



(12) 发明专利申请

(10) 申请公布号 CN 106909404 A

(43) 申请公布日 2017. 06. 30

(21) 申请号 201510973799. 7

(22) 申请日 2015. 12. 22

(71) 申请人 博雅网络游戏开发(深圳)有限公司

地址 518057 广东省深圳市南山区中山园路

1001 号 TCL 产业园国际 E 城 D3 栋 9B-C

(72) 发明人 陈胜 林顺能

(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 何平

(51) Int. Cl.

G06F 9/445(2006. 01)

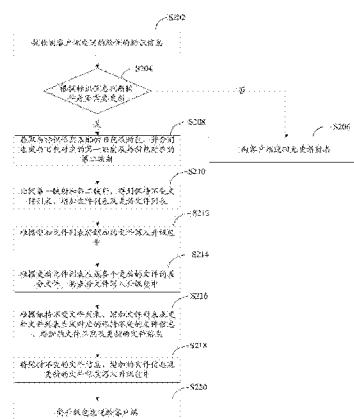
权利要求书3页 说明书12页 附图6页

(54) 发明名称

软件更新方法和系统

(57) 摘要

本发明涉及一种软件更新方法和系统。上述方法包括以下步骤:接收由客户端发送的软件的标识信息;根据标识信息判断软件是否需要更新;若软件需要更新,则获取与标识信息匹配的旧包及新包,并分别生成与旧包对应的第一映射及与新包对应的第二映射;比较第一映射和第二映射,得到保持不变文件列表、增加文件列表及更新文件列表;根据增加文件列表将增加的文件写入升级包中;根据更新文件列表生成各个更新的文件的差分文件并写入升级包中;根据保持不变文件列表、增加文件列表及更新文件列表生成对应的保持不变的文件信息、增加的文件信息及更新的文件信息并写入升级包中;将升级包发送给客户端。上述软件更新方法和系统,能够减少下载消耗的流量。



1. 一种软件更新方法,其特征在于,包括以下步骤:

接收由客户端发送的软件的标识信息;

根据所述标识信息判断所述软件是否需要更新;

若所述软件需要更新,则获取与所述标识信息匹配的旧包及新包,并分别生成与所述旧包对应的第一映射及与所述新包对应的第二映射;

比较所述第一映射和所述第二映射,得到保持不变文件列表、增加文件列表及更新文件列表;

根据所述增加文件列表将增加的文件写入升级包中;

根据所述更新文件列表生成各个更新的文件的差分文件,将所述差分文件写入所述升级包中;

根据所述保持不变文件列表、增加文件列表及更新文件列表生成对应的保持不变的文件信息、增加的文件信息及更新的文件信息;

将所述保持不变的文件信息、增加的文件信息及更新的文件信息写入所述升级包中;

将所述升级包发送给所述客户端。

2. 根据权利要求1所述的软件更新方法,其特征在于,所述标识信息包括标识符和版本号,所述获取与所述标识信息匹配的旧包及新包,并分别生成与所述旧包对应的第一映射及与所述新包对应的第二映射的步骤,具体包括:

获取与所述标识符及版本号匹配的旧包;

读取所述旧包中各个文件的文件名及对应的摘要信息;

根据所述旧包中各个文件的文件名及对应的摘要信息生成第一映射;

获取与所述标识符匹配的新包;

读取所述新包中各个文件的文件名及对应的摘要信息;

根据所述新包中各个文件的文件名及对应的摘要信息生成第二映射。

3. 根据权利要求1或2所述的软件更新方法,其特征在于,在所述将所述升级包发送给所述客户端的步骤之前,还包括:

获取所述新包的签名文件;

将所述签名文件写入所述升级包。

4. 一种软件更新方法,其特征在于,包括以下步骤:

获取软件的标识信息,并根据所述标识信息向服务端发送更新请求;

若所述软件需要更新,则下载升级包,并生成安装包;

根据所述升级包中的保持不变的文件信息从与所述标识信息匹配的旧包中获取对应的保持不变的文件,并将所述保持不变的文件写入所述安装包中;

根据所述升级包中的增加的文件信息从所述升级包中获取增加的文件,并将所述增加的文件写入所述安装包中;

根据所述升级包中的更新的文件信息从所述升级包中获取各个更新的文件的差分文件,将所述差分文件与所述旧包中各个对应的文件合并,并将合并后的文件写入所述安装包中;

调用预设接口安装所述安装包,完成所述软件的更新。

5. 根据权利要求4所述的软件更新方法,其特征在于,在所述调用预设接口完成所述软

件的更新的步骤之前,还包括:

获取所述升级包中的签名文件,并将所述签名文件写入所述安装包中;

读取所述旧包的渠道号,将所述渠道号写入所述安装包中,生成对应的渠道更新包。

6. 一种软件更新系统,其特征在于,包括:

接收模块,用于接收由客户端发送的软件的标识信息;

判断模块,用于根据所述标识信息判断所述软件是否需要更新;

生成映射模块,用于若所述软件需要更新,则获取与所述标识信息匹配的旧包及新包,并分别生成与所述旧包对应的第一映射及与所述新包对应的第二映射;

比较模块,用于比较所述第一映射和所述第二映射,得到保持不变文件列表、增加文件列表及更新文件列表;

写入模块,用于根据所述增加文件列表将增加的文件写入升级包中;

所述写入模块还用于根据所述更新文件列表生成各个更新的文件的差分文件,将所述差分文件写入所述升级包中;

生成文件信息模块,用于根据所述保持不变文件列表、增加文件列表及更新文件列表生成对应的保持不变的文件信息、增加的文件信息及更新的文件信息;

所述写入模块还用于将所述保持不变的文件信息、增加的文件信息及更新的文件信息写入所述升级包中;

发送模块,用于将所述升级包发送给所述客户端。

7. 根据权利要求6所述的软件更新系统,其特征在于,所述标识信息包括标识符和版本号,所述生成映射模块包括:

获取单元,用于获取与所述标识符及版本号匹配的旧包;

读取单元,用于读取所述旧包中各个文件的文件名及对应的摘要信息;

生成单元,用于根据所述旧包中各个文件的文件名及对应的摘要信息生成第一映射;

所述获取单元还用于获取与所述标识符匹配的新包;

所述读取单元还用于读取所述新包中各个文件的文件名及对应的摘要信息;

所述生成单元还用于根据所述新包中各个文件的文件名及对应的摘要信息生成第二映射。

8. 根据权利要求6或7所述的软件更新系统,其特征在于,所述软件更新系统还包括:

获取模块,用于获取所述新包的签名文件;

所述写入模块还用于将所述签名文件写入所述升级包。

9. 一种软件更新系统,其特征在于,包括:

发送模块,用于获取软件的标识信息,并根据所述标识信息向服务端发送更新请求;

下载模块,用于若所述软件需要更新,则下载升级包,并生成安装包;

写入模块,用于根据所述升级包中的保持不变的文件信息从与所述标识信息匹配的旧包中获取对应的保持不变的文件,并将所述保持不变的文件写入所述安装包中;

所述写入模块还用于根据所述升级包中的增加的文件信息从所述升级包中获取增加的文件,并将所述增加的文件写入所述安装包中;

