

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.⁷
H04L 12/28
H04L 29/02



[12] 发明专利申请公开说明书

[21] 申请号 200310118747.9

[43] 公开日 2004年7月7日

[11] 公开号 CN 1510875A

[22] 申请日 2003.12.2

[21] 申请号 200310118747.9

[30] 优先权

[32] 2002.12.2 [33] JP [31] 349994/2002

[71] 申请人 索尼株式会社

地址 日本东京

[72] 发明人 赤鹿秀树 大岛拓哉 铃木唯史

三浦敦史 太田丰一 平野义昭

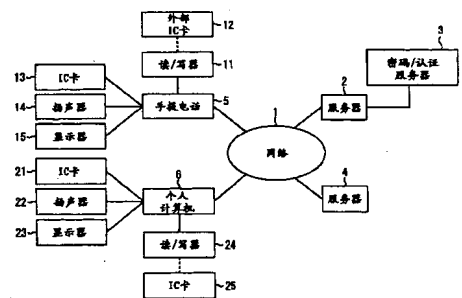
[74] 专利代理机构 中国国际贸易促进委员会专利
商标事务所
代理人 李 勇

权利要求书 5 页 说明书 38 页 附图 24 页

[54] 发明名称 用于控制信息处理设备的系统和方
法

[57] 摘要

设备(例如 IC 卡)连接于其上的客户端(例如 PC、手提电话、PDA、家用电器),开始握手协议,以请求服务器开始通信连接。当通过握手协议建立起通信连接之后,通信连接的主动权传递到服务器,并且状态改变到中间状态。在此中间状态中,包含特定数目的消息和一个已完成消息的控制数据包从服务器被发送到客户端。如果客户端接收到该控制数据包,客户端根据包含在该控制数据包中的消息执行处理。这使得可以通过网络以一种高可靠性和有效性的方式来远程控制设备。



ISSN 1008-4274

1. 包括通过网络互相连接的信息处理设备和信息终端的控制系统,

信息终端包括:

请求装置, 用来在设备被连接到信息终端的状态下, 请求信息处理设备通过网络建立通信连接;

接收装置, 用来通过响应请求装置发出的请求而建立起来的通信连接, 从信息处理设备接收包括用于控制设备的命令的数据包; 以及,

控制装置, 用来根据包含在接收装置接收的数据包中的命令来控制该设备,

信息处理设备包括:

建立装置, 用来在信息处理设备和信息终端之间, 建立通过网络进行的通信连接, 以响应信息终端发出的请求;

发送装置, 在建立装置建立起与信息终端的通信连接之后, 用来从信息处理设备向信息终端传送数据包。

2. 用于控制包括通过网络互相连接的信息处理设备和信息终端的控制系统的的方法, 本方法包括与信息终端相关的信息处理方法, 以及与信息处理设备相关的信息处理方法,

与信息终端相关的信息处理方法包括以下步骤:

在设备与信息终端相连接的状态下, 请求信息处理设备通过网络建立通信连接;

通过响应在请求步骤中发出的请求而建立起来的通信连接, 从信息处理设备接收包含用于控制该设备的命令的数据包;

根据在接收步骤中接收到的数据包中包含的命令, 来控制该设备,

与信息处理设备相关的信息处理方法包括以下步骤:

响应信息终端发出的请求, 在信息处理设备和信息终端之间建

立通过网络执行的通信连接;

在建立步骤中建立起与信息终端的通信连接之后,从信息处理设备向信息终端传送数据包。

3. 信息处理设备,用来通过经由网络与信息处理设备相连的信息终端,来控制与带有特定定时的信息终端相连接的设备,该信息处理设备包括:

建立装置,用来在信息处理设备和信息终端之间建立通过网络执行的通信连接,以响应信息终端在设备连接在其上的状态下发出的请求;

发送装置,在建立装置建立起与信息终端的通信连接之后,用来从信息处理设备向信息终端发送包含用于控制设备的命令的第一数据包。

4. 如权利要求3所述的信息处理设备,还包括接收装置,用来从信息终端接收包含对发送装置传送的命令做出的响应的第二数据包。

5. 如权利要求3所述的信息处理设备,其中命令包含指示当信息终端未能根据命令执行一项处理的时候,该项处理是否应当按照后面的命令继续执行的信息。

6. 如权利要求3所述的信息处理设备,其中发送装置发送的第一数据包包含属于同一协议的多个命令。

7. 如权利要求3所述的信息处理设备,其中发送装置与第一数据包一起发送一个简单的程序,用来引起信息终端判断哪项处理应当被设备执行。

8. 如权利要求3所述的信息处理设备,其中被发送装置发送的第一数据包包含用来根据命令识别被控制设备的识别信息。

9. 如权利要求3所述的信息处理设备,其中当通信装置建立的通信连接包括信息终端的防火墙的时候,发送装置可以使用 HTTP 作为通信协议来发送第一数据包。

10. 如权利要求9所述的信息处理设备,其中在传送多个第一

数据包的期间，发送装置保持建立装置建立起来的、使用 HTTP 的通信连接。

11. 如权利要求 3 所述的信息处理设备，还包括命令装置，用来命令信息终端开始在建立装置建立起来的通信连接中传送预定信息。

12. 信息处理方法，用于信息处理设备通过经由网络与信息处理设备相连接的信息终端，控制与带有特定定时的信息终端相连接的设备，该方法包括以下步骤：

响应信息终端在设备与信息终端相连接的状态下发出的请求，在信息处理设备和信息终端之间建立通过网络执行的通信连接；

在建立步骤中建立起与信息终端的通信连接之后，从信息处理设备向信息终端传送包含用于控制设备的命令的数据包。

13. 包含存储于其上的计算机可读程序的存储介质，用来使计算机通过经由网络与计算机相连接的信息终端执行对与带有特定定时的信息终端相连接的设备进行控制的处理，该程序包括以下步骤，

响应信息终端在设备与信息终端相连接的状态下发出的请求，在信息处理设备和信息终端之间建立通过网络执行的通信连接；

在建立步骤中建立起与信息终端的通信连接之后，从信息处理设备向信息终端传送包含用于控制设备的命令的数据包。

14. 程序，用来使计算机通过经由网络与计算机相连接的信息终端执行对与带有特定定时的信息终端相连接的设备进行控制的处理，该程序包括以下步骤：

响应信息终端在设备与信息终端相连接的状态下发出的请求，在信息处理设备和信息终端之间建立通过网络执行的通信连接；

在建立步骤中建立起与信息终端的通信连接之后，从信息处理设备向信息终端传送包含用于控制设备的命令的数据包。

15. 经由网络与用于远程控制设备的信息处理设备相连接的信息终端，包括：

请求装置，用来在设备被连接到该信息终端的状态下，请求信

息处理设备通过网络建立通信连接;

接收装置, 用来通过响应由请求装置发出的请求而建立的通信连接, 从信息处理设备接收包括用于控制设备的命令的第一数据包;
以及

控制装置, 用来根据包含在接收装置接收的第一数据包中的命令来控制该设备。

16. 如权利要求 15 所述的信息终端, 还包括发送装置, 用于向信息处理设备传送第二数据包, 包含指出控制装置对该设备执行控制的结果的应答。

17. 信息处理方法, 用于通过经由网络与用于远程控制设备的信息处理设备相连接的信息终端, 该方法包括以下步骤:

在设备与信息终端相连接的状态下, 请求信息处理设备通过网络建立通信连接;

通过响应在请求步骤中发出的请求而建立起来的通信连接, 从信息处理设备接收包含用于控制设备的命令的数据包; 以及

根据在接收步骤中接收到的数据包中包含的命令来控制设备。

18. 包含存储于其上的计算机可读程序的存储介质, 用来使计算机执行在计算机和用于远程控制设备的信息处理设备之间进行的信息处理, 该程序包括以下步骤:

在设备与信息终端相连接的状态下, 请求信息处理设备通过网络建立通信连接;

通过响应在请求步骤中发出的请求而建立起来的通信连接, 从信息处理设备接收包含用于控制该设备的命令的数据包;

根据在接收步骤中接收到的数据包中包含的命令来控制该设备。

19. 程序, 用来使计算机执行在计算机和用于远程控制设备的信息处理设备之间进行的信息处理, 该程序包括以下步骤:

在设备与信息终端相连接的状态下, 请求信息处理设备通过网络建立通信连接;

通过响应在请求步骤中发出的请求而建立起来的通信连接，从信息处理设备接收包含用于控制该设备的命令的数据包；

根据在接收步骤中接收到的数据包中包含的命令来控制该设备。

用于控制信息处理设备的系统和方法

技术领域

本发明涉及控制系统和控制方法、信息处理方法和信息处理设备、信息处理终端和用于信息处理终端的方法、存储终端以及程序。更具体来说，本发明涉及控制系统和控制方法、信息处理方法和信息处理设备、信息处理终端和用于信息处理终端的方法、存储终端以及程序，用来控制以更为可靠和有效的方式通过网络连接的设备。

背景技术

近年来，提出了各种系统，用于控制安装在远端位置的设备。

日本未决专利申请 No. 2000-184081 中公开了一种这类系统。在该系统中，安装在家里的家用电器，例如空调或者录像机，根据从家外通过电话线传送的密码而进行控制。

日本未决专利申请 No. 2002-41378 中公开了一种系统，其中一台服务器被设置在控制终端和被控制终端之间，并且控制终端通过该服务器控制被控制终端。

在这些控制系统中，被控制设备需要通过通信装置始终保持与控制设备的连接，这样，安放在远端的控制设备可以访问被控制设备（例如，在日本未决专利申请 No. 2000-184081 中公开的系统，家用电器始终通过电话线连接到远端控制器和适配器）。

也就是说，当做为控制设备的第一个设备和做为被控制设备的第二个设备能够在任何需要通信的时候互相通信时，每个设备都可以主动发起通信。例如，在任何需要发送的时候，第一个设备可以向第二个设备发送各种数据，包括请求，反过来，第二个设备也可以向第一个设备发送各种数据。

这使得，在任何需要控制的时候，都可以通过经由网络与该设备进行通信，来控制该设备。

然而，当例如 IC 卡、显示器或者存储器这样的设备，在不是一直连接到网络的环境中而被远程控制的情况下，该设备通常以这种方式被控制：管理该设备的客户端，例如个人计算机，向服务器发送请求，并且客户端根据服务器返回的应答来控制该设备的操作。然而，在这项技术中，服务器很难直接地主动控制该设备。

也就是说，由于设备不是一直连接到网络，服务器不能在任意时间访问客户端以控制被客户端管理的设备。

更具体来说，当设备被安放在靠近、或者直接安装在，一台连接到网络的设备（例如 IC 卡读/写卡器或者 RF ID 接收器）上时，服务器很难按照特定的时间安排，主动改变设备（例如 IC 卡或者非接触 RF ID 接收）的属性，或者控制非直接连接到服务器上的设备的操作。

此外，在传统的控制系统中，当命令或者应答在服务器和客户端之间传送的时候，一次传送一个命令或者一个应答。这导致在服务器和客户端之间执行传送的次数增加，并由此导致通信成本的增加。

由于，每次传送命令或者应答时，都会附加例如报头和报尾的信息，因而一次传送的数据长度增加，并且每次传送所需的通信时间也增加。特别是，当连接终端，例如手提电话，被远程控制的时候，与其通信的通信速度低并且成本高，因而上述问题就很严重。

此外，在传统的控制系统中，当客户端接收到服务器传送的命令时，如果在客户端出现错误，处理过程就被终止，而不论错误的类型。为了使客户端执行下面的处理，需要服务器和客户端互相连接到一起并传送命令。

发明内容

如上所述，本发明的目标是提供一种通过网络，以一种高度可靠和有效的方式，来对一个设备进行远程控制的技术。

本发明提供了一种包括通过网络互相连接在一起的信息处理设备和信息终端的控制系统。其中信息终端包括：请求装置，用于在设备被连接到信息终端的状态下，请求信息处理设备通过网络建立通信连接；接收装置，用于通过响应请求装置发出的请求而建立起来的通信连接，从信息处理设备接收包括用于控制该设备的命令的数据包；以及控制装置，用于根据包含在接收装置接收的数据包中的命令来控制该设备。而信息处理设备包括：建立装置，用于在信息处理设备和信息终端之间，建立通过网络进行的通信连接，以响应信息终端发出的请求；发送装置，在建立装置建立起与信息终端的通信连接之后，用于从信息处理设备向信息终端传送数据包。

本发明提供了用于控制系统的方法，包括以下步骤：在设备与信息终端相连接的状态下，请求信息处理设备通过网络建立通信连接；通过响应在请求步骤中发出的请求而建立起来的通信连接，从信息处理设备接收包含用于控制该设备的命令的数据包；根据在接收步骤中接收到的数据包中包含的命令，来控制该设备；响应信息终端发出的请求，在信息处理设备和信息终端之间建立通过网络执行的通信连接；在建立步骤中建立起与信息终端的通信连接之后，从信息处理设备向信息终端传送数据包。

本发明还提供了一种信息处理设备，包括：建立装置，用来在信息处理设备和信息终端之间，建立通过网络进行的通信连接，以响应信息终端在设备连接在其上的状态下发出的请求；发送装置，在建立装置建立起与信息终端的通信连接之后，用来从信息处理设备向信息终端发送包含用于控制设备的命令的第一数据包。

信息处理设备此外可以包括接收装置，用来从信息终端接收包含对发送装置传送的命令的响应的第二数据包。

命令可能包含指示当信息终端未能根据命令执行一项处理的时候，该项处理是否应当按照后面的命令继续执行的信息。

发送装置发送的第一数据包可能包含属于同一协议的多个命令。

发送装置可以与第一数据包一起发送一个简单的程序，用来引起信息终端判断哪项处理应当被设备执行。

发送装置发送的第一数据包可能包含用来根据该命令识别被控制设备的识别信息。

当通信装置建立的通信连接包括信息终端的防火墙的时候，发送装置可以使用 HTTP 作为通信协议来发送第一数据包。

在传送多个第一数据包的期间，发送装置可以保持由建立装置建立起来的、通过 HTTP 的通信连接。

信息处理设备此外可以包括命令装置，用来命令信息终端开始在由建立装置建立起来的通信连接中传送预定信息。

本发明还提供了与信息处理设备相关的信息处理方法，包括以下步骤：响应信息终端在设备与信息终端相连接的状态下发出的请求，在信息处理设备和信息终端之间建立通过网络执行的通信连接；在建立步骤中建立起与信息终端的通信连接之后，从信息处理设备向信息终端传送包含用于控制设备的命令的数据包。

本发明还提供了第一个程序，以及包含存储于其上的第一个程序的存储介质，其中程序包含以下步骤：响应信息终端在设备与信息终端相连接的状态下发出的请求，在信息处理设备和信息终端之间建立通过网络执行的通信连接；在建立步骤中建立起与信息终端的通信连接之后，从信息处理设备向信息终端传送包含用于控制设备的命令的数据包。

本发明还提供了信息终端，它包括：请求装置，用来在一个设备被连接到该信息终端的状态下，请求信息处理设备通过网络建立通信连接；接收装置，用来通过响应由请求装置发出的请求而建立的通信连接，从信息处理设备接收包括用于控制该设备的命令的数据包；以及控制装置，用来根据包含在接收装置接收的数据包中的命令来控制该设备。

信息终端此外可以包括发送装置，用于向信息处理设备传送第二数据包，包含指出控制装置对该设备执行的控制的结果的应答。

本发明还提供了信息处理方法，包括以下步骤：在设备与信息终端相连接的状态下，请求信息处理设备通过网络建立通信连接；通过响应在请求步骤中发出的请求而建立起来的通信连接，从信息处理设备接收包含用于控制该设备的命令的数据包；根据在接收步骤中接收到的数据包中包含的命令，来控制该设备。

本发明还提供了第二个程序和包含存储于其上的第二个程序的存储介质，其中第二个程序包括以下步骤：在设备与信息终端相连接的状态下，请求信息处理设备通过网络建立通信连接；通过响应在请求步骤中发出的请求而建立起来的通信连接，从信息处理设备接收包含用于控制该设备的命令的数据包；根据在接收步骤中接收到的数据包中包含的命令来控制该设备。

