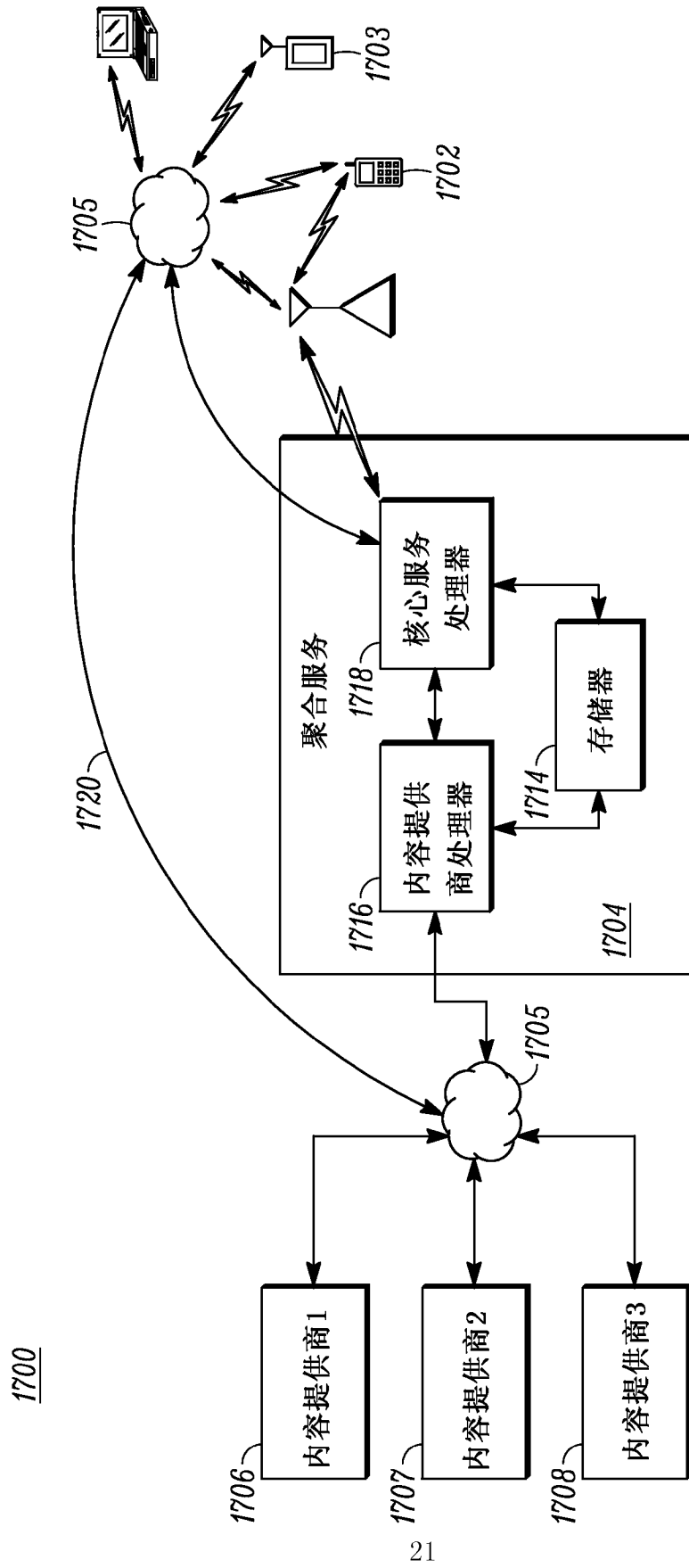


图 17

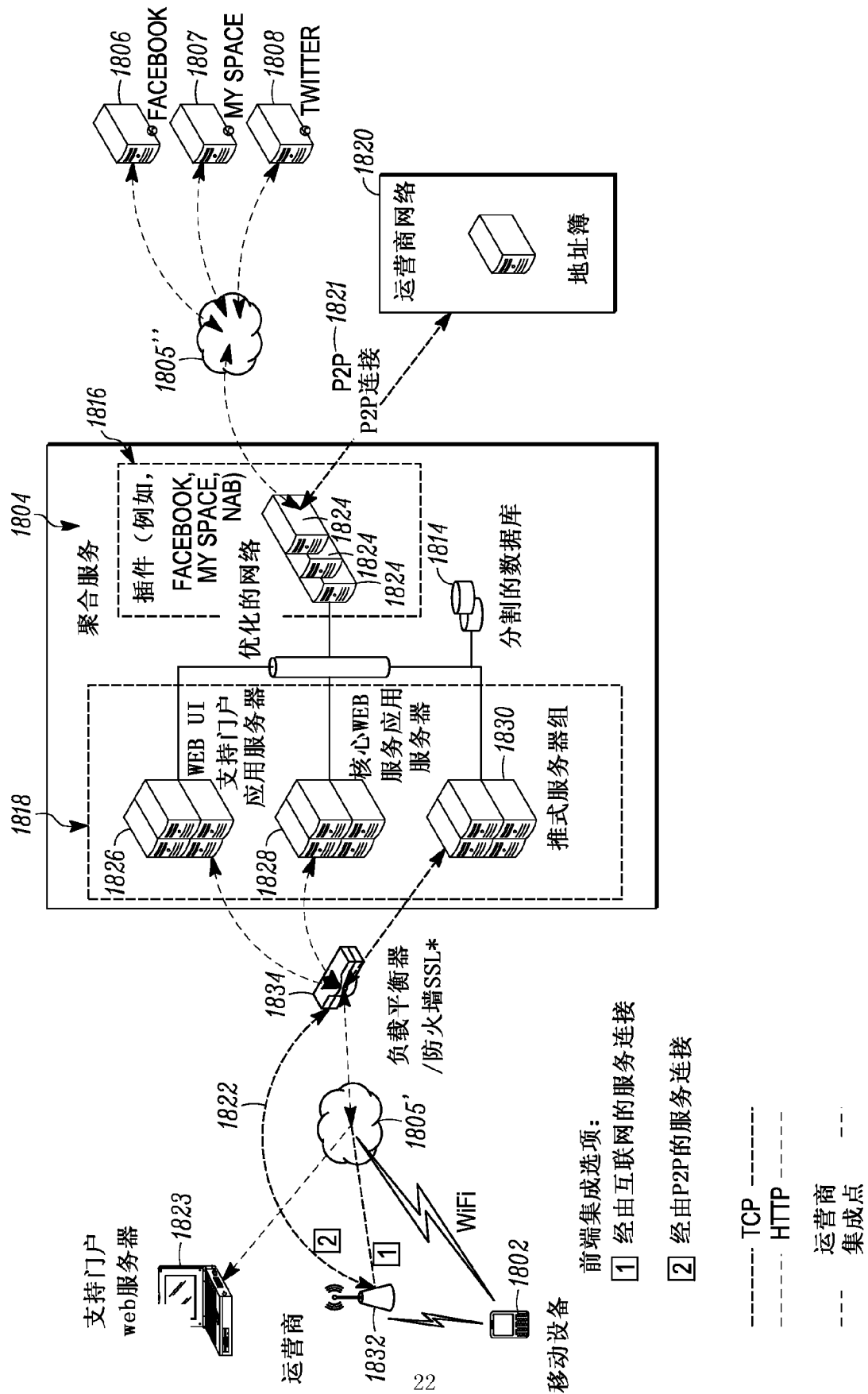


图 18