所述写入模块还用于根据所述升级包中的更新的文件信息从所述升级包中获取各个更新的文件的差分文件,将所述差分文件与所述旧包中各个对应的文件合并,并将合并后

的文件写入所述安装包中；

安装模块,用于调用预设接口安装所述安装包,完成所述软件的更新。

10.根据权利要求9所述的软件更新系统,其特征在于,所述写入模块还用于获取所述升级包中的签名文件,并将所述签名文件写入所述安装包中；

读取模块,用于读取所述旧包的渠道号,将所述渠道号写入所述安装包中,生成对应的渠道更新包。

软件更新方法和系统

技术领域

[0001] 本发明涉及计算机技术领域,特别是涉及一种软件更新方法和系统。

背景技术

[0002] 当软件需要进行更新时,在传统方式中,一般是需要安装软件的终端,例如手机、平板电脑、手提电脑等从服务器上下载完整的最新的安装包,下载完成后再进行安装。但是随着软件的升级更新,完整的最新的安装包会越来越大,在进行下载时往往需要占据较大的带宽并消耗大量的流量,下载的时间也越来越长。

发明内容

[0003] 基于此,有必要提供一种软件更新方法,能够有效减少安装包下载时占据的带宽及消耗的流量,缩短下载时间。

[0004] 此外,还有必要提供一种软件更新系统,能够有效减少安装包下载时占据的带宽及消耗的流量,缩短下载时间。

[0005] 一种软件更新方法,包括以下步骤:

[0006] 接收由客户端发送的软件的标识信息;

[0007] 根据所述标识信息判断所述软件是否需要更新;

[0008] 若所述软件需要更新,则获取与所述标识信息匹配的旧包及新包,并分别生成与所述旧包对应的第一映射及与所述新包对应的第二映射;

[0009] 比较所述第一映射和所述第二映射,得到保持不变文件列表、增加文件列表及更新文件列表;

[0010] 根据所述增加文件列表将增加的文件写入升级包中;

[0011] 根据所述更新文件列表生成各个更新的文件的差分文件,将所述差分文件写入所述升级包中;

[0012] 根据所述保持不变文件列表、增加文件列表及更新文件列表生成对应的保持不变的文件信息、增加的文件信息及更新的文件信息;

[0013] 将所述保持不变的文件信息、增加的文件信息及更新的文件信息写入所述升级包中;

[0014] 将所述升级包发送给所述客户端。

[0015] 在其中一个实施例中,所述标识信息包括标识符和版本号,所述获取与所述标识信息匹配的旧包及新包,并分别生成与所述旧包对应的第一映射及与所述新包对应的第二映射的步骤,具体包括:

[0016] 获取与所述标识符及版本号匹配的旧包;

[0017] 读取所述旧包中各个文件的文件名及对应的摘要信息;

[0018] 根据所述旧包中各个文件的文件名及对应的摘要信息生成第一映射;

[0019] 获取与所述标识符匹配的新包;

- [0020] 读取所述新包中各个文件的文件名及对应的摘要信息；
- [0021] 根据所述新包中各个文件的文件名及对应的摘要信息生成第二映射。
- [0022] 在其中一个实施例中，在所述将所述升级包发送给所述客户端的步骤之前，还包括：
- [0023] 获取所述新包的签名文件；
- [0024] 将所述签名文件写入所述升级包。
- [0025] 一种软件更新方法，包括以下步骤：
- [0026] 获取软件的标识信息，并根据所述标识信息向服务端发送更新请求；
- [0027] 若所述软件需要更新，则下载升级包，并生成安装包；
- [0028] 根据所述升级包中的保持不变的文件信息从与所述标识信息匹配的旧包中获取对应的保持不变的文件，并将所述保持不变的文件写入所述安装包中；
- [0029] 根据所述升级包中的增加的文件信息从所述升级包中获取增加的文件，并将所述增加的文件写入所述安装包中；
- [0030] 根据所述升级包中的更新的文件信息从所述升级包中获取各个更新的文件的差分文件，将所述差分文件与所述旧包中各个对应的文件合并，并将合并后的文件写入所述安装包中；
- [0031] 调用预设接口安装所述安装包，完成所述软件的更新。
- [0032] 在其中一个实施例中，在所述调用预设接口完成所述软件的更新的步骤之前，还包括：
- [0033] 获取所述升级包中的签名文件，并将所述签名文件写入所述安装包中；
- [0034] 读取所述旧包的渠道号，将所述渠道号写入所述安装包中，生成对应的渠道更新包。
- [0035] 一种软件更新系统，包括：
- [0036] 接收模块，用于接收由客户端发送的软件的标识信息；
- [0037] 判断模块，用于根据所述标识信息判断所述软件是否需要更新；
- [0038] 生成映射模块，用于若所述软件需要更新，则获取与所述标识信息匹配的旧包及新包，并分别生成与所述旧包对应的第一映射及与所述新包对应的第二映射；
- [0039] 比较模块，用于比较所述第一映射和所述第二映射，得到保持不变文件列表、增加文件列表及更新文件列表；
- [0040] 写入模块，用于根据所述增加文件列表将增加的文件写入升级包中；
- [0041] 所述写入模块还用于根据所述更新文件列表生成各个更新的文件的差分文件，将所述差分文件写入所述升级包中；
- [0042] 生成文件信息模块，用于根据所述保持不变文件列表、增加文件列表及更新文件列表生成对应的保持不变的文件信息、增加的文件信息及更新的文件信息；
- [0043] 所述写入模块还用于将所述保持不变的文件信息、增加的文件信息及更新的文件信息写入所述升级包中；
- [0044] 发送模块，用于将所述升级包发送给所述客户端。
- [0045] 在其中一个实施例中，所述标识信息包括标识符和版本号，所述生成映射模块包括：

- [0046] 获取单元,用于获取与所述标识符及版本号匹配的旧包;
- [0047] 读取单元,用于读取所述旧包中各个文件的文件名及对应的摘要信息;
- [0048] 生成单元,用于根据所述旧包中各个文件的文件名及对应的摘要信息生成第一映射;
- [0049] 所述获取单元还用于获取与所述标识符匹配的新包;
- [0050] 所述读取单元还用于读取所述新包中各个文件的文件名及对应的摘要信息;
- [0051] 所述生成单元还用于根据所述新包中各个文件的文件名及对应的摘要信息生成第二映射。
- [0052] 在其中一个实施例中,所述软件更新系统还包括:
- [0053] 获取模块,用于获取所述新包的签名文件;
- [0054] 所述写入模块还用于将所述签名文件写入所述升级包。
- [0055] 一种软件更新系统,包括:
- [0056] 发送模块,用于获取软件的标识信息,并根据所述标识信息向服务端发送更新请求;
- [0057] 下载模块,用于若所述软件需要更新,则下载升级包,并生成安装包;
- [0058] 写入模块,用于根据所述升级包中的保持不变的文件信息从与所述标识信息匹配的旧包中获取对应的保持不变的文件,并将所述保持不变的文件写入所述安装包中;
- [0059] 所述写入模块还用于根据所述升级包中的增加的文件信息从所述升级包中获取增加的文件,并将所述增加的文件写入所述安装包中;
- [0060] 所述写入模块还用于根据所述升级包中的更新的文件信息从所述升级包中获取各个更新的文件的差分文件,将所述差分文件与所述旧包中各个对应的文件合并,并将合并后的文件写入所述安装包中;
- [0061] 安装模块,用于调用预设接口安装所述安装包,完成所述软件的更新。
- [0062] 在其中一个实施例中,所述写入模块还用于获取所述升级包中的签名文件,并将所述签名文件写入所述安装包中;
- [0063] 读取模块,用于读取所述旧包的渠道号,将所述渠道号写入所述安装包中,生成对应的渠道更新包。
- [0064] 上述软件更新方法和系统,服务端通过比较软件的旧包及新包的映射后得到不同的文件列表,在升级包中只写入增加的文件及更新文件的差分文件,能够大大减少升级包的数据量,当客户端在进行下载时,无需占用大量的带宽和流量,节约资源,并有效提高了下载速度,节省时间。
- [0065] 此外,客户端能够读取软件的旧有的渠道号并生成对应的渠道更新包,当软件拥有多个渠道进行更新时,服务端只需保存各个版本号对应的一个安装包即可,无需针对不同的渠道生成不同的安装包,简化了软件的多渠道版本的更新步骤,并有效降低了服务端的维护成本。