在根据本发明的控制系统和方法中，在设备连接到信息终端的状态下，通过网络建立通信连接的请求被发送到信息处理设备，一个包含用于控制该设备的命令的数据包从信息处理设备发出，通过响应请求而建立起来的通信连接被信息终端接收，根据包含在接收到的数据包中的命令来控制该设备。在响应信息终端发出的请求而建立起信息处理设备和信息终端之间的通过网络的通信连接之后，数据包被传送到信息终端。

在根据本发明的信息处理设备、信息处理方法、存储介质以及程序中，在设备与信息终端相连接的状态下，为响应信息终端发出的请求而建立起通过网络在信息处理设备和信息终端之间的通信连接。在建立起信息处理设备和信息终端之间的通信连接之后，包含控制该设备的命令的第一数据包被传送到信息终端。

在根据本发明的信息处理终端、用于信息处理终端的方法、存储介质以及程序中，在设备与信息终端相连接的状态下，通过网络建立通信连接的请求被发送到信息处理设备。在响应该请求而建立起通信连接之后，来自于信息处理设备、包含控制该设备的命令的数据包通过建立起来的通信连接而被接收，并且根据接收到的数据包中包含的命令来控制该设备。

附图说明

- 图 1 是根据本发明的控制系统的举例的图示；
- 图 2 是显示图 1 中所示的服务器的配置举例的框图；
- 图 3 是显示该服务器的功能配置举例的框图；
- 图 4 是显示客户端功能配置举例的框图；
- 图 5 是控制协议的状态转换图示；
- 图 6 是数据包的数据格式的举例的图示；
- 图 7 是用于握手协议中的消息的图示；
- 图 8 是使用图 7 中所示的消息来执行的握手协议的图示；
- 图 9 是用于告别协议 (farewell protocol) 中的消息的图示；
- 图 10 是使用图 9 中所示的消息来执行的告别协议的图示；
- 图 11 是用于差错协议 (error protocol) 中的消息的图示；
- 图 12 是使用图 11 中所示的消息来执行的差错协议的图示；
- 图 13 是使用图 11 中所示的消息来执行的差错协议的另一个例子的图示；
- 图 14 是用于更新实体协议 (update entity protocol) 中的消息的图示；
- 图 15 是使用图 14 中所示的消息来执行的更新实体协议的图示；
- 图 16 是用于应用数据传输协议 (application data transfer protocol) 的消息的图示；
- 图 17 是使用图 16 中所示的消息来执行的应用数据传输协议序列的图示；
- 图 18 是用于操作实体协议 (operate entity protocol) 中的消息的图示；
- 图 19 是使用图 18 中所示的消息来执行的操作实体协议序列的图示；
- 图 20 是在中间状态 (neutral state) 下的通信序列的图示；

图 21 是在中间状态下的通信序列的另一个例子的图示；

图 22 是在中间状态下的通信序列的又一个例子的图示；

图 23 是交换协议序列的图示；

图 24 是交换协议序列的图示；

图 25 是控制协议的状态转换图；

图 26 是握手协议序列的另一个例子的图示；

图 27 是告别协议序列的另一个例子的图示。

具体实施方式

图 1 是根据本发明的控制系统的一个举例。

客户端连接到网络 1（设备通过客户端连接到网络 1），用于远程控制客户端的服务器 2 和 4 也连接到网络 1。服务器 2 还连接到用于进行加密、解密和数据认证以配合服务器 2 的密码/认证服务器（cryptography/authentication server）3。

服务器 2 和服务器 4 中的每个都通过客户端控制设备。

做为客户端的手提电话 5 和个人计算机 6 连接到网络 1。如下面将会详细描述，将会使用一个协议，该协议允许服务器（服务器 2 或者服务器 4）通过在服务器和客户端之间的通信连接，主动控制客户端（手提电话 5 或者个人计算机 6）。

在图 1 所示的例子中，读/写器（reader/writer）11、IC 卡 13、扬声器 14 和显示器 15，它们都是通过客户端由服务器 2 或者服务器 4 控制的设备，被连接到手提电话 5 或者设置在手提电话 5 之内。

外部 IC 卡 12 通过读/写器 11 间接地连接到手提电话 5。外部 IC 卡 12 不是像 IC 卡 13 那样直接连接到手提电话 5 或者设置在手提电话 5 之内，而是当外部 IC 卡 12 被设置得靠近读/写器 11 或者直接位于其上的时候，通过使用电磁感应的通信连接间接连接到手提电话 5。

类似，在图 1 中，IC 卡 21、扬声器 22、显示器 23 和读/写器 24，它们都是通过客户端由服务器 2 或者服务器 4 控制的设备，被连

接到个人计算机 6。外部 IC 卡 25 可以在任意时刻被设置得靠近读/写器 11 或者直接位于其上。

这些设备（读/写器 11、外部 IC 卡 12、IC 卡 13、扬声器 14、显示器 15、IC 卡 21、扬声器 22、显示器 23、读/写器 24 以及外部 IC 卡 25）不是通过客户端始终连接到网络，而是在需要的任意时刻连接到网络。

照例，图 1 中所示的系统配置可以按照需要进行改进。例如，除了手提电话 5 或个人计算机 6 之外，或者作为它们的代替，作为客户端的 PDA（个人数字助理）设备或者做为一个设备的存储器也可以通过网络 1 连接到服务器 2 或者服务器 4。

图 2 是显示了图 1 所示的服务器 2 的配置的一个举例的框图。

CPU（中央处理单元）31 根据存储在 ROM（只读存储器）32 中的程序，或者从存储单元 38 载入 RAM（随机存取存储器）33 中的程序，来执行各种处理。RAM 33 也用来存储 CPU 执行的处理过程中所需要的数据。

例如，在图 1 中所示的通信系统中，用于通过使用协议（下面称为控制协议）的通信连接来控制客户端的服务器，通过在 CPU 31 上执行特定的控制程序来实现。

CPU 31、ROM 32 和 RAM 33 通过总线 34 互相连接在一起。总线 34 还连接到输入/输出接口 35。

输入/输出接口 35 还连接到：包括键盘、鼠标以及此类设备的输入单元 36；包括显示器例如 CRT（阴极射线管）或 LCD（液晶显示器）以及扬声器的输出单元 37；诸如硬盘驱动器的存储单元 38；以及诸如调制解调器或终端适配器的通信单元 39。通信单元 39 负责通过网络 1 进行通信。

根据需要，驱动器 40 也连接到输入/输出接口 35，并且，根据需要，磁盘 41、光盘 42、磁光盘 43 或者半导体存储器 44 被安装在驱动器 40 上面，以从此处将计算机程序安装到存储单元 38 中。

下面，服务器 4 以及作为客户端的手提电话 5 和个人计算机 6

与上文参考图 2 描述的服务器 2 具有类似的配置，因此，此处就不对这些服务器和客户端的配置进行重复描述了。在下面的描述中，当讨论手提电话 5 或者个人计算机 6 的时候，也会参考图 2。

下面将会描述用在服务器和客户端之间的通信连接中的协议。下文中，用在服务器和客户端之间的通信连接中的协议被称为控制协议。

在客户端执行的网络应用程序（Web application）的背景下，客户端和服务端之间的通信连接使用控制协议来完成。在用作客户端的手提电话 5 或者类似设备中，网络应用程序不是必需的，只需完成客户端和服务端之间的通信连接。在这种情况下，例如，手提电话 5 的一项应用保留所有待显示的数据，并且数据通过类似于控制设备的方式来控制应用程序和显示器，以进行显示。

可以开发使用控制协议的网络应用程序。在这样的网络应用程序中，例如，一种商业逻辑可以以单一的功能/方法来进行描述。这产生了如下的好处。首先，可以执行该应用程序，而不考虑 HTTP（超文本传输协议）连接的断开。第二，允许服务器向客户端传送请求（也即，HTTP 请求者并不局限于客户端）。第三，允许客户端执行任意的处理过程（没有类似于 Applet 或者 ActiveX 控制之中的限制）。

控制协议包括数据传输协议以及其它子协议。例如，子协议包括握手协议、告别协议、差错协议、更新实体协议、应用数据传输协议以及操作实体协议。

更具体来说，数据传输协议（data transfer protocol）规定了子协议消息的发送/接收步骤，握手协议规定了建立连接的步骤。告别协议规定了释放连接的步骤，差错协议规定了错误通知步骤。

更新实体协议规定了服务器执行的改变客户端状态的步骤。应用数据传输协议规定了一项应用的特定数据的发送/接收步骤。操作实体协议规定了服务器执行的操作客户端的步骤。

包括上面所述这些协议的控制协议仅规定了数据通信的核心

(框架)。为了执行一项应用的特定的处理过程，例如对于特定设备的操作，对该应用的扩展(扩展模块, extension module)是有必要的。就是说，主控制协议和扩展的结合规定了操作步骤。

图 3 是显示了根据控制协议进行通信的服务器(例如, 服务器 2)功能配置的例子框图。图 3 中所示的每个功能框都通过执行服务器 2 的 CPU 31 上面的特定控制程序来实现。

控制器 51 控制整个服务器 2。服务器 2 包括: 包括握手消息管理器 52, 用来管理包含在握手协议中的消息; 告别消息管理器 55, 用来管理包含在告别协议中的消息; 差错消息管理器 58, 用来管理包含在差错协议中的消息; 应用数据传送消息管理器 61, 用来管理包含在应用数据传输协议中的消息; 更新实体消息管理器 64, 用来管理包含在更新实体协议中的消息; 以及操作实体消息管理器 67, 用来管理包含在操作实体协议中的消息。

包含在握手协议中的消息(下面称为握手消息)将参考图 7 在下文中进行详细描述。包含在告别协议中的消息(下面称为告别消息)将参考图 9 在下文中进行详细描述。包含在差错协议中的消息(下面称为差错消息)将参考图 11 在下文中进行详细描述。包含在更新实体协议中的消息(下面称为更新实体消息)将参考图 14 在下文中进行详细描述。包含在应用数据传输协议中的消息(下面称为应用数据传送消息)将参考图 16 在下文中进行详细描述。包含在操作实体协议中的消息(下面称为操作实体消息)将参考图 18 在下文中进行详细描述。

握手消息管理器 52 管理与握手消息相关的处理过程。例如, 如果包含在从客户端发出的数据包中的握手消息被握手消息接收控制器 54 接收, 并传送到握手消息管理器 52, 则握手消息管理器 52 根据接收到的消息而生成握手消息, 并将生成的待发送给客户端的握手消息提供给握手消息发送控制器 53。

握手消息发送控制器 53 控制、并通过如图 2 所示的通信单元 39 来执行握手消息向客户端, 例如手提电话 5 的发送。握手消息接收控

制器 54 控制，并通过如图 2 所示的通信单元 39 来执行，从客户端接收握手消息。

告别消息管理器 55 管理与告别消息相关的处理过程。例如，如果包含在从客户端发出的数据包中的告别消息被告别消息接收控制器 57 接收，并传送到告别消息管理器 55，则告别消息管理器 55 响应接收到的消息而生成告别消息，并将生成的待发送给客户端的告别消息提供给告别消息发送控制器 56。

告别消息发送控制器 56 控制，并通过如图 2 所示的通信单元 39 来执行，告别消息向客户端的发送。告别消息接收控制器 57 控制，并通过如附图 2 所示的通信单元 39 来执行，从客户端接收告别消息。

差错消息管理器 58 管理与差错消息相关的处理过程。例如，如果控制器 51 检测到从客户端发出的数据包中包含了格式错误，差错消息管理器 55 根据检测到的错误生成差错消息，并将生成的待发送给客户端的差错消息提供给差错消息发送控制器 59。

差错消息发送控制器 59 控制，并通过如图 2 所示的通信单元 39 来执行差错消息向客户端的发送。差错消息接收控制器 60 控制，并通过如图 2 所示的通信单元 39 来执行从客户端接收差错消息。

应用数据传送消息管理器 61 管理应用数据传送消息。例如，如果包含在从客户端发出的数据包中的应用数据传送消息被应用数据传送消息接收控制器 63 接收，并传送到应用数据传送消息管理器 61，则应用数据传送消息管理器 61 响应接收到的消息生成应用数据传送消息，并将生成的待发送给客户端的应用数据传送消息提供给应用数据传送消息发送控制器 62。

应用数据传送消息发送控制器 62 控制，并通过如图 2 所示的通信单元 39 来执行应用数据传送消息向客户端的发送。更具体来说，应用数据传送消息发送控制器 62 在应用数据传送消息管理器 61 提供的、一组特定数目的消息上加上一个已结束消息 (finished message)，并将它们以数据包的形式传送到客户端。

应用数据传送消息接收控制器 63 控制、并通过如图 2 所示的通信单元 39 来执行从客户端接收应用数据传送消息。更具体来说，应用数据传送消息接收控制器 63 通过检测包含在数据包中的已结束消息，检测从客户端接收的数据包的末尾，应用数据传送消息接收控制器 63 向应用数据传送消息管理器 61 输出包含在数据包中的应用数据传送消息。

更新实体消息管理器 64 管理与更新实体消息相关的处理过程。例如，如果更新实体消息管理器 64 被控制器 51 命令，向客户端发送特定的更新实体消息，则更新实体消息管理器 64 根据该命令生成更新实体消息，并将生成的待发送给客户端的更新实体消息提供给更新实体消息发送控制器 65。

更新实体消息发送控制器 65 控制、并通过如图 2 所示的通信单元 39 来执行更新实体消息向客户端的发送。更具体来说，更新实体消息发送控制器 65 在更新实体消息管理器 64 提供的、一组特定数目的消息上加上一个已结束消息，并将它们以数据包的形式传送到客户端。

更新实体消息接收控制器 66 控制、并通过如图 2 所示的通信单元 39 来执行从客户端接收更新实体消息。更具体来说，更新实体消息接收控制器 66 通过检测包含在数据包中的已结束消息，检测从客户端接收的数据包的末尾，并且，更新实体消息接收控制器 66 向更新实体消息管理器 64 输出包含在数据包中的更新实体消息。

操作实体消息管理器 67 管理与操作实体消息相关的处理过程。例如，如果控制器 51 命令操作实体消息管理器 67 向客户端发送特定的操作实体消息，则操作实体消息管理器 67 根据该命令生成操作实体消息，并将生成的待发送给客户端的操作实体消息提供给操作实体消息发送控制器 68。

操作实体消息发送控制器 68 控制、并通过如图 2 所示的通信单元 39 来执行操作实体消息向客户端的发送。更具体来说，操作实体消息发送控制器 65 在操作实体消息管理器 67 提供的、一组特定数目

的消息上加上一个已结束消息，并将它们以数据包的形式传送到客户端。

操作实体消息接收控制器 69 控制、并通过如图 2 所示的通信单元 39 来执行从客户端接收操作实体消息。更具体来说，操作实体消息接收控制器 69 通过检测包含在数据包中的已结束消息，检测从客户端接收的数据包的末尾，并且，操作实体消息接收控制器 69 向操作实体消息管理器 67 输出包含在数据包中的操作实体消息。

图 4 是显示根据控制协议进行通信的客户端（例如，手提电话 5）的功能配置的例子框图。图 4 所示的功能配置与上文中参考图 3 描述的服务器功能配置相似，因此，此处不再给出重复的描述。

图 4 中所示的每个功能框都通过执行手提电话 5 的 CPU 31 上的特定控制程序来实现。

控制器 81 控制整个手提电话（客户端），包括握手消息管理器 82，告别消息管理器 85，差错消息管理器 88，应用数据传送消息管理器 91，更新实体消息管理器 94，以及操作实体消息管理器 97。

例如，当从服务器发送的更新实体消息被更新实体消息接收控制器 96 接收，并通过更新实体消息管理器 94 传送到控制器 81 的时候，控制器 81 命令设备控制器 100 来控制设备，以根据接收到的更新实体消息执行一项处理。

在完成了由连接到客户端的设备执行的处理过程之后，控制器 81 从设备控制器 100 要求指示处理结果的信息（例如，指示该项处理是否被成功完成的信息），并且向更新实体消息管理器 94 提供获得的信息。