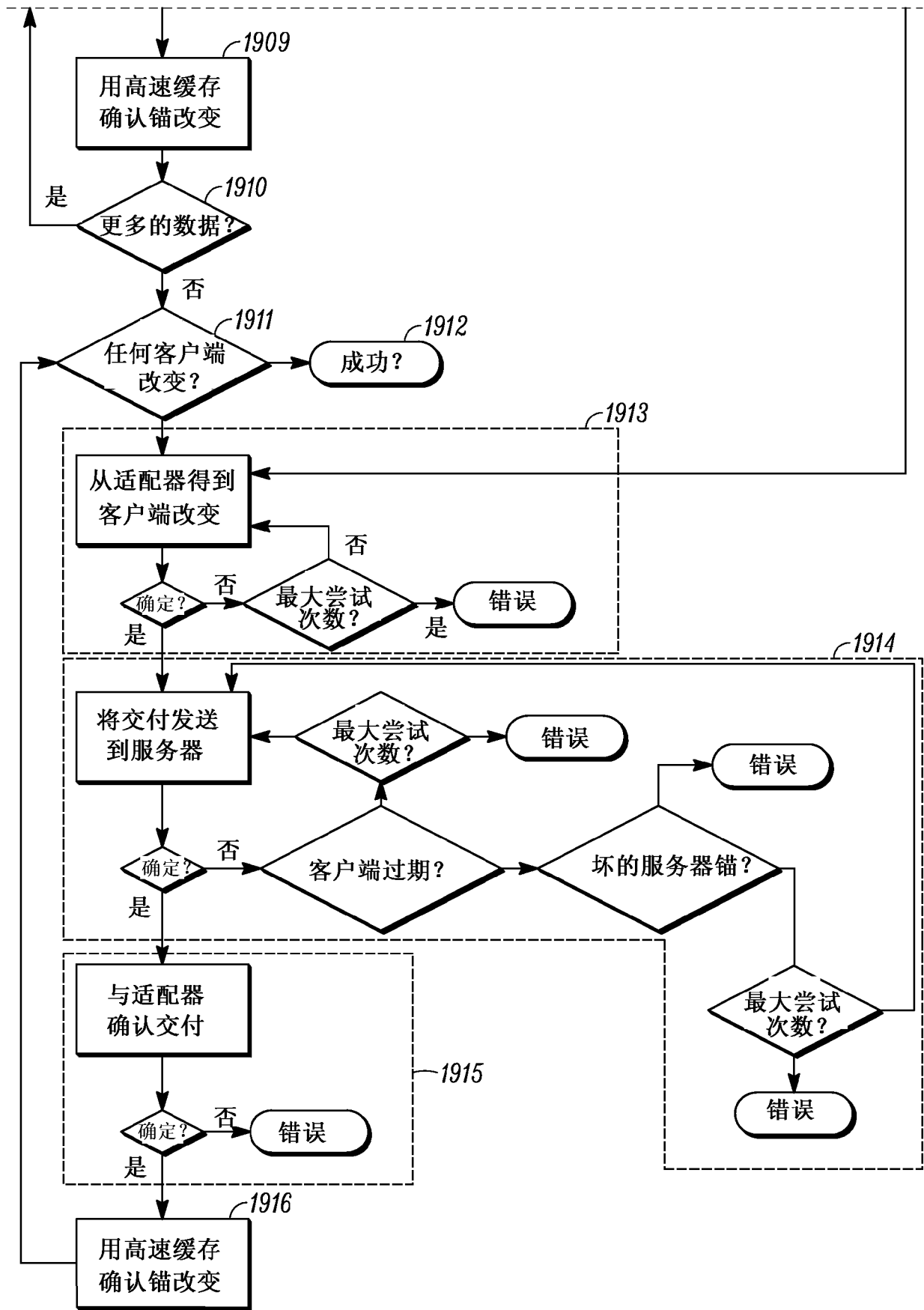


图 19B

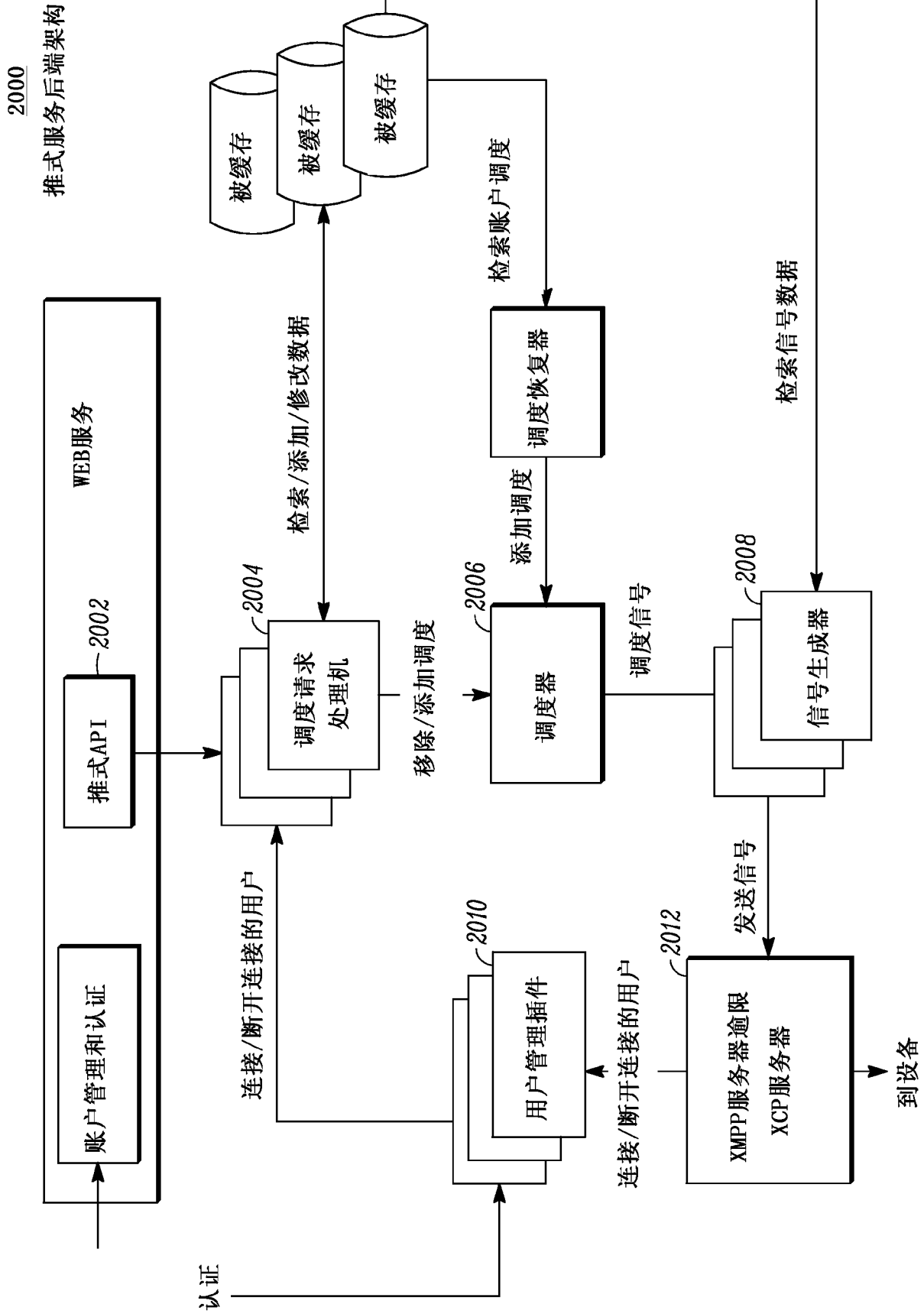


图 20