附图说明

- [0066] 图1为一个实施例中软件更新方法的系统架构图;
- [0067] 图2为一个实施例中软件更新方法的流程示意图;

- [0068] 图3为一个实施例中服务端生成第一映射及第二映射的流程示意图；
- [0069] 图4为另一个实施例中软件更新方法的流程示意图；
- [0070] 图5为一个实施例中客户端生成渠道更新包的流程示意图；
- [0071] 图6为一个实施例中软件更新系统的结构示意图；
- [0072] 图7为一个实施例中生成映射模块的内部结构示意图；
- [0073] 图8为另一个实施例中软件更新系统的结构示意图；
- [0074] 图9为又一个实施例中软件更新系统的结构示意图。

具体实施方式

[0075] 为了使本发明的目的、技术方案及优点更加清楚明白，以下结合附图及实施例，对本发明进行进一步详细说明。应当理解，此处所描述的具体实施例仅用以解释本发明，并不用于限定本发明。

[0076] 如图1所示，图1为一个实施例中软件更新方法的系统架构图，客户端10与服务端20实现双向通信。

[0077] 客户端10获取软件的标识信息，并根据该标识信息向服务端20发送更新请求。服务端20接收由客户端10发送的标识信息，根据标识信息判断软件是否需要更新。若该软件需要更新，则服务端20获取与标识信息匹配的旧包与新包，并分别生成与旧包对应的第一映射和与新包对应的第二映射。服务端20比较第一映射与第二映射，得到保持不变文件列表、增加文件列表及更新文件列表。服务端20根据增加文件列表将增加的文件写入升级包中，根据更新文件列表生成各个更新的文件的差分文件，并将差分文件写入升级包中，再根据保持不变文件列表、增加文件列表及更新文件列表生成对应的保持不变的文件信息、增加的文件信息及更新的文件信息，然后将保持不变的文件信息、增加的文件信息及更新的文件信息写入升级包中。服务端20将升级包发送给客户端10。客户端10下载该升级包，并生成安装包。客户端10根据升级包中的保持不变的文件信息从与标识信息匹配的旧包中获取对应的保持不变的文件，并将保持不变的文件写入安装包中。客户端10根据升级包中的增加的文件信息从升级包中获取增加的文件，并将增加的文件写入安装包中。客户端10根据升级包中的更新的文件信息从升级包中获取各个更新的文件的差分文件，将差分文件与旧包中各个对应的文件合并，并将合并后的文件写入安装包中。然后客户端10调用预设接口安装该安装包，完成软件的更新。

[0078] 如图2所示，一种软件更新方法，包括服务端与客户端，从服务端进行描述，包括以下步骤：

[0079] 步骤S202，接收由客户端发送的软件的标识信息。

[0080] 具体的，客户端向服务端发送带有软件的标识信息的更新请求，软件的标识信息包括标识符和版本号等信息。

[0081] 步骤S204，根据标识信息判断软件是否需要更新，若是，则执行步骤S208，若否，则执行步骤S206。

[0082] 具体的，服务端接收由客户端发送的软件的标识信息后，可在软件数据库中查找与该标识符匹配的软件，并进一步根据版本号判断是否存在更新版本，若存在更新版本，则该软件需要进行更新，若不存在更新版本，则向客户端返回无更新标志。

[0083] 步骤S206,向客户端返回无更新标志。

[0084] 步骤S208,获取与标识信息匹配的旧包及新包,并分别生成与旧包对应的第一映射及与新包对应的第二映射。

[0085] 具体的,若软件需要进行更新,则获取与标识信息匹配的旧包及新包,其中,旧包指的是与标识信息中的版本号对应的旧的安装包,而新包指的是该软件的最新版本的安装包。

[0086] 在一个实施例中,如图3所示,步骤获取与标识信息匹配的旧包及新包,并分别生成与旧包对应的第一映射及与新包对应的第二映射,具体包括以下步骤:

[0087] 步骤S302,获取与标识符及版本号匹配的旧包。

[0088] 具体的,服务端可先根据标识符在软件数据库中查指到对应的软件,再根据版本号获取对应的旧包。

[0089] 步骤S304,读取旧包中各个文件的文件名及对应的摘要信息。

[0090] 具体的,服务端获取旧包后,可将其解压至缓存目录中,再读取其中META-INF目录下的MANIFEST.MF文件,该MANIFEST.MF文件记录有旧包中各个文件的文件名及对应的SHA1 (Secure Hash Algorithm,安全哈希算法)摘要信息。安全哈希算法主要适用于数字签名标准(Digital Signature Standard DSS)里面定义的数字签名算法(Digital Signature Algorithm DSA),SHA1摘要信息可以用于检验数据的完整性,两个不同的文件其产生的SHA1摘要信息也不同。

[0091] 步骤S306,根据旧包中各个文件的文件名及对应的摘要信息生成第一映射。

[0092] 具体的,服务端读取旧包中各个文件的文件名及对应的摘要信息后,可生成第一映射,旧包中各个文件的文件名可作为第一映射的键,而与文件名对应的SHA1摘要信息则作为第一映射的值。

[0093] 步骤S308,获取与标识符匹配的新包。

[0094] 具体的,服务端可根据标识符在软件数据库中查指到对应的软件,并获取该软件最新版本的安装包,即新包。

[0095] 步骤S310,读取新包中各个文件的文件名及对应的摘要信息。

[0096] 具体的,服务端获取新包后,可将其解压至缓存目录中,再读取其中META-INF目录下的MANIFEST.MF文件,该MANIFEST.MF文件记录有新包中各个文件的文件名及对应的SHA1摘要信息。

[0097] 步骤S312,根据新包中各个文件的文件名及对应的摘要信息生成第二映射。

[0098] 具体的,新包中各个文件的文件名可作为第二映射的键,而与文件名对应的SHA1摘要信息则作为第二映射的值。

[0099] 步骤S210,比较第一映射和第二映射,得到保持不变文件列表、增加文件列表及更新文件列表。

[0100] 具体的,服务端通过比较第一映射和第二映射中的键和值是否相同,即能比较旧包与新包中的文件是否相同,可得到保持不变文件列表、增加文件列表及更新文件列表三个文件列表。同时存在于第一映射和第二映射中的文件名,其对应的摘要信息也相同,则该文件即为保持不变的文件;同时存在于第一映射和第二映射中的文件名,但其对应的摘要信息不同,则该文件即为更新的文件;不存在于第一映射只存在于第二映射中的文件名,则

该文件即为增加的文件。根据比较获得的结果,可生成三个不同的文件列表,三个文件列表中分别包含保持不变的文件的文件名、增加的文件文件名及更新的文件的文件名等信息。

[0101] 步骤S212,根据增加文件列表将增加的文件写入升级包中。

[0102] 具体的,服务端可预先建立一个与该软件的安装包格式相同的空的升级包,例如该软件的安装包格式为apk(Android Package,安卓安装包),即可预先建立一个空的以apk为后缀的压缩文件作为升级包。服务端根据增加文件列表中的文件名将新包中对应的增加的文件压缩写入升级包中。在其它实施例中,也可以根据流程及时建立空的升级包。例如,在步骤S204判断软件需要更新时建立空的升级包,或者步骤S212中先建立空的升级包,再根据增加文件列表将增加的文件写入升级包中。

