(19)国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 115134553 A
(43)申请公布日 2022.09.30

(21)申请号 202110277682.0

(22)申请日 2021.03.15

(71)申请人 腾讯科技(深圳)有限公司
地址 518000 广东省深圳市南山区高新区科技中一路腾讯大厦35层

(72)发明人 余一

(74)专利代理机构 北京派特恩知识产权代理有限公司 11270
专利代理师 高天华 张颖玲

(51)Int.Cl.
H04N 7/15(2006.01)
H04N 21/431(2011.01)
H04L 5/00(2006.01)
H04L 9/40(2022.01)
H04L 51/04(2022.01)

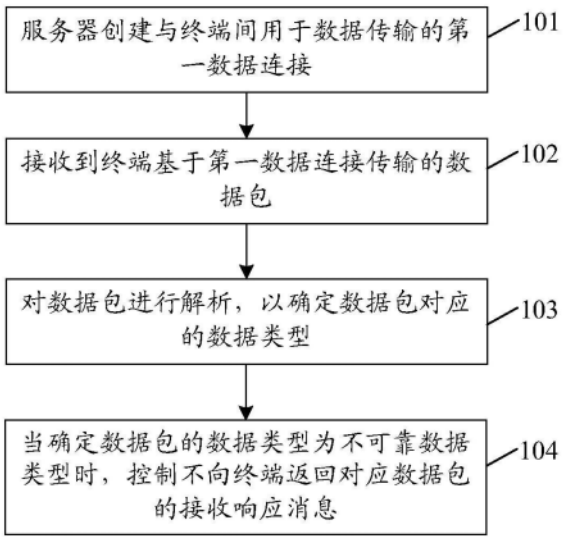
权利要求书3页 说明书24页 附图18页

(54)发明名称

数据传输方法、装置、电子设备及存储介质

(57)摘要

本申请提供了一种数据传输方法、装置、电子设备及存储介质;涉及云技术和通信技术领域,方法包括:创建与终端间用于数据传输的第一数据连接,所述第一数据连接的连接标识,在所述终端的互联网协议地址或端口发生变化时保持不变;接收到所述终端基于所述第一数据连接传输的数据包;对所述数据包进行解析,以确定所述数据包对应的数据类型;当确定所述数据包的数据类型为不可靠数据类型时,控制不向所述终端返回对应所述数据包的接收响应消息;通过本申请,能够减少网络资源的不必要占用,提高网络资源的利用率。



CN 115134553 A

1.一种数据传输方法,其特征在于,所述方法包括:

创建与终端间用于数据传输的第一数据连接,所述第一数据连接的连接标识,在所述终端的互联网协议地址或端口发生变化时保持不变;

接收到所述终端基于所述第一数据连接传输的数据包;

对所述数据包进行解析,以确定所述数据包对应的数据类型;

当确定所述数据包的数据类型为不可靠数据类型时,控制不向所述终端返回对应所述数据包的接收响应消息。

2.如权利要求1所述的方法,其特征在于,所述创建与终端间用于数据传输的第一数据连接之后,所述方法还包括:

获取心跳包发送周期,并按照所述心跳包发送周期,通过所述第一数据连接,发送心跳包至所述终端;

当接收到针对所述心跳包的心跳响应信息时,控制所述第一数据连接对应的连接状态为活跃状态。

3.如权利要求1所述的方法,其特征在于,当所述第一数据连接为首次建立时,所述创建与终端间用于数据传输的第一数据连接,包括:

接收到所述终端发送的请求建立数据连接的第一问询数据包;

基于所述第一问询数据包,发送相应的应答数据包至所述终端;

当接收到所述终端发送的第二问询数据包时,创建与所述终端间用于数据传输的第一数据连接。

4.如权利要求1所述的方法,其特征在于,当所述第一数据连接为非首次建立时,所述创建与终端间用于数据传输的第一数据连接,包括:

接收到所述终端发送的请求建立数据连接的问询数据包;

基于所述问询数据包,创建与所述终端间用于数据传输的第一数据连接。

5.如权利要求1所述的方法,其特征在于,所述对所述数据包进行解析,以确定所述数据包对应的数据类型,包括:

提取所述数据包的包头部分;

对所述包头部分进行解析,得到用于指示数据类型的报文字段;

将所述报文字段所指示的数据类型,确定为所述数据包对应的数据类型。

6.如权利要求1所述的方法,其特征在于,所述方法还包括:

当所述终端的互联网协议地址发生变更时,通过所述终端的第一套接字接口,建立与所述终端间的第二数据连接,以将与所述终端间的数据连接从所述第一数据连接迁移至所述第二数据连接;

其中,所述第二数据连接的连接标识与所述第一数据连接的连接标识相一致。

7.如权利要求6所述的方法,其特征在于,所述建立与所述终端间的第二数据连接之后,所述方法还包括:

基于所述连接标识,通过所述第二数据连接,发送包括第一验证信息的探测数据包至所述终端;

接收到所述终端针对所述探测数据包返回的响应数据包,所述响应数据包包括第二验证信息;

基于所述第二验证信息对所述第一验证信息进行验证，当得到表征验证通过的验证结果时，基于所述第二数据连接进行与所述终端间的数据传输。

8.如权利要求6所述的方法，其特征在于，所述建立与所述终端间的第二数据连接之后，所述方法还包括：

接收到所述终端基于所述连接标识，通过所述第二数据连接传输的包括第三验证信息的探测数据包；

发送针对所述探测数据包的响应数据包至所述终端，所述响应数据包包括第四验证信息；

其中，所述第四验证信息，用于供所述终端基于所述第三验证信息对所述第四验证信息进行验证，当得到表征验证通过的验证结果时，基于所述第二数据连接进行数据传输。

9.如权利要求1所述的方法，其特征在于，所述方法还包括：

对目标互联网协议地址是否发生变更进行监测，得到监测结果，所述目标互联网协议地址归属于与所述终端建立所述第一数据连接的服务器；

当所述监测结果表征所述目标互联网协议地址发生变更时，针对变更后的目标互联网协议地址创建第二套接字接口，并

基于所述第二套接字接口，建立与所述终端间的第三数据连接，以将与所述终端间的数据连接从所述第一数据连接迁移至所述第三数据连接；

其中，所述第三数据连接的连接标识与所述第一数据连接的连接标识相一致。

10.如权利要求9所述的方法，其特征在于，所述对目标互联网协议地址是否发生变更进行监测，得到监测结果，包括：

对所述目标互联网协议地址对应的原始套接字接口的数据传输失败事件、以及所述目标互联网协议地址的变更通知消息中至少之一进行监测；

当监测到所述目标互联网协议地址对应的原始套接字接口的数据传输失败事件、以及所述目标互联网协议地址的变更通知消息中至少之一时，得到表征所述目标互联网协议地址发生变更的监测结果。

11.如权利要求9所述的方法，其特征在于，所述建立与所述终端间的第三数据连接之后，所述方法还包括：

基于所述连接标识，对所述第三数据连接的数据可达性进行验证，得到验证结果；

当所述验证结果表征所述第三数据连接的数据可达时，基于所述第三数据连接进行与所述终端间的数据传输。

12.一种数据传输方法，其特征在于，所述方法包括：

创建与服务器间用于数据传输的第一数据连接，所述第一数据连接的连接标识，在所述服务器的互联网协议地址或端口发生变化时保持不变；

基于所述第一数据连接，发送数据包至所述服务器；

其中，所述数据包的数据类型包括不可靠数据类型，所述不可靠数据类型，用于当所述服务器确定所述数据包的数据类型为不可靠数据类型时，控制不返回对应所述数据包的接收响应消息。

13.一种数据传输装置，其特征在于，所述装置包括：

创建模块，用于创建与终端间用于数据传输的第一数据连接，所述第一数据连接的连

接标识,在所述终端的互联网协议地址或端口发生变化时保持不变;

接收模块,用于接收到所述终端基于所述第一数据连接传输的数据包;

解析模块,用于对所述数据包进行解析,以确定所述数据包对应的数据类型;

控制模块,用于当确定所述数据包的数据类型为不可靠数据类型时,控制不向所述终端返回对应所述数据包的接收响应消息。

14.一种电子设备,其特征在于,所述电子设备包括:

存储器,用于存储可执行指令;

处理器,用于执行所述存储器中存储的可执行指令时,实现如权利要求1至12任一项所述的数据传输方法。

15.一种计算机可读存储介质,其特征在于,存储有可执行指令,所述可执行指令被执行时,用于实现如权利要求1至12任一项所述的数据传输方法。

数据传输方法、装置、电子设备及存储介质

技术领域

[0001] 本申请涉及云技术和通信技术领域，尤其涉及一种数据传输方法、装置、电子设备及存储介质。

背景技术

[0002] 相关技术中，针对音视频数据、频繁移动的坐标数据等无需可靠传输的不可靠数据，往往和需要可靠传输的可靠数据复用同一条通道传输，即均通过可靠传输方式进行传输。而网络较差时，可靠传输方式的重传策略会给网络带来较大压力，并且会影响信令等可靠数据的正常传输。

发明内容

[0003] 本申请实施例提供一种数据传输方法、装置、电子设备及存储介质，能够减少网络资源的不必要占用，提高网络资源的利用率。

[0004] 本申请实施例的技术方案是这样实现的：

[0005] 本申请实施例提供一种数据传输方法，包括：

[0006] 创建与终端间用于数据传输的第一数据连接，所述第一数据连接的连接标识，在所述终端的互联网协议地址或端口发生变化时保持不变；

[0007] 接收到所述终端基于所述第一数据连接传输的数据包；

[0008] 对所述数据包进行解析，以确定所述数据包对应的数据类型；

[0009] 当确定所述数据包的数据类型为不可靠数据类型时，控制不向所述终端返回对应所述数据包的接收响应消息。

[0010] 本申请实施例提供一种数据传输方法，包括：

[0011] 创建与服务器间用于数据传输的第一数据连接，所述第一数据连接的连接标识，在所述服务器的互联网协议地址或端口发生变化时保持不变；

[0012] 基于所述第一数据连接，发送数据包至所述服务器；

[0013] 其中，所述数据包的数据类型包括不可靠数据类型，所述不可靠数据类型，用于当所述服务器确定所述数据包的数据类型为不可靠数据类型时，控制不返回对应所述数据包的接收响应消息。

[0014] 本申请实施例还提供一种数据传输装置，包括：

[0015] 创建模块，用于创建与终端间用于数据传输的第一数据连接，所述第一数据连接的连接标识，在所述终端的互联网协议地址或端口发生变化时保持不变；

[0016] 接收模块，用于接收到所述终端基于所述第一数据连接传输的数据包；

[0017] 解析模块，用于对所述数据包进行解析，以确定所述数据包对应的数据类型；

[0018] 控制模块，用于当确定所述数据包的数据类型为不可靠数据类型时，控制不向所述终端返回对应所述数据包的接收响应消息。

[0019] 上述方案中，所述装置还包括：

[0020] 连接状态控制模块，用于获取心跳包发送周期，并按照所述心跳包发送周期，通过所述第一数据连接，发送心跳包至所述终端；

[0021] 当接收到针对所述心跳包的心跳响应信息时，控制所述第一数据连接对应的连接状态为活跃状态。

[0022] 上述方案中，当所述第一数据连接为首次建立时，所述创建模块，还用于接收到所述终端发送的请求建立数据连接的第一问询数据包；

[0023] 基于所述第一问询数据包，发送相应的应答数据包至所述终端；

[0024] 当接收到所述终端发送的第二问询数据包时，创建与所述终端间用于数据传输的第一数据连接。

[0025] 上述方案中，当所述第一数据连接为非首次建立时，所述创建模块，还用于接收到所述终端发送的请求建立数据连接的问询数据包；

[0026] 基于所述问询数据包，创建与所述终端间用于数据传输的第一数据连接。

[0027] 上述方案中，所述解析模块，还用于提取所述数据包的包头部分；

[0028] 对所述包头部分进行解析，得到用于指示数据类型的报文字段；

[0029] 将所述报文字段所指示的数据类型，确定为所述数据包对应的数据类型。

[0030] 上述方案中，所述装置还包括：

[0031] 第一迁移模块，用于当所述终端的互联网协议地址发生变更时，通过所述终端的第一套接字接口，建立与所述终端间的第二数据连接，以将与所述终端间的数据连接从所述第一数据连接迁移至所述第二数据连接；

[0032] 其中，所述第二数据连接的连接标识与所述第一数据连接的连接标识相一致。

[0033] 上述方案中，所述第一迁移模块，还用于基于所述连接标识，通过所述第二数据连接，发送包括第一验证信息的探测数据包至所述终端；

[0034] 接收到所述终端针对所述探测数据包返回的响应数据包，所述响应数据包包括第二验证信息；

[0035] 基于所述第二验证信息对所述第一验证信息进行验证，当得到表征验证通过的验证结果时，基于所述第二数据连接进行与所述终端间的数据传输。

[0036] 上述方案中，所述第一迁移模块，还用于接收到所述终端基于所述连接标识，通过所述第二数据连接传输的包括第三验证信息的探测数据包；

[0037] 发送针对所述探测数据包的响应数据包至所述终端，所述响应数据包包括第四验证信息；

[0038] 其中，所述第四验证信息，用于供所述终端基于所述第三验证信息对所述第四验证信息进行验证，当得到表征验证通过的验证结果时，基于所述第二数据连接进行数据传输。

[0039] 上述方案中，所述装置还包括：

[0040] 第二迁移模块，用于对目标互联网协议地址是否发生变更进行监测，得到监测结果，所述目标互联网协议地址归属于与所述终端建立所述第一数据连接的服务器；

[0041] 当所述监测结果表征所述目标互联网协议地址发生变更时，针对变更后的目标互联网协议地址创建第二套接字接口，并

[0042] 基于所述第二套接字接口，建立与所述终端间的第三数据连接，以将与所述终端

间的数据连接从所述第一数据连接迁移至所述第三数据连接；

[0043] 其中，所述第三数据连接的连接标识与所述第一数据连接的连接标识相一致。

[0044] 上述方案中，所述第二迁移模块，还用于对所述目标互联网协议地址对应的原始套接字接口的数据传输失败事件、以及所述目标互联网协议地址的变更通知消息中至少之一进行监测；

[0045] 当监测到所述目标互联网协议地址对应的原始套接字接口的数据传输失败事件、以及所述目标互联网协议地址的变更通知消息中至少之一时，得到表征所述目标互联网协议地址发生变更的监测结果。

[0046] 上述方案中，所述第二迁移模块，还用于基于所述连接标识，对所述第三数据连接的数据可达性进行验证，得到验证结果；

[0047] 当所述验证结果表征所述第三数据连接的数据可达时，基于所述第三数据连接进行与所述终端间的数据传输。

[0048] 本申请实施例还提供一种数据传输装置，包括：

[0049] 连接创建模块，用于创建与服务器间用于数据传输的第一数据连接，所述第一数据连接的连接标识，在所述服务器的互联网协议地址或端口发生变化时保持不变；

[0050] 发送模块，用于基于所述第一数据连接，发送数据包至所述服务器；

[0051] 其中，所述数据包的数据类型包括不可靠数据类型，所述不可靠数据类型，用于当所述服务器确定所述数据包的数据类型为不可靠数据类型时，控制不返回对应所述数据包的接收响应消息。

[0052] 本申请实施例还提供一种电子设备，包括：

[0053] 存储器，用于存储可执行指令；

[0054] 处理器，用于执行所述存储器中存储的可执行指令时，实现本申请实施例提供的数据传输方法。

[0055] 本申请实施例还提供一种计算机可读存储介质，存储有可执行指令，所述可执行指令被处理器执行时，实现本申请实施例提供的数据传输方法。

[0056] 本申请实施例具有以下有益效果：

[0057] 创建与终端间用于数据传输的第一数据连接，该第一数据连接的连接标识，在终端的互联网协议地址或端口发生变化时保持不变；当接收到该终端基于第一数据连接传输的数据包时，对该数据包进行解析，以确定该数据包对应的数据类型；当确定数据包的数据类型为不可靠数据类型时，则控制不向该终端返回对应数据包的接收响应消息。如此，通过解析数据包的数据类型，针对不可靠数据类型的数据包，则不返回对应的接收响应消息，从而避免了终端重传不可靠数据类型的数据包的情况，减少网络资源的不必要占用，提高网络资源的利用率。

附图说明

[0058] 图1是本申请实施例提供的数据传输系统100的架构示意图；

[0059] 图2是本申请实施例提供的数据传输方法的电子设备500的结构示意图；

[0060] 图3A是本申请实施例提供的数据传输方法的流程示意图；

[0061] 图3B是本申请实施例提供的数据传输方法的流程示意图；

[0062] 图4是本申请实施例提供的数据传输方法的流程示意图；

[0063] 图5是本申请实施例提供的用于数据传输的长连接的创建方法的流程示意图；

[0064] 图6是本申请实施例提供的客户端的应用示意图；

[0065] 图7是本申请实施例提供的TCP连接的建立流程示意图；

[0066] 图8是本申请实施例提供的TLS连接的建立流程示意图；

[0067] 图9是本申请实施例提供的互联网协议地址变更时客户端的表现示意图；

[0068] 图10是本申请实施例提供的QUIC连接的建立流程示意图；

[0069] 图11是本申请实施例提供的用于数据传输的长连接的创建方法的流程示意图；

[0070] 图12是本申请实施例提供的数据连接的连接迁移的流程示意图；

[0071] 图13是本申请实施例提供的连接迁移的流程示意图；

[0072] 图14是本申请实施例提供的路径验证的流程示意图；

[0073] 图15是本申请实施例提供的可靠帧格式和不可靠帧格式的示意图；

[0074] 图16是本申请实施例提供的应用于远程控制的数据传输的流程示意图；

[0075] 图17A是本申请实施例提供的画中画模式的示意图；

[0076] 图17B是本申请实施例提供的声音大小显示的示意图；

[0077] 图18是本申请实施例提供的客户端指标的优化示意图；

[0078] 图19是本申请实施例提供的数据传输装置555的结构示意图；

[0079] 图20是本申请实施例提供的数据传输装置2000的结构示意图。

具体实施方式

[0080] 为了使本申请的目的、技术方案和优点更加清楚，下面将结合附图对本申请作进一步地详细描述，所描述的实施例不应视为对本申请的限制，本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其它实施例，都属于本申请保护的范围。

