(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 109069924 A
(43)申请公布日 2018.12.21

(21)申请号 201580085424.5
(22)申请日 2015.12.21
(85)PCT国际申请进入国家阶段日
2018.06.20
(86)PCT国际申请的申请数据
PCT/EP2015/002585 2015.12.21
(87)PCT国际申请的公布数据
WO2017/108067 DE 2017.06.29
(71)申请人 格瑞拉伯克斯有限公司
地址 德国汉堡
(72)发明人 F·彼得 S·克哈里尔
R·维斯特曼
(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 11038
代理人 吕晨芳
(51)Int.Cl.
A63F 13/355(2006.01)
A63F 13/352(2006.01)
A63F 13/2145(2006.01)

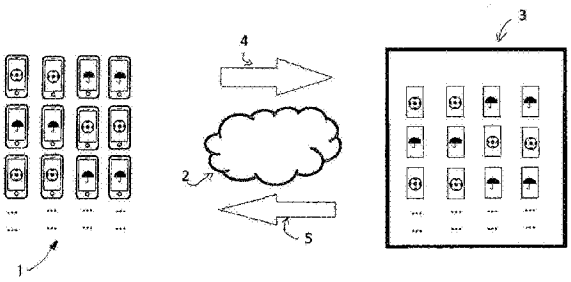
权利要求书10页 说明书15页 附图2页
按照条约第19条修改的权利要求书10页

(54)发明名称

用于播放云端中的应用程序的方法以及用于经由确定的远程通信系统来流处理和再现应用程序(APP)的远程通信网络以及用于流处理和再现应用程序(APP)的远程通信网络的应用

(57)摘要

按照本发明描述如何利用在远程通信终端设备(移动无线电终端设备、平板、手提电脑)中的小计算性能也给不发达的地区提供这样的远程通信终端设备并且因此使其能够参与数字的信息交换。



CN 109069924 A

1.一种用于借助配备有触摸屏的移动无线电终端设备来播放云端中的应用程序的方法,所述移动无线电终端设备具有触摸数字转换器和LCT显示器,其中,计算性能连同图形处理性能被放置在所述云端中并且在所述触摸屏上通过触摸/语音控制在所述触摸屏那里显示的符号来调用应用程序。

2.一种用于提供与平台无关的流技术的方法,所述流技术被一次编程并且能被移植到任意的远程通信终端设备上,在所述方法中,对各个应用程序(Applikation)、例如视频游戏的流处理经由WAN来实现,使得:

a)借助远程通信终端设备(小应用程序)来实施与会话服务器的通信;

b)为确定的终端用户,针对所涉及的应用程序、例如游戏的在地理上与远程通信终端设备最近的流服务器,实施确定的会话;

c)通过所涉及的会话服务器将会话信息通知给远程通信终端设备和流服务器;

d)在远程通信终端设备与所涉及的应用程序、例如视频游戏的流服务器之间实施直接连接;

e)在远程通信终端设备与所涉及的流服务器之间建立直接连接时引入下列步骤:

i.经由所涉及的流服务器记录运行的应用程序、例如游戏的音频/视频数据,所述游戏在所述流服务器上运行;

ii.通过高质量的硬件编码器压缩所述音频/视频数据;

iii.经由WAN传输所压缩的音频/视频数据;

iv.在远程通信终端设备侧接收所述音频/视频数据;

v.解压缩所述音频/视频数据;

vi.在远程通信终端设备(小)上可视化所述音频/视频数据;

vii.在远程通信终端设备(小)上记录远程通信终端设备的用户、例如游戏者的动作(输入);

viii.将所述输入高效地往回传输给游戏的所涉及的流服务器,以及

ix.在流服务器上再现所传输的输入。

3.根据权利要求1或2所述的方法,其特征在于,具有不同操作系统的终端设备、例如具有Android软件的终端设备和/或具有iOS软件的终端设备在一个以及同一个远程通信系统中以应用程序工作,其中,在一个终端设备上已加载的应用程序在变换至以另外的软件、例如Android软件工作的另一个终端设备时在新终端设备上不变地被继续使用,之前在较早终端设备上已加载的应用程序由新终端设备在没有重新获得所述应用程序的情况下经由以另外的软件工作的较早的终端设备的识别方法在新终端设备上不变地被继续使用。