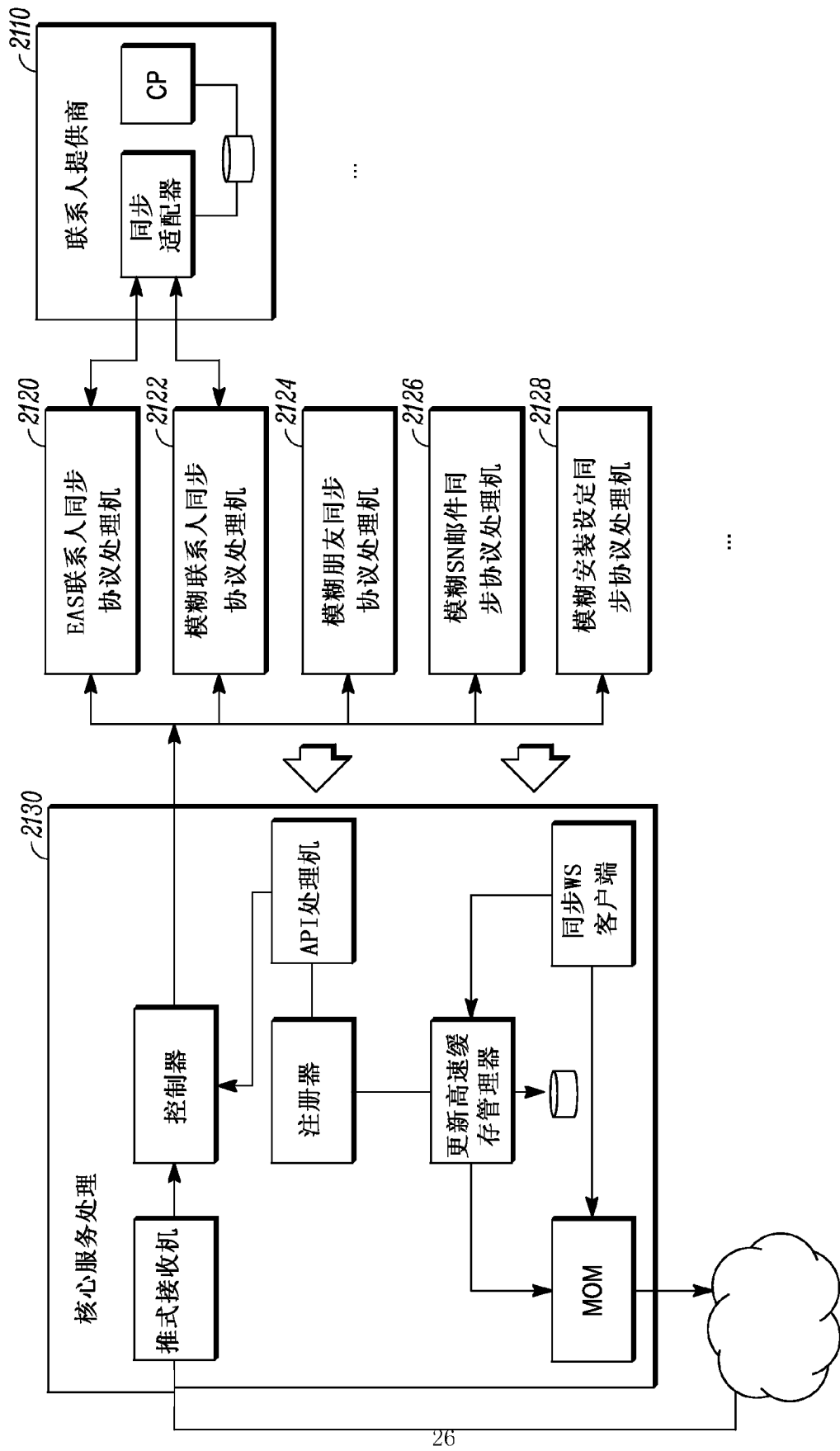


图 21

2200

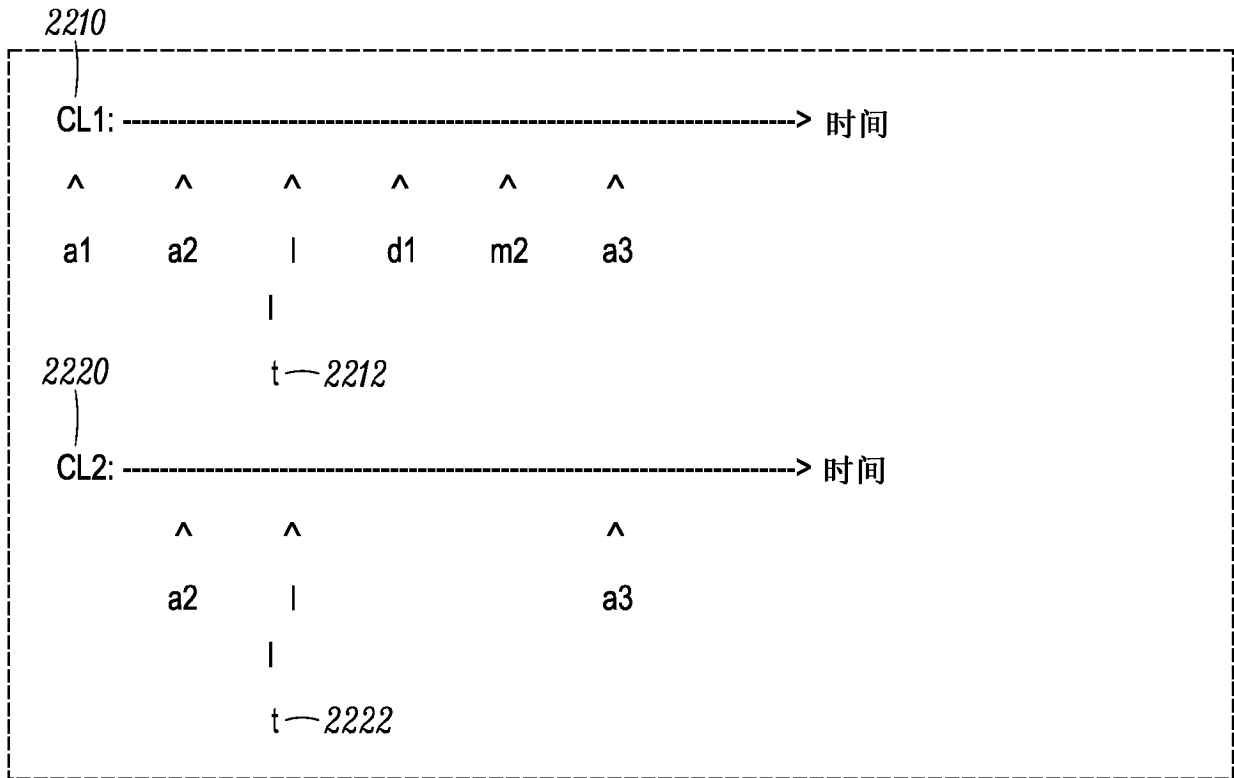


图 22

2300