更新实体消息管理器 94 根据处理的结果生成应答消息（更新实体消息），并将生成的应答消息通过更新实体消息发送控制器 95 发送到服务器。

设备控制器 100 根据可能在下列情况之一之下发出的命令，来控制对于连接到客户端的设备的操作。

首先，当应用数据传送消息接收控制器 93 获得的应用数据传送

消息被传送到控制器 81 的时候，用来引起设备根据应用数据传送消息执行一项处理的命令由控制器 81 发出。其次，当更新实体消息接收控制器 96 获得的更新实体消息被传送到控制器 81 的时候，用来引起设备根据更新实体消息执行一项处理的命令由控制器 81 发出。第三，当操作实体消息接收控制器 99 获得的操作实体消息被传送到控制器 81 的时候，用来引起设备根据操作实体消息执行一项处理的命令由控制器 81 发出。

设备控制器 100 向控制器 81 提供指示控制结果（设备执行的处理的结果）的信息。根据处理的结果，应用数据传送消息管理器 91，更新实体消息管理器 94，或者操作实体消息管理器 97，生成一个发送到服务器的应答。

现在，参考图 5，下面将描述控制协议的状态转换。之后将描述各个状态下进行通信的细节。

在步骤 S1，客户端（例如手提电话 5）的 CPU 31 开始执行客户端程序。结果是，开始状态转换。当客户端程序被启动的时候，需要下列信息：

- * 服务器 URL（统一资源定位器 Uniform Resource Locator）：控制客户端的服务器的 URL（例如，<http://ww.oge.com/webapp/Sever>）。

- * 浏览者网络应用程序 URL：网络应用程序的 URL，为屏幕前的浏览者提供在终端用户处执行的应用程序（例如，<http://ww.oge.com/webapp/ViewDispatcher>）。

当客户端（例如手提电话 5）不需要网络应用程序的时候，不需要该 URL。

- * 点心文件（cookie）（可选）：用来识别会话（session）。

客户端使用特定的会话 ID 来向由服务器 URL 指定的服务器（例如，服务器 2）发送 HTTP 请求。在上述例子中，用于在“/webapp/Sever”发送(posting)消息的 HTTP 请求被发送到“ww.oge.com”的特定端口号。

在客户端程序启动时设置了点心文件信息的情况下，客户端向 HTTP 请求的点心文件报头 (cookie header) 添加特定的点心文件信息。

从服务器返回的应答消息在 HTTP 应答的内容部分描述。

在启动客户端程序之后，在步骤 S2 中，客户端传送一个使用握手子协议的客户端问候消息 (client-hello message)，来请求建立通信连接。

即，在握手状态 (图 5 中的状态 1)，客户端问候消息被发送，并且连接被建立起来。

在完成了握手协议之后，在步骤 S3 中，客户向服务器发送一个已完成消息。作为响应，连接被建立起来，控制的主动权转移到服务器。

在一些情况下，连接被建立起来，以响应从客户端向服务器发送客户端问候完成消息 (client-hello-done message) (附图 25)。

即，特定的命令 (消息) 被从服务器发送到客户端，指示根据该命令执行的处理的结果的应答被从客户端返回给服务器。在连接被建立起来并且主动权转移到服务器之后，状态改变到中间状态 (图 5 中的状态 2)。

在步骤 S4 中，在中间状态下，服务器控制客户端的状态，或者根据应用的逻辑发送/接收该项应用的特定数据。

在这个处理过程中，使用更新实体子协议来进行对客户端状态的控制。在另一方面，应用数据传送子协议被用于发送/接收该项应用的特定数据，并且操作实体子协议被用于操作连接到客户端的设备。

只有在中间状态下，上述三个子协议的任意组合才被允许以任意顺序同时被发送/接收。例如，如同后面将参考图 20 至 22 进行描述的，一个数据包只能包含一种类型的子协议 (即，一个数据包包括一个报头和一种类型的子协议的消息序列)。在数据包的末尾，设置了一个该子协议的已完成消息。

中间状态下，在每个使用子协议（更新实体子协议，应用数据传送子协议，或者操作实体子协议）从服务器向客户端发送的消息中，有一个包含的信息，指示如果该项处理失败，后续的处理过程是否被继续。

例如，当客户端没能处理起始比特为“0”的消息的时候，就不再处理后续的消息。另一方面，当客户端没能处理起始比特为“1”的消息的时候，后续的消息仍被处理。这也用于其它的子协议扩展消息（subprotocol extension message）。

如果请求的逻辑已经完成，之后，在步骤 S5，服务器使用告别子协议，向客户端发送一个服务器告别消息（server-good-bye message）。发送服务器告别消息之后、释放连接之前的状态被称为告别状态（图 5 中的状态 3）。

如果服务器判断告别协议已完成，之后，在步骤 S6 中，服务器向客户端发送一个已完成消息，并结束连接。作为响应，在步骤 S7 中，客户端释放资源，结束处理过程。

在一些情况下，在服务器向客户端发送服务器告别完成消息（server-good-bye-done message）的时候，连接即被终止，而不经告别状态。作为响应，客户端释放资源，结束处理过程（附图 25）。

当在任意步骤（状态）中出现错误，并且不可能再继续后面的处理的时候，服务器的客户端跳到步骤 S8，并使用错误子协议向服务器或者客户端发送差错消息。从出现错误到使用控制协议结束通信连接之间的状态被称为错误状态（附图 5 中的状态 4）。

在步骤 S9，例如，当已完成消息从消息服务器被发送到客户端，错误状态结束，连接被关闭。

如上所述，在客户端和服务器之间的通信连接通过其状态以上述方式改变的控制协议来进行。

现在，下面描述通过数据传输协议定义的数据包的数据格式。

图 6 显示了数据包的数据格式的一个例子。

如图 6 所示，数据包包含报头和预定数据的消息（子协议消息）序列。

报头包括三个字段：版本字段，子协议类型（SPT）字段，以及长度字段。

控制协议的版本号在版本字段中描述。更具体地说，主版本号由版本号的高阶字节来代表，副版本号由低阶字节来代表。如果数据传输协议改变，则主版本号提高，而当子协议被添加或者改变，则副版本号提高。

子协议类型表示数据包中使用的子协议的类型，其中类型可以是握手协议、告别协议、差错协议、应用数据传输协议、更新实体协议或操作实体协议其中之一。

长度表示消息序列的长度。

包含在消息序列中的每个消息包括扩展（Ext）字段、设备 ID（DID）字段、消息类型（MT）字段、长度字段和数据字段。

在扩展字段，描述了该应用范围（application domain）的特定处理的扩展的类型。

在设备 ID 字段，描述了有待于按照消息而被控制的设备的 ID。例如，在多个设备连接到一个客户端的情况下，ID 可以被设定成，使其一般性地表示所有这些有待根据消息而被控制的设备，或者，也可以设定 ID，使得服务器和客户端可以处理该消息而不涉及特定设备。

在消息类型字段，描述了消息的类型。如后面将要描述的，握手协议可以包括从客户端发往服务器的客户端问候消息，或者从服务器发往客户端的服务器问候消息，并且这种消息的类型由消息类型字段指出。

在长度字段，描述了数据部分的数据长度。消息的数据部分的数据格式取决于消息类型。

下面，进一步详细描述了子协议。

图 7 显示了使用握手协议发送的消息。

如图 7 所示，已完成消息表示子协议的完成，警告消息（warning message）表示警告。警告消息的数据部分包含由特定字符串所代表的警告。

客户端问候消息表示开始从客户端向服务器传送握手数据，客户端问候完成消息表示握手数据的结束。

服务器问候消息表示开始从服务器向客户端传送握手数据，服务器问候完成消息表示握手数据的结束。

设备消息是设备清单通知。在设备消息的数据部分，设备 ID 被唯一地分配给各个设备，并且描述了许可（permission）、类型名称长度、类型名称、设备名称长度、设备名称。许可表示是否允许访问设备。当许可设为“0”，允许访问设备（操作设备）。然而，当许可设为“1”的时候，不允许访问（操作）设备。

设备消息例如被用来通知服务器有什么设备连接到客户端并可以被远程操作，以及作为设备可以执行什么处理以响应给定命令。

参考图 8 所示的序列流（sequence flow），下面描述了握手协议，图 7 所示的信息在其中发送。在这个序列中，通过举例，由做为客户端的手提电话 5，和作为通过通信连接、使用控制协议来控制设备的服务器的服务器 2 来执行处理过程。

在步骤 S21 中，例如当手提电话 5（客户端）的握手消息发送控制器 83（附图 4）例如向服务器 2 发送了一个客户端问候消息的时候，握手协议序列开始。更具体来说，为响应控制器 81 发出的命令，握手消息管理器 82 生成了客户端问候消息，并将其通过握手消息发送控制器 83 发送到服务器 2。

在步骤 S22 中，手提电话 5 的握手消息发送控制器 83 向服务器 2 发送设备消息，以提供与所有连接到手提电话 5 并可以被服务器 2 控制的设备相关的信息。这个设备消息也是在控制器 81 的控制之下，由握手消息管理器 82 生成的。

例如，关于连接到手提电话 5 的读/写器 11，以及通过读/写器 11 连接到手提电话 5 的外部 IC 卡 12，的信息，通过设备消息被提供

给服务器 2。

在步骤 S23 中，发送了该应用的特定消息。之后，当在步骤 S24 中，发出了客户端问候完成消息的时候，也完成了握手数据从手提电话 5 到服务器 2 的发送。

如果在步骤 S41 到 S44 中，服务器 2 的握手消息接收控制器 54 获得了从手提电话 5 发送的客户端问候消息、设备消息、应用的特定消息和客户端问候完成消息，那么，在步骤 S45 中，握手消息发送控制器 53（图 3）向手提电话 5 发送服务器问候消息。

在下一个步骤 S46 中，服务器 2 的握手消息发送控制器 53 发送一个应用的特定消息。之后，在步骤 S47 中，握手消息发送控制器 53 发送一个服务器问候完成消息。在上面的步骤中，包括应用的特定消息在内的所有消息，在握手消息发送控制器 53 的控制下，在握手协议中发送。

从服务器 2 发送的服务器问候消息、应用特定消息（application-specific）和服务器问候完成消息，分别在步骤 S25 到 S27 中被手提电话 5 的握手消息接收控制器 84 接收。

如果手提电话 5 和服务器 2 都已经发送了问候完成消息（客户端问候完成消息和服务器问候完成消息），那么，在步骤 S48 中，服务器 2 的握手消息发送控制器 53 向手提电话 5 发送一个已完成消息。

在步骤 S28 中，从服务器 2 发送的已完成消息被手提电话 5 的握手消息接收控制器 84 接收。

在步骤 S29，手提电话 5 的握手消息发送控制器 83 发送一个应用的特定消息。在下一个步骤 S30，手提电话 5 的握手消息发送控制器 83 向服务器 2 发送一个已完成消息。从手提电话 5 发送的应用的特定消息和已完成消息分别在步骤 S49 和 S50 中被服务器 2 的握手消息接收控制器 54 接收。此时，在手提电话 5 和服务器 2，握手协议均被完成。

在图 8 中，虚线表示只有当需要的时候应用的特定消息才被发

送。注意，发送了需要数目的应用的特定消息。例如，在步骤 S23 中，在一些情况下没有消息被发送，但是在另一些情况下会发送多个消息。在参考图 10 所描述的序列中也是如此。

尽管在图 8 中，当在步骤 S50 中服务器 2 接收到了已完成消息的时候，握手协议被完成，但是握手协议也可以在步骤 S44 中，当服务器 2 接收到客户端问候完成消息的时候被完成。

此外，尽管在图 8 中，服务器 2 和手提电话 5 都发送问候消息和问候完成消息，这些消息也可以只从手提电话 5 发送到服务器 2。这使得快速结束握手协议并开始在中状态下的处理过程成为可能，尽管握手协议的可靠性会稍微降低。换言之，如果手提电话 5 和服务器 2 发送/接收问候消息和问候完成消息，则握手协议在一种更为可靠的方式下被执行。

图 9 显示了用在告别协议中的消息。

已完成消息表示子协议的结束，警告消息表示警告。警告消息的数据部分包括由特定字符串代表的警告消息。

客户端告别消息表示开始从客户端向服务器传送告别数据，客户端告别完成消息表示告别数据的结束。

服务器告别消息表示开始从服务器向客户端传送告别数据，服务器告别完成消息表示告别数据的结束。

返回编码消息 (return-code message) 是对于来自服务器的编码结束的通知，其中它的数据部分包括结束编码 (end code)。

参考图 10 所示的序列流，下面描述告别协议，在其中图 9 所示的消息被发送。

在步骤 S81 中，当服务器 2 的告别消息发送控制器 56 向手提电话 5 发送了一个服务器告别消息的时候，告别协议开始。就是说，当在通信连接中占据主动的服务器 2 发送一个消息的时候（在握手协议完成之后，如前面参考图 5 所描述的，服务器 2 在通信连接中占据主动），告别协议开始。

在步骤 S82 中，服务器 2 的告别消息发送控制器 56 向手提电话

5 (客户端) 发送一个返回编码消息。在下一个步骤 S83 中, 应用的特定消息 (在扩展字段中指定的告别扩展消息) 被发送。

在完成了发送所有的应用的特定消息之后, 处理过程进行到步骤 S84。在步骤 S84, 服务器 2 的告别消息发送控制器 56 向手提电话 5 发送服务器告别完成消息。

在步骤 S81 到步骤 S84 中从服务器 2 发送的每条信息都在步骤 S61 到步骤 S64 中被手提电话 5 的告别消息接收控制器 87 接收。

在步骤 S65, 手提电话 5 的告别消息发送控制器 86 向服务器 2 发送客户端告别消息。

在下一个步骤 S66, 手提电话 5 的告别消息发送控制器 86 发送一条应用的特定消息。之后, 在步骤 S67, 手提电话 5 的告别消息发送控制器 86 发送客户端告别完成消息。

在客户端告别完成消息发送完成之后, 在步骤 S68 中, 手提电话 5 的告别消息发送控制器 86 发送一条已完成消息。在步骤 S65 至 S68 中, 从手提电话 5 的告别消息发送控制器 86 发送的每条消息都在步骤 S85 至 S88 中被服务器 2 的告别消息接收控制器 57 接收。

如果从手提电话 5 发出的已完成消息在步骤 S88 中被服务器 2 的告别消息接收控制器 57 接收, 则处理进行到步骤 S89。在步骤 S89 中, 告别消息发送控制器 56 向手提电话 5 发送已完成消息, 并结束告别协议。从服务器 2 的告别消息发送控制器 56 发出的已完成消息在步骤 S69 中被手提电话 5 接收。

或者也可以, 当在步骤 S84 中发送了服务器告别完成消息的时候结束告别协议。这使得有可能快速结束告别协议。

图 11 显示了用于差错协议中的消息。

已完成消息表示子协议的结束, 警告消息表示警告。警告消息的数据部分包括由特定字符串代表的警告消息。

数据包格式差错消息表示数据包包含了格式错误。非法状态差错消息表示检测到了非法消息。

意外差错消息表示发生了意外错误。

数据包格式差错消息、非法状态差错消息、以及意外差错消息的数据部分，包括表示由特定字符串代表的错误的内容的差错消息。

参考图 12 和图 13 所示的序列流，描述了差错协议，在其中图 11 所示的消息被发送。

图 12 显示了当手提电话 5 检测到错误的时候，执行的差错协议序列。

如果手提电话 5 的控制器 81 检测到使其不能够进行进一步处理的错误，在步骤 S101 中，控制器 81 命令差错消息管理器 88 发送一条如附图 11 所示的消息（数据包格式差错消息、非法状态差错消息、或者意外差错消息），以通知服务器 2 出现了错误（以及检测到的错误的的内容）。更具体来说，在控制器 81 的控制下，差错消息管理器 88 生成特定消息，并通过差错消息发送控制器 89 将其发送。

之后，在步骤 S102 中，从手提电话 5 发送已完成消息，并在步骤 S112 中，该消息被服务器接收。在步骤 S113 中，已完成消息从服务器 113 发送，并在步骤 S103 中被手提电话 5 接收。这样，如上文中参考附图 5 所述，在服务器和客户端之间的通信连接被终止。