4.根据权利要求3所述的方法,其特征在于,以不同的操作系统工作的终端设备——例如iOS终端设备和/或Android终端设备——在使用认证方法的情况下使用同样的应用商店、例如iOS商店和/或Google-Play商店。

5.一种用于经由确定的用于实施根据权利要求1或跟随该权利要求的各权利要求之一所述的方法的远程通信系统来流处理和再现应用程序(APP)的远程通信网络,在所述远程通信网络中,一个流服务器或多个能通过远程通信相互连接的流服务器实施所涉及的应用程序,并且所述流服务器与相应的远程通信终端设备位置靠近地连接,其中,所涉及的远程通信终端设备从位置靠近的服务器调用所期望的应用程序(Applikation),该位置靠近的

服务器为所涉及的应用程序的呈现和编码提供计算性能。

6.根据权利要求5所述的用于在对应用程序陌生的系统环境上再现应用程序的远程通信网络，所述系统环境要么以不同的硬件组件要么以不同的软件组件区分，其中，流服务器承担对不同应用程序的处理以及对应用程序及其音频和视频信号的呈现和/或编码，其中，将数据传输给相应的远程通信终端设备——移动无线电设备、平板、手提电脑、个人电脑、电视并且借助修改的h.254协议来实施所述传输，使用WAN作为用于音频/视频包的经由UDP/TCP的传输方式并且由所涉及的流服务器承担全部的计算性能，其中，仅在远程通信终端设备中对所打包的数据进行解码。

7.用于实施根据权利要求1或跟随该权利要求的权利要求之一所述的方法的远程通信网络，其具有电话终端设备、用于启动能在触摸屏上显示的流客户端的处理器和另外的通信设备、例如WLAN接收机或无线电接收机以及具有本地的存储模块，所述电话终端设备具有用于输入和再现信号的带有符号的触摸屏，所述处理器负责客户端的呈现，所述另外的通信设备用于云端与终端设备之间的通信，其中，远程通信终端设备经由所谓的接口用于两个或更多个设备之间的通信以及用于播放云端中的应用程序或者用于播放本地存储的存储器模块中的交易数据。

8.根据权利要求5或跟随该权利要求的权利要求之一所述的用于经由确定的远程通信系统来流处理和再现应用程序(APP)的远程通信网络，其特征在于，具有不同操作系统的终端设备、例如具有Android软件的终端设备或具有iOS软件的终端设备在一个以及同一个远程通信系统中以应用程序工作，其中，在一个终端设备上已加载的应用程序在变换至以另外的软件、例如Android软件工作的另一个终端设备时在新终端设备上不变地被继续使用，之前在较早终端设备上已加载的应用程序由新终端设备在没有重新获得所述应用程序的情况下经由以另外的软件工作的较早的终端设备的识别方法在新终端设备上不变地被继续使用。

9.根据权利要求8所述的用于经由确定的远程通信系统来流处理和再现应用程序(APP)的远程通信网络，其特征在于，以不同操作系统工作的终端设备——例如iOS终端设备和/或Android终端设备——在使用认证方法的情况下使用同样的应用商店、例如iOS商店和/或Google-Play商店。

10.根据权利要求5或跟随该权利要求的权利要求之一所述的用于经由确定的远程通信系统来流处理和再现应用程序(APP)的远程通信网络的应用，在所述远程通信网络中，一个流服务器或多个能通过远程通信相互连接的流服务器实施所涉及的应用程序，并且所述流服务器与相应的远程通信终端设备位置靠近地连接，其中，所涉及的远程通信终端设备从位置靠近的服务器调用所期望的应用程序(Applikation)，该位置靠近的服务器为所涉及的应用程序的呈现和编码提供计算性能。

11.用于在对应用程序陌生的系统环境上再现应用程序的远程通信网络的应用，以用于实施根据权利要求1或2所述的方法，所述系统环境要么以不同硬件组件要么以不同的软件组件区分，其中，流服务器承担对不同应用程序的处理以及对应用程序及其各个图像(帧)的音频和视频信号的呈现和/或编码，其中，将数据传输给相应的远程通信终端设备——移动无线电设备、平板、手提电脑、个人电脑、电视并且借助修改的h.254协议来实施所述传输，使用WAN作为用于音频/视频包的经由UDP/TCP的传输方式并且由所涉及的流服