[0081] 在以下的描述中，涉及到“一些实施例”，其描述了所有可能实施例的子集，但是可以理解，“一些实施例”可以是所有可能实施例的相同子集或不同子集，并且可以在不冲突的情况下相互结合。

[0082] 在以下的描述中，所涉及的术语“第一\第二\第三”仅仅是是区别类似的对象，不代表针对对象的特定排序，可以理解地，“第一\第二\第三”在允许的情况下可以互换特定的顺序或先后次序，以使这里描述的本申请实施例能够以除了在这里图示或描述的以外的顺序实施。

[0083] 除非另有定义，本文所使用的所有的技术和科学术语与属于本申请的技术领域的技术人员通常理解的含义相同。本文中所使用的术语只是为了描述本申请实施例的目的，不是旨在限制本申请。

[0084] 对本申请实施例进行进一步详细说明之前，对本申请实施例中涉及的名词和术语进行说明，本申请实施例中涉及的名词和术语适用于如下的解释。

[0085] 1)客户端，终端中运行的用于提供各种服务的应用程序，例如即时通讯客户端、视频播放客户端。

[0086] 2)响应于，用于表示所执行的操作所依赖的条件或者状态，当满足所依赖的条件或状态时，所执行的一个或多个操作可以是实时的，也可以具有设定的延迟；在没有特别说

明的情况下,所执行的多个操作不存在执行先后顺序的限制。

[0087] 3)TCP:Transmission Control Protocol,是一种面向连接的、可靠的、基于字节流的传输层通信协议,由IETF(国际互联网工程任务组)的RFC 793定义。

[0088] 4)TLS:Transport Layer Security,是一种安全协议,目的是为互联网通信提供安全及数据完整性保障

[0089] 5)QUIC:Quick UDP Internet Connection,是谷歌制定的一种基于UDP的低时延的可靠互联网传输层协议,2016年11月由IETF(国际互联网工程任务组)开始进行标准化,预计2021年从基础草案发布成为RFC。

[0090] 6)Cellular:蜂窝技术,一种无线通信技术,也就是我们通常说的的GPRS/3G/4G/5G网络。

[0091] 7)长连接:客户端和后台之间建立并维护的一条长时间有效的可靠的数据传输通道。

[0092] 8)XMPP:XMPP是一种基于标准通用标记语音的子集XML的协议,它继承了在XML环境中灵活的发展性。可用于服务类实时通讯、表示和需求响应服务中的XML数据元流式传输。因此,基于XMPP的应用具有超强的可扩展性,由IETF(国际互联网工程任务组)的RFC 3920定义。

[0093] 9)RTT:Round-Trip Time,往返时延。在计算机网络中它是一个重要的性能指标,表示从发送端发送数据开始,到发送端收到来自接收端的确认(接收端收到数据后便立即发送确认),总共经历的时延。

[0094] 基于上述对本申请实施例中涉及的名词和术语的解释,下面说明本申请实施例提供的数据传输系统。参见图1,图1是本申请实施例提供的数据传输系统100的架构示意图,为实现支撑一个示例性应用,终端(示例性示出了终端400-1)通过网络300连接服务器200,网络300可以是广域网或者局域网,又或者是二者的组合,使用无线或有线链路实现数据传输。

[0095] 终端(如终端400-1),用于创建与服务器200间用于数据传输的第一数据连接;基于第一数据连接,发送数据包至服务器200;

[0096] 服务器200,用于创建与终端间用于数据传输的第一数据连接;接收到终端基于所述第一数据连接传输的数据包;对数据包进行解析,以确定数据包对应的数据类型;当确定数据包的数据类型为不可靠数据类型时,控制不向终端返回对应数据包的接收响应消息。

[0097] 本申请实施例可以借助于云技术(Cloud Technology)实现,云技术是指在广域网或局域网内将硬件、软件、网络等系列资源统一起来,实现数据的计算、储存、处理和共享的一种托管技术。

[0098] 云技术是基于云计算商业模式应用的网络技术、信息技术、整合技术、管理平台技术、以及应用技术等的总称,可以组成资源池,按需所用,灵活便利。云计算技术将变成重要支撑。技术网络系统的后台服务需要大量的计算、存储资源。

[0099] 在实际应用中,服务器200可以是独立的物理服务器,也可以是多个物理服务器构成的服务器集群或者分布式系统,还可以是提供云服务、云数据库、云计算、云函数、云存储、网络服务、云通信、中间件服务、域名服务、安全服务、CDN、以及大数据和人工智能平台等基础云计算服务的云服务器。终端(如终端400-1)可以是智能手机、平板电脑、笔记本电

脑、台式计算机、智能音箱、智能电视、智能手表等，但并不局限于此。终端(如终端400-1)以及服务器200可以通过有线或无线通信方式进行直接或间接地连接，本申请在此不做限制。

[0100] 参见图2，图2是本申请实施例提供的数据传输方法的电子设备500的结构示意图。在实际应用中，电子设备500可以为图1示出的服务器或终端，以电子设备500为图1示出的终端为例，对实施本申请实施例的数据传输方法的电子设备进行说明，本申请实施例提供的电子设备500包括：至少一个处理器510、存储器550、至少一个网络接口520和用户接口530。电子设备500中的各个组件通过总线系统540耦合在一起。可理解，总线系统540用于实现这些组件之间的连接通信。总线系统540除包括数据总线之外，还包括电源总线、控制总线和状态信号总线。但是为了清楚说明起见，在图2中将各种总线都标为总线系统540。

[0101] 处理器510可以是一种集成电路芯片，具有信号的处理能力，例如通用处理器、数字信号处理器(DSP，Digital Signal Processor)，或者其他可编程逻辑器件、分立门或者晶体管逻辑器件、分立硬件组件等，其中，通用处理器可以是微处理器或者任何常规的处理器等。

[0102] 用户接口530包括使得能够呈现媒体内容的一个或多个输出装置531，包括一个或多个扬声器和/或一个或多个视觉显示屏。用户接口530还包括一个或多个输入装置532，包括有助于用户输入的用户接口部件，比如键盘、鼠标、麦克风、触屏显示屏、摄像头、其他输入按钮和控件。

[0103] 存储器550可以是可移除的，不可移除的或其组合。示例性的硬件设备包括固态存储器，硬盘驱动器，光盘驱动器等。存储器550可选地包括在物理位置上远离处理器510的一个或多个存储设备。

[0104] 存储器550包括易失性存储器或非易失性存储器，也可包括易失性和非易失性存储器两者。非易失性存储器可以是只读存储器(ROM，Read Only Me mory)，易失性存储器可以是随机存取存储器(RAM，Random Access Memor y)。本申请实施例描述的存储器550旨在包括任意适合类型的存储器。

[0105] 在一些实施例中，存储器550能够存储数据以支持各种操作，这些数据的示例包括程序、模块和数据结构或者其子集或超集，下面示例性说明。

[0106] 操作系统551，包括用于处理各种基本系统服务和执行硬件相关任务的系统程序，例如框架层、核心库层、驱动层等，用于实现各种基础业务以及处理基于硬件的任务；

[0107] 网络通信模块552，用于经由一个或多个(有线或无线)网络接口520到达其他计算设备，示例性的网络接口520包括：蓝牙、无线相容性认证(WiFi)、和通用串行总线(USB，Universal Serial Bus)等；

[0108] 呈现模块553，用于经由一个或多个与用户接口530相关联的输出装置531(例如，显示屏、扬声器等)使得能够呈现信息(例如，用于操作外围设备和显示内容和信息的用户接口)；

[0109] 输入处理模块554，用于对一个或多个来自一个或多个输入装置532之一的一个或多个用户输入或互动进行检测以及翻译所检测的输入或互动。

[0110] 在一些实施例中，本申请实施例提供的数据传输装置可以采用软件方式实现，图2示出了存储在存储器550中的数据传输装置555，其可以是程序和插件等形式的软件，包括以下软件模块：创建模块5551、接收模块5552、解析模块5553和控制模块5554，这些模块是

逻辑上的,因此根据所实现的功能可以进行任意的组合或进一步拆分,将在下文中说明各个模块的功能。

[0111] 在另一些实施例中,本申请实施例提供的数据传输装置可以采用软硬件结合的方式实现,作为示例,本申请实施例提供的数据传输装置可以是采用硬件译码处理器形式的处理器,其被编程以执行本申请实施例提供的数据传输方法,例如,硬件译码处理器形式的处理器可以采用一个或多个应用专用集成电路(ASIC,Application Specific Integrated Circuit)、DSP、可编程逻辑器件(PLD,Programmable Logic Device)、复杂可编程逻辑器件(CPLD,Complex Progra mmable Logic Device)、现场可编程门阵列(FPGA,Field-Programmable GateArray)或其他电子元件。

[0112] 基于上述对本申请实施例提供的数据传输系统及电子设备的说明,下面说明本申请实施例提供的数据传输方法。在一些实施例中,本申请实施例提供的数据传输方法可由服务器或终端单独实施,或由服务器及终端协同实施,下面以服务器实施为例说明本申请实施例提供的数据传输方法。参见图3A,图3A是本申请实施例提供的数据传输方法的流程示意图,本申请实施例提供的数据传输方法包括:

[0113] 步骤101:服务器创建与终端间用于数据传输的第一数据连接。

[0114] 其中,第一数据连接的连接标识,在终端的互联网协议地址或端口发生变化时保持不变。

[0115] 这里,服务器在与终端进行数据通信之前,可以预先创建服务器与终端之间的用于数据传输的第一数据连接。在第一数据连接创建完成后,服务器与终端通过该第一数据连接进行数据包的发送与接收。在实际应用中,终端可以设置有应用客户端,比如会议客户端、学习客户端等,该服务器可以为相应的应用客户端的后台服务器。

[0116] 在本申请实施例中,该第一数据连接的连接标识,在终端和服务器中任一方的互联网协议地址(即IP地址)或端口发送变化时保持不变,即在终端和服务器中任一方的IP地址或端口发送变化时,该终端与服务器之间的数据通道并不会断,双方均可基于该第一数据连接的连接标识进行数据包的传输。如此,当终端或者服务器的IP地址发生变更时,并不影响双方之间的数据传输,而对于用户来说,IP地址的变更是无感知的,也不会出现网络断连、需要重新连接等情况。

[0117] 在实际应用中,该第一数据连接是基于QUIC协议建立的,QUIC连接的连接标识是一个64位的连接ID,用户在Wi-Fi和蜂窝网络(即Celluar)之间切换时,无论是IP地址或者端口(Port)发生变化,QUIC连接中的连接ID保持不变,终端与服务器双方仍可基于该QUIC连接的连接标识进行数据包的传输,因此不需要重新经过协议握手等过程创建连接,如此QUIC连接迁移可达到用户无感知的网络类型切换的技术效果。

[0118] 在一些实施例中,当第一数据连接为首次建立时,服务器可通过如下方式创建与终端间用于数据传输的第一数据连接:接收到终端发送的请求建立数据连接的第一问询数据包;基于第一问询数据包,发送相应的应答数据包至终端;当接收到终端发送的第二问询数据包时,创建与终端间用于数据传输的第一数据连接。

[0119] 在实际应用中,该第一数据连接可以是基于QUIC协议实现的。具体地,当第一数据连接为首次建立时,服务器与终端之间可以通过如下方式创建第一数据连接:当终端需要建立与服务器的数据连接时,向服务器发送请求建立数据连接的第一问询数据包;服务器

接收到该第一问询数据包后，基于该第一问询数据包，发送相应的应答数据包至终端，该应答数据包中可以包括密钥交换算法的公钥信息、算法信息、证书信息、协议可用版本信息等；终端接收到服务器发送的响应数据包并存储其中包含的信息，然后向服务器发送第二问询数据包，该第二问询数据包中可以携带终端的加密信息，比如终端所运行客户端的公钥信息；服务器接收到该第二问询数据包后，若接受建立数据连接，则创建与终端间的第一数据连接，若不接受则返回拒绝建立的响应信息。这里，上述终端和服务器对应的加密信息用于对第一数据连接中传输的数据进行加密，以保证数据传输的安全性。

[0120] 在实际实施时，在终端发送第二问询数据包时，可以将终端的加密信息和应用数据一起发送；服务器在接收到第二问询数据包时，基于该第二问询数据包建立与终端间的第一数据连接，从而可以针对第二问询数据包中的应用数据进行相应的响应。如此，该第一数据连接则可以在一个RTT内完成协议握手和数据加密的操作，既减少了数据连接的握手耗时，也完成了数据加密传输，保证了数据的安全性。

[0121] 在一些实施例中，当第一数据连接为非首次建立时，服务器可通过如下方式创建与终端间用于数据传输的第一数据连接：接收到终端发送的请求建立数据连接的问询数据包；基于问询数据包，创建与终端间用于数据传输的第一数据连接。

[0122] 在实际应用中，该第一数据连接可以是基于QUIC协议实现的。具体地，当第一数据连接为非首次建立时，服务器与终端之间可以通过如下方式创建第一数据连接：由于是非首次建立终端与服务器间的数据连接，终端侧存储有服务器的相关信息，即首次建立数据连接时存储的第二问询数据包中的密钥交换算法的公钥信息、算法信息、证书信息、协议可用版本信息等。此时，当终端再次需要和服务器建立数据连接时，终端仅需要发送请求建立数据连接的问询数据包至服务器，该问询数据包中可以直接携带加密的应用数据；服务器接收到该问询数据包后，若接受建立连接，则直接创建与终端间的第一数据连接，从而可以针对问询数据包中的应用数据进行相应的响应。如此，该第一数据连接则可以在0个RTT内完成建立，既减少了数据连接的握手耗时，也完成了数据加密传输，保证了数据的安全性。

[0123] 在一些实施例中，创建与终端间用于数据传输的第一数据连接之后，服务器可通过如下方式维持第一数据连接的连接状态为活跃状态：获取心跳包发送周期，并按照心跳包发送周期，通过第一数据连接，发送心跳包至终端；当接收到针对心跳包的心跳响应信息时，控制第一数据连接对应的连接状态为活跃状态。

[0124] 这里，服务器在和终端间建立第一数据连接后，还需要维持该第一数据连接的连接状态为活跃状态。即设置心跳机制，通过发送心跳包至终端，判断是否在预设的时长阈值内接收到该心跳包的心跳响应信息，以确定是否需要维持该第一数据连接的连接状态为活跃状态，从而维持服务器与终端之间的第一数据连接为长连接。具体地，可以预先设置心跳包的发送周期以及时长阈值；然后按照心跳包发送周期，通过第一数据连接向终端发送心跳包；判断是否在预设的时长阈值内接收到终端针对该心跳包返回的心跳响应信息；当确定在预设的时长阈值内接收到终端针对该心跳包返回的心跳响应信息，则控制第一数据连接对应的连接状态为活跃状态，当确定在预设的时长阈值内未接收到终端针对该心跳包返回的心跳响应信息，则控制第一数据连接对应的连接状态为断开状态，即断开与终端间的第一数据连接，减少网络资源浪费。

[0125] 步骤102：接收到终端基于第一数据连接传输的数据包。

[0126] 这里，服务器建立与终端间用于数据传输的第一数据连接后，通过该第一数据连接进行数据包的传输。在本申请实施例中，该第一数据连接既可支持可靠数据传输，也可支持不可靠数据传输。当需要发送可靠数据(比如握手时期的数据、握手成功后的信令数据等)时，可以将数据封装成可靠数据类型的数据包；当需要发送不可靠数据(比如音视频数据、频繁移动的坐标数据等)时，可以将数据封装成不可靠数据类型的数据包。具体地，可靠数据类型的数据包和不可靠数据类型的数据包采用不同的封包格式，且数据包中携带用于指示数据类型的报文字段，该报文字段用于存放用于指示数据类型的数据标识，即不同数据类型的数据包的数据标识是不同的。

[0127] 这里，该可靠数据类型的数据包，即为传输过程中严格要求的不可丢弃的数据(比如握手时期的数据、握手成功后的信令数据等)，具体是需要采用可靠传输方式进行传输的数据；该不可靠数据类别的数据包，即为传输过程中可以部分丢弃的数据(比如音视频数据、频繁移动的坐标数据等)，具体是需要采用不可靠传输方式进行传输的数据。这里，可靠传输方式所传输的数据，如果因为网络抖动或者拥塞等原因，接收端没有接收到数据，则发送方需要重传该数据，其中，是否接收到该数据根据接收方是否回复ACK等响应消息来判断；而不可靠传输方式是与可靠传输方式相对应的一种传输方式，不可靠传输方式的发送方不管接收方是否接收到数据，只管发送。

[0128] 步骤103：对数据包进行解析，以确定数据包对应的数据类型。

[0129] 这里，服务器在接收到终端基于第一数据连接发送的数据包后，对数据包进行解析，以确定数据包对应的数据类型。从而针对不同数据类型的数据包，采用不同的数据处理方式进行处理。

[0130] 在一些实施例中，服务器可通过如下方式确定数据包对应的数据类型：提取数据包的包头部分；对包头部分进行解析，得到用于指示数据类型的报文字段；将报文字段所指示的数据类型，确定为数据包对应的数据类型。

[0131] 在实际应用中，可靠数据类型的数据包和不可靠数据类型的数据包采用不同的封包格式，且数据包中携带用于指示数据类型的报文字段，该报文字段用于存放用于指示数据类型的数据标识，即不同数据类型的数据包的数据标识是不同的。当服务器接收到终端发送的数据包后，提取数据包的包头部分，然后对包头部分进行解析，得到用于指示数据类型的报文字段，该报文字段中存放有用于指示数据类型的数据标识。此时，将报文字段中数据标识所指示的数据类型，确定为所接收的数据包对应的数据类型。

[0132] 步骤104：当确定数据包的数据类型为不可靠数据类型时，控制不向终端返回对应数据包的接收响应消息。

[0133] 基于上述实施例，当确定所接收的数据包的数据类型为不可靠数据类型时，则服务器控制不向该终端返回对应数据包的接收响应消息。相应的，终端针对不可靠数据类型的数据包尽管发送即可。

[0134] 当确定所接收的数据包的数据类型为可靠数据类型时，则服务器向该终端返回对应数据包的接收响应消息，以通知终端已接收到该数据包。相应的，若终端未接收到针对该数据包的接收响应消息，则重新发送该数据包至服务器。

[0135] 在一些实施例中，当终端的互联网协议地址发生变更时，通过终端的第一套接字接口，建立与终端间的第二数据连接，以将与终端间的数据连接从第一数据连接迁移至第