附图 13 显示了当服务器 2 检测到错误的时候，执行的差错协议。

如果服务器 2 的控制器 51 检测到使其不能够进行进一步处理的错误，在步骤 S131 中，控制器 51 命令差错消息管理器 58 根据检测到的错误的的内容，向手提电话 5 发送一条消息，以通知出现了错误。更具体来说，在控制器 51 的控制下，差错消息管理器 58 生成特定消息，并通过差错消息发送控制器 59 将其发送。

在步骤 S132 中，在发送表示出现错误的消息之后，服务器 2 的差错消息发送控制器 59 向手提电话 5 发送已完成消息。

从服务器发出的该消息在步骤 S121 和步骤 S122 中，被手提电话 5 接收，通信连接被终止。

如上所述，当检测到错误的时候，服务器 2 和手提电话 5 都能够任意时间向其它设备发送差错消息。

为取代使用错误子协议发送消息，可以发送在应用中定义的差错消息，以通知出现了错误。

不同于差错消息，警告消息不影响序列。当服务器 2 或者手提电话 5 中的任何一个收到警告消息的时候，消息的内容可能被写入日志，或者被忽略。

图 14 显示了在更新实体协议中发送的消息。

在更新实体协议中发送的消息用来改变设备的状态（属性）。例如，这些消息被用于打开安装在图 1 所示的读/写器表面上的 LED（发光二极管），或者打开特定设备的电源。

已完成消息表示子协议的结束，警告消息表示警告。警告消息的数据部分包括由特定字符串代表的警告消息。

设定属性消息从服务器被传送到客户端，以设定连接到客户端的设备的属性。

此设定属性消息是其失败会导致不能进行后续处理的消息之一。因此，设定属性消息的第一个比特被设定为表示如果出现错误则处理过程被终止。

设定属性消息的数据部分包括与待设定到设备的属性相关的信息，例如属性名称长度、属性名称、属性值长度和属性值。

作为对服务器发出的设定属性消息的应答而从客户端返回的属性已设定消息，其数据部分包括一个标识，表示由设定属性消息指定的处理过程是否被客户端成功执行。

获得属性消息由服务器发送到客户端，以获得连接到客户端的设备的属性值。获得属性消息是作为对属性消息的应答，从客户端返回的消息。

获得属性消息是其失败会导致不能进行后续处理的消息之一。因此，获得属性消息的第一个比特被设定为表示如果出现错误则处理过程被终止。

获得属性消息的数据部分包括，表示需要获得其属性值的设备的属性名称长度、属性名称以及诸如此类的信息。

属性消息是作为对从服务器发送的获得属性消息的应答，从客户端返回的消息。属性消息的数据部分包括，表示服务器所请求的设备的属性名称长度、属性名称以及诸如此类的信息。

设定网络超时消息表示在客户端设定的网络超时时间。

此设定网络超时消息是其在客户端的失败不会导致后续处理终止的消息之一。因此，设定网络超时消息的第一个比特被设定为表示如果出现错误则处理过程应当继续。

设定网络超时消息的数据部分包括有待设定在客户端的超时时间值。

参考图 15 所示的序列，下面描述更新实体协议，如图 14 所示的信息在其中传送。

当服务器 2 希望改变连接到手提电话 5 的设备的状态的时候，在步骤 S151 中，在控制器 51 的控制下，服务器 2 的更新实体协议管理器 64 生成一条更新实体消息，以传送到手提电话 5。生成的更新实体协议被提供给更新实体消息发送控制器 65。

在步骤 S152 中，服务器 2 的更新实体消息发送控制器 65 在步骤 S151 中生成的更新实体消息序列的末尾添加一条已结束消息，并向手提电话 5 发送包含更新实体消息序列和已完成消息的数据包。

就是说，包含在步骤 S151 中生成的特定数目的更新实体消息和在步骤 S152 中添加的已完成消息的数据包，被发送到手提电话 5。

因为在一个数据包中结合在一起的多个消息可以在一次发送到手提电话 5（在这种情况下，需要在每次发送多个消息的时候添加一个报头和一个报尾），比起在一个数据包中只有一条消息和一个报头、一个报尾放在一起，并且消息被一条一条地发送的情况，数据长度可以被缩短。这也使得通信时间和通信成本都缩减了。这一点对于所有的协议都是这样。

包含在数据包中并从服务器 2 发出的更新实体消息在步骤 S141 中，被手提电话 5 的更新实体消息接收控制器 96 接收，并且，添加在更新实体消息序列末尾的已完成消息在步骤 S142 中被接收。

在手提电话 5 中，当包含在从服务器发出的数据包中的消息被接收的时候，这些消息被处理的顺序与这些消息被获得的顺序是一样的。按照要求，表示处理结果的应答被控制器 81 生成。

手提电话 5 的更新实体消息接收控制器 96 通过检测服务器 2 发送的已完成消息，来检测数据包的末尾。在步骤 S143 中，更新实体消息管理器 94 对于接收到的更新实体消息生成应答（更新实体消息）。

例如，如果从服务器 2 接收的更新实体消息表示一个设备的属性值应当改变，则设备控制器 100 根据更新实体消息改变该属性值，并且控制器 81 通知更新实体消息管理器 94 属性值是否被成功改变。基于该通知，更新实体消息管理器 94 生成一个应答，并且将生成的应答提供给更新实体消息发送控制器 65。

在步骤 S144 中，手提电话 5 的更新实体消息发送制器 95 向生成的应答添加一个已完成消息，并向服务器 2 发送包含了应答和已完成消息的数据包。是否返回应答，以及返回多少个应答，都取决于从服务器 2 接收的消息。

在步骤 S153，从手提电话 5 发出的、包含了更新实体消息序列的数据包被服务器 2 的更新实体消息接收控制器 66 接收。在步骤 S154 中，添加在更新实体消息序列的末尾的已完成消息被获得。

下面，在中间状态下从服务器 2 发送到手提电话 5 的数据包被称为控制数据包（控制数据包的具体例子是：不包含、包含一个或包含多个更新实体消息和一个已完成消息的数据包，不包含、包含一个或包含多个应用数据传送消息和一个已完成消息的数据包，不包含、包含一个或包含多个操作实体消息和一个已完成消息的数据包）。

根据包含在控制数据包中的消息，从手提电话 5 返回到服务器 2，以通知执行的处理的结果的数据包，被称为应答数据包（其中每个应答数据包不包含、包含一个或包含多个应答和一个已完成消息）。

如上所述，包含更新实体消息和一个已完成消息的控制数据

包，通过在服务器 2 占据主动之下进行的通信连接，从服务器 2 被传送到手提电话 5，并且，手提电话 5 根据该更新实体消息执行一项处理过程。此外，在手提电话 5 中，生成对于从服务器 2 接收的更新实体消息的应答，并且，包含该应答和一个已完成消息的应答数据包被返回给服务器 2。

更具体来说，如果包含如图 14 所示的设定属性消息的控制数据包被从服务器 2 传送到手提电话 5，那么，在手提电话 5 中，根据该设定属性消息来设定设备的属性值，并且包含一个属性已设定消息（应答）的应答数据包被返回给服务器 2。

从手提电话 5 或者服务器 2 发出的更新实体消息序列可以包括特定数目的应用特定消息（在扩展字段中指定的更新实体扩展消息）。

图 16 显示了用在应用数据传输协议中的消息。

应用数据传输协议被用于，例如，当用于控制通过读/写器 11 连接到手提电话 5 的外部 IC 卡 12 的应用数据或者应用命令在服务器和客户端（设备）之间被发送的时候。

已完成消息表示协议结束，警告消息表示警告。警告消息的数据部分包括由特定字符串代表的警告消息。

参考图 17 所示的序列流，下面描述应用数据传输协议，在其中图 16 所示的消息被发送。

在步骤 S171 中，服务器 2 的应用数据传输协议管理器 61 生成一组应用数据传送消息。

生成的应用数据传送消息被提供给应用数据传送消息发送控制器 62。

在步骤 S172 中，服务器 2 的应用数据传送消息发送控制器 62 在生成的应用数据传送消息的末尾添加一个已完成消息，并将包含该应用数据传送消息序列和已完成消息的控制数据包发送到手提电话 5。

包含在控制数据包中的应用数据传送消息在步骤 S161 中，被手

提电话 5 的应用数据传送消息接收控制器 93 接收，并且，已完成消息在步骤 S162 中被接收。

手提电话 5 的应用数据传送消息接收控制器 93 通过检测已完成消息，检测到控制数据包的末尾，手提电话 5 的应用数据传送消息接收控制器 93 将包含在控制数据包中的应用数据传送消息提供给应用数据传送消息管理器 91。

在步骤 S163 中，生成了对于包含在数据包中的应用数据传送消息的应答。更具体来说，表示设备控制器 100 执行的处理的结果的消息，被提供给应用数据传送消息管理器 91，并且应用数据传送消息管理器 91 根据处理结果生成应答。生成的应答被提供给应用数据传送消息发送控制器 92。

在步骤 S164 中，手提电话 5 的应用数据传送消息发送控制器 92 在生成的应答（应用数据传送消息）的末尾添加一个已完成消息，并将包含应答序列和已完成消息的应答数据包发送给服务器 2。

在步骤 S173 中，包含在从手提电话 5 发送的数据包中的应用数据传送消息被服务器 2 的应用数据传送消息接收控制器 63 获得，而添加在应用数据传送消息序列末尾的已完成消息在步骤 S174 中获得。

是否返回应答，以及返回多少应答，都取决于从服务器 2 接收的消息。

由于，在上述的更新实体协议的情况下，从手提电话 5 或者服务器 2 发出的应用数据传送消息可能包含特定数目的应用特定消息（在扩展字段中指定的应用数据传送扩展消息）。

图 18 显示了用于操作实体协议的消息。

操作实体协议用于控制连接到客户端的设备。例如，从连接到手提电话 5 的读/写器 11 启动或者停止发射无线电波，或者显示特定的屏幕用来让用户输入命令/数据，这些都通过操作实体协议来控制。

已完成消息表示协议的结束，警告消息表示警告。警告消息的

数据部分包括由特定字符串代表的警告消息。

更新显示消息 (update-view message) 从服务器被发往客户端, 以用来更新客户端的显示 (显示屏)。对更新显示消息的应答作为显示已更新消息, 被返回到服务器。

更新显示消息是其失败会导致不能进行后续处理的消息之一。因此, 更新显示消息的第一个比特被设定为表示如果出现错误则处理过程被终止。

显示已更新消息是作为对于从服务器接收的更新显示消息的应答而返回的消息。显示已更新消息的数据部分包含一个标识, 表示显示是否被成功更新。

开始数据输入消息由服务器被发往客户端, 以请求客户端开始向设备输入数据。结束数据输入消息从客户端返回, 作为对于开始数据输入消息的应答。

开始数据输入消息是其失败会导致不能进行后续处理的消息之一。因此, 开始数据输入消息的第一个比特被设定为表示如果出现错误则处理过程被终止。

作为对开始数据输入消息的应答而从客户端返回的结束数据输入消息的数据部分包含一个表示数据是否被成功输入的标识, 还包含表示属性名称长度、属性名称、属性值长度和属性值的信息。

操作设备消息由服务器被发往客户端, 以请求客户端开始对连接到客户端的设备进行操作。

设备应答消息从客户端返回, 作为对于操作设备消息的应答。

操作设备消息是其失败会导致不能进行后续处理的消息之一。因此, 操作设备消息的第一个比特被设定为表示如果出现错误则处理过程被终止。

操作设备消息的数据部分包含表示操作名称的字符数目、操作名称、参数长度和参数的信息。

设备应答消息从客户端被发往服务器, 以通知在设备上执行的操作的结果。设备应答消息的数据部分包含表示应答长度和应答的信

息。

播放声音消息由服务器被发往客户端，以请求客户端根据服务器指定的逻辑名称来生成声音。没有对播放声音消息的应答返回。

播放声音消息是其在客户端的失败不会导致处理被终止的消息之一。因此，播放声音消息的第一个比特被设定为表示如果出现错误，则处理过程应当继续。

播放声音消息的数据单元包含表示待生成的声音的逻辑名称的信息。

参考图 19 所示的序列流，下面描述操作实体协议，在其中图 18 所示的消息被发送。

在步骤 S191 中，服务器 2 的操作实体消息管理器 67 生成一组有意义的操作实体消息，待发送到手提电话 5。生成的操作实体消息被提供给操作实体消息发送控制器 68。

在步骤 S192 中，服务器 2 的操作实体消息发送控制器 68 向生成的消息序列添加一个已完成消息，并向手提电话 5 发送包含消息序列和已完成消息的数据包。

包含在发往手提电话 5 的控制数据包中的操作实体消息在步骤 S181 中被接收，已完成消息在在步骤 S182 中被接收。

在步骤 S183 中，在控制器 81 的控制下，手提电话 5 的操作实体消息管理器 97 生成对于从服务器 2 接收的操作实体消息的应答。是否返回应答，以及返回多少应答，都取决于从服务器 2 接收的消息。

在步骤 S184 中，手提电话 5 的操作实体消息发送控制器 95 向操作实体消息管理器 97 生成的应答添加一个已完成消息，并服务器 2 发送包含应答和已完成消息的数据包。

包含在从手提电话 5 发出的数据包中的操作实体消息在步骤 S193 中被服务器 2 的操作实体消息接收控制器 69 接收，添加在操作实体消息序列末尾的已完成消息在步骤 S194 中被接收。

从手提电话 5 或者服务器 2 发出的操作实体消息序列可以包括

特定数目的应用特定消息（在扩展字段中指定的操作实体扩展消息）。

如上所述，在中间状态（图 5），在手提电话 5 和服务器 2 之间成功执行了握手之后，服务器 2 在通信连接中取得主动权。就是说，在中间状态，服务器 2 向手提电话 5 发送使用更新实体协议、应用数据传输协议或者操作实体协议的消息（注意，在任何使用上面参考附图 15 描述的更新实体协议的通信序列、使用参考图 17 描述的更新实体协议的通信序列、以及使用参考图 19 描述的操作实体协议的通信序列中，通信连接都是由服务器 2 发起的）。

就是说，在中间状态，通信连接不是以从服务器 2 返回到手提电话 5 一个应答，并根据该应答来控制连接到手提电话 5 的设备这种方式而完成的，而是，以根据直接从服务器 2 发往手提电话 5 的命令来控制设备的方式。

这意味着，一旦通信连接被建立起来，服务器 2 可以通过通信连接，直接控制设备。

参考图 20，下面描述了在中间状态下，服务器（服务器 2）和客户端（手提电话 5），通过使用上述子协议的通信连接，来执行的处理序列。

在图 20 的步骤 S211 中，服务器 2 向手提电话 5 发送控制数据包（请求数据传送数据包）201，包含应用数据传送消息 A（MsgA），应用数据传送消息 B（MsgB），以及放在消息序列 MsgA 和 MsgB 的末尾的已完成消息。

在使用应用数据传输协议的通信连接中，如上面参考图 17 所描述的，服务器 2 的应用数据传送消息发送控制器 62 通过在由特定数目的应用数据传送消息构成的序列的末尾添加一个已完成消息而生成控制数据包，并将生成的控制数据包传送到手提电话 5。

在步骤 S211 中，通过在操作实体消息 X（MsgX）的末尾添加一个已完成消息，生成了控制数据包（操作实体数据包）202，并且生成的控制数据包从服务器 2 被传送到手提电话 5。

在使用操作实体协议的通信连接中，如上面参考图 19 所描述的，服务器 2 通过在由特定数目的操作实体消息构成的序列的末尾添加一个已完成消息而生成控制数据包，并将生成的控制数据包传送到手提电话 5。

此外，在此步骤 S211 中，通过在应用数据传送消息 C (MsgC) 的末尾添加一个已完成消息，而生成控制数据包 (请求数据传送数据包) 203，并且生成的控制数据包从服务器 2 被传送到手提电话 5。

如上所述，多个控制数据包被一次发送到手提电话 5，这样，比起控制数据包被分别传送的情况，进行通信的时间被缩短了。

包含在从服务器 2 发送的控制数据包中的消息被手提电话 5 处理，并按照与发送过来的消息同样的顺序生成应答。手提电话 5 生成的应答则按照与生成应答同样的顺序，被返回到服务器 2。