[0103] 步骤S214,根据更新文件列表生成各个更新的文件的差分文件,将差分文件写入升级包中。

[0104] 具体的,服务端根据更新文件列表中的文件名先获取新包中更新的文件,同时获取旧包中对应的旧版文件,并利用bsdiff等二进制差量工具,将更新的文件及旧版文件进行二进制对比,生成两个文件的差分文件,然后将差分文件压缩写入升级包中。在对比第一映射及第二映射的基础上,通过二进制差量工具生成每个更新的文件的差分文件,可大大减小升级包中更新文件的数据量,以此节省客户端下载时所需的流量,并缩短时长,且又可避免单独采用增量升级时MD5(Message Digest Algorithm MD5,消息摘要算法第五版)校验不通过导致无法完成增量升级的问题。

[0105] 步骤S216,根据保持不变文件列表、增加文件列表及更新文件列表生成对应的保持不变的文件信息、增加的文件信息及更新的文件信息。

[0106] 具体的,服务端可根据保持不变文件列表、增加文件列表及更新文件列表生成对应的保持不变的文件信息、增加的文件信息及更新的文件信息的JSON(JavaScript Object Notation,JavaScript对象表示法)字符串,保持不变的文件信息、增加的文件信息及更新的文件信息中即包含对应的文件列表的内容。

[0107] 步骤S218,将保持不变的文件信息、增加的文件信息及更新的文件信息写入升级包中。

[0108] 具体的,服务端可将保持不变的文件信息、增加的文件信息及更新的文件信息的JSON字符串压缩写入升级包中,方便客户端根据该文件信息进行软件更新。

[0109] 步骤S220,将升级包发送给客户端。

[0110] 具体的,服务端将升级包发送给客户端,使得客户端完成软件的更新。

[0111] 上述软件更新方法,服务端通过比较软件的旧包及新包的映射后得到不同的文件列表,在升级包中只写入增加的文件及更新文件的差分文件,能够大大减少升级包的数据量,当客户端在进行下载时,无需占用大量的带宽和流量,节约资源,并有效提高了下载速度,节省时间。

[0112] 在一个实施例中,上述软件更新方法,在步骤将升级包发送给客户端之前,还包括:获取新包的签名文件,将签名文件写入升级包。

[0113] 具体的,许多不同的安装包中都有其独特的签名文件,该签名文件中包含有包名、厂商的数字证书等信息,签名文件中的信息可作为各个安装包的标识。服务端可获取新包

的签名文件,并将签名文件压缩写入升级包中,使升级包带有新包的标识,即可容易识别该升级包对应的软件及版本号。

[0114] 上述软件更新方法,服务端通过比较软件的旧包及新包的映射后得到不同的文件列表,在升级包中只写入增加的文件及更新文件的差分文件,能够大大减少升级包的数据量,当客户端在进行下载时,无需占用大量的带宽和流量,节约资源,并有效提高了下载速度,节省时间。

[0115] 如图4所示,一种软件更新方法,包括服务端与客户端,从客户端进行描述,包括以下步骤:

[0116] 步骤S402,获取软件的标识信息,并根据标识信息向服务端发送更新请求。

[0117] 具体的,客户端可获取软件的标识符、版本号等标识信息,并向服务端发送带有该标识信息的更新请求,由服务端根据该标识信息判断软件是否需要更新。当软件不需要进行更新时,即客户端若接收到由服务端发送的无更新标志,可提示软件当前版本即为最新版本,无需进行更新。

[0118] 步骤S404,若软件需要更新,则下载升级包,并生成安装包。

[0119] 具体的,若软件需要进行更新,则客户端从服务端下载对应的升级包,并生成一个与升级包格式相同的空的安装包。

[0120] 步骤S406,根据升级包中的保持不变的文件信息从与标识信息匹配的旧包中获取对应的保持不变的文件,并将保持不变的文件写入安装包中。

[0121] 具体的,客户端下载升级包后,可从升级包中解压出保持不变的文件信息、增加的文件信息及更新的文件信息的JSON字符串。客户端可读取保持不变的文件信息,并根据该保持不变的文件信息从旧包中解压出对应的保持不变的文件压缩写入到安装包中,其中,旧包即为客户端当前版本软件对应的安装包。

[0122] 步骤S408,根据升级包中的增加的文件信息从升级包中获取增加的文件,并将增加的文件写入安装包中。

[0123] 具体的,客户端可从升级包中读取增加的文件信息,并根据增加的文件信息从升级包中解压出对应的增加的文件,再将增加的文件压缩写入安装包中。

[0124] 步骤S410,根据升级包中的更新的文件信息从升级包中获取各个更新的文件的差分文件,将差分文件与旧包中各个对应的文件合并,并将合并后的文件写入安装包中。

[0125] 具体的,客户端可从升级包中读取更新的文件信息,再根据更新的文件信息从升级包中解压出对应的各个更新的文件的差分文件,并根据更新的文件信息从旧包中解压出对应的旧版文件。客户端可利用bspatch等二进制差分合并工具将差分文件及对应的旧版文件进行合并,并将合并后的文件压缩写入安装包中。

[0126] 步骤S412,调用预设接口安装安装包,完成软件的更新。

[0127] 具体的,安装包写入完成后,客户端可调用预设接口进行安装,完成软件的更新。

[0128] 上述软件更新方法,服务端通过比较软件的旧包及新包的映射后得到不同的文件列表,在升级包中只写入增加的文件及更新文件的差分文件,能够大大减少升级包的数据量,当客户端在进行下载时,无需占用大量的带宽和流量,节约资源,并有效提高了下载速度,节省时间。

[0129] 如图5所示,在一个实施例中,在步骤S412调用预设接口安装安装包,完成软件的

更新之前,还包括:

[0130] 步骤S502,获取升级包中的签名文件,并将签名文件写入安装包中。

[0131] 具体的,客户端可从升级包中解压出签名文件,并将签名文件压缩写入安装包中,签名文件中包含有包名、厂商的数字证书等信息,签名文件中的信息可作为安装包的标识。

[0132] 步骤S504,读取旧包的渠道号,将渠道号写入安装包中,生成对应的渠道更新包。

[0133] 具体的,当软件拥有多渠道更新时,不同的渠道的安装包中携带有不同的渠道号,例如不同渠道的apk文件,其渠道号通常写入文件末尾的EOCD(End of central directory record)字段中。客户端可读取旧包的渠道号,并将渠道号写入安装包的特定字段中,生成对应的渠道更新包,该特定字段需满足可修改且不会影响安装包的正确打开及使用。例如apk文件的EOCD字段中有一个Comment字段,Comment字段通常位于文件的末尾,且其长度不固定,Comment字段的内容可任意写入并修改,只需同时改变对应的长度即不影响apk文件的正确打开及使用。因此,客户端可从软件对应的旧的apk文件的EOCD字段中读取渠道号,并将渠道号写入安装包的EOCD字段的Comment字段中,生成对应的渠道更新包。

[0134] 上述软件更新方法,客户端能够读取软件的旧有的渠道号并生成对应的渠道更新包,当软件拥有多渠道进行更新时,服务端只需保存各个版本号对应的一个安装包即可,无需针对不同的渠道生成不同的安装包,简化了软件的多渠道版本的更新步骤,并有效降低了服务端的维护成本。