二数据连接;其中,第二数据连接的连接标识与第一数据连接的连接标识相一致。

[0136] 这里,当终端的IP地址发生变更时,比如当终端的网络类型由Wi-Fi变更为蜂窝网络时,由于和原始IP地址绑定的套接字接口(即socket)已经不能使用,此时,则需要针对变更后的IP地址,重新创建相应的第一套接字接口,并将重新创建的第一套接字接口与变更后的IP地址进行绑定,从而通过该重新创建的第一套接字接口进行数据包的传输。

[0137] 基于此,终端可通过重新创建的第一套接字接口,基于第一数据连接的连接标识,与服务器进行数据传输,以建立第二数据连接;同样的,服务器也基于该第二数据连接与终端进行数据传输。如此,将终端与服务器间的数据连接由第一数据连接迁移至第二数据连接。

[0138] 这里,第二数据连接的连接标识与第一数据连接的连接标识相一致,保证在IP地址发生变更时,终端和服务器之间仍可通过连接标识进行数据传输,避免出现由于IP地址变更导致数据连接断连的情况。该第二数据连接和第一数据连接,实则为终端与服务器之间所建立的数据连接对应的多个网络路径,这里将终端与服务器间的数据连接由第一数据连接迁移至第二数据连接,实则是通过连接标识,将终端与服务器间的数据连接从原始网络路径迁移至新的网络路径、且保证数据连接迁移过程中数据能够正常传输,从而实现了IP地址的无感知切换。

[0139] 这里,终端可通过如下方式确定终端的IP地址是否发生变更:对终端的IP地址对应的原始套接字接口的数据传输失败事件、以及IP地址的变更通知消息中至少之一进行监听;当监听到IP地址对应的原始套接字接口的数据传输失败事件、以及IP地址的的变更通知消息中至少之一时,则确定终端的IP地址发生了变更。

[0140] 在实际应用中,可在每个操作系统(包括Windows/Mac/Android/iOS系统)下,根据各操作系统的代码逻辑,分别实现IP地址的变更通知消息的监听。从而当终端监听到IP地址的变更通知消息时,则确定终端的IP地址发生了变更。

[0141] 在实际应用中,还可针对各操作系统,设置通用的IP地址变更监听方式,即对IP地址对应的原始套接字接口的数据传输失败事件进行监听。当监听到IP地址对应的原始套接字接口的数据传输失败事件时,则确定终端的IP地址发生了变更。如此,则不需要针对不同的操作系统分别实现相应的IP地址的变更通知消息的监听,降低实现逻辑的复杂性,提高效率。

[0142] 在一些实施例中,建立与终端间的第二数据连接之后,服务器可通过如下方式对所建立的第二数据连接进行路径验证:基于连接标识,通过第二数据连接,发送包括第一验证信息的探测数据包至终端;接收到终端针对探测数据包返回的响应数据包,响应数据包包括第二验证信息;基于第二验证信息对第一验证信息进行验证,当得到表征验证通过的验证结果时,基于第二数据连接进行与终端间的数据传输。

[0143] 在实际应用中,当终端与服务器间的数据连接发生迁移后,即由第一数据连接迁移至第二数据连接后,需要对第二数据连接进行路径验证,即验证第二数据连接的数据可达性。这里,数据可达性的验证,即为验证第二数据连接是否能够支持数据在终端和服务器之间传输,即终端或服务器分别作为接收方时,是否能够接收到发送方所发送的数据。

[0144] 具体地,服务器可基于终端与服务器间的数据连接的连接标识,通过第二数据连接,发送包含第一验证信息的探测数据包至终端,在实际实施时,该第一验证信息为一个不

可预测的随机值;终端接收到包含第一验证信息的探测数据包后,获取其中包含的第一验证信息,将该第一验证信息作为第二验证信息携带于针对探测数据包的响应数据包中返回至服务器。服务器接收到该响应数据包,获取该数据包中包含的第二验证信息,然后通过第一验证信息对第二验证信息进行验证,即判断第二验证信息与第一验证信息是否一致;当第二验证信息与第一验证信息一致时,则得到表征验证通过的验证结果,此时,则认为第二数据连接的数据可达。在后续的数据传输过程中,则基于第二数据连接进行与终端间的数据传输。

[0145] 在一些实施例中,建立与终端间的第二数据连接之后,终端可通过如下方式对所建立的第二数据连接进行路径验证:接收到终端基于连接标识,通过第二数据连接传输的包括第三验证信息的探测数据包;发送针对探测数据包的响应数据包至终端,响应数据包包括第四验证信息;其中,该第四验证信息,用于供终端基于第三验证信息对第四验证信息进行验证,当得到表征验证通过的验证结果时,基于第二数据连接进行数据传输。

[0146] 在实际应用中,不只服务器需要针对第二数据连接的数据可达性进行验证,终端在IP地址变更后,也需要针对第二数据连接的数据可达性进行验证。具体地,终端基于终端与服务器间的数据连接的连接标识,通过第二数据连接,发送包含第三验证信息的探测数据包至服务器,在实际实施时,该第三验证信息为一个不可预测的随机值;服务器接收到包含第三验证信息的探测数据包后,获取其中包含的第三验证信息,将该第三验证信息作为第四验证信息携带于针对探测数据包的响应数据包中返回至终端。终端接收到该响应数据包,获取该数据包中包含的第四验证信息,然后通过第三验证信息对第四验证信息进行验证,即判断第四验证信息与第三验证信息是否一致;当第四验证信息与第三验证信息一致时,则得到表征验证通过的验证结果,此时,则认为第二数据连接的数据可达。在后续的数据传输过程中,则基于第二数据连接进行与终端间的数据传输。

[0147] 在一些实施例中,服务器还对目标互联网协议地址是否发生变更进行监测,得到监测结果,该目标互联网协议地址归属于与终端建立第一数据连接的服务器;当监测结果表征目标互联网协议地址发生变更时,针对变更后的目标互联网协议地址创建第二套接字接口,并基于第二套接字接口,建立与终端间的第三数据连接,以将与终端间的数据连接从第一数据连接迁移至第三数据连接;其中,第三数据连接的连接标识与第一数据连接的连接标识相一致。

[0148] 这里,服务器还可以对自身的IP地址,即目标互联网协议地址是否发生变更进行监测,得到监测结果。当服务器的目标IP地址发生变更时,由于和原始IP地址绑定的套接字接口(即socket)已经不能使用,此时,则需要针对变更后的IP地址,重新创建相应的第二套接字接口,并将重新创建的第二套接字接口与变更后的目标IP地址进行绑定,从而通过该重新创建的第二套接字接口进行数据包的传输。

[0149] 基于此,服务器可通过重新创建的第二套接字接口,基于第一数据连接的连接标识,与终端进行数据传输,以建立第三数据连接;同样的,终端也基于该第三数据连接与服务器进行数据传输。如此,将终端与服务器间的数据连接由第一数据连接迁移至第三数据连接。

[0150] 这里,第三数据连接的连接标识与第一数据连接的连接标识相一致,保证在IP地址发生变更时,终端和服务器之间仍可通过连接标识进行数据传输,避免出现由于IP地址

变更导致数据连接断连的情况。该第三数据连接和第一数据连接，实则为终端与服务器之间所建立的数据连接对应的多个网络路径，这里将终端与服务器间的数据连接由第一数据连接迁移至第三数据连接，实则是通过连接标识，将终端与服务器间的数据连接从原始网络路径迁移至新的网络路径、且保证数据连接迁移过程中数据能够正常传输，从而实现了IP地址的无感知切换。

[0151] 在一些实施例中，服务器可通过如下方式对目标互联网协议地址是否发生变更进行监测，得到监测结果：对目标互联网协议地址对应的原始套接字接口的数据传输失败事件、以及目标互联网协议地址的变更通知消息中至少之一进行监测；当监测到目标互联网协议地址对应的原始套接字接口的数据传输失败事件、以及目标互联网协议地址的变更通知消息中至少之一时，得到表征目标互联网协议地址发生变更的监测结果。

[0152] 在一些实施例中，服务器建立与终端间的第三数据连接之后，可通过如下方式对第三数据连接的数据可达性进行验证：基于连接标识，对第三数据连接的数据可达性进行验证，得到验证结果；当验证结果表征第三数据连接的数据可达时，基于第三数据连接进行与终端间的数据传输。

[0153] 在实际应用中，当终端与服务器间的数据连接发生迁移后，即由第一数据连接迁移至第三数据连接后，需要对第三数据连接进行路径验证，即验证第三数据连接的数据可达性。这里，数据可达性的验证，即为验证第三数据连接是否能够支持数据在终端和服务器之间传输，即终端或服务器分别作为接收方时，是否能够接收到发送方所发送的数据。具体地，第三数据连接的路径验证方式也可以采用第二数据连接的路径验证方式，在本申请实施例中将不再赘述。

[0154] 应用本申请上述实施例，创建与终端间用于数据传输的第一数据连接，该第一数据连接的连接标识，在终端的互联网协议地址或端口发生变化时保持不变；当接收到该终端基于第一数据连接传输的数据包时，对该数据包进行解析，以确定该数据包对应的数据类型；当确定数据包的数据类型为不可靠数据类型时，则控制不向该终端返回对应数据包的接收响应消息。如此，通过解析数据包的数据类型，针对不可靠数据类型的数据包，则不返回对应的接收响应消息，从而避免了终端重传不可靠数据类型的数据包的情况，减少网络资源的不必要占用，提高网络资源的利用率。

[0155] 下面继续说明本申请实施例提供的数据传输方法。在一些实施例中，本申请实施例提供的数据传输方法可由服务器或终端单独实施，或由服务器及终端协同实施，下面以终端实施为例说明本申请实施例提供的数据传输方法。参见图3B，图3B是本申请实施例提供的数据传输方法的流程示意图，本申请实施例提供的数据传输方法包括：

[0156] 步骤201：终端创建与服务器间用于数据传输的第一数据连接。

[0157] 其中，该第一数据连接的连接标识，在服务器的互联网协议地址或端口发生变化时保持不变。

[0158] 这里，终端在与服务器进行数据通信之前，可以预先创建服务器与终端之间的用于数据传输的第一数据连接。在第一数据连接创建完成后，服务器与终端通过该第一数据连接进行数据包的发送与接收。在实际应用中，终端可以设置有应用客户端，比如会议客户端、学习客户端等，该服务器可以为相应的应用客户端的后台服务器。

[0159] 在本申请实施例中，该第一数据连接的连接标识，在终端和服务器中任一方的互

联网协议地址(即IP地址)或端口发送变化时保持不变,即在终端和服务器中任一方的IP地址或端口发送变化时,该终端与服务器之间的数据通道并不会断,双方均可基于该第一数据连接的连接标识进行数据包的传输。如此,当终端或者服务器的IP地址发生变更时,并不影响双方之间的数据传输,而对于用户来说,IP地址的变更是无感知的,也不会出现网络断连、需要重新连接等情况。

[0160] 在实际应用中,该第一数据连接是基于QUIC协议建立的,QUIC连接的连接标识是一个64位的连接ID,用户在Wi-Fi和蜂窝网络(即Celluar)之间切换时,无论是IP地址或者端口(Port)发生变化,QUIC连接中的连接ID保持不变,终端与服务器双方仍可基于该QUIC连接的连接标识进行数据包的传输,因此不需要重新经过协议握手等过程创建连接,如此QUIC连接迁移可达到用户无感知的网络类型切换的技术效果。

[0161] 步骤202:基于第一数据连接,发送数据包至服务器。

[0162] 其中,该数据包的数据类型包括不可靠数据类型,该不可靠数据类型,用于当服务器确定数据包的数据类型为不可靠数据类型时,控制不返回对应数据包的接收响应消息。

[0163] 当终端与服务器间的第一数据连接建立后,终端通过第一数据连接,发送数据包至服务器。在本申请实施例中,该第一数据连接既可支持可靠数据传输,也可支持不可靠数据传输。当需要发送可靠数据(比如握手时期的数据、握手成功后的信令数据等)时,可以将数据封装成可靠数据类型的数据包;当需要发送不可靠数据(比如音视频数据、频繁移动的坐标数据等)时,可以将数据封装成不可靠数据类型的数据包。具体地,可靠数据类型的数据包和不可靠数据类型的数据包采用不同的封包格式,且数据包中携带用于指示数据类型的报文字段,该报文字段用于存放用于指示数据类型的数据标识,即不同数据类型的数据包的数据标识是不同的。

[0164] 这里,该可靠数据类型的数据包,即为传输过程中严格要求的不可丢弃的数据(比如握手时期的数据、握手成功后的信令数据等),具体是需要采用可靠传输方式进行传输的数据;该不可靠数据类别的数据包,即为传输过程中可以部分丢弃的数据(比如音视频数据、频繁移动的坐标数据等),具体是需要采用非可靠传输方式进行传输的数据。这里,可靠传输方式所传输的数据,如果因为网络抖动或者拥塞等原因,接收端没有接收到数据,则发送方需要重传该数据,其中,是否接收到该数据根据接收方是否回复ACK等响应消息来判断;而非可靠传输方式是与可靠传输方式相对应的一种传输方式,非可靠传输方式的发送方不管接收方是否接收到数据,只管发送。

[0165] 服务器在接收到终端基于第一数据连接发送的数据包后,对数据包进行解析,以确定数据包对应的数据类型。从而针对不同数据类型的数据包,采用不同的数据处理方式进行处理。当确定所接收的数据包的数据类型为不可靠数据类型时,则服务器控制不向该终端返回对应数据包的接收响应消息。相应的,终端针对不可靠数据类型的数据包尽管发送即可。

[0166] 当确定所接收的数据包的数据类型为可靠数据类型时,则服务器向该终端返回对应数据包的接收响应消息,以通知终端已接收到该数据包。相应的,若终端未接收到针对该数据包的接收响应消息,则重新发送该数据包至服务器。

[0167] 应用本申请上述实施例,创建与终端间用于数据传输的第一数据连接,该第一数据连接的连接标识,在终端的互联网协议地址或端口发生变化时保持不变;当接收到该终

端基于第一数据连接传输的数据包时，对该数据包进行解析，以确定该数据包对应的数据类型；当确定数据包的数据类型为不可靠数据类型时，则控制不向该终端返回对应数据包的接收响应消息。如此，通过解析数据包的数据类型，针对不可靠数据类型的数据包，则不返回对应的接收响应消息，从而避免了终端重传不可靠数据类型的数据包的情况，减少网络资源的不必要占用，提高网络资源的利用率。

[0168] 接下来继续对本申请实施例提供的数据传输方法进行说明，本申请实施例提供的数据传输方法可以由终端及服务器协同实施。参见图4，图4为本申请实施例提供的数据传输方法的流程示意图，本申请实施例提供的数据传输方法包括：

[0169] 步骤301：终端与服务器建立用于数据传输的第一数据连接。

[0170] 步骤302：终端基于第一数据连接，发送数据包至服务器。

[0171] 步骤303：服务器接收到终端基于第一数据连接传输的数据包，对数据包进行解析，确定数据包对应的数据类型。

[0172] 步骤304：判断数据包对应的数据类型是否为不可靠数据类型，若否，执行步骤305；若是，执行步骤306。

[0173] 步骤305：向终端返回对应数据包的接收响应消息。

[0174] 步骤306：控制不向终端返回对应数据包的接收响应消息。

[0175] 步骤307：终端接收到对应数据包的接收响应消息。

[0176] 应用本申请上述实施例，创建与终端间用于数据传输的第一数据连接，该第一数据连接的连接标识，在终端的互联网协议地址或端口发生变化时保持不变；当接收到该终端基于第一数据连接传输的数据包时，对该数据包进行解析，以确定该数据包对应的数据类型；当确定数据包的数据类型为不可靠数据类型时，则控制不向该终端返回对应数据包的接收响应消息。如此，通过解析数据包的数据类型，针对不可靠数据类型的数据包，则不返回对应的接收响应消息，从而避免了终端重传不可靠数据类型的数据包的情况，减少网络资源的不必要占用，提高网络资源的利用率。

[0177] 下面将说明本申请实施例在一个实际的应用场景中的示例性应用。

[0178] 接下来在对本申请实施例提供的数据传输方法进行说明之前，首先说明本申请实施例提供的用于数据传输的长连接的创建方法。如图5所示，图5是本申请实施例提供的用于数据传输的长连接的创建方法的流程示意图，包括：

[0179] 步骤401：客户端发送SYN握手信号至服务器。

[0180] 步骤402：服务器返回应答信号SYN+ACK至客户端，以表示接收到SYN握手信号。

[0181] 步骤403：客户端发送ACK响应信号至服务器，以建立与服务器间的TCP连接。

[0182] 这里，在实际应用中，该客户端可以是会议客户端，用户可通过终端安装并运行该客户端，实现相应的客户端功能。作为示例，参见图6，图6是本申请实施例提供的客户端的应用示意图，这里，该客户端为会议客户端中，包括预定会议、加入会议、会议中打开摄像头、麦克风等功能场景，该每个场景下均需要客户端和服务器建立长连接来进行数据的交互和传输。因此，客户端需要与服务器之间建立长连接，在本申请实施例中，该长连接可通过TCP+TLS+XMPP共同建立实现，其中，上述步骤401-403即为TCP协议的握手流程，在经过步骤401-403后，则客户端与服务器之间建立了TCP连接。

[0183] 具体地，参见图7，图7是本申请实施例提供的TCP连接的建立流程示意图，包括：

[0184] 步骤501:客户端发送SYN握手信号至服务器;步骤502:服务器返回应答信号SYN+ACK至客户端,以表示接收到SYN握手信号;步骤503:客户端发送ACK响应信号至服务器,以建立与服务器间的TCP连接;步骤504:客户端发送应用数据(Application Data)至服务器;步骤505:服务器返回应用数据(Application Data)至客户端。如此实现基于TCP连接的数据传输。