在图 20 所示的例子中，当控制数据包 201 和 203 在步骤 S201 中被手提电话 5 接收，手提电话 5 根据接收到的消息，按照应用数据传送消息 A、应用数据传送消息 B、操作实体消息 X 和应用数据传送消息 C 的顺序，执行处理。

如上所述，如果在根据消息而执行的处理过程中出现了错误，后面的处理过程是继续还是终止取决于消息的第一个比特的值。

在图 20 所示的例子中，假设所有发送的消息都被成功执行。应用数据传送消息 A 和操作实体消息 X 需要应答，而应用数据传送消息 B 和应用数据传送消息 C 不需要应答。

对于必须被服务器 2 管理的消息，按照该项处理是否被手提电话 5 成功完成，手提电话 5 生成应答。

这样，在完成了根据应用数据传送消息 A 进行的处理之后，在步骤 S202 中，手提电话 5 生成了对于应用数据传送消息 A 的应答 A (ResA)，并且，把通过在应答 A 上添加已完成消息而生成的应答数据包 (请求数据传送数据包) 211 返回到服务器 2。

在使用应用数据传输协议的通信连接中，如上面参考图 17 所描

述的，应用数据传送消息管理器 91 对于包含在从服务器 2 发送的控制数据包中的应用数据传送消息生成应答，并将包含生成的应答和已完成消息的应答数据包，通过应用数据传送消息发送控制器 92，返回到服务器 2。

此外，在完成了根据操作实体协议 X 进行的处理之后，在步骤 S202 中，手提电话 5 生成了对于操作实体消息 X 的应答 X (ResX)，并且，把通过在应答 X 上面添加已完成消息而生成的应答数据包 (操作实体数据包) 212 返回到服务器 2。

在使用应用数据传输协议的通信连接中，如上面参考附图 19 所描述的，操作实体消息管理器 97 对于包含在从服务器 2 发送的控制数据包中的操作实体消息生成应答，并将包含生成的应答和已完成消息的应答数据包，通过操作实体消息发送控制器 98，返回到服务器 2。

对于每个从服务器 2 接收的控制数据包，手提电话 5 必须向服务器 2 返回应答数据包，而不论应答数据包是否包含应答消息。这样，在步骤 S202 中，即使在应用数据传送消息 C 不需要应答的情况下，对于控制数据包 203，也会返回给服务器 2 一个仅包含一个已完成消息的应答数据包 (请求数据传送数据包) 213。

在步骤 S212 中，从手提电话 5 发送的应答数据包被服务器 2 接收。从包含在应答数据包中的应答，服务器 2 识别出手提电话 5 已经根据该消息成功执行了处理。

如上所述，由于应答消息是按照与消息从服务器 2 发出相同的顺序生成的，服务器 2 和手提电话 5 可以按照已发送的消息和应答之间的相互关系来管理它们。

图 21 显示了另一个由服务器 2 和手提电话 5 执行的处理序列。

在图 21 的步骤 S231 中，如同在图 20 中所示的序列的情况，控制数据包 221 和 223 从服务器 2 被发送到手提电话 5。

在图 21 所示的例子中，假定手提电话 5 根据从服务器发往手提电话 5 的应用数据传送消息 A (MsgA) 来执行的处理失败。此外，

还假定应用数据传送消息 A 的第一个比特表示，表示如果在应用数据传送消息 A 的处理过程中出现错误，则处理过程被终止。

这样，当在步骤 S221 中，手提电话 5 获得包含在控制数据包 221 中的应用数据传送消息 A 的时候，如果手提电话 5 根据获得的消息执行的处理失败，那么手提电话 5 不再根据后面的消息，即应用数据传送消息 B、操作实体消息 X 和应用数据传送消息 C，来执行处理。

当手提电话 5 没能根据应用数据传送消息 A (MsgA) 来进行处理的时候，手提电话 5 生成应答 A (ResA)，以通知服务器 2 处理失败。在步骤 S222 中，手提电话 5 通过在生成的应答 A 上面添加已完成消息，而生成应答数据包 (请求数据传送数据包) 231，并将生成的应答数据包 231 返回到服务器 2。

在步骤 S232 中，服务器 2 接收到从手提电话 5 发送的应答数据包 231。从包含在应答数据包 231 中的应答 A，服务器 2 知道手提电话 5 按照应用数据传送消息 A 来执行的处理失败。服务器 2 也认识到，根据应用数据传送消息 B、操作实体消息 X 和应用数据传送消息 C 的处理过程不会被执行。

附图 22 显示了由服务器 2 和手提电话 5 执行的处理序列的另一个例子。

在附图 22 的步骤 S251 中，如同在附图 20 和 21 中所示的序列的情况，控制数据包 241 和 242 从服务器 2 被发送到手提电话 5。

在附图 21 所示的例子中，假定手提电话 5 根据从服务器发往手提电话 5 的应用数据传送消息 B (MsgB) 来执行的处理失败。此外，还假定应用数据传送消息 B 的第一个比特被设定为表示，如果在应用数据传送消息 B 的处理过程中出现错误，则处理过程应当继续。

应用数据传送消息 A 和操作实体消息 X 需要应答，而应用数据传送消息 C 不需要应答。

这样，在步骤 S241 中，手提电话 5 接收到控制数据包 241，根据应用数据传送消息 A 执行处理，并生成应答 A (ResA)。

如果手提电话 5 根据下一个消息，即应用数据传送消息 B，来执行的处理失败，手提电话 5 继续进行后面的处理，这是因为应用数据传送消息 B 的第一个比特表示，后续处理过程应当继续。就是说，手提电话 5 根据包含在控制数据包 242 中的操作实体消息 X 而执行处理，并生成对操作实体消息 X 的应答 X (ResX)。

在完成生成应答 X 之后，手提电话 5 根据包含在控制数据包 243 中的应用数据传送消息 C 执行处理。

在步骤 S242 中，包含应答 A 和已完成数据包的应答数据包（请求数据传送数据包）251、包含应答 X 和已完成数据包的应答数据包（操作实体数据包）252、以及包含已完成数据包的应答数据包（请求数据传送数据包）253 被手提电话 5 生成，并被传送到服务器 2。

在步骤 S252 中，从手提电话 5 发出的应答数据包被服务器 2 接收。从这些应答数据包中，服务器 2 知道根据应用数据传送消息 A 和操作实体消息 X 的处理已经被手提电话 5 成功执行，而根据应用数据传送消息 B 的处理则已经失败。此后，处理序列被终止。

在中间状态，数据包在服务器 2 和手提电话 5 之间，按照上述方式，被反复地发送/接收，并且客户端（设备）由服务器控制。

尽管在上面描述的实施例中，消息向手提电话 5 的发送以及应答到服务器 2 的返回被反复地执行，以控制特定设备，但服务器 2 也可能向手提电话 5 与控制数据包一起传送脚本（简单的程序），手提电话 5 可以根据脚本来判断下一步执行什么处理以及根据消息进行的前面的处理的结果。

这样，脚本指示根据哪条消息应当进行什么处理，取决于对每条消息生成的应答（设备的应答）。

在完成了根据脚本的处理之后，从设备向手提电话 5 返回的应答都在一个应答数据包中进行描述，并发送到服务器 2。

根据发送到服务器 2 的脚本执行历史（表示有什么消息按照什么顺序从手提电话 5 被传送到设备（什么消息被手提电话 5 处理）），服务器 2 可以判断从手提电话 5 发出的哪个应答相应于哪条

消息。

尽管在上述实施例中，在通信连接根据握手协议被建立起来之后，在服务器 2 和手提电话 5 之间的通信连接在服务器 2 占据主动下进行，但是在特定时间，手提电话 5 也有可能获得通信连接的主动权。

通信连接主动权的改变使用，例如，服务器-客户端传送方向翻转 (reverse) 协议，来进行。

例如，当通信连接在服务器 2 的主动之下进行，如果一条交换消息 (exchange message) 从服务器 2 被发送到手提电话 5，使用带有特定定时的服务器-客户端传送方向翻转协议，后面的通信连接按照与上面描述的相类似的方式进行，但是在手提电话 5 的主动之下。

当通信连接在手提电话 5 的主动之下进行时，如果交换消息从手提电话 5 被发送到服务器 2，并使用带有特定定时的服务器-客户端传送方向翻转协议，则主动权被传递到服务器 2，下面的通信连接在服务器 2 的主动下进行。

图 23 和 24 显示了交换协议序列。

在如图 23 所示的序列的情况下，开始，手提电话 5 (客户端) 掌握通信连接的主动权。在步骤 S301 中，一个交换消息从手提电话 5 被发送，并在步骤 S311 中被服务器 2 接收。之后，在服务器 2 和手提电话 5 之间的通信连接在服务器 2 的主动之下被继续。

在如图 24 所示的序列的情况之下，开始，服务器 2 掌握通信连接的主动权。在步骤 S331 中，一个交换消息从服务器 2 被发送，并在步骤 S321 中手提电话 5 被接收。之后，在服务器 2 和手提电话 5 之间的通信连接在手提电话 5 的主动之下被继续。

上面描述了在中间状态下，通信连接的主动权交换的方式。

当手提电话 5 的防火墙存在于建立在服务器 2 和手提电话 5 之间的通信信道中的情况下，在服务器 2 和手提电话 5 之间的通信连接通过防火墙使用 HTTP 来进行。在这种情况下，所有使用上述协议的数据包在，保持在整个通信连接期间中的单一会话中，被发送/接

收。

图 25 显示了控制协议的状态转换的另一个例子，它可以用来代替图 5 中所示的状态转换。在下面的讨论中，没有描述与图 5 类似的处理。

在步骤 S343 中，在完成了握手协议之后，客户端向服务器发送客户端问候完成消息。作为应答，连接被建立起来，控制的主动权被传递到了服务器。

在步骤 S344，在中间状态下，通信连接在服务器的主动之下，使用更新实体子协议、更新实体子协议以及操作实体子协议，被执行。在此通信连接中，按照要求，主动权通过交换消息被传递。

在步骤 S346 中，如果服务器判断告别协议已完成，服务器向客户端发送一个服务器告别完成消息，并终止连接。客户端的资源被释放，处理过程被终止。

如上所述，也可以通过发送服务器告别完成消息来终止处理过程。

在如图 26 所示的例子中，当客户端告别完成消息从客户端被发往服务器的时候，连接被终止。

就是说，在步骤 S361 中，客户端问候消息由客户端发往服务器。在下一个步骤 S362 中，设备消息被传送。

在步骤 S363 中，应用特定消息被发送。在下一个步骤 S364 中，客户端问候完成消息被发送，在服务器和客户端的连接被建立起来。

同时，在步骤 S371 至 S374 中，服务器接收从客户端发出的客户端问候消息、客户端问候消息、应用特定消息和应用特定消息。

例如，在客户端是手提电话或者类似设备的情况下，在客户端和服务器之间的连接可以通过上述序列建立起来。比起图 8 所示的序列，这个序列更为简化。

附图 27 显示了序列的另一个例子，在其中，当服务器问候完成消息从服务器被发往客户端时，告别协议被终止。

在这个序列中，在步骤 S391 中，当服务器发送服务器告别消息时，告别协议开始。在下一个步骤 S392，返回编码消息被发送。在步骤 S393 中，应用特定消息被发送。

在步骤 S394 中，服务器向客户端发送服务器告别完成消息，并且连接被关闭。

从服务器发出的服务器告别消息、返回编码消息、应用特定消息和服务器告别完成消息，在步骤 S381 到 S384 中，被客户端接收。为响应接收到的服务器告别完成消息，客户端资源被释放。

在客户端是手提电话或者类似设备的情况下，如同在握手协议中，在客户端和服务器之间的连接可以通过上述序列被终止。比起图 10 所示的序列，这个序列更为简化。

上述处理序列可以由硬件或者软件来执行。

当处理序列由软件来执行的时候，形成软件的程序可以从存储介质或者通过网络安装到作为专用硬件的计算机上，或者安装到能够根据安装在其上的各种程序来执行各种处理的通用计算机上。

可以用于上述目的的存储介质的具体例子包括，如附图 2 所示，磁盘 41（例如软盘）、光盘 42（例如 CD-ROM（压缩光盘只读存储器）、DVD（数字通用光盘）、磁光盘 43（例如 MD（迷你光盘（注册商标）））或者半导体存储器 44，以数据包介质的形式，程序存储于其上，并将其从计算机分别提供给用户。程序也可以通过预先安装在内嵌的 ROM 32 或者存储单元 38（例如安装在计算机中的硬盘）上面，而提供给用户。

在本描述中，在存储于存储介质中的程序中描述的步骤可以按照程序中描述的顺序，以时间序列来执行，或者也可以以并行或者单独的方式来执行。

在本描述中，术语“系统”用来指整台的设备。

如同可以从上面的描述中理解的，本发明提供了很大的进步。就是说，本发明使得信息处理设备和信息终端可以互相通信，特别是，以一种信息处理设备掌握通信的主动权的方式互相通信。