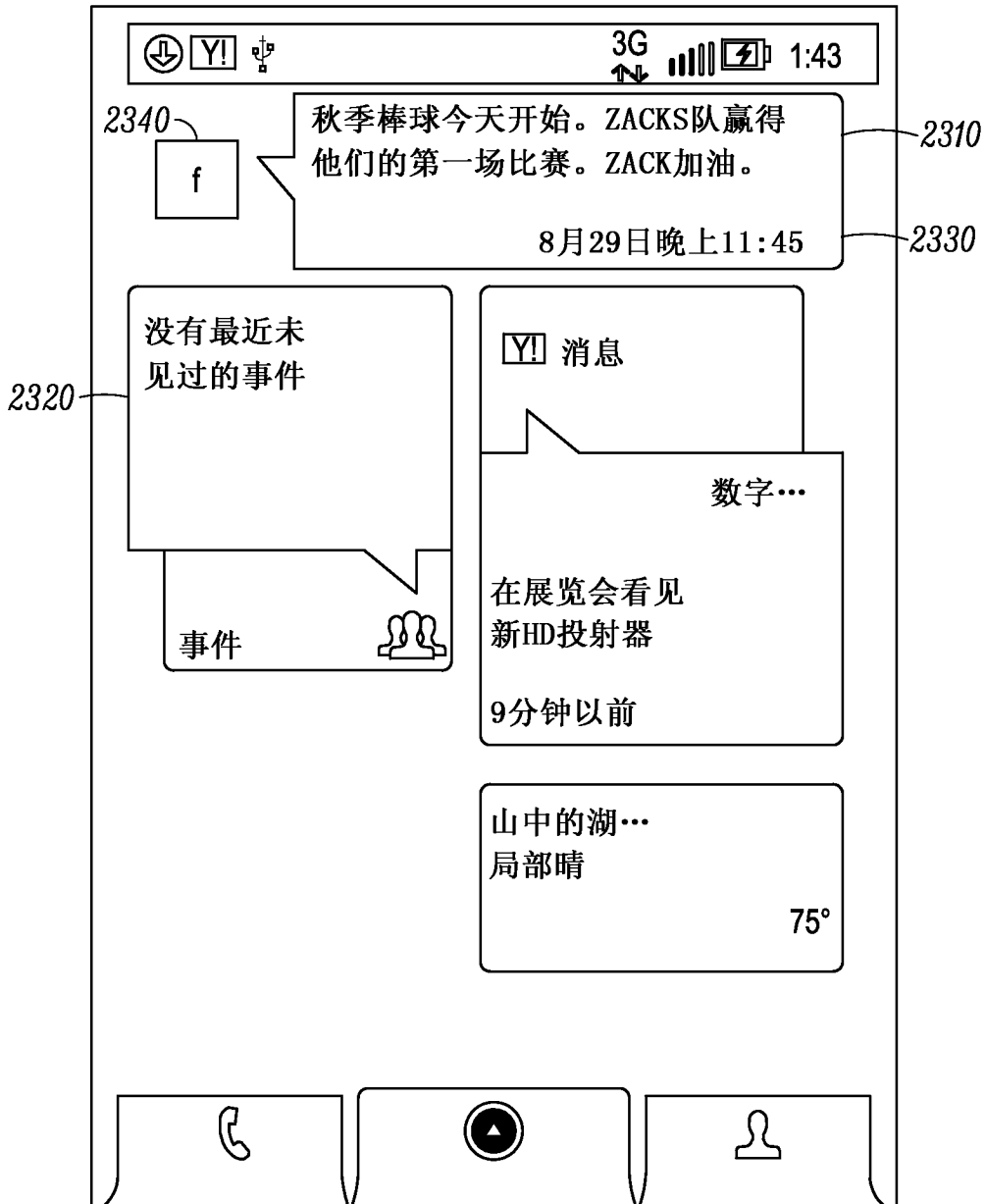


图 23

2400

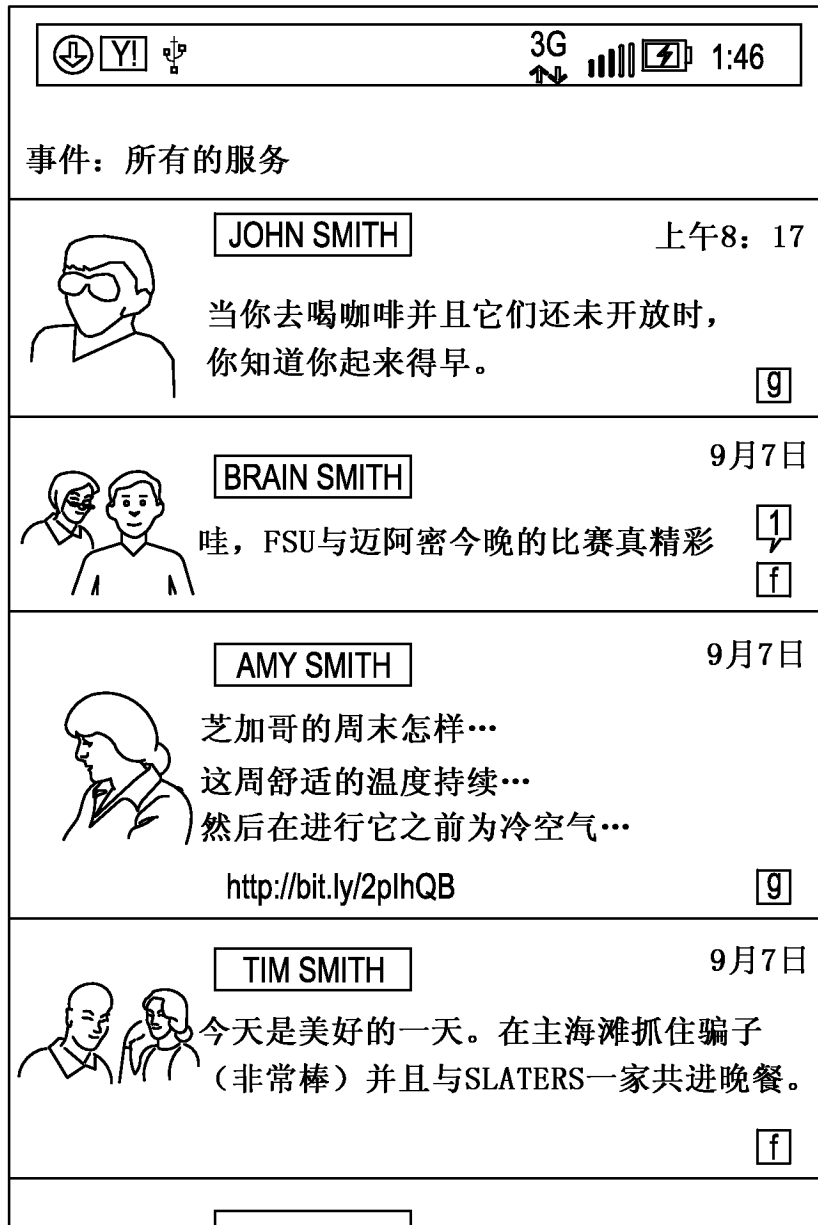


图 24

2500