[0185] 步骤404:客户端发送问询数据包(XMPP Stream Start)至服务器,以请求服务器的流特征字段。

[0186] 这里,客户端与服务器之间建立了TCP连接后,开始执行XMPP握手流程。

[0187] 步骤405:服务器返回携带会话ID的应答数据包ACK。

[0188] 这里,该会话ID即为服务器为客户端本次会话所分配的标识。

[0189] 步骤406:服务器下发特征字段Server Feature(TLS)至客户端。

[0190] 步骤407:客户端发送TLS握手的握手请求(TLS start)至服务器。

[0191] 步骤408:服务器发送针对TLS握手的握手请求的响应信息ACK至客户端。

[0192] 步骤409:客户端发送客户端问询数据包Client Hello至服务器。

[0193] 这里,该Client Hello数据包携带客户端的随机数random、密码套件ciph ers以及可用的协议版本号version等信息。

[0194] 步骤410:服务器返回服务器问询数据包Server Hello至客户端。

[0195] 这里,该Server Hello数据包携带服务器的随机数random、密码套件ciph ers以及证书cert等信息。

[0196] 步骤411:客户端发送客户端问询结束数据包Client Hello Finish至服务器。

[0197] 步骤412:服务器返回服务器问询结束数据包Server Hello Finish至客户端。

[0198] 这里,客户端和服务器之间执行标准TLS握手,并在握手完成后,建立TLS连接,以实现通道的TLS加密。

[0199] 具体地,参见图8,图8是本申请实施例提供的TLS连接的建立流程示意图,包括:步骤601:客户端发送客户端问询数据包Client Hello至服务器;步骤602:服务器返回服务器问询数据包Server Hello至客户端;步骤603:客户端发送客户端问询结束数据包Client Hello Finish至服务器;步骤604:服务器返回服务器问询结束数据包Server Hello Finish至客户端;步骤605:客户端发送应用数据至服务器;步骤606:服务器返回应用数据至客户端。如此实现基于TLS连接的数据加密传输,保证数据安全性。

[0200] 步骤413:客户端发送问询数据包Stream Request至服务器,以请求服务器下发流特征字段并进行SASL认证。

[0201] 步骤414:服务器发送包含流标志的响应数据包ACK至客户端。

[0202] 步骤415:服务器发送流特征字段Server Feature(SASL)至客户端,并携带所支持的SASL加密方式。

[0203] 步骤416:客户端选择加密方式,并发送携带用户信息的SASL验证开始数据包(SASL challenge start)至服务器。

[0204] 这里,在实际应用中,客户端从所支持的SASL加密方式中选择PLAIN加密方式。

[0205] 步骤417:服务器发送针对接收到的数据包的SASL验证结束数据包(SA SL challenge end)至客户端。

[0206] 这里,服务器针对SASL的验证成功。

[0207] 步骤418:客户端发送初始流Stream Request至服务器。

[0208] 步骤419:服务器发送给客户端流标志作为响应数据包ACK。

[0209] 步骤420:服务器响应并返回支持的流特征字段Server Feature,以指示说明客户端需要开始绑定资源bind resource流程。

[0210] 步骤421:客户端发送绑定资源请求Bind Rosource Request至服务器。

[0211] 步骤422:服务器判断并返回绑定资源结果Bind Resourca Response至客户端。

[0212] 步骤423:客户端发送应用数据Application Data至服务器。

[0213] 步骤424:服务器返回应用数据Application Data至客户端。

[0214] 如此,上述用于数据传输的长连接是基于TCP+TLS协议和的长连接通信技术。其中,TCP协议提供了可靠传输通道,TLS加密协议为通道提供了安全保障。然而针对本申请实施例提供的上述用于数据传输的长连接的创建方法,申请人进一步分析发现,从图7和图8可以看出,在正式开始数据通信之前,TCP建立连接需要1.5个RTT,完成通道的TLS加密需要2个RTT,而基于上述流程建立用于数据传输的长连接(即客户端与服务器之间的长连接)一共需要11个RTT。

[0215] 而统计数据表明,在正常网络情况下,只有69.9%的用户能够在1s之内完成握手过程,开始客户端的其他会议功能使用,整个握手过程,考虑到用户体验,连接建立的超时时间是30s,而所有登录失败的用户中,有41.53%的用户是因为连接建立超时导致登录失败,因为弱网情况下,RTT较大,如果连接建立的流程复杂的话必然导致完成握手的耗时较多,超时的可能性更大,因此,简化握手环节,减少登录耗时,提高登录成功率和弱网抗性是亟待解决的问题。

[0216] 且,目前客户端与服务器之间的长连接是基于TCP协议的,TCP协议使用一个四元组(源IP地址、目的IP地址、源端口和目的端口)来标识一个长连接。当用户设备网络在wifi和cellular之间切换时,源IP地址会发生变化,因此TCP协议连接无法支持在wifi和cellular之间无缝切换,也就导致一旦用户切换网络,整个长连接必须断开重连,否则数据无法继续传输,表现在客户端侧就是会出现“网络不稳定,正在连接”等掉线重连的情况,参见图9,图9是本申请实施例提供的互联网协议地址变更时客户端的表现示意图。在长连接断开重连期间,所有指令数据都无法发送接收,由于日常使用过程中,进出电梯等或者偶现wifi信号不好,导致wifi和cellular互相频繁切换的场景是非常常见的,如果每次都需要断开重连,会非常影响用户的体验。

[0217] 基于此,本申请实施例还提供一种用于数据传输的长连接的创建方法,该用于数据传输的长连接的创建方法是基于QUIC协议建立的。接下来首先对本申请提供的QUIC连接的建立流程进行说明,如图10所示,图10是本申请实施例提供的QUIC连接的建立流程示意图,包括:

[0218] 步骤701:客户端发送初始问询数据包Inchoate Client Hello至服务器。

[0219] 步骤702:服务器返回针对初始问询数据包Inchoate Client Hello响应数据包Rejection至客户端。

[0220] 步骤703:客户端发送完整问询数据包Complete Client Hello至服务器。

[0221] 这里,该Complete Client Hello可以携带客户端的加密信息,并与加密的应用数

据Encript Application Data一起发送至服务器,即表明在一个RTT内完成QUIC握手和数据加密的操作。

[0222] 步骤704:服务器基于Complete Client Hello返回加密应用数据Encript Application Data。

[0223] 如图10所示,可知QUIC连接在建立时,只需要1个RTT,客户端接收到服务器发送的针对问询数据包Rejection之后,会在下次发送数据包时带上客户端的加密信息,并与应用数据一起发送至服务器,从而在一个RTT内完成QUI C握手和数据加密的操作,相较于TCP+TLS,减省了2.5个RTT。

[0224] 如此,会议客户端与服务器之间的长连接可基于QUIC协议来实现,参见图11,图11是本申请实施例提供的用于数据传输的长连接的创建方法的流程示意图,包括:

[0225] 步骤801:客户端发送初始问询数据包Inchoate Client Hello至服务器。

[0226] 步骤802:服务器返回针对Inchoate Client Hello响应数据包Rejection至客户端。

[0227] 步骤803:客户端发送问询数据包至服务器,以请求服务器的流特征字段;同时发送完整问询数据包Complete Client Hello至服务器。

[0228] 步骤804:服务器返回携带会话ID的应答数据包ACK;同时基于Comple te Client Hello返回加密应用数据Encript Application Data。

[0229] 步骤805:客户端发送问询数据包Stream Request至服务器,以请求服务器的流特征字段。

[0230] 步骤806:服务器发送流标志作为响应数据包ACK至客户端。

[0231] 步骤807:服务器发送流特征字段Server Feature(SASL)至客户端,并携带所支持的SASL加密方式。

[0232] 步骤808:客户端选择加密方式,并发送携带用户信息的SASL验证开始数据包(SASL challenge start)至服务器。

[0233] 这里,在实际应用中,客户端从所支持的SASL加密方式中选择PLAIN加密方式。

[0234] 步骤809:服务器发送针对接收到的数据包的SASL验证结束数据包(SA SL challenge end)至客户端。

[0235] 这里,服务器针对SASL的验证成功。

[0236] 步骤810:客户端发送初始流Stream Request至服务器。

[0237] 步骤811:服务器发送给客户端流标志作为响应数据包ACK。

[0238] 步骤812:服务器响应并返回支持的流特征字段Server Feature,以指示说明客户端需要开始绑定资源bind resource流程。

[0239] 步骤813:客户端发送绑定资源请求Bind Rosource Request至服务器。

[0240] 步骤814:服务器判断并返回绑定资源结果Bind Resourca Response至客户端。

[0241] 步骤815:客户端发送应用数据Application Data至服务器。

[0242] 步骤816:服务器返回应用数据Application Data至客户端。

[0243] 由此可见,引入QUIC协议的长连接建立的过程已经缩减到7个RTT,相较于上述图5所示的长连接建立的过程减少了4个RTT,而理论上长连接建立耗时则能够减少36%。

[0244] 同时,由于QUIC协议是使用连接标识(ConnectionID)来唯一标识一个连接的,当

客户端源IP地址发生变更时(比如网络在wifi和cellular之间切换时),基于QUIC协议的长连接的连接标识不变,如此保持客户端和服务器之间的数据通道不断。具体地,参见图12,图12是本申请实施例提供的数据连接的连接迁移的流程示意图,包括:

[0245] 步骤901:客户端针对IP1建立第一套接字接口socket,并将第一套接字接口socket与IP1绑定。

[0246] 步骤902:客户端发送应用数据请求Application Data Request至服务器。

[0247] 步骤903:服务器返回针对应用数据请求的应用数据响应Application Data Response至客户端。

[0248] 步骤904:客户端监听到网络状态更改通知network state change notify,由IP1变更为IP2。

[0249] 步骤905:客户端针对IP2建立第二套接字接口socket,并将第二套接字接口socket与IP2绑定。

[0250] 这里,当客户端的网络在wifi和cellular之间切换时,客户端会收到系统的网络状态更改通知,此时IP1变更为IP2,原始IP1绑定的第一套接字接口soc ket已经无法继续通信。因此需要新建一个socket跟变更后的IP2进行绑定,以继续进行网络数据包的收发,同时还对第一套接字接口socket缓存中发送失败的数据包进行重发,从而完成客户端与服务器之间的数据连接的网络通道的迁移。在实际应用中,可在每个操作系统(包括Windows/Mac/Android/iOS系统)下,根据各操作系统的代码逻辑,分别实现IP地址的网络状态更改通知的监听。从而当客户端监听到IP地址的网络状态更改通知时,则确定客户端的IP地址发生了变更,从而及时创建新的socket以完成数据连接的迁移。

[0251] 步骤906:基于第二套接字接口socket实现数据连接的连接迁移connectio n migration。

[0252] 步骤907:客户端重传第一套接字接口socket缓存中发送失败的数据包。

[0253] 步骤908:服务器返回应用数据响应至客户端。

[0254] 进一步地,参见图13,图13是本申请实施例提供的连接迁移的流程示意图,包括:

[0255] 步骤1001:客户端针对IP1建立第一套接字接口socket,并将第一套接字接口socket与IP1绑定。

[0256] 步骤1002:客户端发送应用数据请求Application Data Request至服务器。

[0257] 步骤1003:服务器返回针对应用数据请求的应用数据响应Application Dat a Response至客户端。

[0258] 步骤1004:客户端发送应用数据请求至服务器。

[0259] 此时,客户端的网络在wifi和cellular之间切换,由IP1变更为IP2。

[0260] 步骤1005:服务器返回应用数据响应发送失败,客户端监听到socket读取失败事件。

[0261] 步骤1006:客户端针对IP2建立第二套接字接口socket,并将第二套接字接口socket与IP2绑定。

[0262] 步骤1007:基于第二套接字接口socket实现数据连接的连接迁移connecti on migration。

[0263] 步骤1008:客户端重传第一套接字接口socket缓存中发送失败的数据包。

[0264] 步骤1009:服务器返回应用数据响应至客户端。

[0265] 这里,客户端不再对操作系统的物理网络的网络状态变更通知进行监听,而是对socket读写失败事件进行监听,由此来判断当前是否需要重建socket绑定新的ip以进行网络数据传输。如此,既不需要分别监听4个平台(Windows/Mac/Android/iOS)的物理网络变化,又可以成功监测到网络切换导致的socket读写失败,并且及时建立新的传输通道完成连接迁移。

[0266] 继续地,在客户端和服务器之间的连接迁移后,还需要对迁移后的网络路径进行数据可达性验证。参见图14,图14是本申请实施例提供的路径验证的流程示意图,包括:

[0267] 步骤1101:连接迁移之前,客户端通过IP1使用非探测包(Non-probing Packet)和服务器进行数据通信。

[0268] 这里,非探测包(Non-probing Packet)即为传输的应用数据相关的数据包。

[0269] 步骤1102:客户端的IP变成IP2,客户端发送包含路径验证帧PATH_CH ALLENGE帧的探测包(Probing Packet)至服务器,该PATH_CHALLENGE帧包含一个不可预测的随机值。

[0270] 这里,当客户端的IP地址发生变更后并完成连接迁移后,客户端在迁移后的新的网络路径启动路径验证,验证新网络路径的数据可达性(即reachability)。

[0271] 步骤1103:服务器接收到客户端发送的包含PATH_CHALLENGE帧的探测包,返回响应探测包(Probing Packet)至客户端,并在响应探测包包含的路径响应帧PATH_RESPONSE帧携带通过PATH_CHALLENGE接收到的随机值。

[0272] 这里,客户端接收到包含的PATH_RESPONSE的响应探测包,验证响应探测包的数据体payload里面的随机值是否与PATH_CHALLENGE帧的随机值一致,若一致,则表征路径的数据可达。

[0273] 步骤1104:服务器发送包含PATH_CHALLENGE帧的探测包(Probing Pa cket)至客户端,该PATH_CHALLENGE帧包含一个不可预测的随机值。

[0274] 这里,服务器也需要发送PATH_CHALLENGE帧对客户端进行路径验证。

[0275] 步骤1105:客户端接收到客户端发送的包含PATH_CHALLENGE帧的探测包,返回响应探测包(Probing Packet)至服务器,并在响应探测包包含的PATH_RESPONSE帧携带通过PATH_CHALLENGE接收到的随机值。

[0276] 这里,服务器接收到包含的PATH_RESPONSE的响应探测包,验证响应探测包的数据体payload里面的随机值是否与PATH_CHALLENGE帧的随机值一致,若一致,则表征路径的数据可达。

[0277] 步骤1106:连接迁移之后,客户端通过IP2使用非探测包(Non-probing Packet)和服务器进行数据通信。

[0278] 由此可见,基于QUIC协议之后的长连接能够做到在IP地址发生变更时实现连接迁移,即实现网络wifi和cellular之间的无缝切换,而不必重新建立连接。在实际应用中,用户在使用会议客户端的过程中若发生网络切换场景,音视频数据和开关麦克风/摄像头等等网络操作都可以直接续传,用户是完全无感知的,提高了用户体验。

[0279] 在对本申请实施例提供的用于数据传输的长连接的创建方法说明结束后,接下来在对本申请实施例提供的数据传输方法进行说明。在本申请实施例中,该数据传输方法是基于QUIC协议的长连接实现的。

[0280] 相关技术中，对于信令等不可丢弃的数据，采用可靠传输方式进行传输；而对于其他不严格要求，可以部分丢弃的不可靠数据(例如音量大小的显示数据，频繁移动的坐标数据等)，也同样和可靠数据复用同一条通道都进行可靠传输；那么在网络较差时，可靠传输通道的重传策略会给网络带来较大压力，并且会影响信令等可靠数据的正常传输。而如果为这种不可靠数据专门建立另一条非可靠通道，例如UDP通道等，客户端和服务器就需要同时维护两条通道的状态，导致增加新的通道握手时延，并且实现逻辑复杂，维护成本高。

[0281] 基于此，本申请实施例提供一种数据传输方法，基于QUIC协议建立终端与服务器(即客户端与服务器)之间的数据连接，采用同一数据连接通道同时进行可靠传输和不可靠传输。具体地，在本申请实施例中，该客户端与服务器之间的数据连接既可支持可靠数据传输，也可支持不可靠数据传输。当需要发送可靠数据(比如握手时期的数据、握手成功后的信令数据，包括会议过程中摄像头、麦克风的开关操作指令等)时，可以将数据封装成可靠数据类型的数据包；当需要发送不可靠数据(比如音量大小的显示数据、频繁移动的坐标数据等)时，可以将数据封装成不可靠数据类型的数据包。具体地，可靠数据类型的数据包和不可靠数据类型的数据包采用不同的封包格式，且数据包中携带用于指示数据类型(包括可靠数据类型和不可靠数据类型)的标识，即可靠数据类型的数据包的标识、与不靠数据类型的数据包的标识是不同的。如图15所示，图15是本申请实施例提供的可靠帧格式和不可靠帧格式的示意图，其中图15中A图为可靠帧格式，图15中B图为不可靠帧格式。

[0282] 这里，该可靠数据类型的数据包，即为传输过程中严格要求的不可丢弃的数据(比如握手时期的数据、握手成功后的信令数据等)，具体是需要采用可靠传输方式进行传输的数据；该不可靠数据类别的数据包，即为传输过程中可以部分丢弃的数据(比如音量大小的显示数据、频繁移动的坐标数据等)，具体是需要采用非可靠传输方式进行传输的数据。这里，可靠传输方式所传输的数据，如果因为网络抖动或者拥塞等原因，接收端没有接收到数据，则发送方需要重传该数据，其中，是否接收到该数据根据接收方是否回复ACK等响应消息来判断；而不可靠传输方式是与可靠传输方式相对应的一种传输方式，不可靠传输方式的发送方不管接收方是否接收到数据，只管发送。

[0283] 具体地，当服务器确定所接收的数据包的数据类型为不可靠数据类型时，则服务器不向该终端返回对应数据包的接收响应消息。相应的，终端针对不可靠数据类型的数据包尽管发送即可。当服务器确定所接收的数据包的数据类型为可靠数据类型时，则服务器向该终端返回对应数据包的接收响应消息，以通知终端已接收到该数据包。相应的，若终端未接收到针对该数据包的接收响应消息，则重新发送该数据包至服务器。