本发明可以以高可靠性和有效性的方式发送/接收各种数据包，这样就使得信息处理设备可以以高可靠性和有效性的方式控制信息终端。这种高效率的通信使得通信成本可以降低。

此外，本发明使得即使在处理中间出现错误时，客户端也可以继续处理过程。

图1

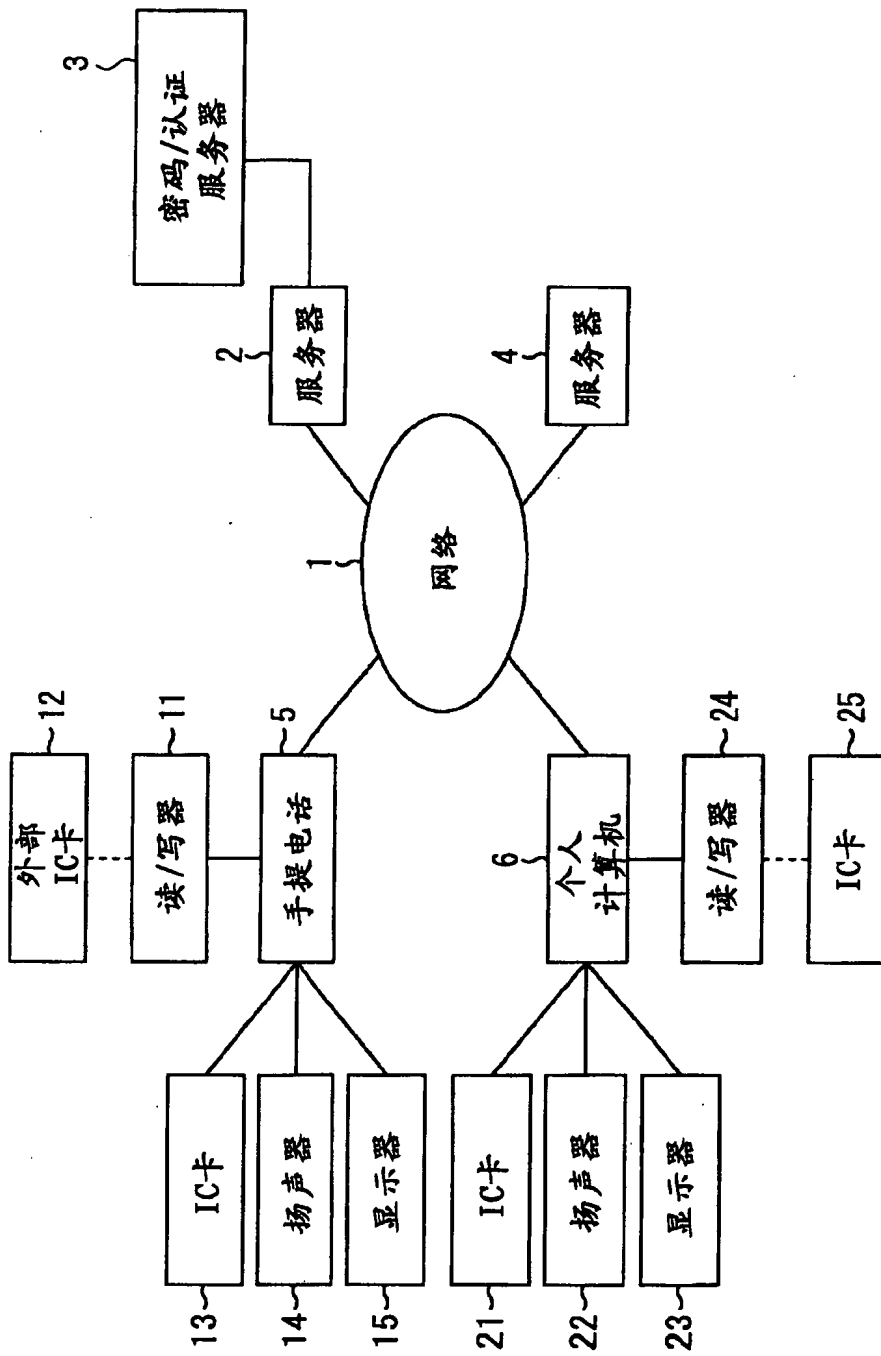


图2

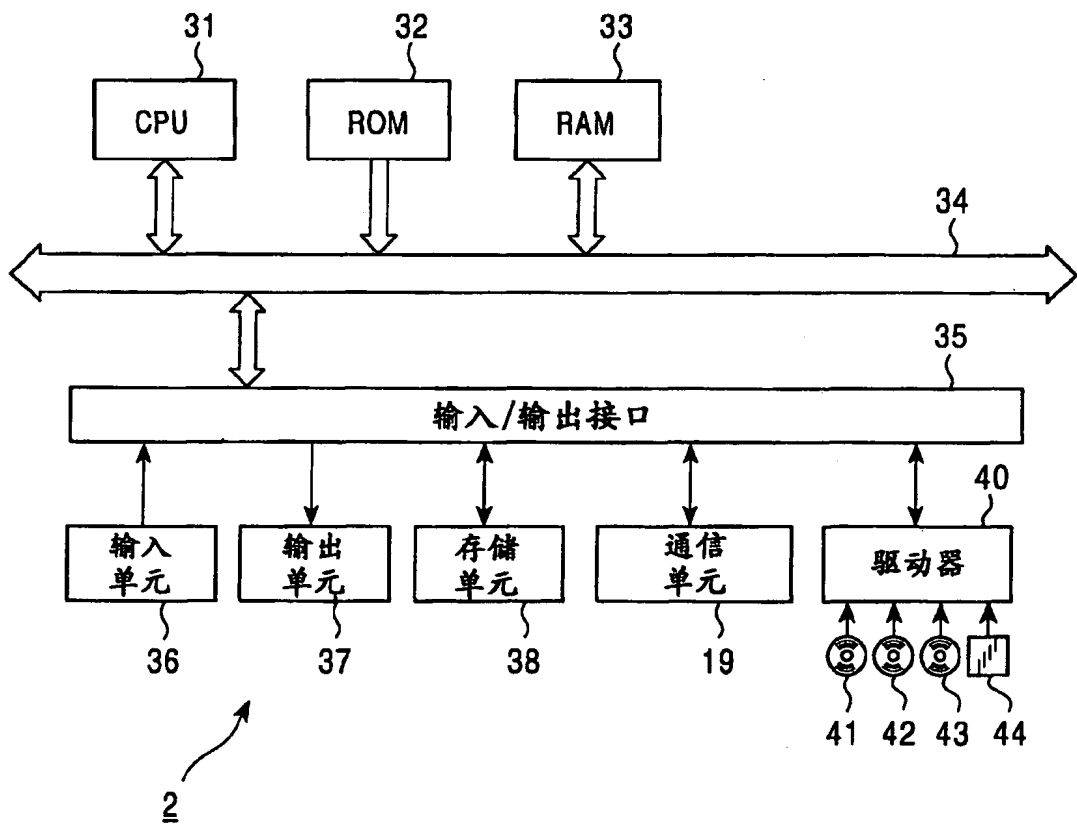


图3

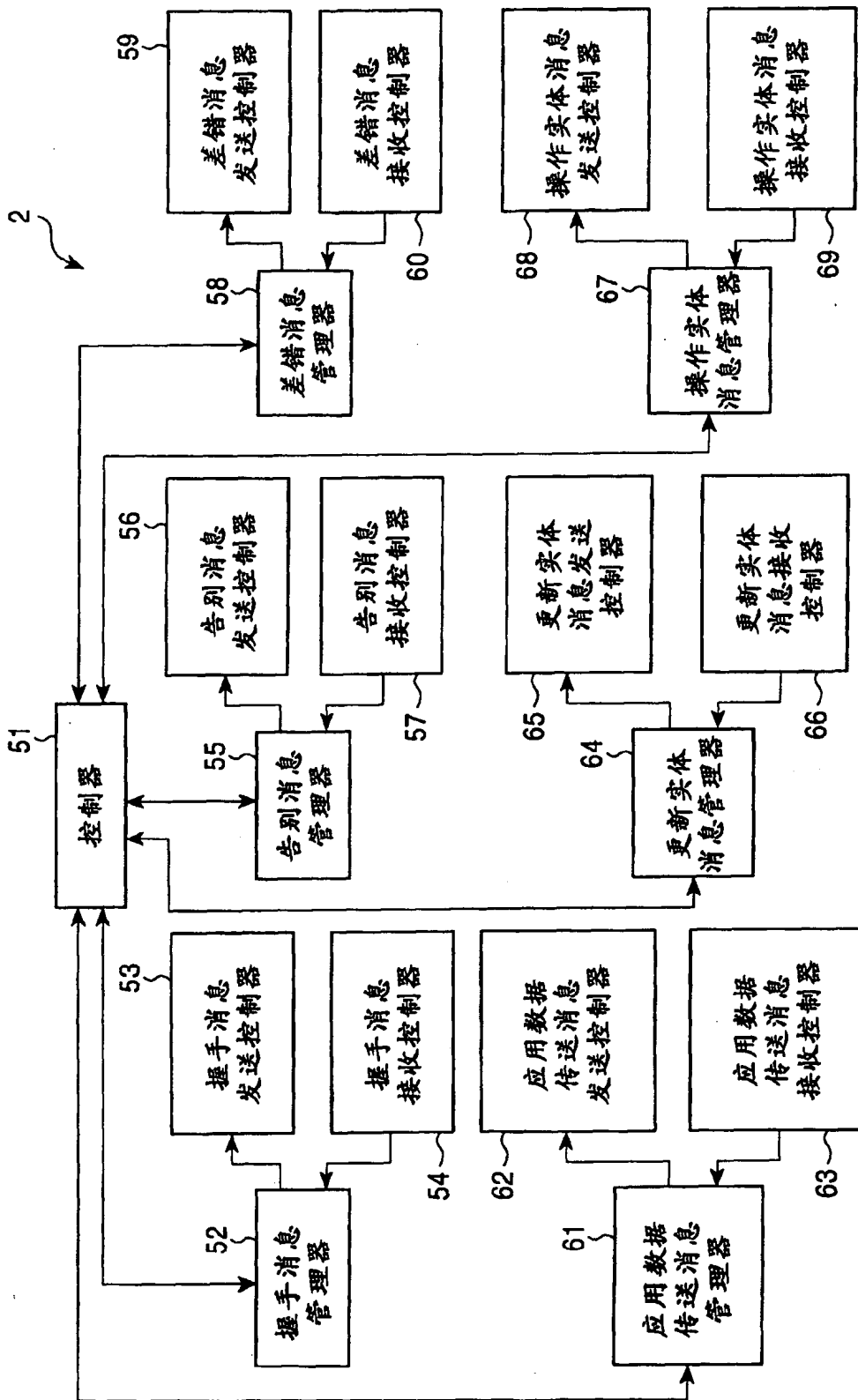


图 4

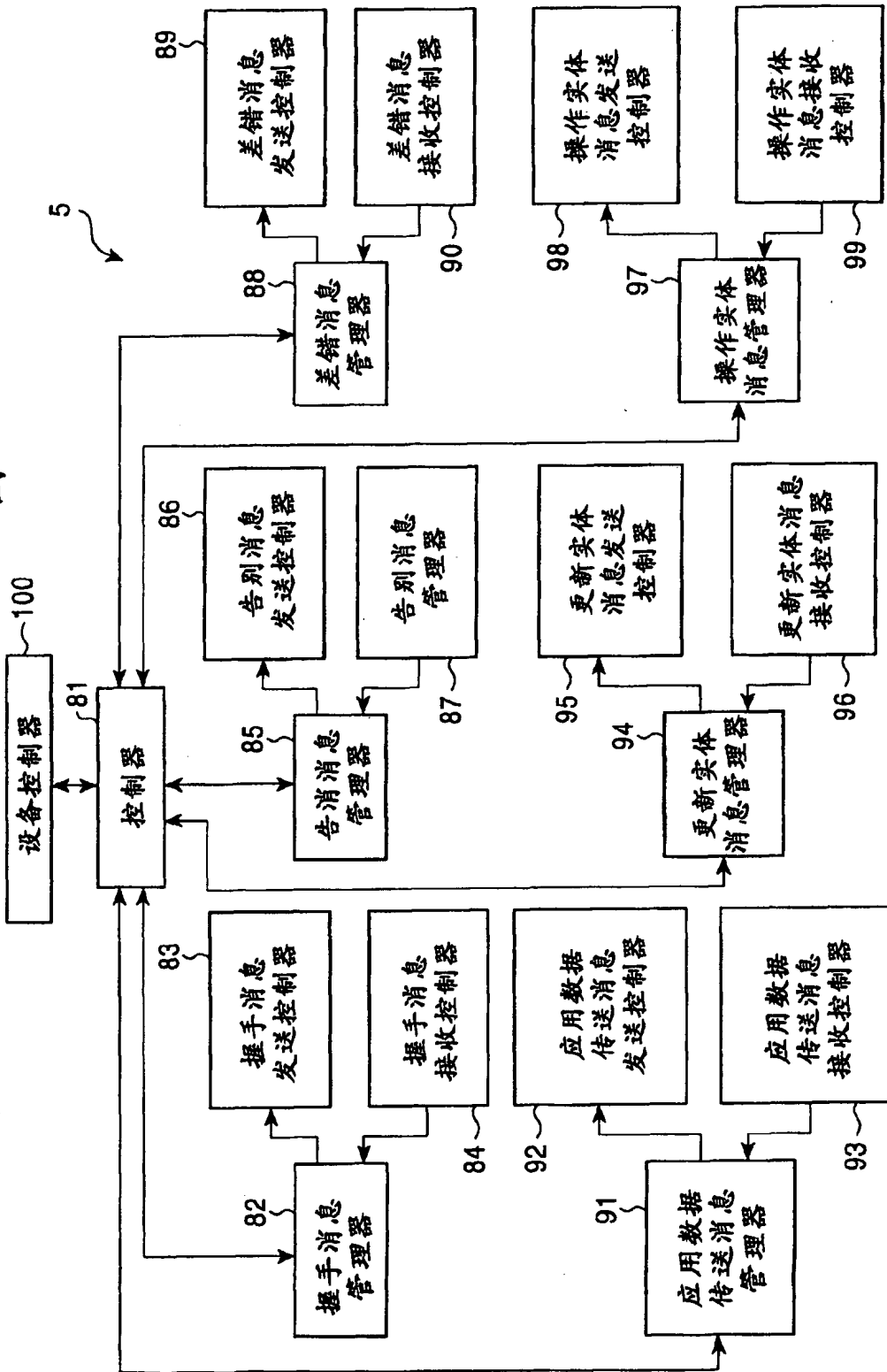


图5

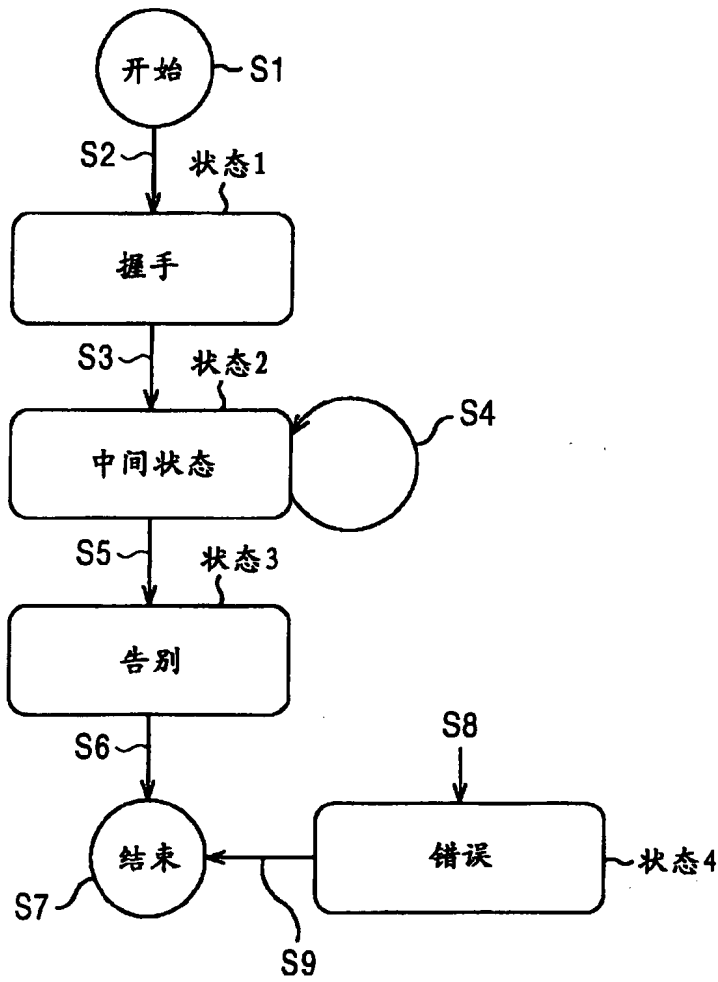


图6

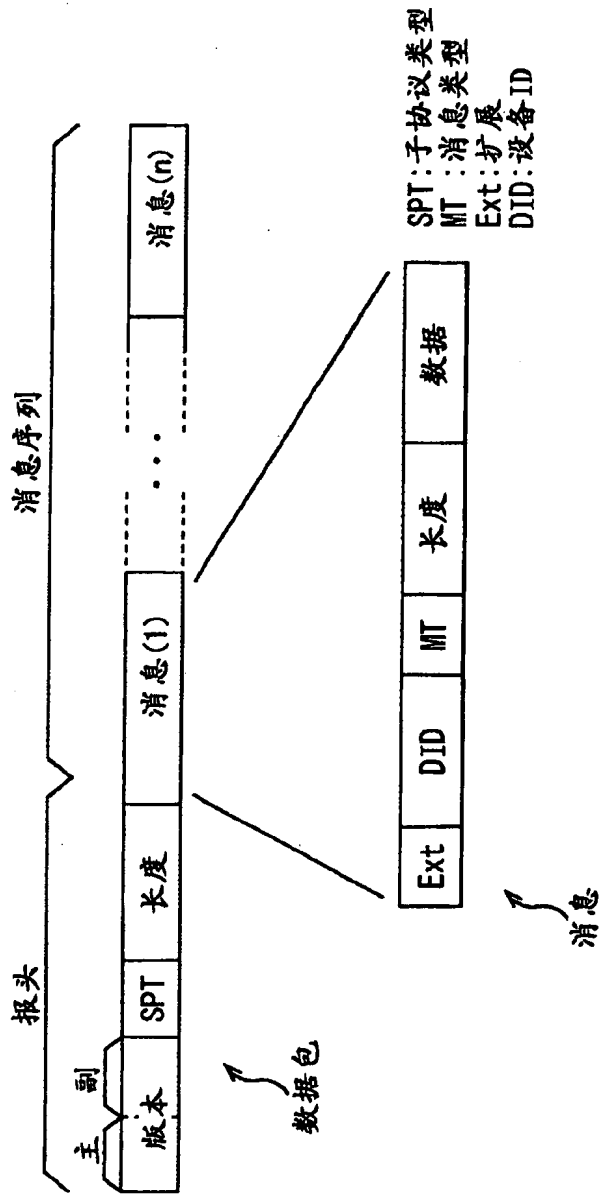


图7

握手协议		
消息名称	描述	数据部分
已完成	子协议结束	无
警告	警告	警告消息: 字符串
客户端问候	向服务器发送握手数据开始	无
客户端问候完成	向服务器发送握手数据结束	无
服务器问候	向客户端发送握手数据开始	无
服务器问候完成	向客户端发送握手数据结束	无
设备	设备清单通知	<ul style="list-style-type: none"> • 设备ID • 许可 • 类型名称长度 • 类型名称 • 设备名称长度 • 设备名称