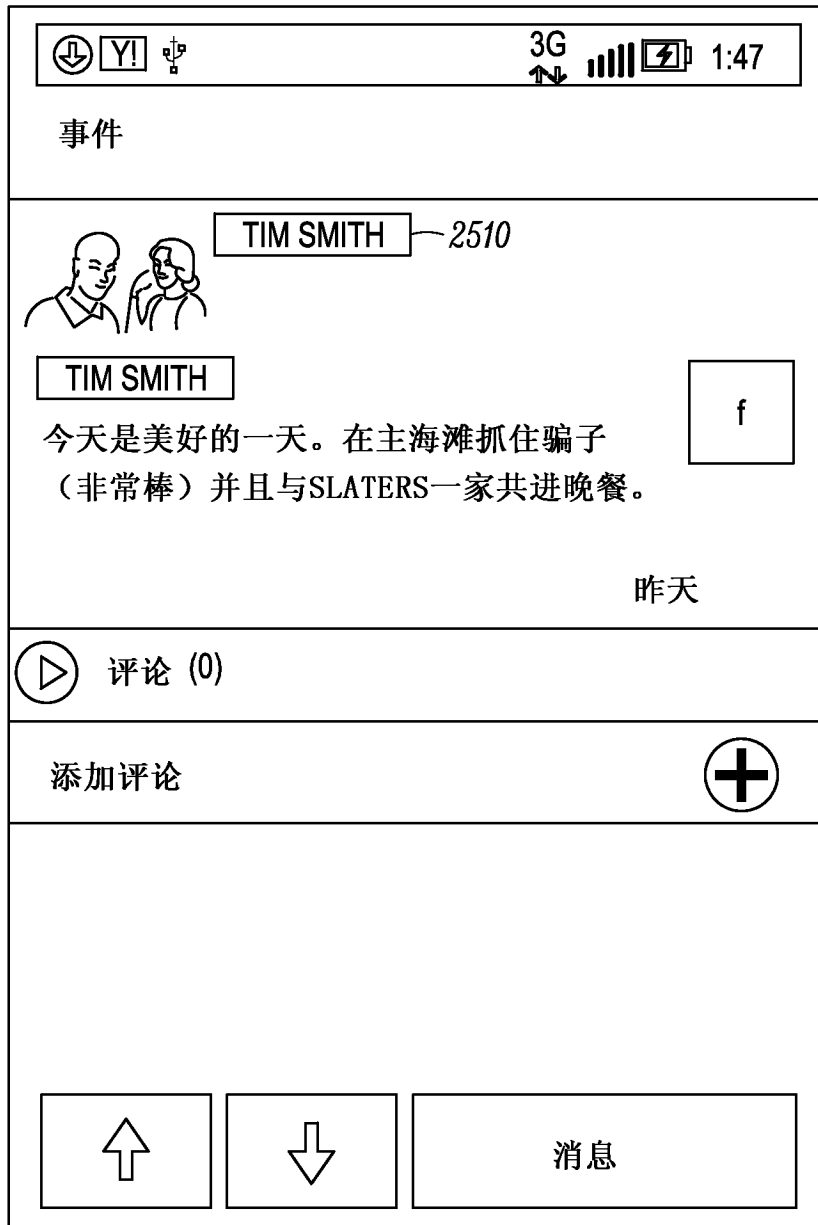


图 25

2600

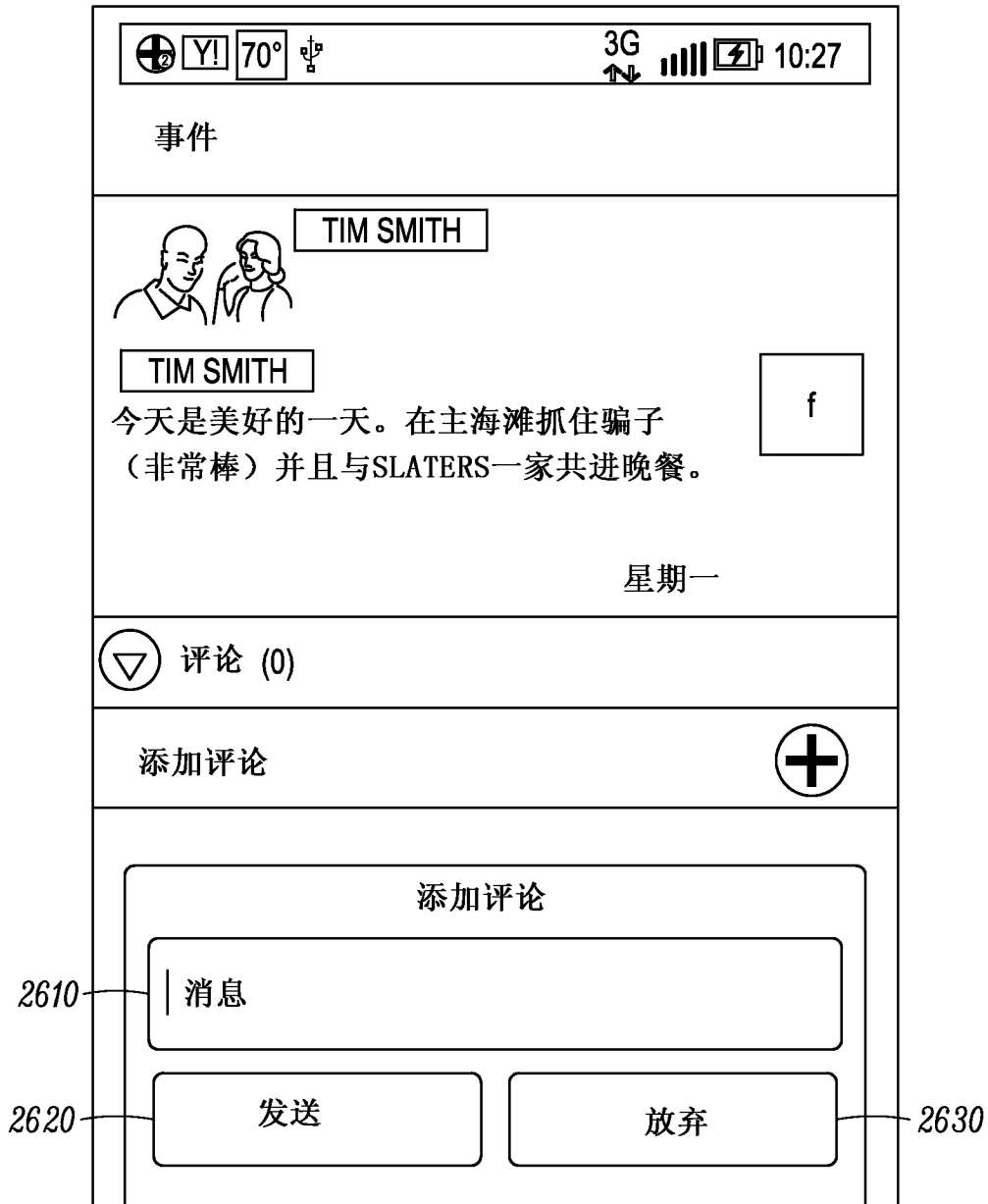


图 26

2700