[0135] 如图6所示,一种软件更新系统,包括接收模块610、判断模块620、生成映射模块630、比较模块640、写入模块650、生成文件信息模块660和发送模块670。

[0136] 接收模块610,用于接收由客户端发送的软件的标识信息。

[0137] 具体的,客户端向服务端发送带有软件的标识信息的更新请求,软件的标识信息包括标识符和版本号等信息。

[0138] 判断模块620,用于根据标识信息判断软件是否需要更新。

[0139] 具体的,服务端接收由客户端发送的软件的标识信息后,可在软件数据库中查找与该标识符匹配的软件,并进一步根据版本号判断是否存在更新版本,若存在更新版本,则该软件需要进行更新,若不存在更新版本,则向客户端返回无更新标志。

[0140] 生成映射模块630,用于若该软件需要更新,则获取与标识信息匹配的旧包及新包,并分别生成与旧包对应的第一映射及与新包对应的第二映射。

[0141] 具体的,若软件需要进行更新,则获取与标识信息匹配的旧包及新包,其中,旧包指的是与标识信息中的版本号对应的旧的安装包,而新包指的是该软件的最新版本的安装包。

[0142] 如图7所示,在一个实施例中,生成映射模块630包括获取单元632、读取单元634和生成单元636。

[0143] 获取单元632,用于获取与标识符及版本号匹配的旧包。

[0144] 具体的,服务端可先根据标识符在软件数据库中查指到对应的软件,再根据版本号获取对应的旧包。

[0145] 读取单元634,用于读取旧包中各个文件的文件名及对应的摘要信息。

[0146] 具体的,服务端获取旧包后,可将其解压至缓存目录中,再读取其中META-INF目录下的MANIFEST.MF文件,该MANIFEST.MF文件记录有旧包中各个文件的文件名及对应的SHA1

摘要信息。安全哈希算法主要适用于数字签名标准里面定义的数字签名算法,SHA1摘要信息可以用于检验数据的完整性,两个不同的文件其产生的SHA1摘要信息也不同。

[0147] 生成单元636,用于根据旧包中各个文件的文件名及对应的摘要信息生成第一映射。

[0148] 具体的,服务端读取旧包中各个文件的文件名及对应的摘要信息后,可生成第一映射,旧包中各个文件的文件名可作为第一映射的键,而与文件名对应的SHA1摘要信息则作为第一映射的值。

[0149] 获取单元632还用于获取与标识符匹配的新包。

[0150] 具体的,服务端可根据标识符在软件数据库中查指到对应的软件,并获取该软件最新版本的安装包,即新包。

[0151] 读取单元634还用于读取新包中各个文件的文件名及对应的摘要信息。

[0152] 具体的,服务端获取新包后,可将其解压至缓存目录中,再读取其中META-INF目录下的MANIFEST.MF文件,该MANIFEST.MF文件记录有新包中各个文件的文件名及对应的SHA1摘要信息。

[0153] 生成单元636还用于根据新包中各个文件的文件名及对应的摘要信息生成第二映射。

[0154] 具体的,新包中各个文件的文件名可作为第二映射的键,而与文件名对应的SHA1摘要信息则作为第二映射的值。

[0155] 比较模块640,用于比较第一映射和第二映射,得到保持不变文件列表、增加文件列表及更新文件列表。

[0156] 具体的,服务端通过比较第一映射和第二映射中的键和值是否相同,即能比较旧包与新包中的文件是否相同,可得到保持不变文件列表、增加文件列表及更新文件列表三个文件列表。同时存在于第一映射和第二映射中的文件名,其对应的摘要信息也相同,则该文件即为保持不变的文件;同时存在于第一映射和第二映射中的文件名,但其对应的摘要信息不同,则该文件即为更新的文件;不存在于第一映射只存在于第二映射中的文件名,则该文件即为增加的文件。根据比较获得的结果,可生成三个不同的文件列表,三个文件列表中分别包含保持不变的文件的文件名、增加的文件文件名及更新的文件的文件名等信息。

[0157] 写入模块650,用于根据增加文件列表将增加的文件写入升级包中。

[0158] 具体的,服务端可预先建立一个与该软件的安装包格式相同的空的升级包,例如该软件的安装包格式为apk,即可预先建立一个空的以apk为后缀的压缩文件作为升级包。服务端根据增加文件列表中的文件名将新包中对应的增加的文件压缩写入升级包中。在其它实施例中,也可以根据流程及时建立空的升级包。例如,在判断出软件需要更新时建立空的升级包,或是需要写入增加的文件时再建立空的升级包。

[0159] 写入模块650还用于根据更新文件列表生成各个更新的文件的差分文件,将差分文件写入升级包中。

[0160] 具体的,服务端根据更新文件列表中的文件名先获取新包中更新的文件,同时获取旧包中对应的旧版文件,并利用bsdiff等二进制差量工具,将更新的文件及旧版文件进行二进制对比,生成两个文件的差分文件,然后将差分文件压缩写入升级包中。在对比第一

映射及第二映射的基础上,通过二进制增量工具生成每个更新的文件的差分文件,可大大减小升级包中更新文件的数据量,以此节省客户端下载时所需的流量,并缩短时长,且又可避免单独采用增量升级时MD5校验不通过导致无法完成增量升级的问题。

[0161] 生成文件信息模块660,用于根据保持不变文件列表、增加文件列表及更新文件列表生成对应的保持不变的文件信息、增加的文件信息及更新的文件信息。

[0162] 具体的,服务端可根据保持不变文件列表、增加文件列表及更新文件列表生成对应的保持不变的文件信息、增加的文件信息及更新的文件信息的JSON字符串,保持不变的文件信息、增加的文件信息及更新的文件信息中即包含对应的文件列表的内容。

[0163] 写入模块650还用于将保持不变的文件信息、增加的文件信息及更新的文件信息写入升级包中。

[0164] 具体的,服务端可将保持不变的文件信息、增加的文件信息及更新的文件信息的JSON字符串压缩写入升级包中,方便客户端根据该文件信息进行软件更新。

[0165] 发送模块670,用于将升级包发送给客户端。

[0166] 具体的,服务端将升级包发送给客户端,使得客户端完成软件的更新。

[0167] 上述软件更新系统,服务端通过比较软件的旧包及新包的映射后得到不同的文件列表,在升级包中只写入增加的文件及更新文件的差分文件,能够大大减少升级包的数据量,当客户端在进行下载时,无需占用大量的带宽和流量,节约资源,并有效提高了下载速度,节省时间。

[0168] 在一个实施例中,上述软件更新系统,除了包括接收模块610、判断模块620、生成映射模块630、比较模块640、写入模块650、生成文件信息模块660和发送模块670,还包括获取模块。

[0169] 获取模块,用于获取新包的签名文件。

[0170] 写入模块650还用于将该签名文件写入升级包。

[0171] 具体的,许多不同的安装包中都有其独特的签名文件,该签名文件中包含有包名、厂商的数字证书等信息,签名文件中的信息可作为各个安装包的标识。服务端可获取新包的签名文件,并将签名文件压缩写入升级包中,使升级包带有新包的标识,即可容易识别该升级包对应的软件及版本号。

[0172] 上述软件更新系统,服务端通过比较软件的旧包及新包的映射后得到不同的文件列表,在升级包中只写入增加的文件及更新文件的差分文件,能够大大减少升级包的数据量,当客户端在进行下载时,无需占用大量的带宽和流量,节约资源,并有效提高了下载速度,节省时间。