务器承担全部的计算性能，其中，仅在远程通信终端设备中对所打包的数据进行解码。

12.用于提供与平台无关的流技术的远程通信网络的应用，所述流技术被一次编程并且能被移植到任意的远程通信终端设备上，以用于实施根据权利要求1或2所述的方法，在所述方法中经由WAN实现对各个应用程序(Applikationen)、例如视频游戏的流处理，使得：

a)借助远程通信终端设备(小应用程序)来实施与会话服务器的通信；

b)为确定的终端用户，针对所涉及的应用程序、例如游戏的在地理上与远程通信终端设备最近的流服务器，实施确定的会话；

c)通过所涉及的会话服务器将会话信息通知给远程通信终端设备和流服务器；

d)在远程通信终端设备与所涉及的应用程序、例如视频游戏的流服务器之间实施直接连接；

e)在远程通信终端设备与所涉及的流服务器之间建立直接连接时引入下列步骤：

i.经由所涉及的流服务器记录运行的应用程序、例如游戏的音频/视频数据，所述游戏在所述流服务器上运行；

ii.通过高质量的硬件编码器压缩所述音频/视频数据；

iii.经由WAN传输所压缩的音频/视频数据；

iv.在远程通信终端设备侧接收所述音频/视频数据；

v.解压缩所述音频/视频数据；

vi.在远程通信终端设备(小)上可视化所述音频/视频数据

vii.在远程通信终端设备(小)上记录远程通信终端设的用户、例如游戏者的动作(输入)；

viii.将所述输入高效地往回传输给游戏的所涉及的流服务器，以及

ix.在流服务器上再现所传输的用于应用程序的输入。

13.根据权利要求1或跟随该权利要求的权利要求之一所述的应用，具有源代码——用于与客户端(用户、终端设备)通信——如下：

```
/**********AddPortAsynchronisation.java**************************

    Responsible for adding relevant ports to network devices to make
ensure smooth communication, technology targets a technique that
can run independent of network hardware [负责将相关的端口添加至
网络设备（例如路由器）以便这样确保流畅的通信。该技术能够实现
与用户的网络硬件无关的普遍使用]
**********************************************************************/

package org.cloundgaming4u.client.portforwarding;
import java.io.lOException;
```

```
import net.sbbi.upnp.messages.UPNPResponseException;
import android.content.Context;
import android.os.AsyncTask;
import android.util.Log;
public class AddPortAsync extends AsyncTask<Void, Void, Void> {
private Context context;
private UPnPPortMapper uPnPPortMapper;
private String externall P; private String internall P;
private int externalPort;
private int internalPort;
public AddPortAsync(Context context,UPnPPortMapper
uPnPPortMapper,
String
externallP, String internallP, int externalPort, int internalPort) {
this.context = context;
this.uPnPPortMapper = uPnPPortMapper;
this. externallP = externallP;
this.internallP = internallP;
this. externalPort = externalPort;
this.internalPort = internalPort;
}
@Override
protected void onPreExecute() {
super.onPreExecute();
if(uPnPPortMapper == null)
uPnPPortMapper = new UPnPPortMapper();
}
@Override
protected Void dolnBackground(Void... params) {
```

```
if(uPnPPortMapper != null)
{
try
{
Log.d("cg4u_log","Contacting Router for setting network configurations"); if(uPnPPortMapper.openRouterPort(externallP, externalPort,internallP,internalPort, "CG4UGames"))
{
Log.d("cg4u_log",String.format("Setting network configurations successful IP:%s Port:%d",externallP,externalPort));
Log.d("cg4u_log",String.format("Setting network configurations successful
IP:%s Port:%d",internallP,intemalPort));
}
}
catch (lOException e)
{
e.printStackTrace();
}
catch (UPNPResponseException e)
{
e.printStackTrace();
}
}
return null;
}
@Override
protected void onPostExecute(Void result) {
super.onPostExecute(result);
```

```
//Send broadcast for update in the main activity
//Intent i = new Intent(ApplicationConstants.APPLICATION_ENCODING_TEXT);
//context.sendBroadcast(i);
}
}
/**********UniversalPortMapper.java**********
Responsible for making sure that random port generated by server is dynamically mapped at client end [负责服务器的一般端口指派]