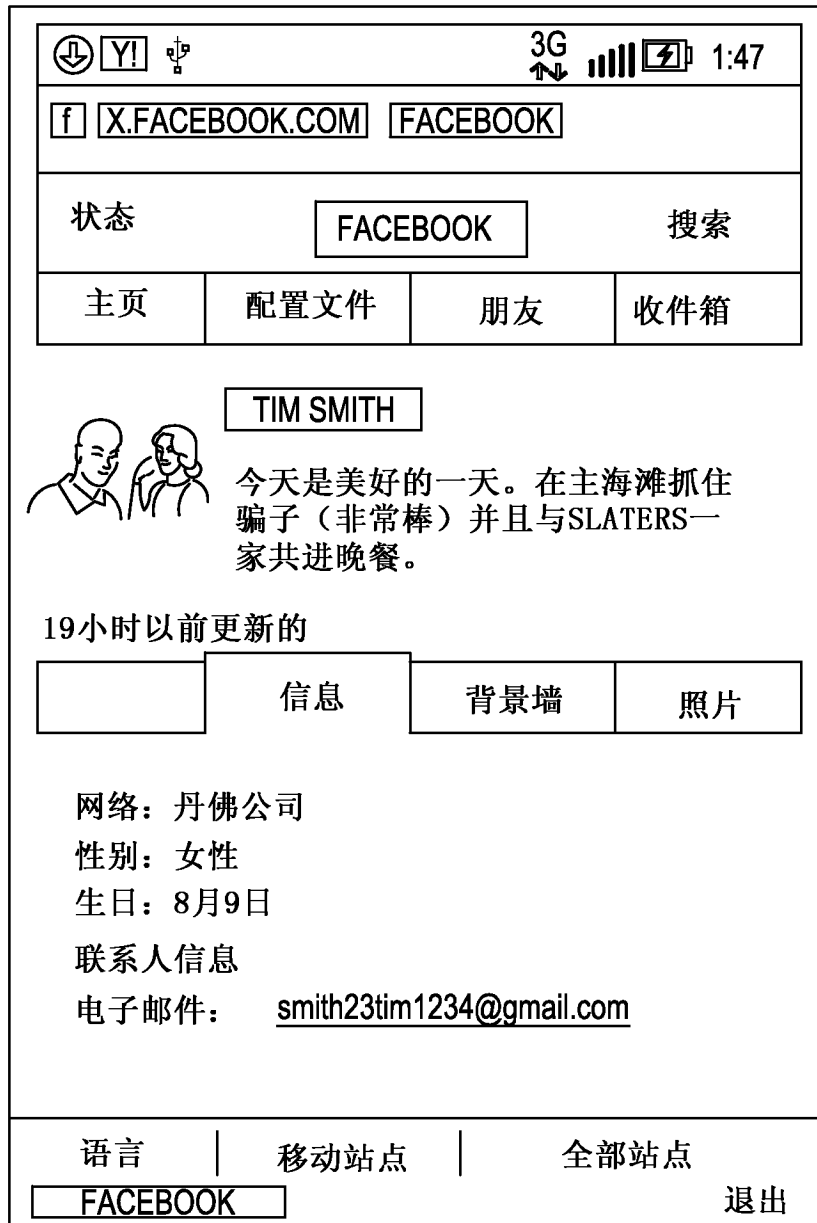


图 27

2800

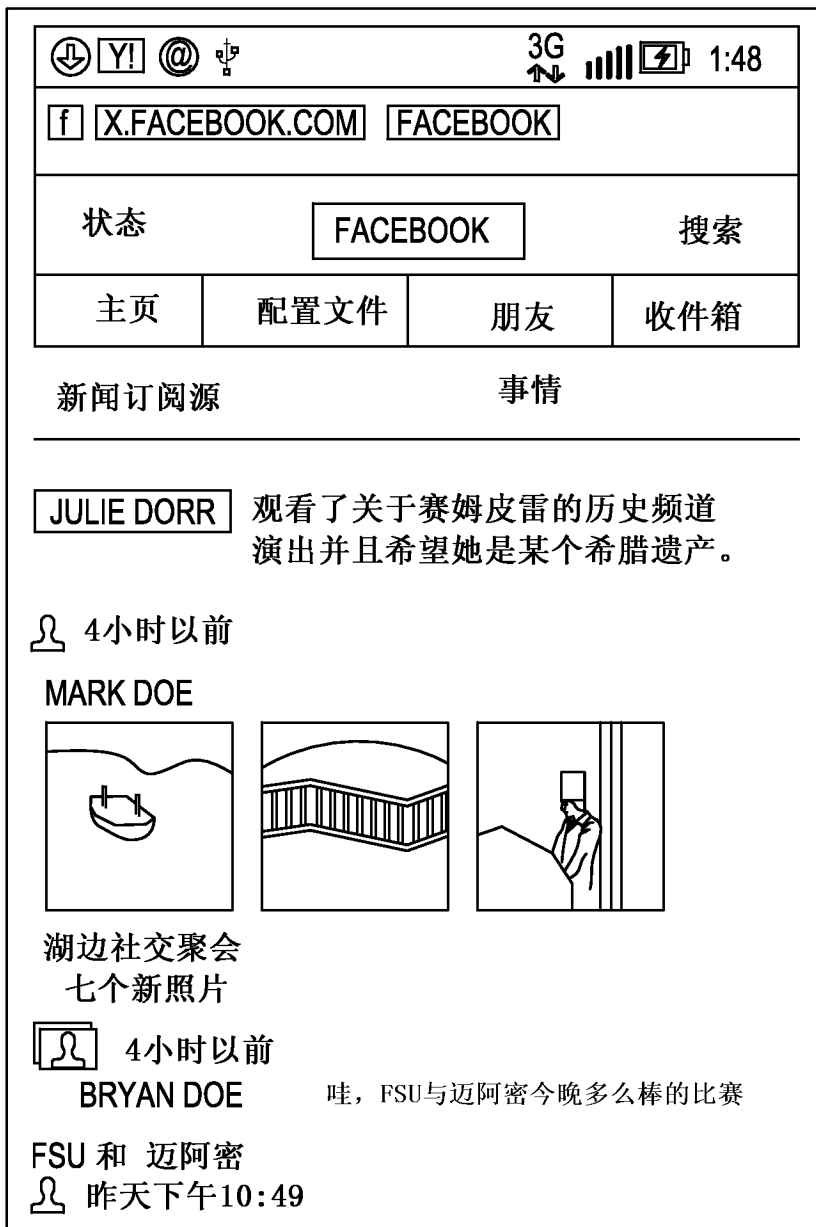


图 28