[0284] 以远程控制为例，参见图16，图16是本申请实施例提供的应用于远程控制的数据传输的流程示意图，包括：

[0285] 步骤1201：第一客户端和第二客户端均与服务器建立QUIC长连接。

[0286] 这里，第一客户端对应用户A，第二客户端对应用户B，该QUIC长连接是可以是基于QUIC协议和XMPP协议共同建立。

[0287] 步骤1202：第一客户端发送远程控制请求Remote Control Request至服务器。

[0288] 这里，该第一客户端利用QUIC长连接发送远程控制请求的信令，其中携带目的用户B的标识信息，要求开启远程控制服务，对用户B的第二客户端进行远程控制。

[0289] 步骤1203：服务器返回ACK接收响应信息至第一客户端。

[0290] 步骤1204:服务器将第一客户端的远程控制请求通过远程控制推送信息Remote Control Push发送至第二客户端。

[0291] 步骤1205:第二客户端返回第一远程控制响应至服务器,该第一远程控制响应中携带拒绝接受远程控制的通知消息。

[0292] 步骤1206:服务器将远程控制响应消息发送至第一客户端,以通知第一客户端远程控制连接失败。

[0293] 步骤1207:第二客户端返回第二远程控制响应至服务器,该第二远程控制响应中携带同意接受远程控制的通知消息。

[0294] 步骤1208:服务器将远程控制响应消息发送至第一客户端,以通知第一客户端远程控制连接成功。

[0295] 这里,该远程控制响应消息中携带鉴权token,用于后续具体指令控制数据鉴权。

[0296] 步骤1209:第一客户端发送远程控制开始请求至服务器,以要求开始远程控制。

[0297] 步骤1210:服务器将远程控制开始请求发送至第二客户端。

[0298] 步骤1211:返回指令下发成功的远程控制开始响应消息至第一客户端。

[0299] 这里,服务器接收到第一客户端的远程控制开启信令,通过长连接通道向目的用户B发送远程控制开始的push指令,然后向用户A回包指令下发成功的通知消息。

[0300] 步骤1212:第一客户端接收到指令下发成功的远程控制开始响应消息后,拦截鼠标和键盘操作。

[0301] 步骤1213:第二客户端接收到远程控制开始请求,拦截鼠标和键盘操作。

[0302] 步骤1214:第一客户端将控制指令数据封装成不可靠数据包,通过QUIC长连接通道发送至服务器,同时携带鉴权token和目的用户信息。

[0303] 步骤1215:服务器将控制指令数据通过QUIC长连接通道转发至第二客户端。

[0304] 这里,该控制指令数据可以包括鼠标坐标,鼠标操作指令,和键盘操作指令等。但是这些数据中,不需要保证百分百可靠,例如鼠标移动过程中,即使当前网络状态差,有些坐标信息丢失了,只要移动停止的坐标发送成功了,或者通过再次操作,再次发送,远程控制也能成功。因此将远程控制中的控制指令数据可以封装为不可靠数据包,并通过建立的既支持可靠传输也支持不可靠传输的QUIC长连接进行传输。

[0305] 这里,不可靠数据类型的数据还包括会议客户端的画中画模式(即开启屏幕分享同时开启摄像头,摄像头画面可以放在窗口中,如图17A所示,图17A是本申请实施例提供的画中画模式的示意图)中涉及的数据,比如用户指定的任何地方,就是用户可以调整摄像头画面大小,拖动画面位置,画面的坐标和尺寸数据,该类数据也是可以通过不可靠传输方式进行传输的不可靠数据。另外,会议客户端中有音量数据,即指示用户的声音大小的显示数据(如图17B所示,图17B是本申请实施例提供的声音大小显示的示意图),这些数据量都比较大,而且网络差时,丢掉一些也不会有太大影响,该类数据也是可以通过不可靠传输方式进行传输的。

[0306] 应用本申请上述实施例,通过将QUIC协议引入到建立客户端和服务器之间的长连接过程中,利用QUIC的快速握手并进行通道加密的特性和连接迁移特性,取得了较大的优化效果。如图18所示,图18是本申请实施例提供的客户端指标的优化示意图,这里,基于TCP+TLS+XMPP建立长连接、与基于QUIC+XMPP建立长连接相比,如图18中A图所示,成功降低了

登录耗时，根据线上数据统计，登录平均耗时从1224ms降低为555ms，优化效果达54.6％；如图18中B图所示，因为登录耗时高产生的超时问题也得到优化，登录成功率从99.89％提升到99.98％，提升了0.09％；如图18中C图所示，会议客户端中开关麦成功率从98.59％提升到99.84％，优化效果达1.26％。

[0307] 下面继续说明本申请实施例提供的数据传输装置555，在一些实施例中，数据传输装置可采用软件模块的方式实现。参见图19，图19是本申请实施例提供的数据传输装置555的结构示意图，本申请实施例提供的数据传输装置555包括：

[0308] 创建模块5551，用于创建与终端间用于数据传输的第一数据连接，所述第一数据连接的连接标识，在所述终端的互联网协议地址或端口发生变化时保持不变；

[0309] 接收模块5552，用于接收到所述终端基于所述第一数据连接传输的数据包；

[0310] 解析模块5553，用于对所述数据包进行解析，以确定所述数据包对应的数据类型；

[0311] 控制模块5554，用于当确定所述数据包的数据类型为不可靠数据类型时，控制不向所述终端返回对应所述数据包的接收响应消息。

[0312] 在一些实施例中，所述装置还包括：

[0313] 连接状态控制模块，用于获取心跳包发送周期，并按照所述心跳包发送周期，通过所述第一数据连接，发送心跳包至所述终端；

[0314] 当接收到针对所述心跳包的心跳响应信息时，控制所述第一数据连接对应的连接状态为活跃状态。

[0315] 在一些实施例中，当所述第一数据连接为首次建立时，所述创建模块5551，还用于接收到所述终端发送的请求建立数据连接的第一问询数据包；

[0316] 基于所述第一问询数据包，发送相应的应答数据包至所述终端；

[0317] 当接收到所述终端发送的第二问询数据包时，创建与所述终端间用于数据传输的第一数据连接。

[0318] 在一些实施例中，当所述第一数据连接为非首次建立时，所述创建模块5551，还用于接收到所述终端发送的请求建立数据连接的问询数据包；

[0319] 基于所述问询数据包，创建与所述终端间用于数据传输的第一数据连接。

[0320] 在一些实施例中，所述解析模块5553，还用于提取所述数据包的包头部分；

[0321] 对所述包头部分进行解析，得到用于指示数据类型的报文字段；

[0322] 将所述报文字段所指示的数据类型，确定为所述数据包对应的数据类型。

[0323] 在一些实施例中，所述装置还包括：

[0324] 第一迁移模块，用于当所述终端的互联网协议地址发生变更时，通过所述终端的第一套接字接口，建立与所述终端间的第二数据连接，以将与所述终端间的数据连接从所述第一数据连接迁移至所述第二数据连接；

[0325] 其中，所述第二数据连接的连接标识与所述第一数据连接的连接标识相一致。

[0326] 在一些实施例中，所述第一迁移模块，还用于基于所述连接标识，通过所述第二数据连接，发送包括第一验证信息的探测数据包至所述终端；

[0327] 接收到所述终端针对所述探测数据包返回的响应数据包，所述响应数据包包括第二验证信息；

[0328] 基于所述第二验证信息对所述第一验证信息进行验证，当得到表征验证通过的验

证结果时，基于所述第二数据连接进行与所述终端间的数据传输。

[0329] 在一些实施例中，所述第一迁移模块，还用于接收到所述终端基于所述连接标识，通过所述第二数据连接传输的包括第三验证信息的探测数据包；

[0330] 发送针对所述探测数据包的响应数据包至所述终端，所述响应数据包包括第四验证信息；

[0331] 其中，所述第四验证信息，用于供所述终端基于所述第三验证信息对所述第四验证信息进行验证，当得到表征验证通过的验证结果时，基于所述第二数据连接进行数据传输。

[0332] 在一些实施例中，所述装置还包括：

[0333] 第二迁移模块，用于对目标互联网协议地址是否发生变更进行监测，得到监测结果，所述目标互联网协议地址归属于与所述终端建立所述第一数据连接的服务器；

[0334] 当所述监测结果表征所述目标互联网协议地址发生变更时，针对变更后的目标互联网协议地址创建第二套接字接口，并

[0335] 基于所述第二套接字接口，建立与所述终端间的第三数据连接，以将与所述终端间的数据连接从所述第一数据连接迁移至所述第三数据连接；

[0336] 其中，所述第三数据连接的连接标识与所述第一数据连接的连接标识相一致。

[0337] 在一些实施例中，所述第二迁移模块，还用于对所述目标互联网协议地址对应的原始套接字接口的数据传输失败事件、以及所述目标互联网协议地址的变更通知消息中至少之一进行监测；

[0338] 当监测到所述目标互联网协议地址对应的原始套接字接口的数据传输失败事件、以及所述目标互联网协议地址的变更通知消息中至少之一时，得到表征所述目标互联网协议地址发生变更的监测结果。

[0339] 在一些实施例中，所述第二迁移模块，还用于基于所述连接标识，对所述第三数据连接的数据可达性进行验证，得到验证结果；

[0340] 当所述验证结果表征所述第三数据连接的数据可达时，基于所述第三数据连接进行与所述终端间的数据传输。

[0341] 应用本申请上述实施例，创建与终端间用于数据传输的第一数据连接，该第一数据连接的连接标识，在终端的互联网协议地址或端口发生变化时保持不变；当接收到该终端基于第一数据连接传输的数据包时，对该数据包进行解析，以确定该数据包对应的数据类型；当确定数据包的数据类型为不可靠数据类型时，则控制不向该终端返回对应数据包的接收响应消息。如此，通过解析数据包的数据类型，针对不可靠数据类型的数据包，则不返回对应的接收响应消息，从而避免了终端重传不可靠数据类型的数据包的情况，减少网络资源的不必要占用，提高网络资源的利用率。

[0342] 下面继续说明本申请实施例提供的数据传输装置2000，参见图20，图20是本申请实施例提供的数据传输装置2000的结构示意图，本申请实施例提供的数据传输装置2000包括：

[0343] 连接创建模块2010，用于创建与服务器间用于数据传输的第一数据连接，所述第一数据连接的连接标识，在所述服务器的互联网协议地址或端口发生变化时保持不变；

[0344] 发送模块2020，用于基于所述第一数据连接，发送数据包至所述服务器；

[0345] 其中，所述数据包的数据类型包括不可靠数据类型，所述不可靠数据类型，用于当所述服务器确定所述数据包的数据类型为不可靠数据类型时，控制不返回对应所述数据包的接收响应消息。

[0346] 应用本申请上述实施例，创建与终端间用于数据传输的第一数据连接，该第一数据连接的连接标识，在终端的互联网协议地址或端口发生变化时保持不变；当接收到该终端基于第一数据连接传输的数据包时，对该数据包进行解析，以确定该数据包对应的数据类型；当确定数据包的数据类型为不可靠数据类型时，则控制不向该终端返回对应数据包的接收响应消息。如此，通过解析数据包的数据类型，针对不可靠数据类型的数据包，则不返回对应的接收响应消息，从而避免了终端重传不可靠数据类型的数据包的情况，减少网络资源的不必要占用，提高网络资源的利用率。

[0347] 本申请实施例还提供一种电子设备，所述电子设备包括：

[0348] 存储器，用于存储可执行指令；

[0349] 处理器，用于执行所述存储器中存储的可执行指令时，实现本申请实施例提供的数据传输方法。

[0350] 本申请实施例还提供一种计算机程序产品或计算机程序，该计算机程序产品或计算机程序包括计算机指令，该计算机指令存储在计算机可读存储介质中。计算机设备的处理器从计算机可读存储介质读取该计算机指令，处理器执行该计算机指令，使得该计算机设备执行本申请实施例提供的数据传输方法。

[0351] 本申请实施例还提供一种计算机可读存储介质，存储有可执行指令，所述可执行指令被处理器执行时，实现本申请实施例提供的数据传输方法。

[0352] 在一些实施例中，计算机可读存储介质可以是FRAM、ROM、PROM、EP ROM、EEPROM、闪存、磁表面存储器、光盘、或CD-ROM等存储器；也可以是包括上述存储器之一或任意组合的各种设备。

[0353] 在一些实施例中，可执行指令可以采用程序、软件、软件模块、脚本或代码的形式，按任意形式的编程语言(包括编译或解释语言，或者声明性或过程性语言)来编写，并且其可按任意形式部署，包括被部署为独立的程序或者被部署为模块、组件、子例程或者适合在计算环境中使用的其它单元。

[0354] 作为示例，可执行指令可以但不一定对应于文件系统中的文件，可以可被存储在保存其它程序或数据的文件的一部分，例如，存储在超文本标记语言(HTML，Hyper Text Markup Language)文档中的一个或多个脚本中，存储在专用于所讨论的程序的单个文件中，或者，存储在多个协同文件(例如，存储一个或多个模块、子程序或代码部分的文件)中。

[0355] 作为示例，可执行指令可被部署为在一个计算设备上执行，或者在位于一个地点的多个计算设备上执行，又或者，在分布在多个地点且通过通信网络互连的多个计算设备上执行。

[0356] 以上所述，仅为本申请的实施例而已，并非用于限定本申请的保护范围。凡在本申请的精神和范围之内所作的任何修改、等同替换和改进等，均包括在本申请的保护范围之内。

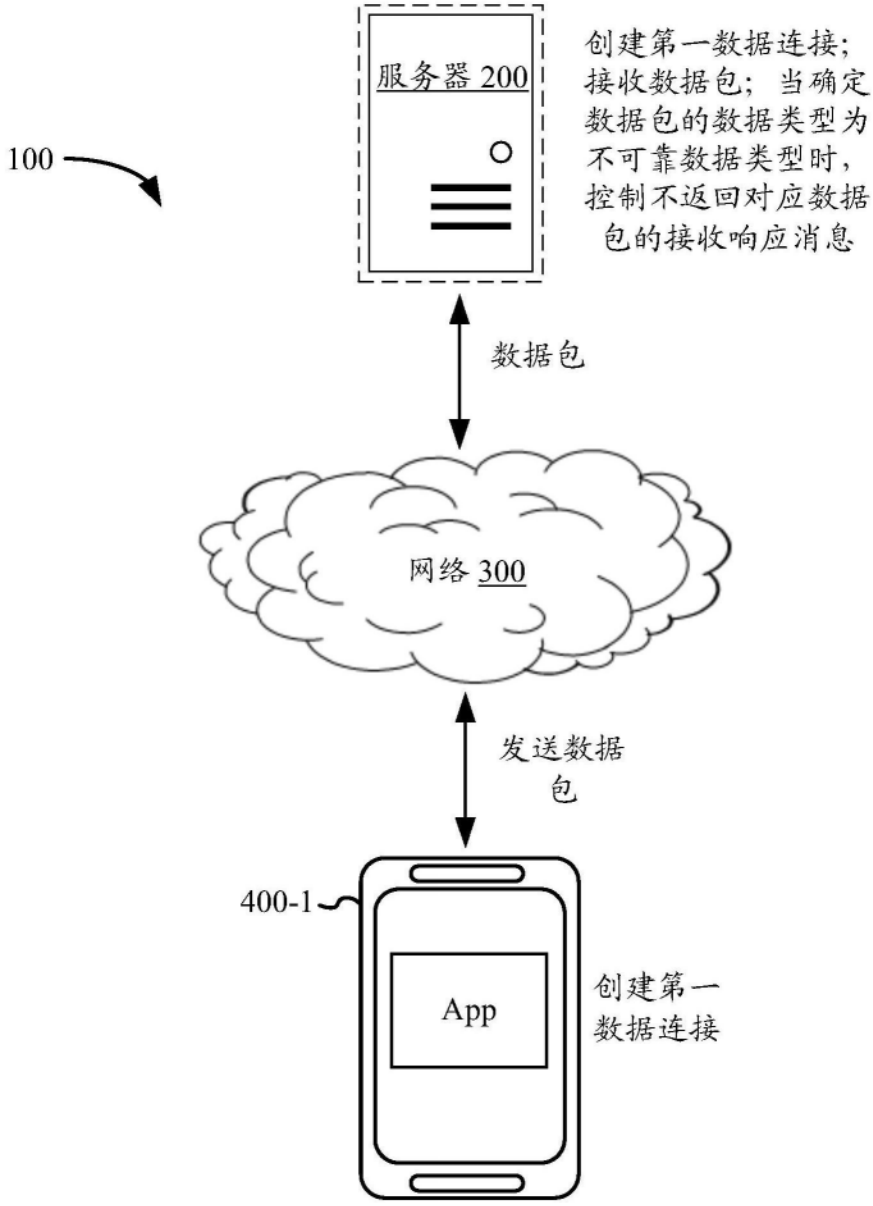

100
服务器 200
创建第一数据连接；
接收数据包；当确定
数据包的数据类型为
不可靠数据类型时，
控制不返回对应数据
包的接收响应消息
数据包
网络 300
发送数据
包
400-1
App
创建第一
数据连接

图1

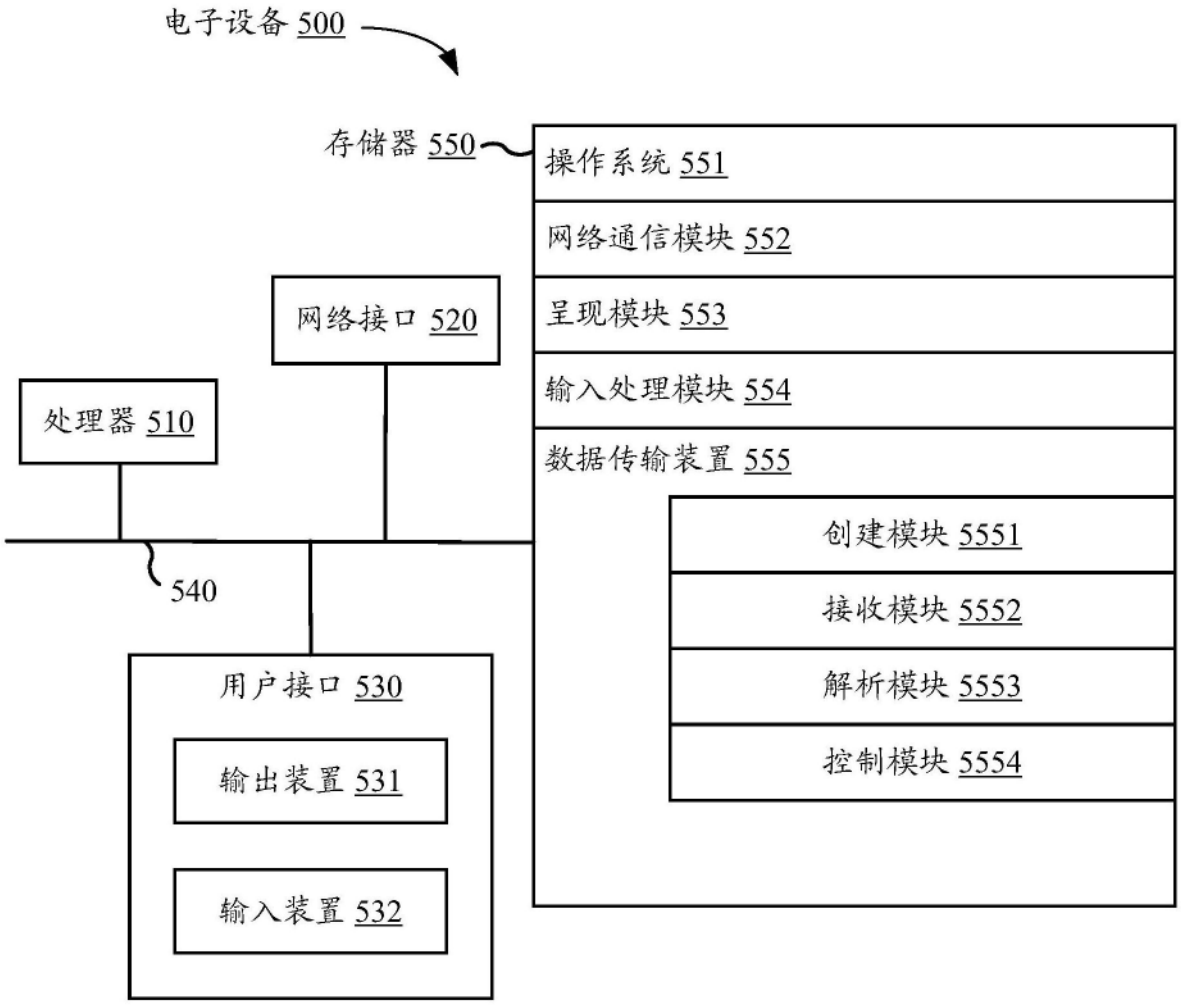

电子设备 500
存储器 550
操作系统 551
网络通信模块 552
呈现模块 553
输入处理模块 554
数据传输装置 555
创建模块 5551
接收模块 5552
解析模块 5553
控制模块 5554
网络接口 520
处理器 510
540
用户接口 530
输出装置 531
输入装置 532