图 8

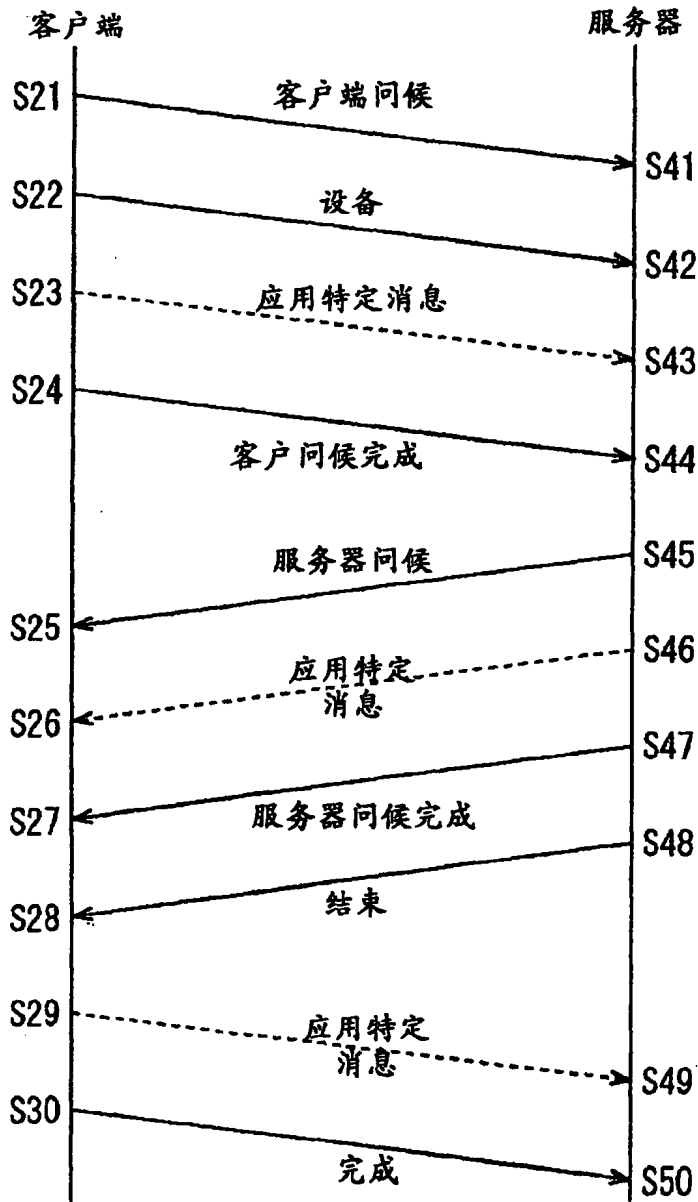


图9

告别协议		无
消息名称	描述	
已完成	子协议结束	警告消息:字符串
警告	警告	无
客户端告别	向服务器发送告别 数据开始	无
客户端告别 完成	向服务器发送告别 数据结束	无
服务器告别	向客户端发送告别 数据开始	无
服务器告别 完成	向客户端发送告别 数据结束	无
返回编码	通知服务器 结束编码	结束编码

图10

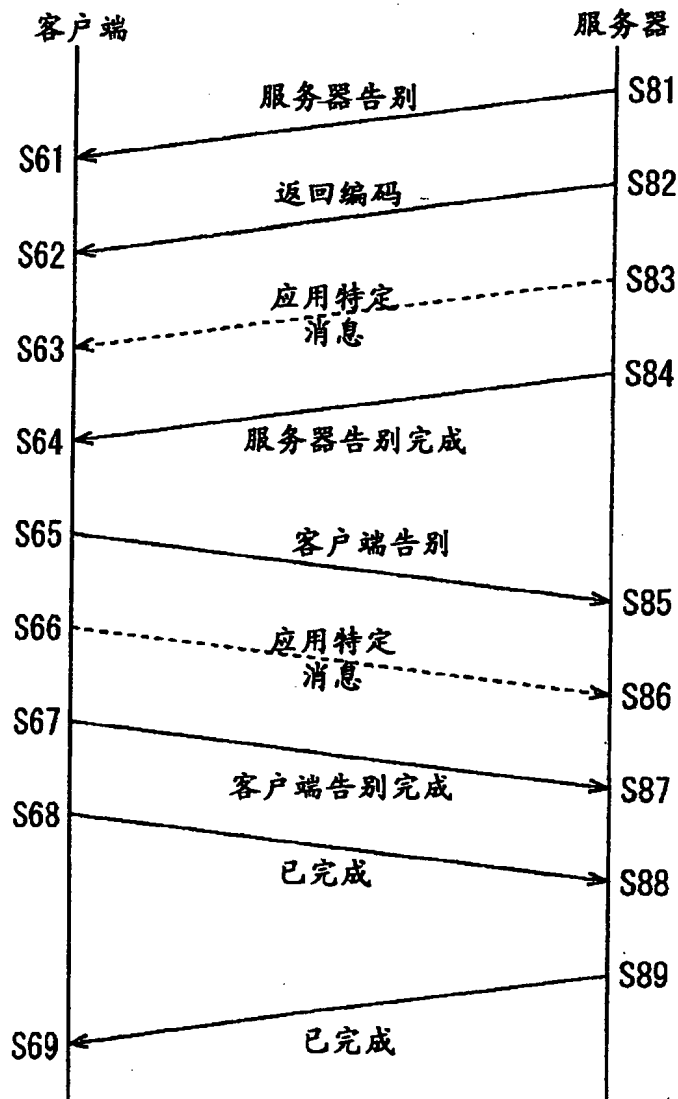


图11

差错协议		
消息名称	描述	数据部分
已完成	子协议结束	无
警告	警告	警告消息: 字符串
数据包格式错误	格式错误	差错消息: 字符串
非法状态错误	检测到的非法状态错误	差错消息: 字符串
意外错误	出现的意外错误	差错消息: 字符串

图12

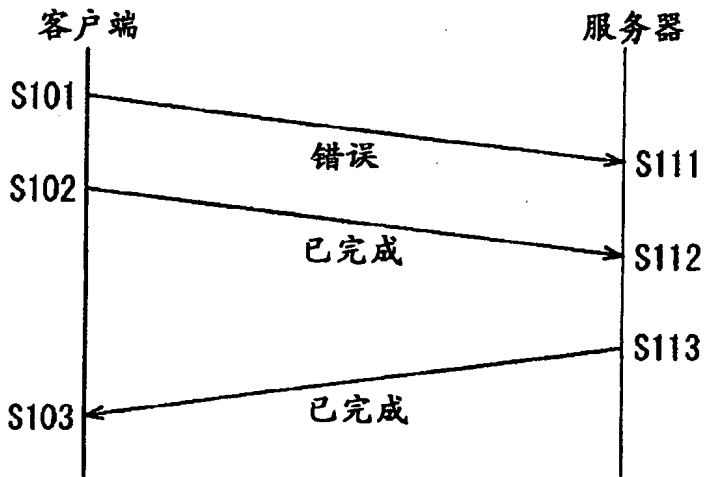


图13

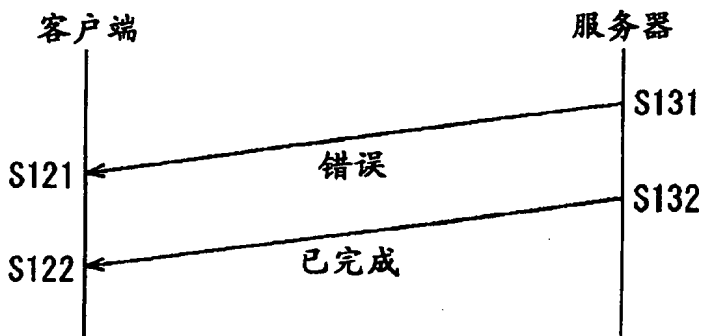


图14

更新实体协议	
消息名称	描述
已完成	数据包结束
警告	警告
设定属性	设定客户端设备的属性 应答: 属性已设定 失败时: 终止
属性已设定	对设定属性的应答
获得属性	获得客户端设备的属性 应答: 属性已设定 失败时: 终止
属性	对于获得属性的应答
设定网络超时	设定客户端的网络超时值 应答: 无 失败时: 继续
	数据部分
	无
	警告消息: 子符串
	<ul style="list-style-type: none"> • 属性名称长度 • 属性名称 • 属性长度 • 属性
	成功标识
	<ul style="list-style-type: none"> • 属性名称长度 • 属性名称
	<ul style="list-style-type: none"> • 属性长度 • 属性
	<ul style="list-style-type: none"> • 超时值 (毫秒)

图15

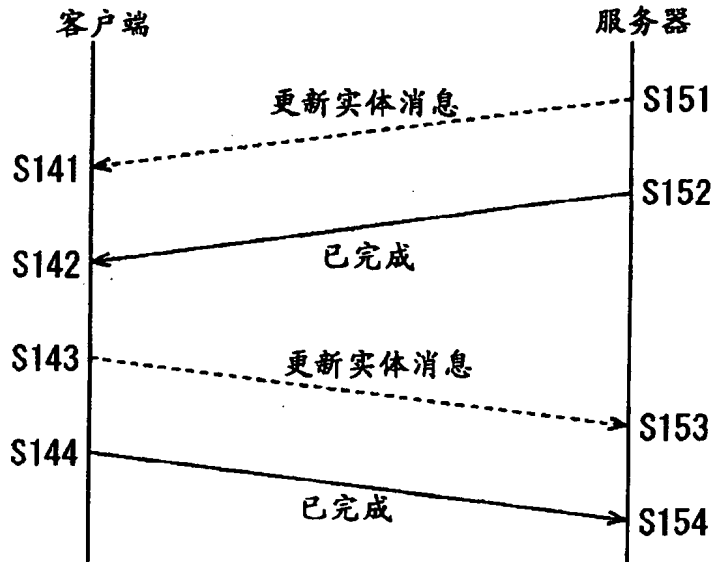


图16

应用数据传送协议	
消息名称	描述
已完成	数据包末尾
警告	警告

图17

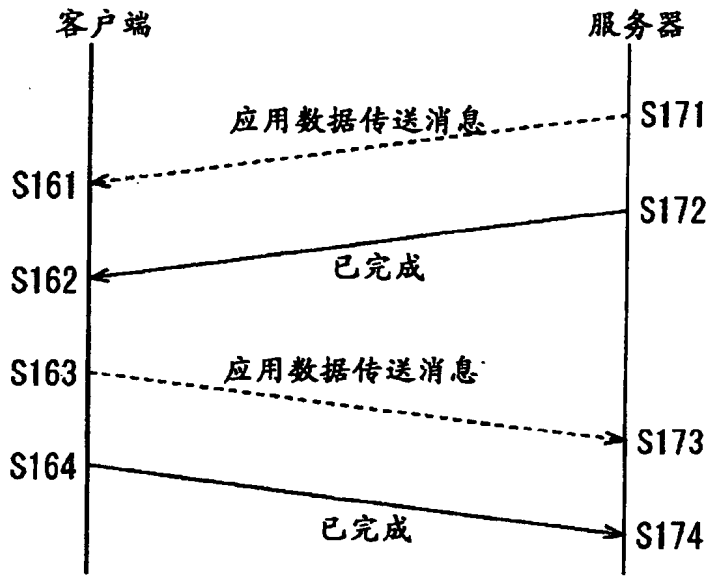


图18

操作实体协议		数据部分
消息名称	描述	
已完成	数据包结束	无
警告	警告	警告消息: 字符串
更新显示	更新显示 应答: 显示已更新 失败时: 终止	无
显示已更新	对更新显示的应答	成功标识
开始数据输入	开始数据输入 应答: 结束数据输入 失败时: 终止	无
结束数据输入	对开始数据输入的应答	<ul style="list-style-type: none"> • 成功标识 • 属性名称长度 • 属性名称 • 属性长度 • 属性
操作设备	操作客户端设备 应答: 设备应答 失败时: 终止	<ul style="list-style-type: none"> • 操作名称的字符 • 数目 • 操作名称 • 参数长度 • 参数
设备应答	对设备操作的应答	<ul style="list-style-type: none"> • 应答长度 • 应答
播放声音	根据客户端指定的 物理名称生成声音 应答: 无 失败时: 继续	声音的物理名称: 字符串