2900

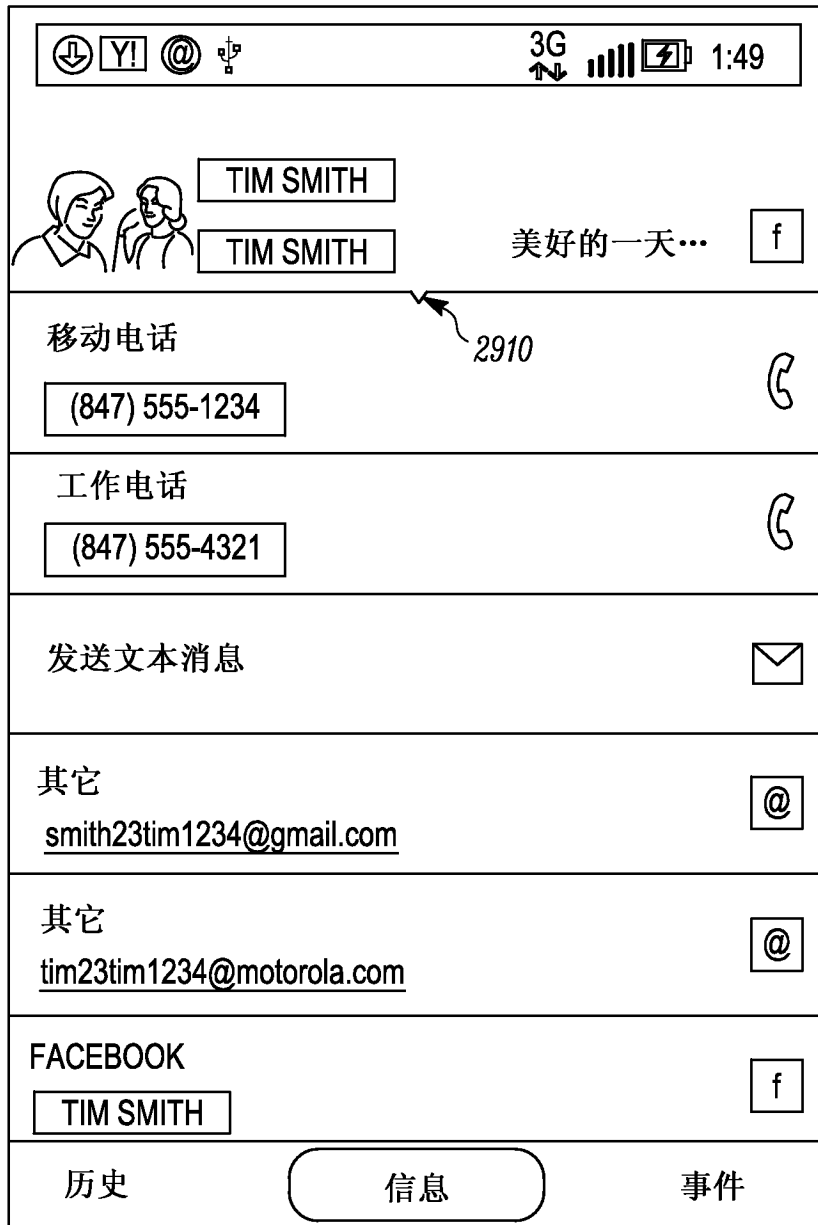


图 29

3000

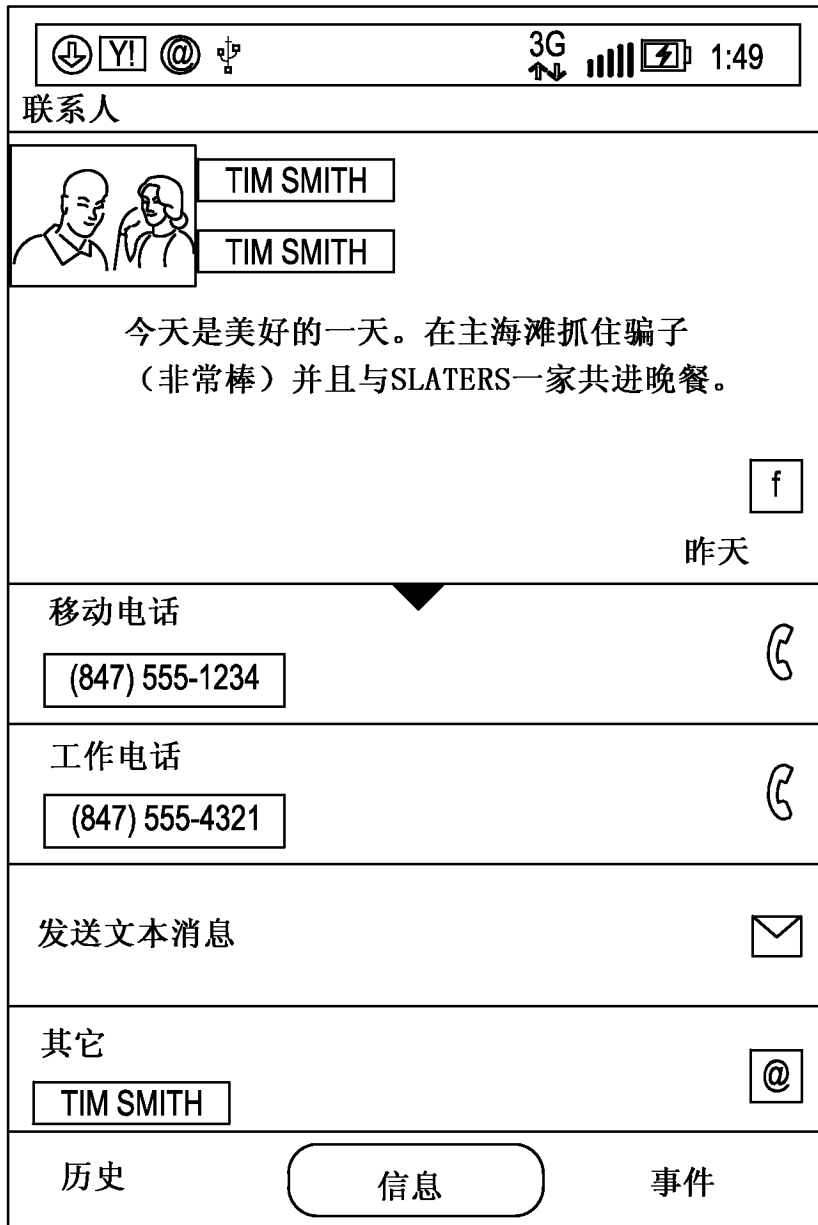