图2

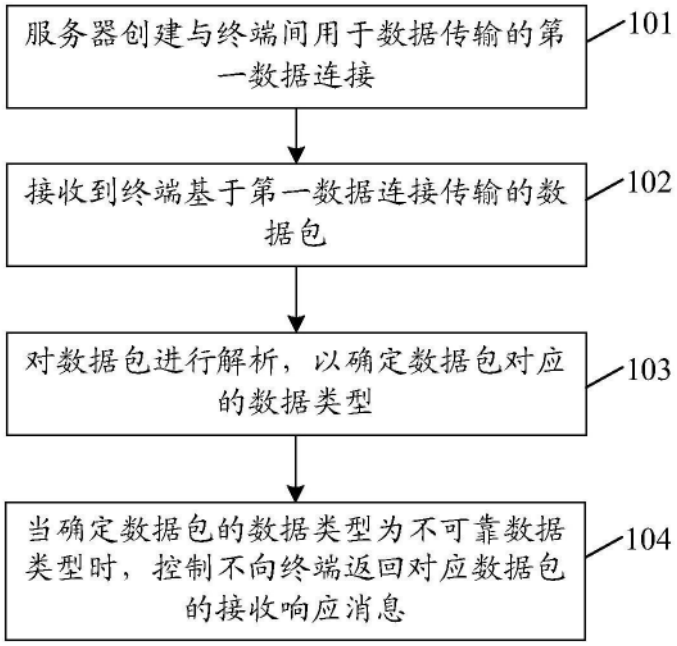

服务器创建与终端间用于数据传输的第一数据连接
101
接收到终端基于第一数据连接传输的数据包
102
对数据包进行解析，以确定数据包对应的数据类型
103
当确定数据包的数据类型为不可靠数据类型时，控制不向终端返回对应数据包的接收响应消息
104

图3A

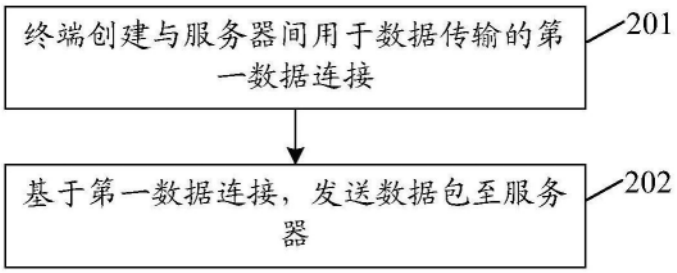

终端创建与服务器间用于数据传输的第一数据连接
201
基于第一数据连接，发送数据包至服务器
202

图3B

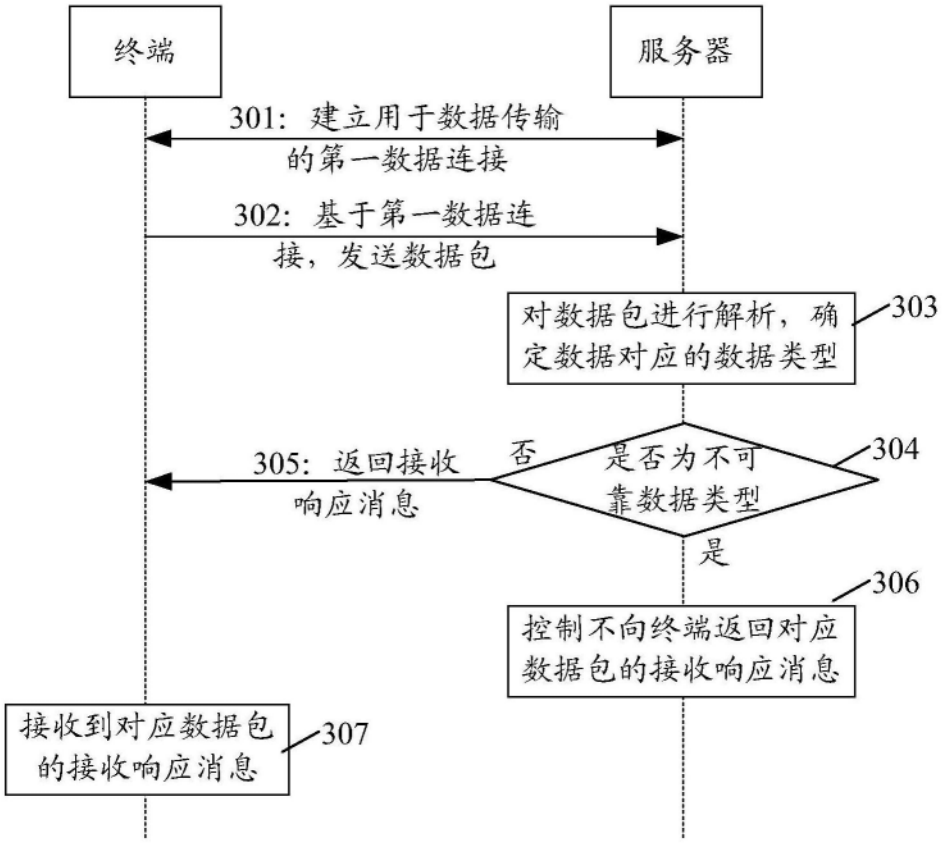

终端
服务器
301：建立用于数据传输的第一数据连接
302：基于第一数据连接，发送数据包
对数据包进行解析，确定数据对应的数据类型
303
是否为不可靠数据类型
304
否
305：返回接收响应消息
是
306
控制不向终端返回对应数据包的接收响应消息
接收到对应数据包的接收响应消息
307

图4


客户端
服务器
401: SYN握手信号
402: SYN+ACK应答信号
403: ACK响应信号
TCP握手
404: XMPP 问询数据包
405: 应答数据包
406: 特征流字段 (TLS)
407: TLS 握手请求
408: 响应信息
409: 客户端问询数据包
410: 服务器问询数据包
411: 客户端问询结束数据包
412: 服务器问询结束数据包
TLS握手
413: 问询数据包
414: 响应数据包
415: 流特征字段 (SASL)
416: SASL验证开始数据包
417: SASL验证结束数据包
418: 初始流请求
419: ACK响应数据包
420: 流特征字段（绑定资源）
421: 绑定资源请求
422: 绑定资源结果
XMPP握手
423: 应用数据
424: 应用数据

图5


无SIM卡
17:33
100%
XX
加入会议
快速会议
预定会议
28 01月今天
历史会议
17: 30-18: 00 744 615 265待开始
XX预定的会议
17: 30-18: 00 726 430 223待开始
XX预定的会议
16:33
离开
正在进入会议...
16:33
YY会议
00:01
结束
点击可快速查看会议信息
XX
邀请您的联系人参加会议
静音
停止视频
共享屏幕
管理成员(1)
更多

图6

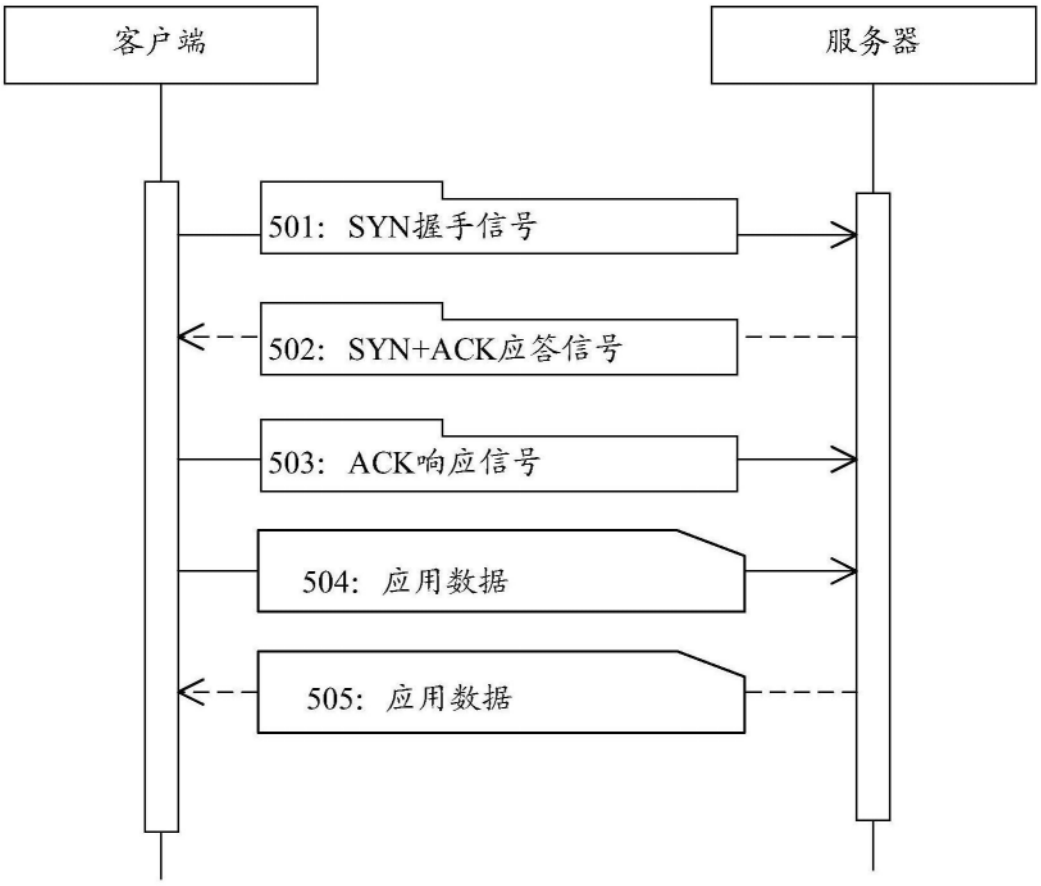

客户端
服务器
501: SYN握手信号
502: SYN+ACK应答信号
503: ACK响应信号
504: 应用数据
505: 应用数据

图7

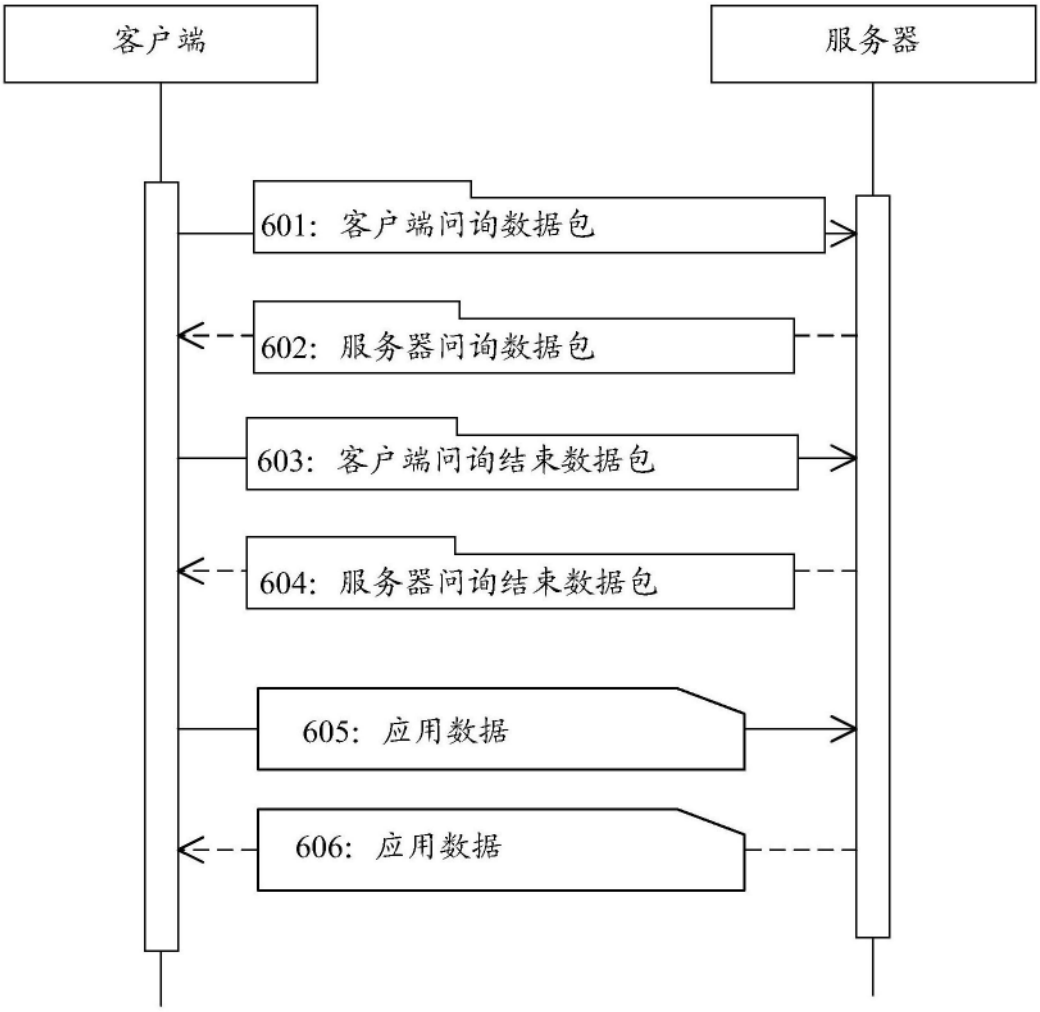

客户端
服务器
601：客户端问询数据包
602：服务器问询数据包
603：客户端问询结束数据包
604：服务器问询结束数据包
605：应用数据
606：应用数据