**********/
package org.cloundgaming4u.client.portforwarding;
import net.sbbi.upnp.impls.InternetGatewayDevice;
import net.sbbi.upnp.messages.UPNPResponseException;
import java.io.IOException;
public class UPnPPortMapper {
private InternetGatewayDevice[] internetGatewayDevices; private InternetGatewayDevice foundGatewayDevice;
/**
* Search for IGD External Address
* @ return String
*/
public String findExternallPAddress () throws IOException, UPNPResponseException {
/** Upnp devices router
search*/
if(internetGatewayDevices == null)
{
internetGatewayDevices =
```

```
InternetGatewayDevice.getDevices(ApplicationConstants.SCAN_TIM
EOUT);
}
if(internetGatewayDevices != null)
{
for (InternetGatewayDevice IGD : internetGatewayDevices)
{
foundGatewayDevice = IGD;
return IGD.getExternallPAddress().toString();
}
}
return null;
}
/**
* Search Found Internet Gateway Device Friendly Name
* @return
*/
public String findRouterName(){
if(foundGatewayDevice != null){
return
foundGatewayDevice.getlGDRootDevice().getFriendlyName().toString
();
}
return "null";
}
/**
* Open Router Port
* IGD == Internet Gateway Device
*
```

```
* @param internallP
* @param internalPort
* @param externalRouterIP
* @param externalRouterPort
* @param description
* @ return
* @throws lOException
* @throws UPNPResponseException
*/
public boolean openRouterPort(String externalRouterIP,int
externalRouterPort,
String internallP,int internalPort, String description)
throws lOException, UPNPResponseException {
/** Upnp devices router
search*/
if(internetGatewayDevices == null){
internetGatewayDevices =
InternetGatewayDevice.getDevices(ApplicationConstants.SCAN_TIM
EOUT); }
if(internetGatewayDevices != null){
for (InternetGatewayDevice addIGD : internetGatewayDevices) {
/** Open port for TCP protocol and also for UDP protocol
* Both protocols must be open this is a MUST*/
//addlGD.addPortMapping(description, externalRouterIP,
internalPort, externalRouterPort, internallP, 0,
ApplicationConstants.TCP_PROTOCOL);
addlGD.addPortMapping(description, externalRouterIP, internalPort,
externalRouterPort, internallP, 0,
ApplicationConstants.UDP_PROTOCOL);
```

```
}
return true;
}else{
Return false;
}
}
public boolean removePort(String extemallP,int port) throws
lOException,
UPNPResponseException{
/** Upnp devices router
search*/
if(internetGatewayDevices == null){
internetGatewayDevices = lnternetGatewayDevice.getDevices(5000);
}
/**Remote port mapping for all routers*/
if(internetGatewayDevices != null){
for (InternetGatewayDevice removelGD : internetGatewayDevices) {
//          removelGD.deletePortMapping(externallP,          port,
ApplicationConstants.TCP_PROTOCOL);
removelGD.deletePortMapping(externallP, port, "UDP");
}
return true;
}else{
return false;
}
}
}
```

****

End of ClientNetworkCommunication

客户端网络通信结束

****

14.远程通信网络的根据权利要求10或跟随该权利要求的权利要求之一所述的应用，用于经由确定的远程通信系统来流处理和再现应用程序(APP)，其特征在于，具有不同操作系统的终端设备、例如具有Android软件的终端设备和/或具有iOS软件的终端设备在一个以及同一个远程通信系统中以应用程序工作，其中，在一个终端设备上已加载的应用程序在变换至以另外的软件、例如Android软件工作的另一个终端设备时在新终端设备上不变地被继续使用，之前在较早终端设备上已加载的应用程序由新终端设备在没有重新获得所述应用程序的情况下经由以另外的软件工作的较早的终端设备的识别方法在新终端设备上不变地被继续使用。