[0173] 如图8所示,在另一个实施例中,一种软件更新系统,包括发送模块810、下载模块820、写入模块830和安装模块840。

[0174] 发送模块810,用于获取软件的标识信息,并根据标识信息向服务端发送更新请求。

[0175] 具体的,客户端可获取软件的标识符、版本号等标识信息,并向服务端发送带有该标识信息的更新请求,由服务端根据该标识信息判断软件是否需要更新。当软件不需要进行更新时,即客户端若接收到由服务端发送的无更新标志,可提示软件当前版本即为最新版本,无需进行更新。

[0176] 下载模块820,用于若软件需要更新,则下载升级包,并生成安装包。

[0177] 具体的,若软件需要进行更新,则客户端从服务端下载对应的升级包,并生成一个与升级包格式相同的空的安装包。

[0178] 写入模块830,用于根据升级包中的保持不变的文件信息从与标识信息匹配的旧包中获取对应的保持不变的文件,并将保持不变的文件写入安装包中。

[0179] 具体的,客户端下载升级包后,可从升级包中解压出保持不变的文件信息、增加的文件信息及更新的文件信息的JSON字符串。客户端可读取保持不变的文件信息,并根据该保持不变的文件信息从旧包中解压出对应的保持不变的文件压缩写入到安装包中,其中,旧包即为客户端当前版本软件对应的安装包。

[0180] 写入模块830还用于根据升级包中的增加的文件信息从升级包中获取增加的文件,并将增加的文件写入安装包中。

[0181] 具体的,客户端可从升级包中读取增加的文件信息,并根据增加的文件信息从升级包中解压出对应的增加的文件,再将增加的文件压缩写入安装包中。

[0182] 写入模块830还用于根据升级包中的更新的文件信息从升级包中获取各个更新的文件的差分文件,将差分文件与旧包中各个对应的文件合并,并将合并后的文件写入安装包中。

[0183] 具体的,客户端可从升级包中读取更新的文件信息,再根据更新的文件信息从升级包中解压出对应的各个更新的文件的差分文件,并根据更新的文件信息从旧包中解压出对应的旧版文件。客户端可利用bspatch等二进制差分合并工具将差分文件及对应的旧版文件进行合并,并将合并后的文件压缩写入安装包中。

[0184] 安装模块840,用于调用预设接口安装安装包,完成软件的更新。

[0185] 具体的,安装包写入完成后,客户端可调用预设接口进行安装,完成软件的更新。

[0186] 上述软件更新系统,服务端通过比较软件的旧包及新包的映射后得到不同的文件列表,在升级包中只写入增加的文件及更新文件的差分文件,能够大大减少升级包的数据量,当客户端在进行下载时,无需占用大量的带宽和流量,节约资源,并有效提高了下载速度,节省时间。

[0187] 如图9所示,在一个实施例中,上述软件更新系统,除了包括发送模块810、下载模块820、写入模块830和安装模块840,还包括读取模块850。

[0188] 写入模块830还用于获取升级包中的签名文件,并将签名文件写入安装包中。

[0189] 具体的,客户端可从升级包中解压出签名文件,并将签名文件压缩写入安装包中,签名文件中包含有包名、厂商的数字证书等信息,签名文件中的信息可作为安装包的标识。

[0190] 读取模块850,用于读取旧包的渠道号,将渠道号写入安装包中,生成对应的渠道更新包。

[0191] 具体的,当软件拥有多渠道更新时,不同的渠道的安装包中携带有不同的渠道号,例如不同渠道的apk文件,其渠道号通常写入文件末尾的EOCD字段中。客户端可读取旧包的渠道号,并将渠道号写入安装包的特定字段中,生成对应的渠道更新包,该特定字段需满足可修改且不会影响安装包的正确打开及使用。例如apk文件的EOCD字段中有一个Comment字段,Comment字段通常位于文件的末尾,且其长度不固定,Comment字段的内容可任意写入并修改,只需同时改变对应的长度即不影响apk文件的正确打开及使用。因此,客户端可从软

件对应的旧的apk文件的EOCD字段中读取渠道号,并将渠道号写入安装包的EOCD字段的Comment字段中,生成对应的渠道更新包。

[0192] 上述软件更新系统,客户端能够读取软件的旧有的渠道号并生成对应的渠道更新包,当软件拥有多渠道进行更新时,服务端只需保存各个版本号对应的一个安装包即可,无需针对不同的渠道生成不同的安装包,简化了软件的多渠道版本的更新步骤,并有效降低了服务端的维护成本。

[0193] 以上所述实施例的各技术特征可以进行任意的组合,为使描述简洁,未对上述实施例中的各个技术特征所有可能的组合都进行描述,然而,只要这些技术特征的组合不存在矛盾,都应当认为是本说明书记载的范围。

[0194] 以上所述实施例仅表达了本发明的几种实施方式,其描述较为具体和详细,但并不能因此而理解为对发明专利范围的限制。应当指出的是,对于本领域的普通技术人员来说,在不脱离本发明构思的前提下,还可以做出若干变形和改进,这些都属于本发明的保护范围。因此,本发明专利的保护范围应以所附权利要求为准。