图8

图9

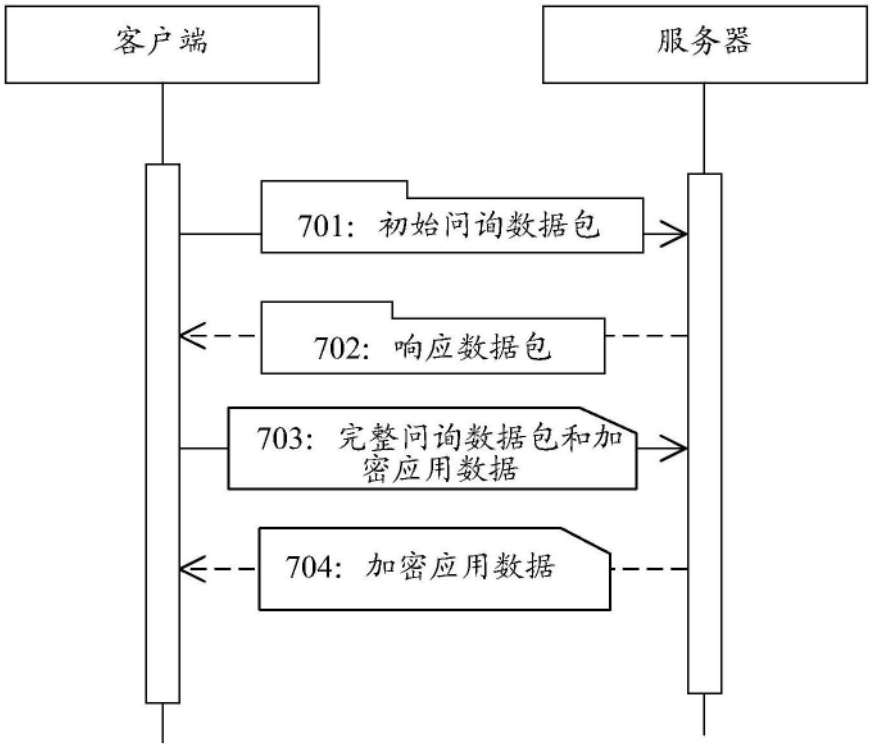

客户端
服务器
701：初始问询数据包
702：响应数据包
703：完整问询数据包和加密应用数据
704：加密应用数据

图10

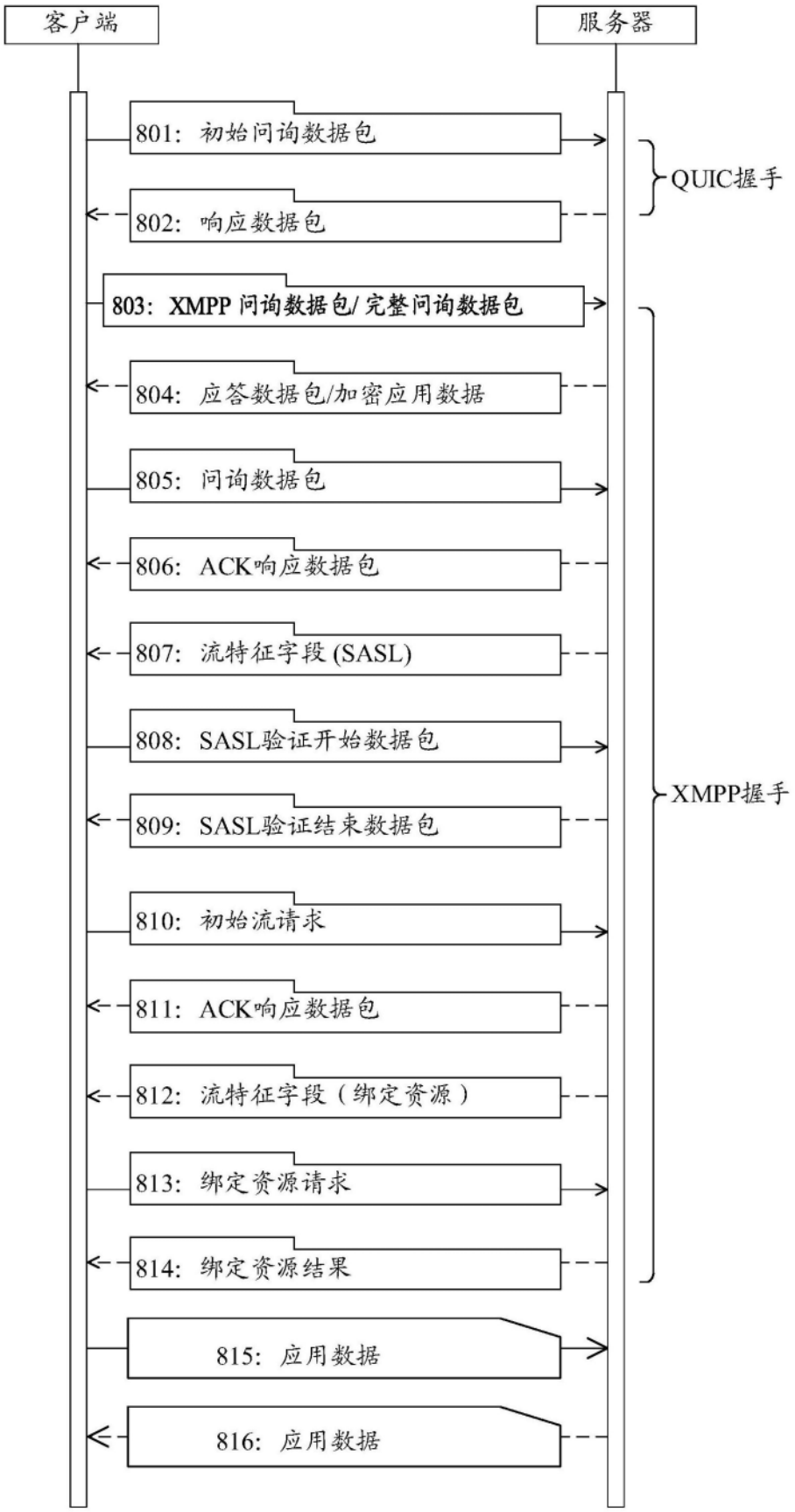

客户端
服务器
801: 初始问询数据包
802: 响应数据包
QUIC握手
803: XMPP 问询数据包/ 完整问询数据包
804: 应答数据包/加密应用数据
805: 问询数据包
806: ACK响应数据包
807: 流特征字段 (SASL)
808: SASL验证开始数据包
809: SASL验证结束数据包
XMPP握手
810: 初始流请求
811: ACK响应数据包
812: 流特征字段（绑定资源）
813: 绑定资源请求
814: 绑定资源结果
815: 应用数据
816: 应用数据

图11

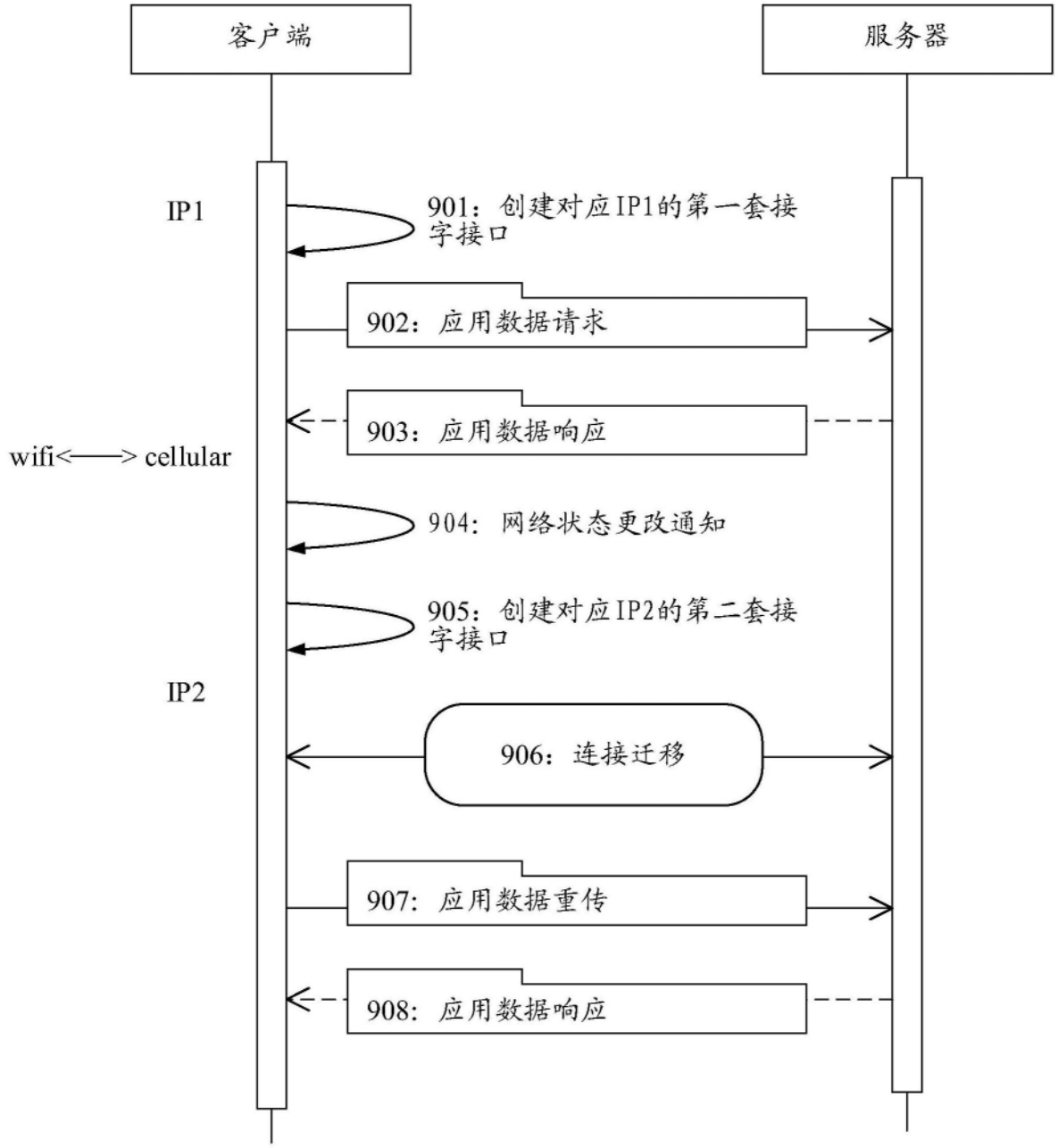

客户端
服务器
IP1
901：创建对应IP1的第一套接字接口
902：应用数据请求
903：应用数据响应
wifi<——> cellular
904：网络状态更改通知
905：创建对应IP2的第二套接字接口
IP2
906：连接迁移
907：应用数据重传
908：应用数据响应

图12

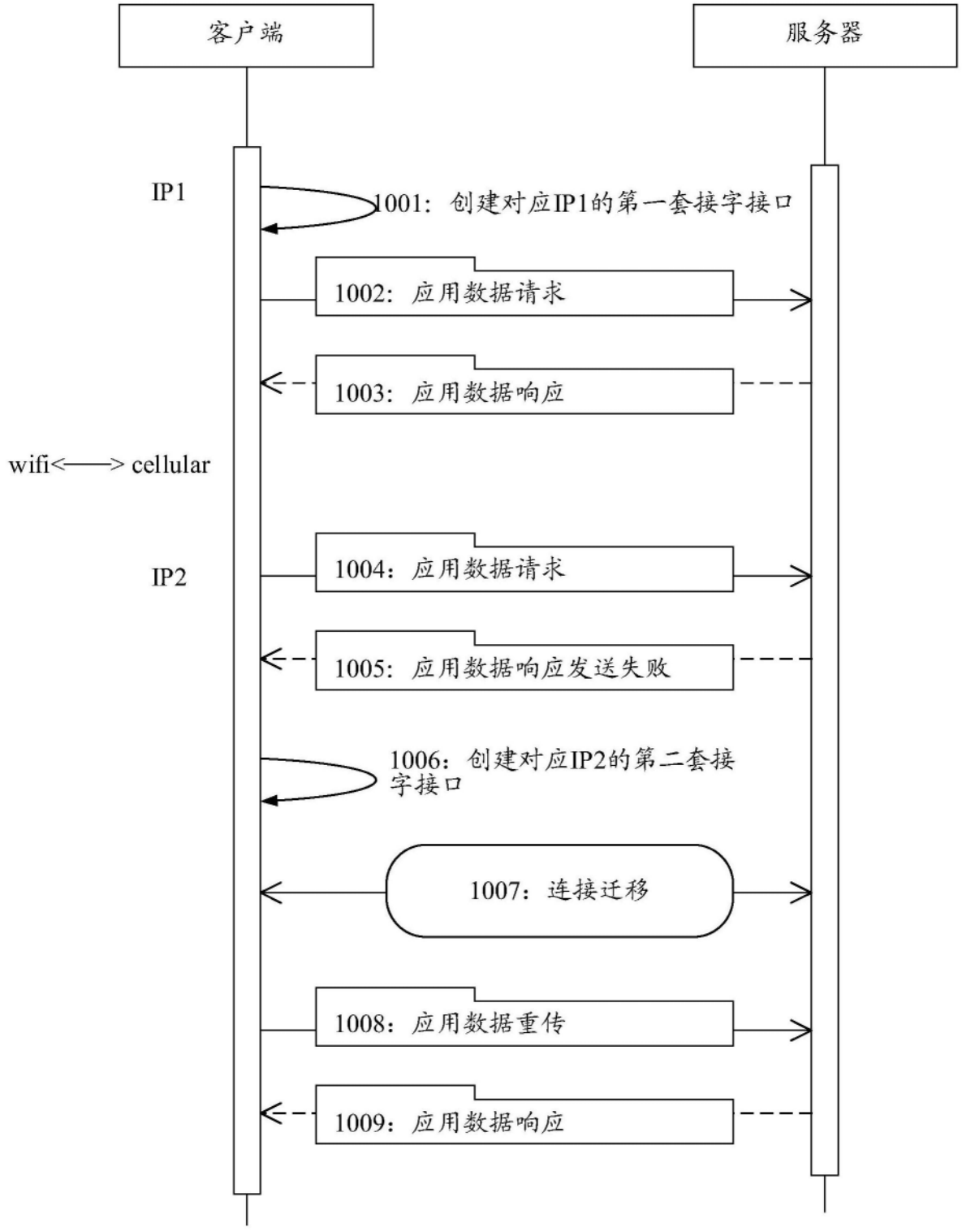

客户端
服务器
IP1
1001：创建对应IP1的第一套接字接口
1002：应用数据请求
1003：应用数据响应
wifi<——> cellular
IP2
1004：应用数据请求
1005：应用数据响应发送失败
1006：创建对应IP2的第二套接字接口
1007：连接迁移
1008：应用数据重传
1009：应用数据响应

图13

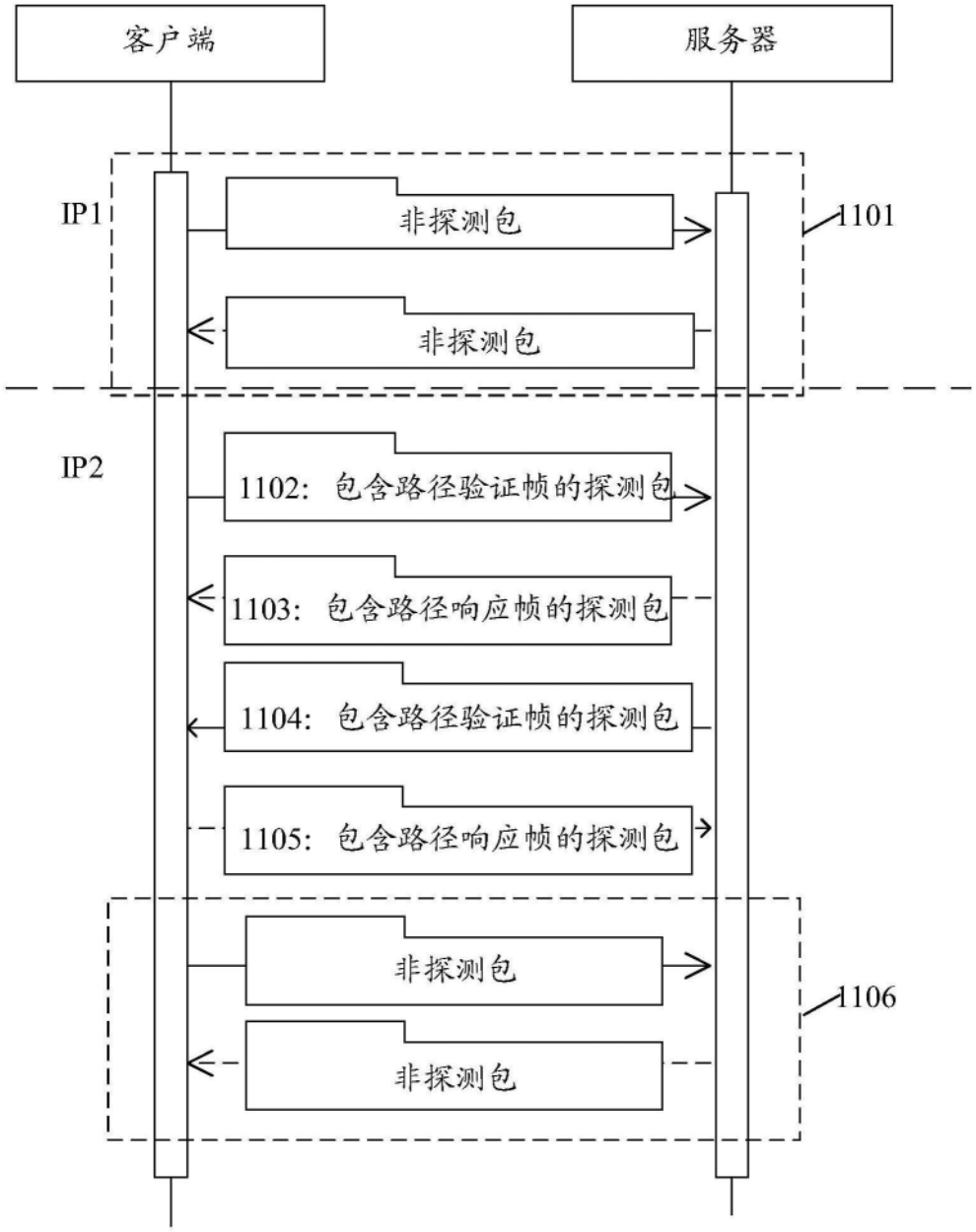
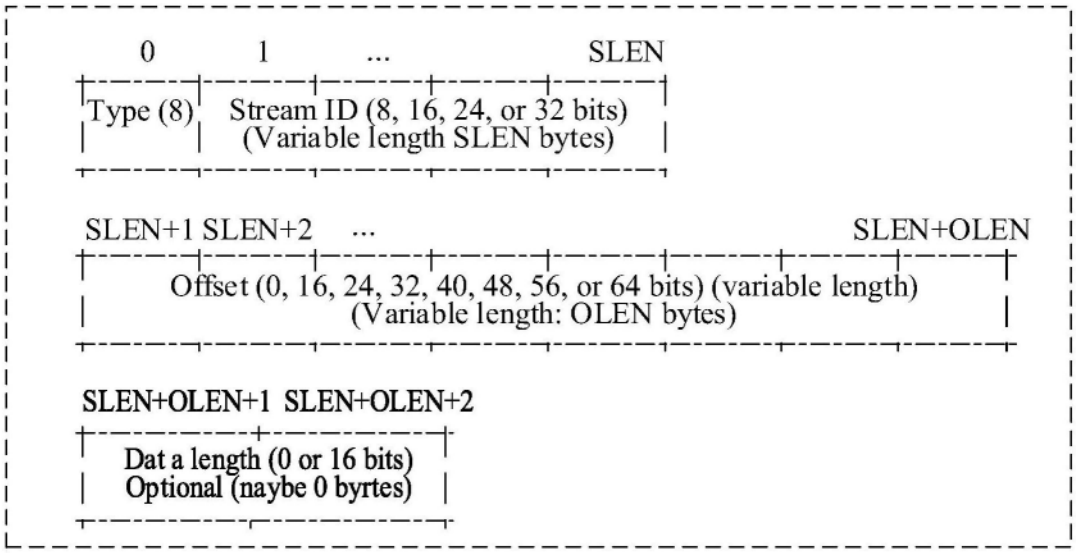

客户端
服务器
IP1
非探测包
非探测包
1101
IP2
1102: 包含路径验证帧的探测包
1103: 包含路径响应帧的探测包
1104: 包含路径验证帧的探测包
1105: 包含路径响应帧的探测包
非探测包
非探测包
1106