15.用于流处理和再现应用程序(APP)的远程通信网络的根据权利要求14所述的应用，其特征在于，以不同操作系统工作的终端设备——例如iOS终端设备和/或Android终端设备——在使用认证方法的情况下使用同样的应用商店、例如iOS商店和/或Google-Play商店。

用于播放云端中的应用程序的方法以及用于经由确定的远程通信系统来流处理和再现应用程序(APP)的远程通信网络以及用于流处理和再现应用程序(APP)的远程通信网络的应用

技术领域

[0001] 本发明涉及一种用于播放云端(Cloud)中的应用程序的方法。

[0002] 此外，本发明涉及一种用于经由确定的远程通信系统来流处理和再现应用程序(APP)的远程通信网络。

[0003] 此外，本发明涉及用于经由确定的远程通信系统来流处理和再现应用程序(APP)的远程通信网络的应用。

背景技术

[0004] 数据量的数字化在过去数年中已经呈爆炸式增加并且在接下来的数年中继续简直雪崩式上升。人与人之间的、更确切地说不仅在私人领域中而且尤其是在职业领域中的数字通信也扮演着越来越重要的角色。用于此的工具是所谓的智能手机，更确切地说尤其是在亚洲、在美国以及在欧洲。在这些国家地区中，大部分人都拥有这样的终端设备。

[0005] 如果考虑到这些区域的经济强度及其国民生产总值，则这点不再令人感到惊奇，因为人们亦即行业公司具有足够的财政资金来购置这样的终端设备。然而，在例如非洲地区中或在所谓的新工业化国家地区和我们称为第三世界的国家地区(在那里由于收入少而因此不可能实现智能手机的流行)中情况不同。这不仅妨碍人与人之间的通信，而且也总体上妨碍知识的传播，尤其是也在商业领域中。

[0006] 如今越来越重要的是，在本地开发应用程序。然而，本地的开发总是单独与确定的平台适配。然而问题在于，总是有更新的且更现代的平台出现在市场上并且用户不仅使用一个平台、而是使用多个不同的平台。

[0007] 另一个问题是处于背后的硬件。特定的应用程序也基于特定的硬件。该硬件需要满足对应用程序的确定要求，例如图形负载、处理器容量、存储器、能量消耗。但是反过来地，应用程序也可能要求比平台的硬件所能够提供的更多的计算性能或图形处理性能。这特别是在使用大量图形的应用程序、例如游戏中可能导致用户由于系统不兼容而不能使用所述应用程序。基本上存在三种不同的方式用于将应用程序传输到平台未知的环境上。

[0008] 首先是所谓的本地开发(移植)。在未知平台的角度下重新开发应用程序。这是所有三种方法中最复杂且最耗时的途径，但该途径提供了利用该新平台的所有功能的可能性。然而，该方法的一个问题是，应用程序受到平台的框架条件限制。因此，例如图形要求高的游戏不能被移植到移动平台上。在未知平台之内的不同的硬件前提条件也构成问题，因为例如不是每个用户都具有相同的移动无线电设备。

[0009] 对此补充地已经存在如下软件，所述软件应该使得开发者能够更轻易地实现本地开发。在使用确定的软件的情况下如下地进行移植，即：取代已存在的软件的一部分，以便这样实现对未知系统的兼容性。该步骤不总是可能的，因为一些平台在架构设计上彼此过于不同。在这样的情况下也大部分缺少平台运营商的支持，出于该原因大多采用本地开发。

[0010] 网页应用程序(Web-Apps)是如下的应用程序,这些应用程序基于网页浏览器被开发并且由此能在几乎所有平台上被使用。为此也常常使用WCM系统(Webcontent Management)。然而,仅能经由必须由平台提供的相应浏览器来实现这些应用程序。在该方法中不利的是,不是所有应用程序都能够利用该方法被移植。必须使用如下浏览器,该浏览器不总是确保应用程序的本地实现。

[0011] 流处理(Steaming):这意味着应用程序在服务器上运行并且借助客户端在未知平台上仅播放所述应用程序。然而,该技术当前被限制于确定的时间不关键(关键字在此为"延迟")的应用程序。