图19

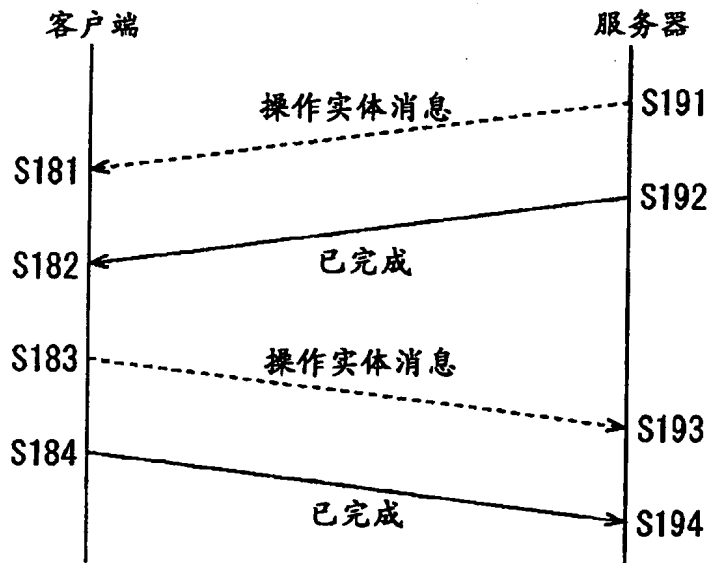


图20

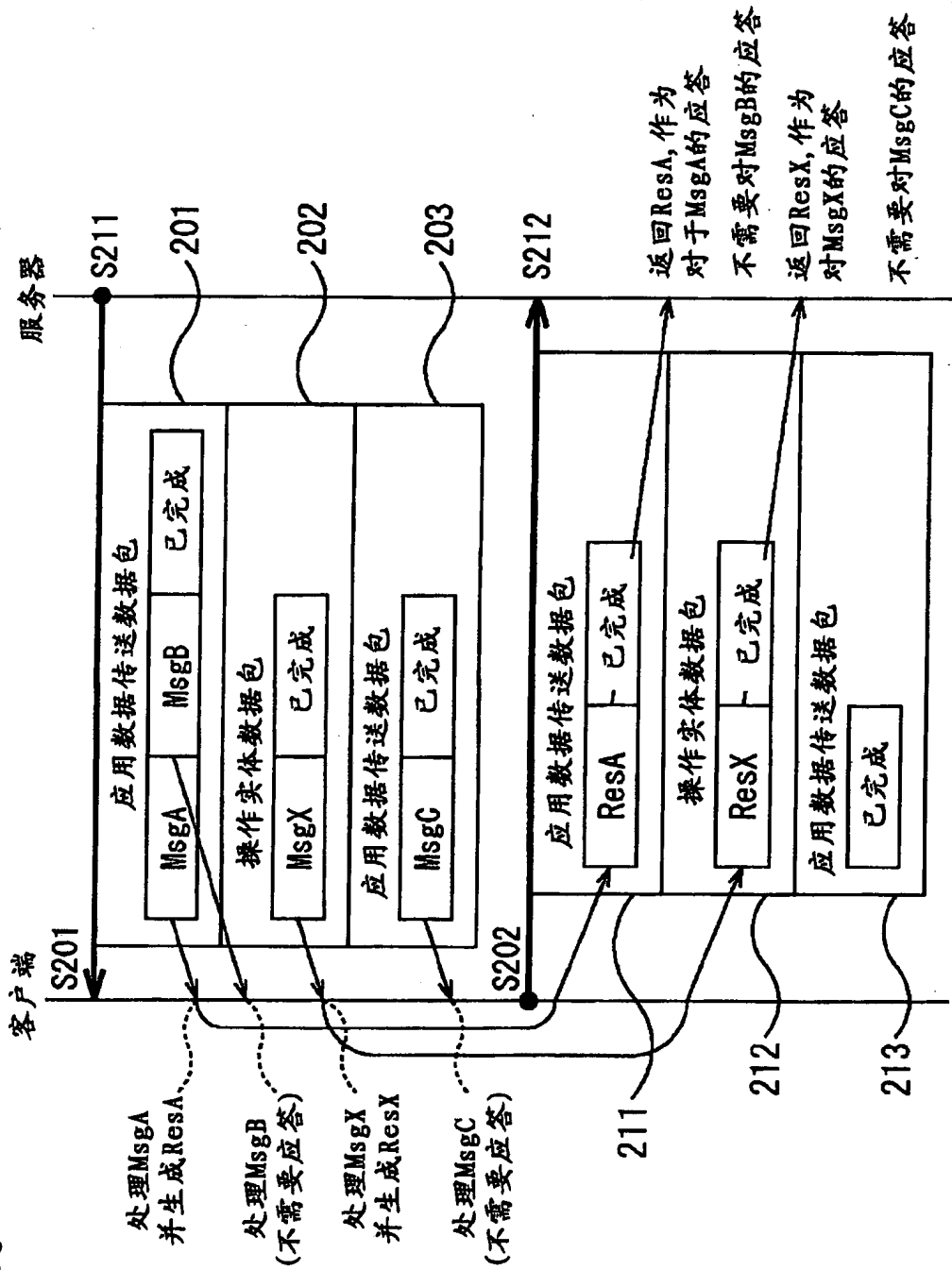
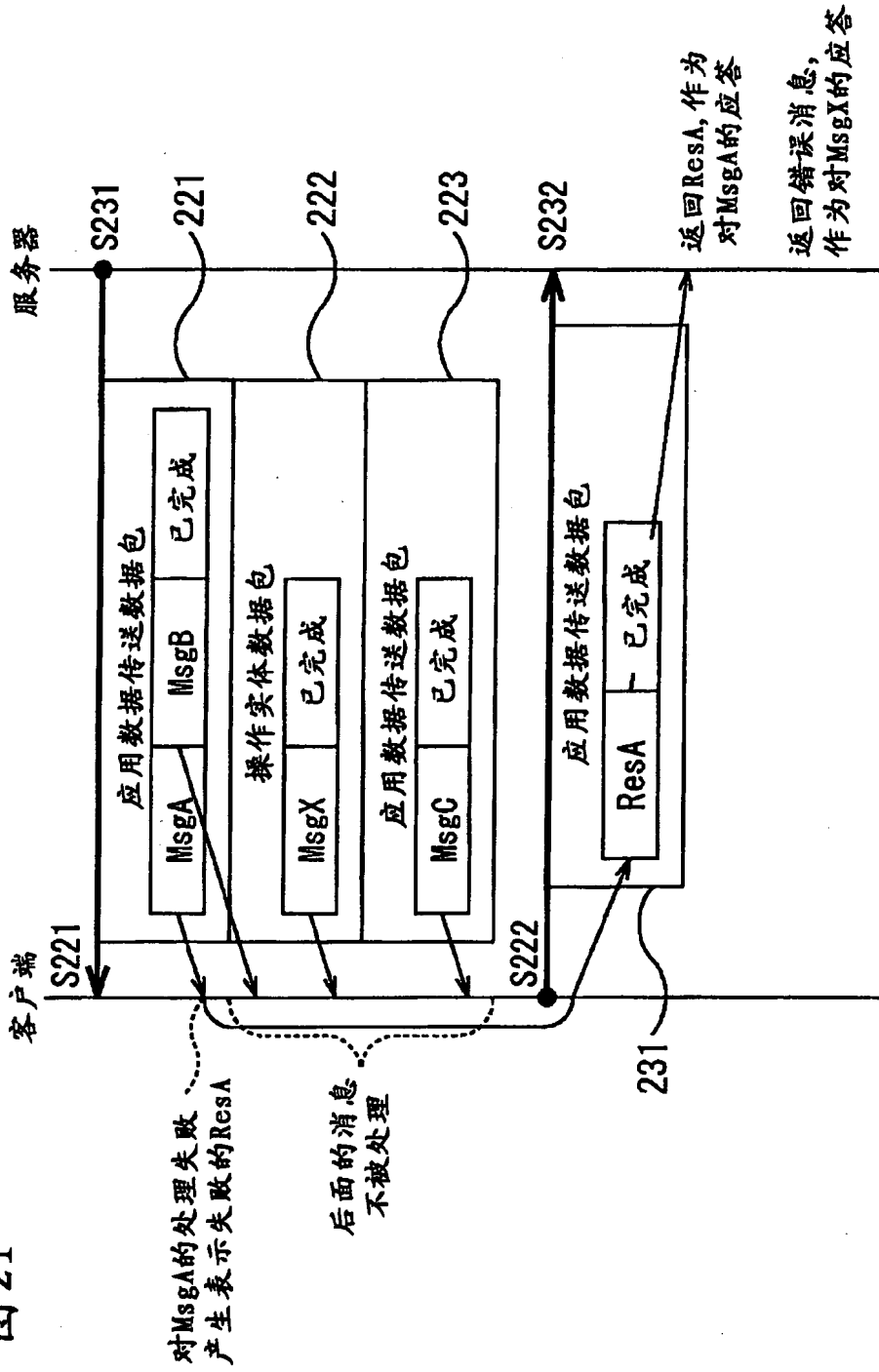


图 21



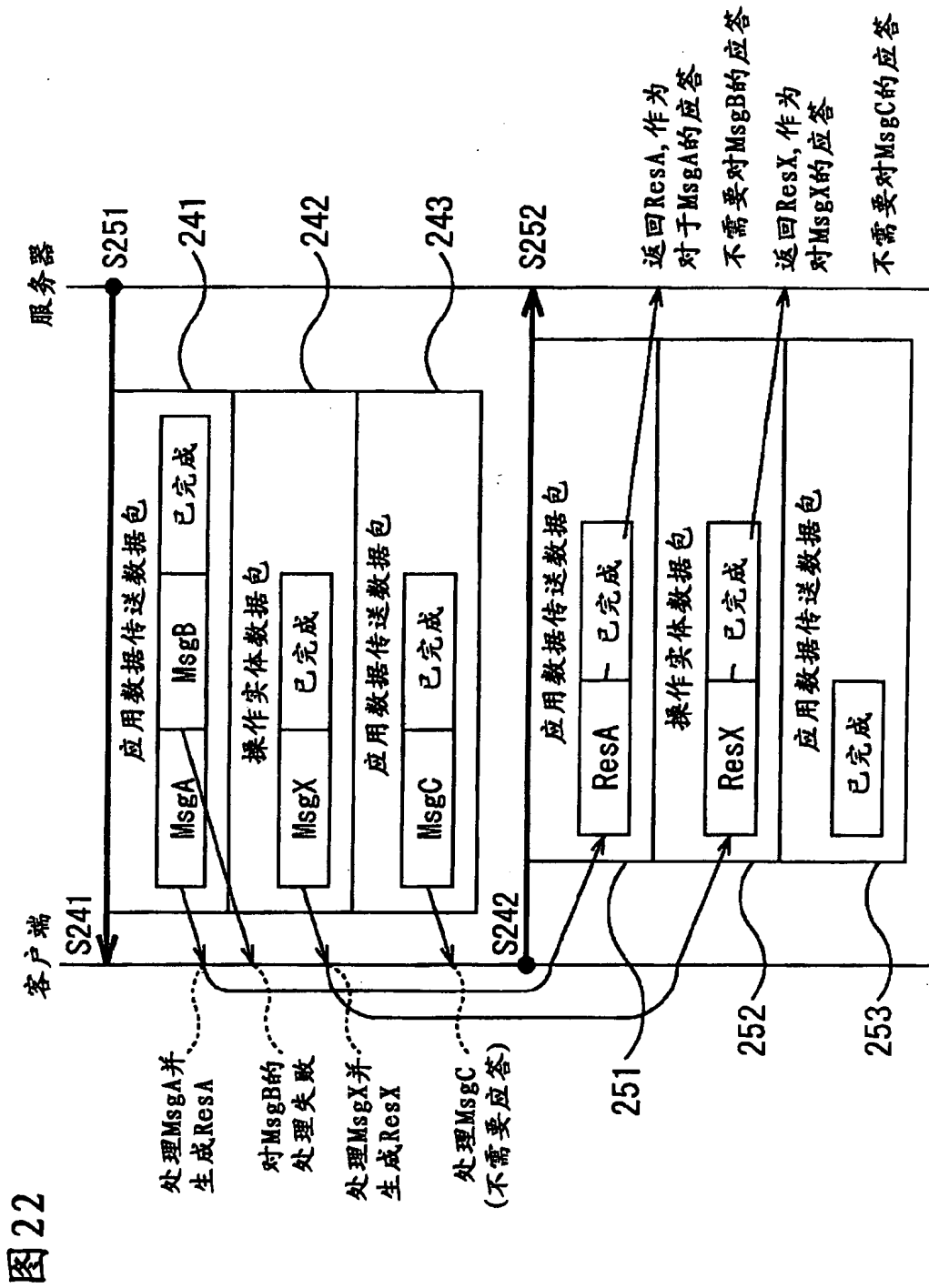


图 23

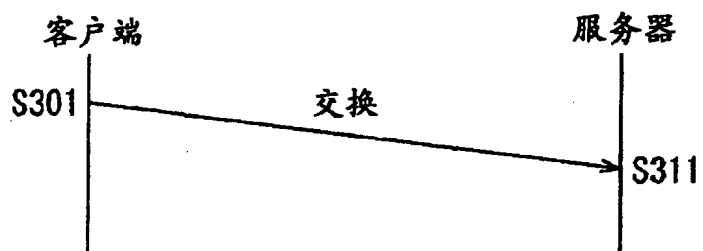


图 24

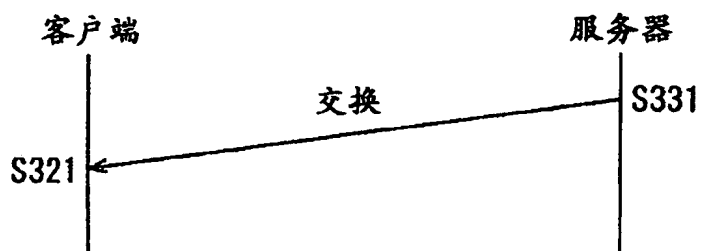


图 25

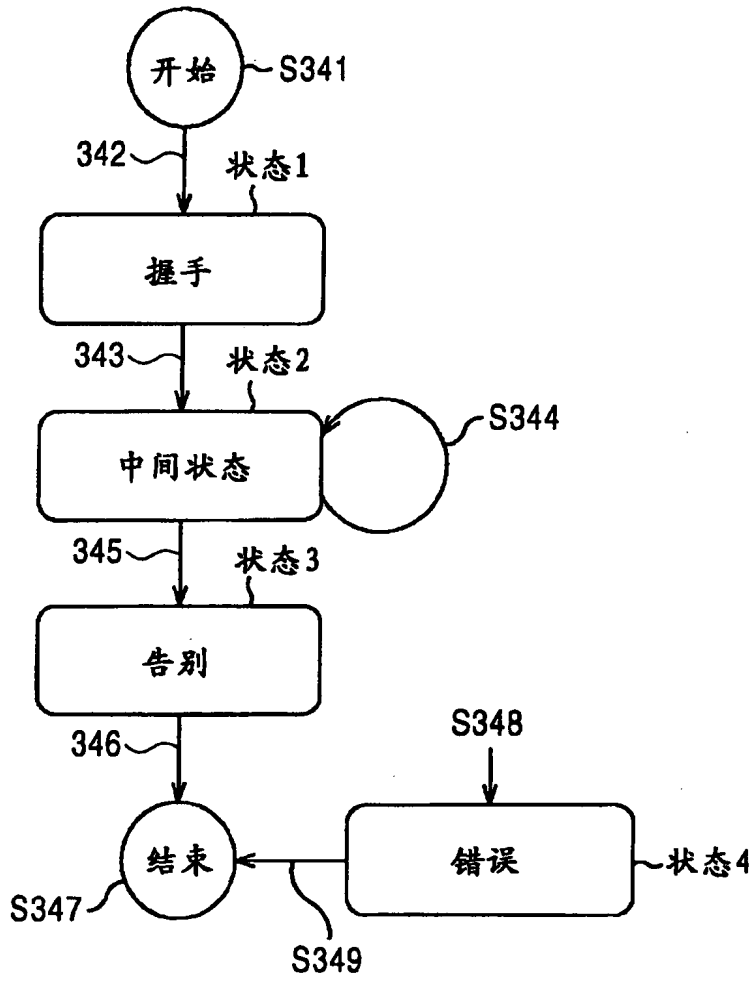


图26

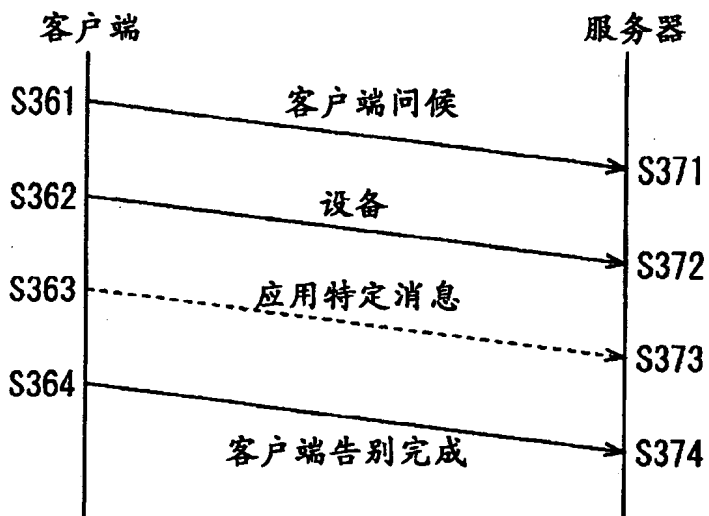
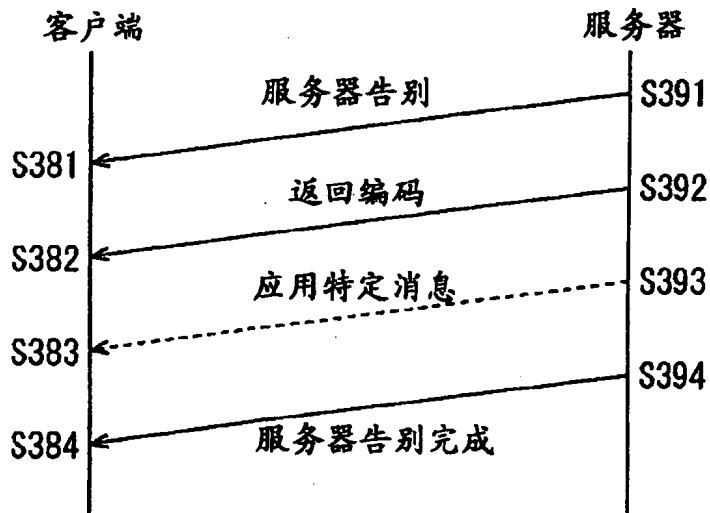


图 27



System and method for controlling the information processing device**Abstract**

A client terminal (such as a PC, portable telephone, PDA, electrical appliances), to which a device (such as a IC card) connected, begins a handshake protocol to request a server to start communication. When communication is established via the handshake protocol, the initiative of communication is transferred to a server, and the state is changed into a neutral state. In the neutral state, a control packet including a particular number of messages and a finished message is transmitted from the server to the client terminal. If the client terminal receives the control packet, the client terminal performs a process according to the messages included in the control packet. This makes it possible to remotely control a device via a network in a highly reliable and efficient manner.