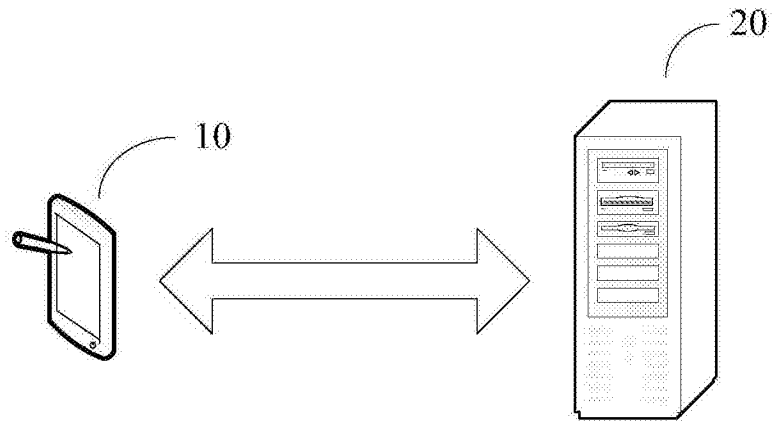


图1

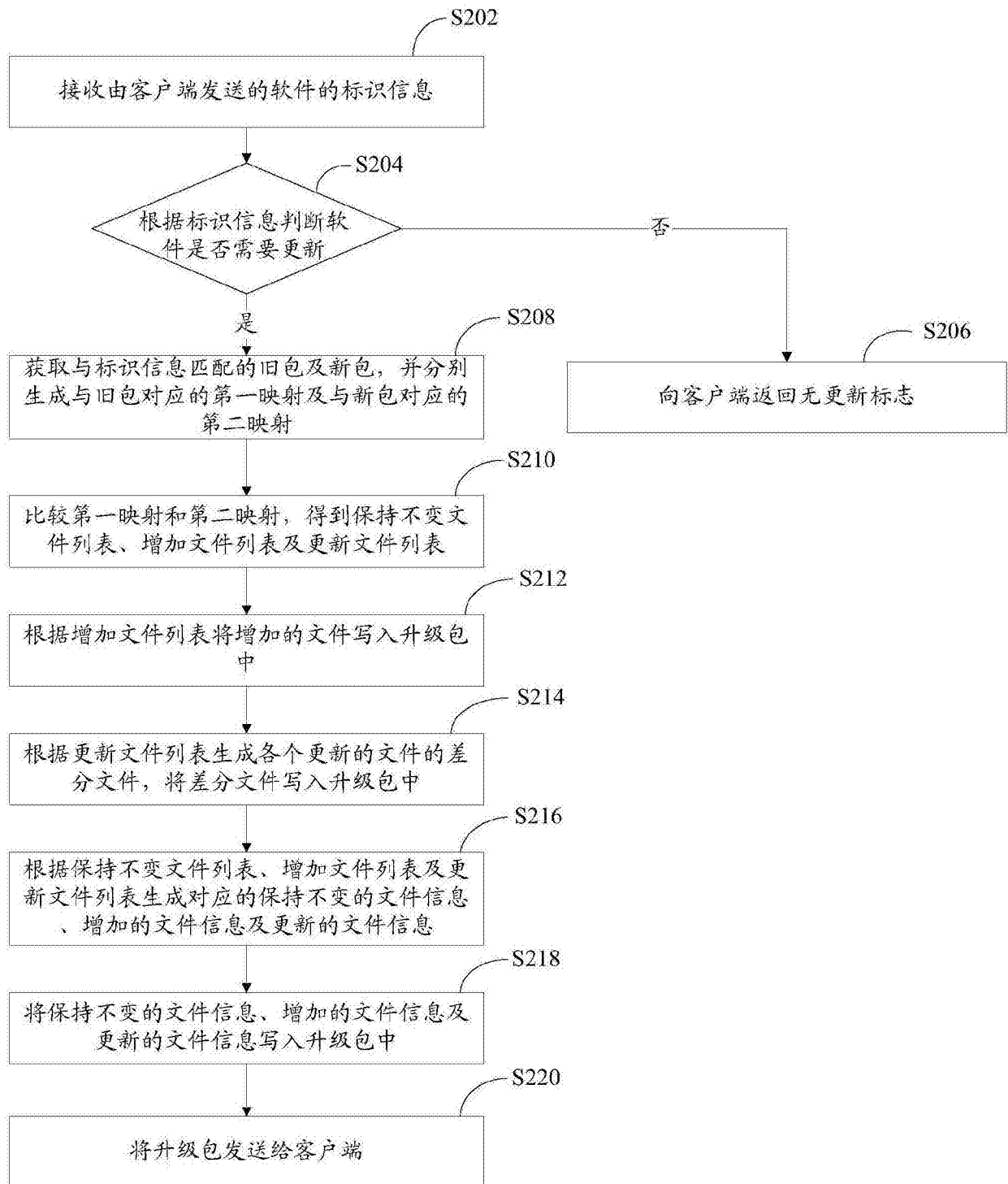


图2

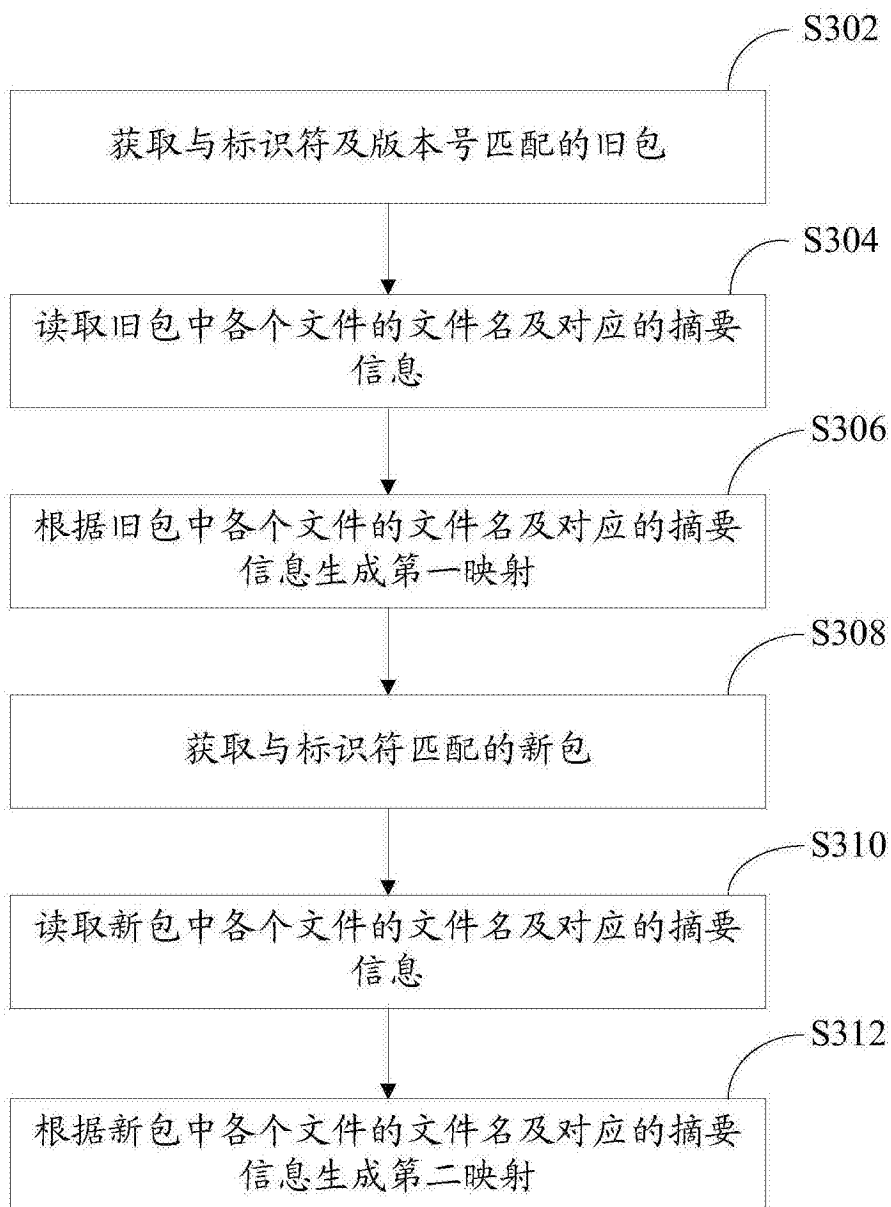


图3

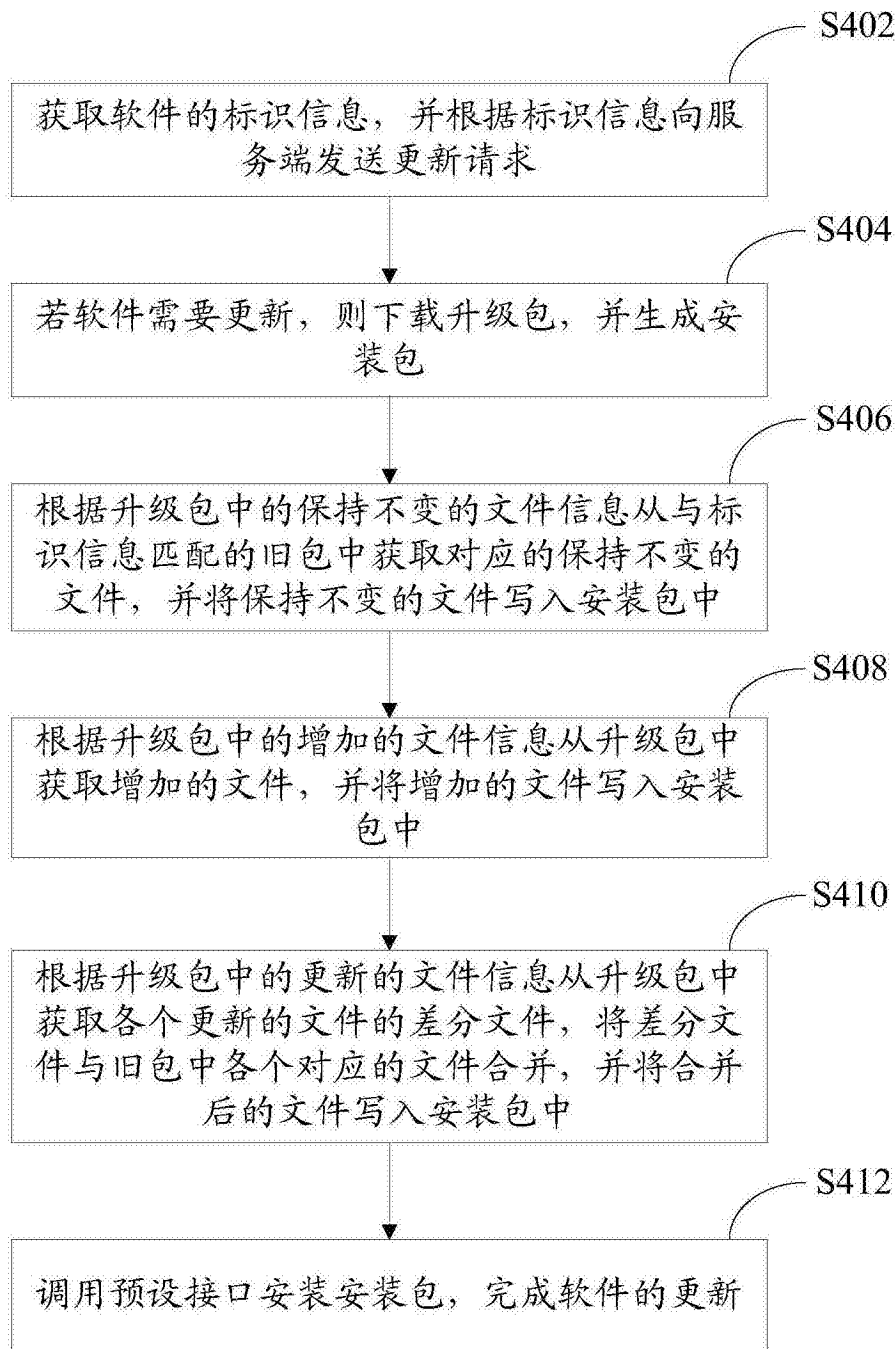


图4