图14

A: 可靠数据帧



B: 不可靠数据帧

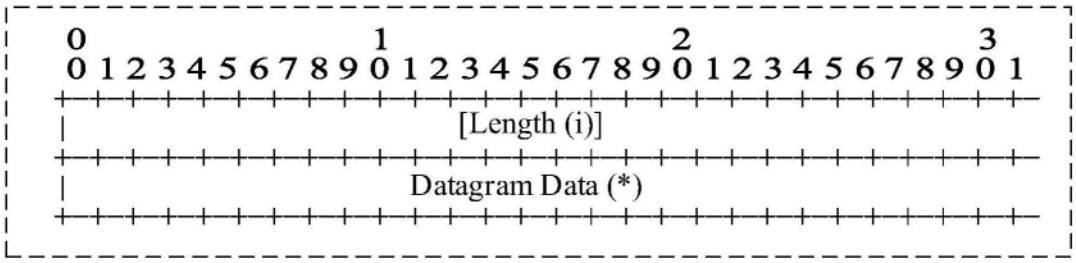


图15

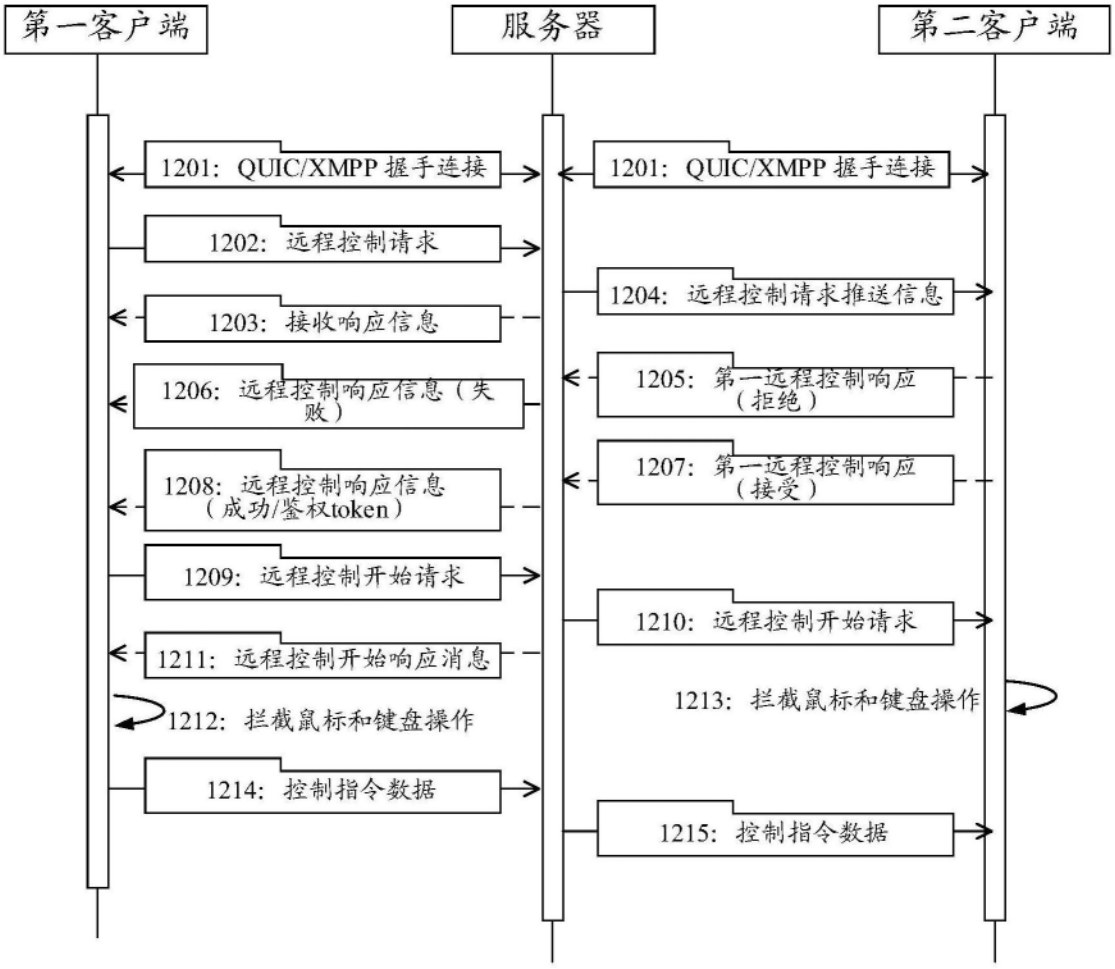

第一客户端
服务器
第二客户端
1201：QUIC/XMPP 握手连接
1201：QUIC/XMPP 握手连接
1202：远程控制请求
1204：远程控制请求推送信息
1203：接收响应信息
1205：第一远程控制响应（拒绝）
1206：远程控制响应信息（失败）
1207：第一远程控制响应（接受）
1208：远程控制响应信息（成功/鉴权token）
1209：远程控制开始请求
1210：远程控制开始请求
1211：远程控制开始响应消息
1213：拦截鼠标和键盘操作
1212：拦截鼠标和键盘操作
1214：控制指令数据
1215：控制指令数据

图16

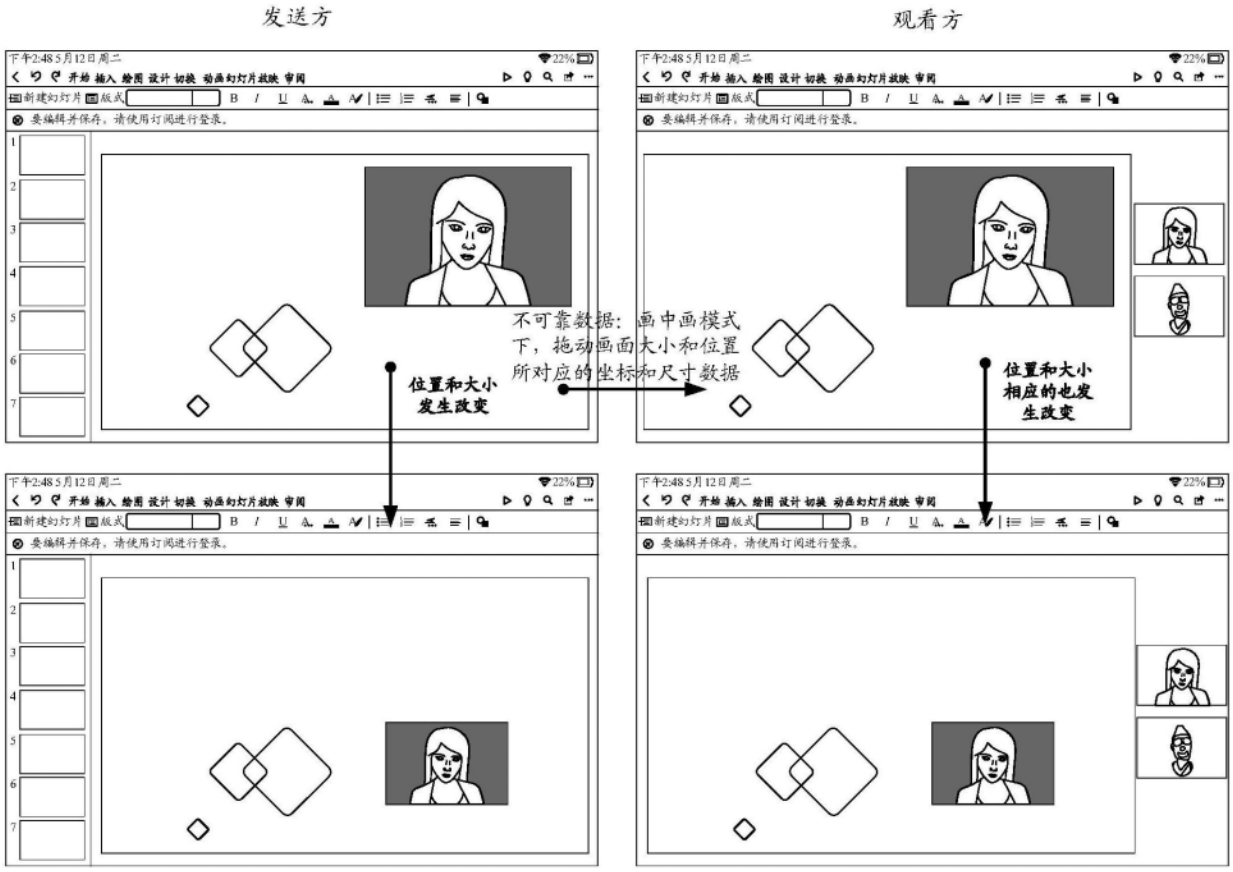

发送方
观看方
不可靠数据：画中画模式下，拖动画面大小和位置所对应的坐标和尺寸数据
位置和大小发生改变
位置和大小相应的也发生改变

图17A

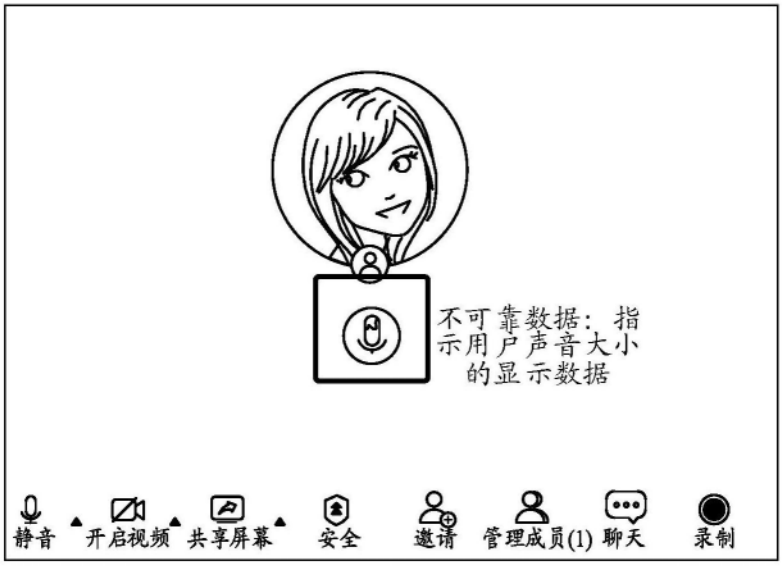

不可靠数据：指示用户声音大小的显示数据
静音
开启视频
共享屏幕
安全
邀请
管理成员(1)
聊天
录制

图17B

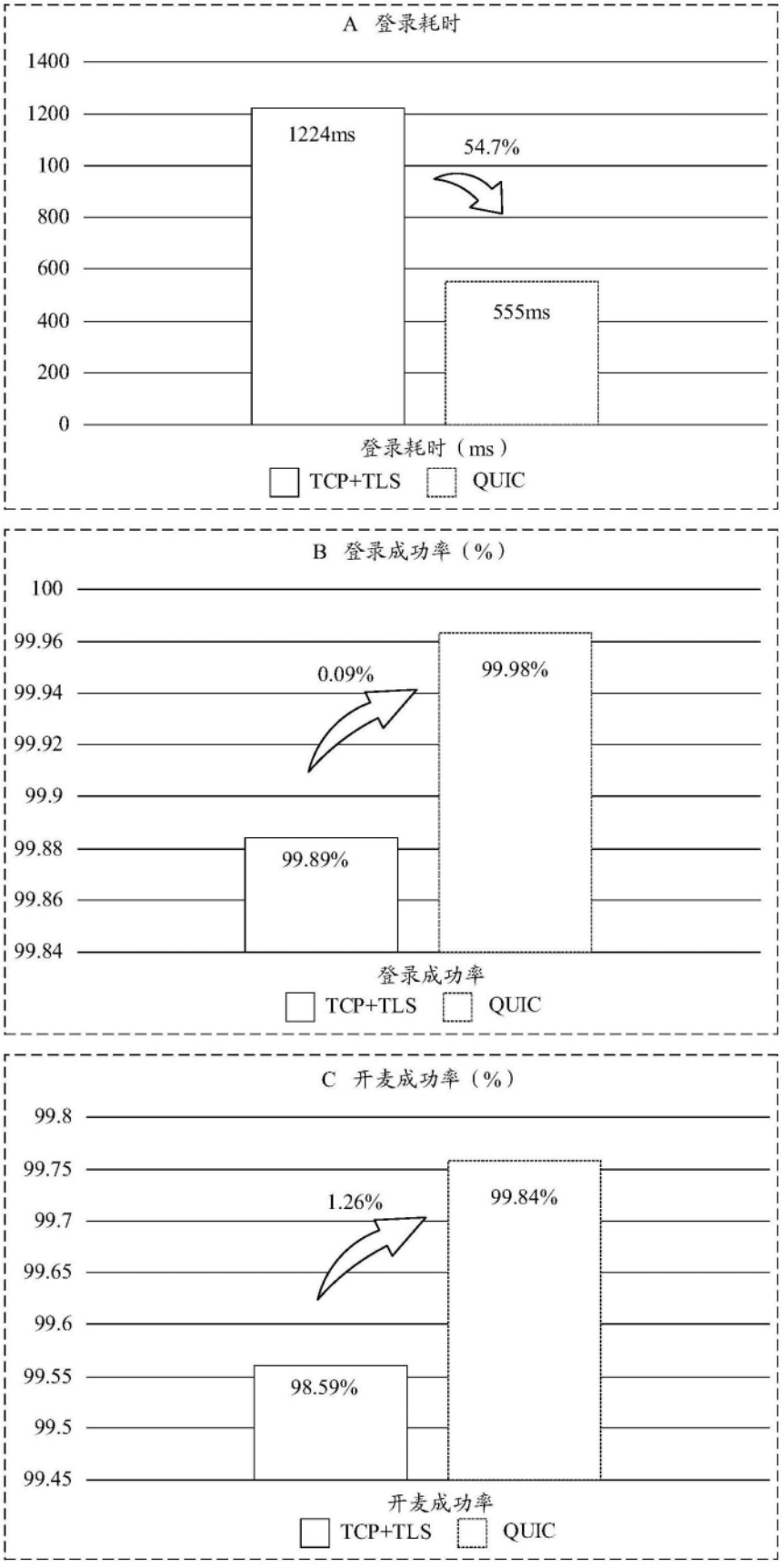

A 登录耗时
1400
1200
100
800
600
400
200
0
1224ms
54.7%
555ms
登录耗时（ms）
TCP+TLS
QUIC
B 登录成功率（%）
100
99.96
99.94
99.92
99.9
99.88
99.86
99.84
0.09%
99.98%
99.89%
登录成功率
TCP+TLS
QUIC
C 开麦成功率（%）
99.8
99.75
99.7
99.65
99.6
99.55
99.5
99.45
1.26%
99.84%
98.59%
开麦成功率
TCP+TLS
QUIC

图18

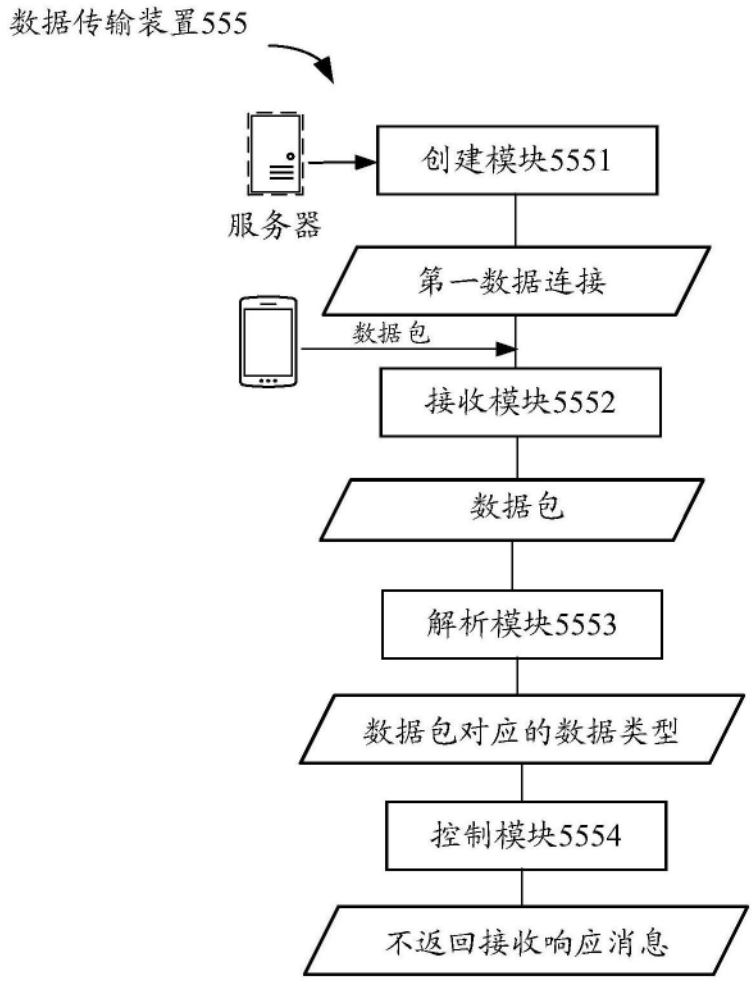

数据传输装置555
创建模块5551
服务器
第一数据连接
数据包
接收模块5552
数据包
解析模块5553
数据包对应的数据类型
控制模块5554
不返回接收响应消息

图19

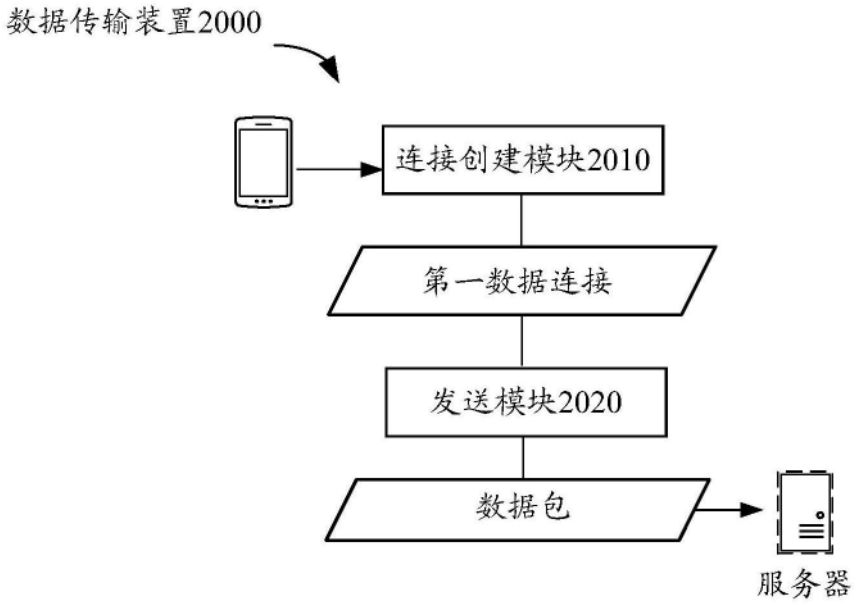

数据传输装置2000
连接创建模块2010
第一数据连接
发送模块2020
数据包
服务器

图20