图 30

3100

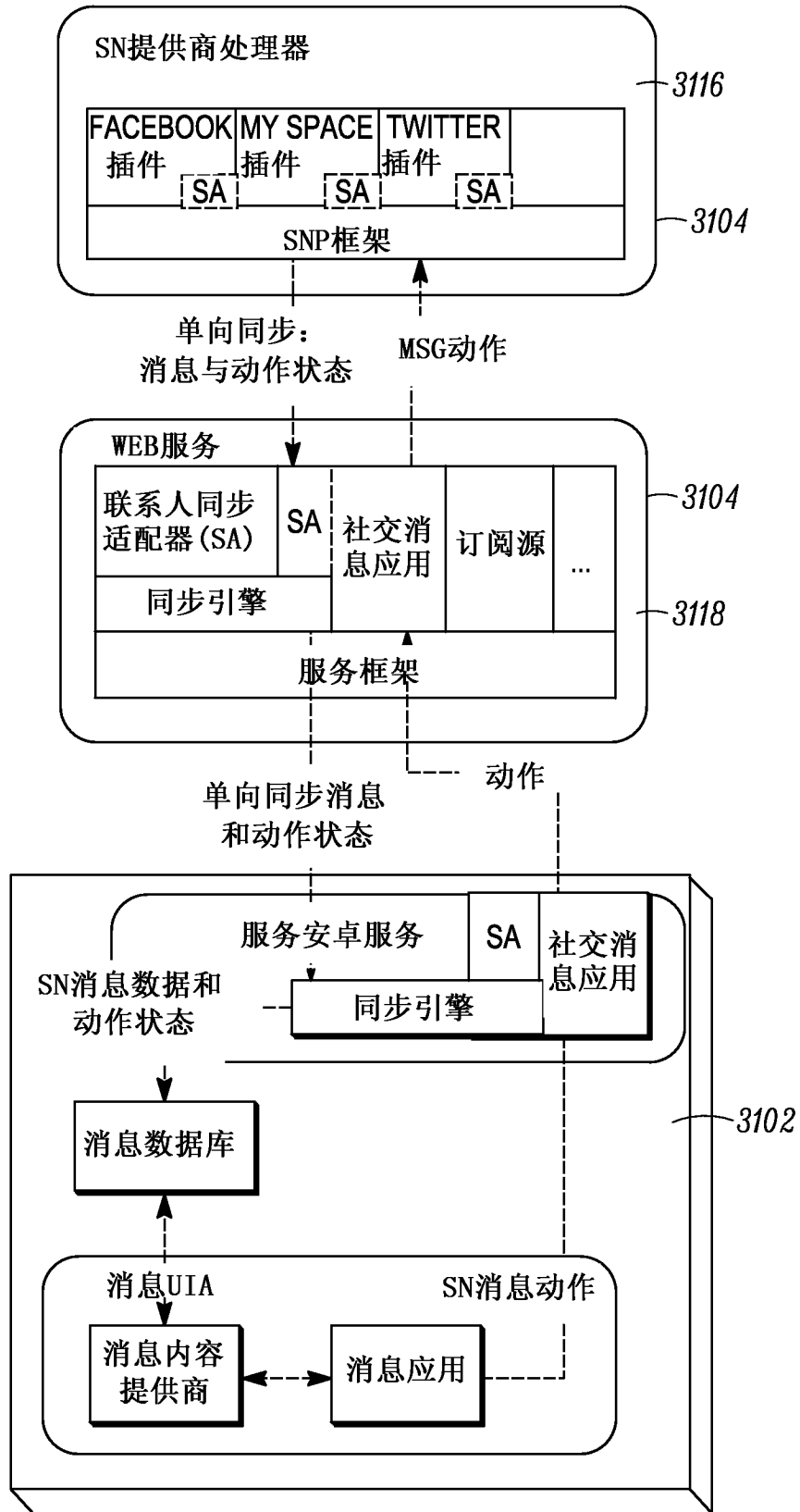


图 31

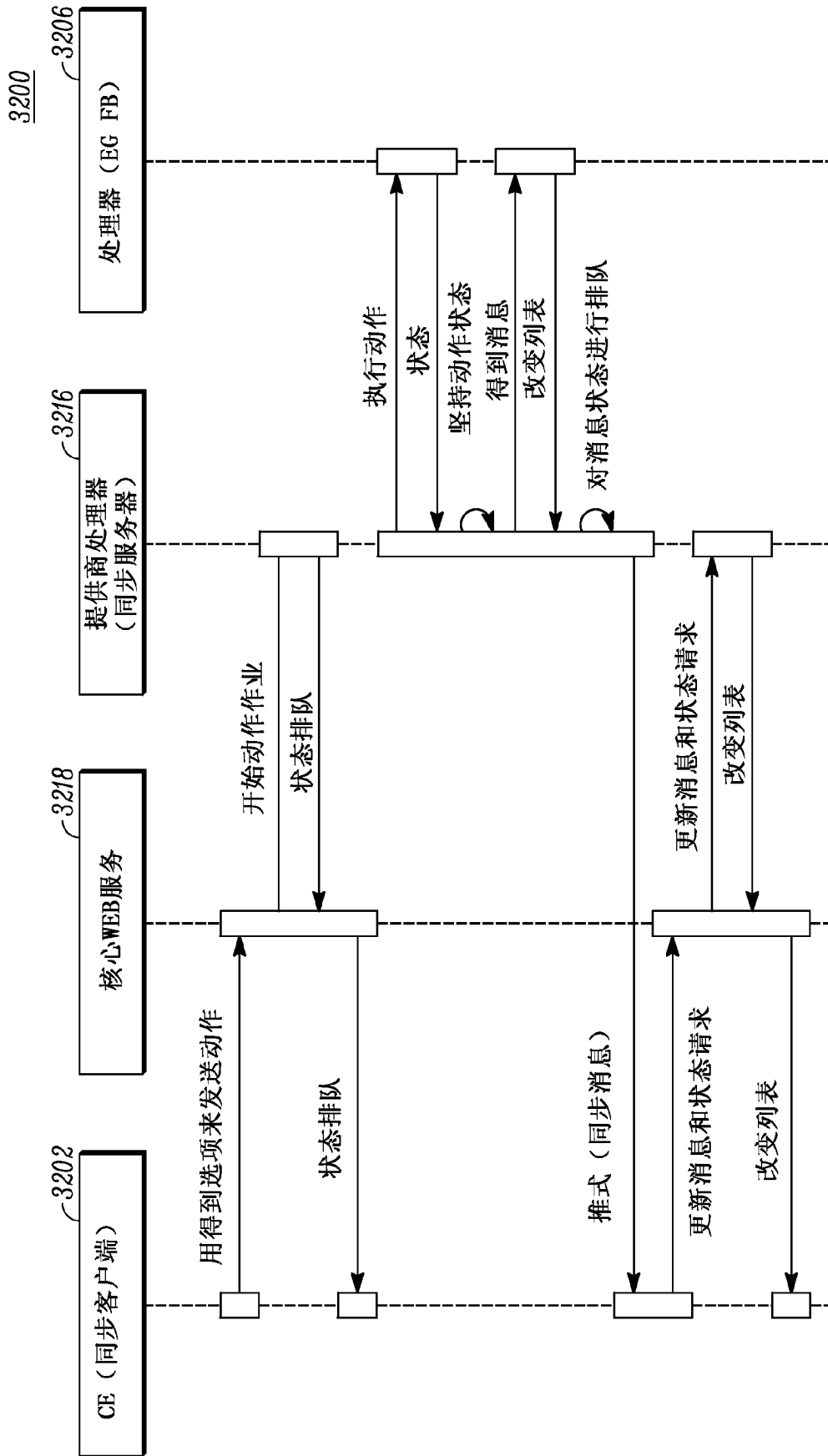


图 32