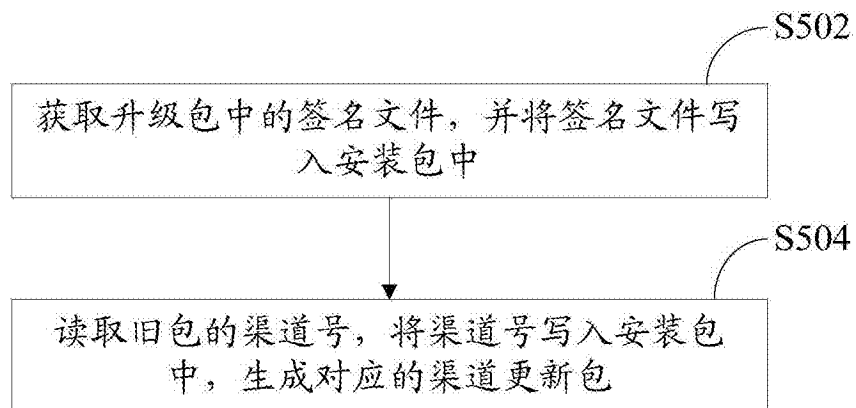


图5

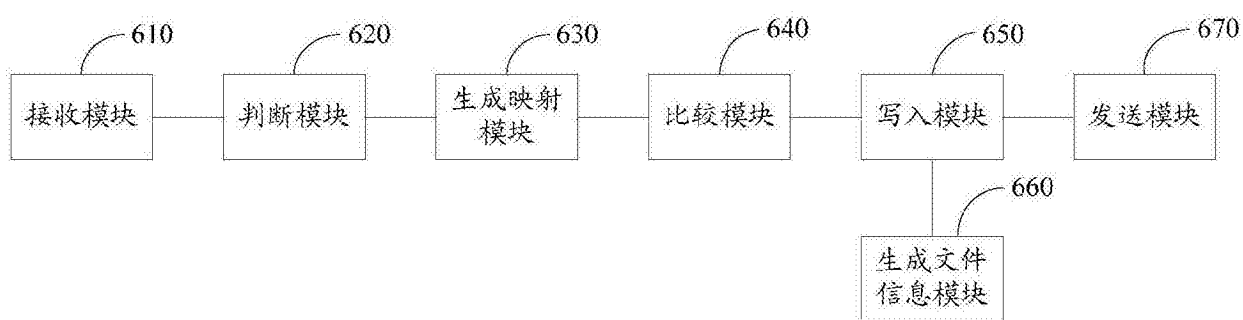


图6

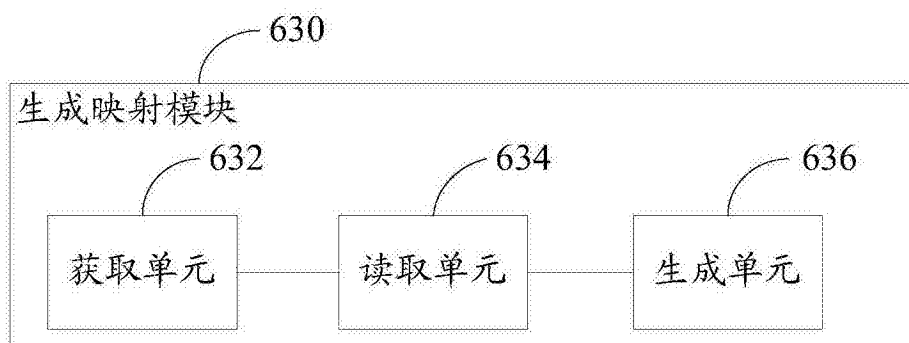


图7



图8

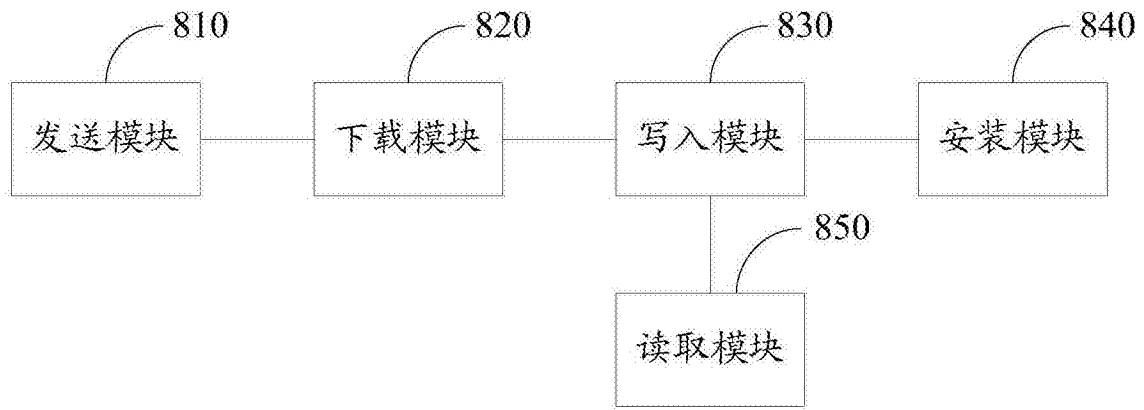


图9