[0012] 通过WO 2012/037170 A1已知的是,与流处理并行地将应用程序代码传输到客户端,以便一旦应用程序能在客户端上运行则能够结束所述流处理,从而应用程序直接在客户端上运行,以便这样能够节省流处理资源。这对于例如控制台可以是有利的,但在硬件特定的前提条件(约束)中是不可能的。

[0013] 在WO 2009/073830中说明一种系统,该系统基于"订阅费用"给用户提供对服务器的访问。在该情况下,确定的流服务器(Streaming-Server)在预定的时间段内被指派给客户。然而,我们的系统将地理上最优的流服务器指派给用户,而不需要"订阅费用"。

[0014] 补充于此地,WO 2010/141522 A1使用一种游戏服务器,经由该游戏服务器部分地进行客户端与流服务器之间的流通信。此外,交互层(Interactive layer)的功能经由视频源来反映,该视频源在该开发中经由单独的服务器来处理,以便也提供对例如广告区域的第三方访问。

发明内容

[0015] 本发明基于如下任务:尤其是、但不仅仅为具有较少收入的国家地区的居民提供一种方法,他们能够利用该方法普遍参与数字化数据量的交换。

[0016] 此外,本发明基于如下任务:使这样的结构弱的国家地区装备有如下的远程通信网络,该远程通信网络即使在人均收入小时也使得数字化数据的交换变得容易。

[0017] 最后,本发明基于如下任务:使远程通信网络的应用变得容易,收入差的人口阶层也能够借其参与数字化数据的交换。

[0018] 涉及方法的解决方案

[0019] 该任务通过权利要求1的特征来解决。

[0020] 一些优点

[0021] 按照本发明的方法提供成本有利的变型方案,因为收入差的人口由此借助非常简单的设备也能够基本上不受限制地参与数字化的数据交换。他们为此按照本发明仅需要非常简单的并且因此便宜的以具有触摸屏的移动无线电设备形式的终端设备,其中,移动无线电终端设备仅必须包含小的呈现客户端的处理器以及可能也包含局域网(WLAN)或无线(Wifi)接收机/发射机用于传输信号。计算机仅必须能够承担最小的基本功能、例如经由触摸屏处理用户输入以及播放和呈现视频流。处理器的规格仅需要非常小。在此,"呈现"理解为在终端设备上再现在云端中产生的图像的过程。因此,这意味着计算性能和呈现仅在云端中实现。仅仅为具有移动无线电终端设备的终端用户提供设有简单的菜单引导、例如指示图形的触摸屏,终端用户必须用手指敲击所述触摸屏以便从云端调用相应的信息。

[0022] 亦即总之，在客户侧的硬件成本是非常低的。由此，即使在收入差的国家地区也能够出售所述设备。而这基于流技术提供类似于智能手机的功能。由于计算性能或图形处理性能被放置在云端，所以也不产生与硬件的兼容问题并且不必每两年购买一个移动无线电终端设备。省去了完整的重新购买周期。因此，所述设备为了实施所述方法而仅包括四个构件：a)触摸屏，其负责输入和再现信号并且具有触摸数字转换器和LCT显示器，

[0023] b)处理器，其需要用于启动流客户端。该流客户端在触摸屏上显示。所述处理器负责呈现客户端和数据。

[0024] c)第三单元负责网络与设备之间的通信。在此可以涉及WLAN接收机、Wifi接收机或无线电接收机。

[0025] d)此外需要小的本地存储器模块。该模块可以针对不同的功能被使用。因此，例如个人数据可以被存储或该区域可以被声明为私人存储器，其方式为存储确定的信息。因此可想到的是，从所述设备中实现虚拟钱包。

[0026] 因此，通过按照本发明的方法能够经由上面提及的接口实现两个设备之间的远程通信。

[0027] 此外存在交互。也就是说，所述设备能够实现播放云端中的应用程序并且也可想到虚拟钱包，在该虚拟钱包中，本地存储在存储器模块中的交易数据类似于比特币系统(BitCon-System)那样用于在各个设备之间交换信息。

[0028] 进一步的创造性的设计方案

[0029] 权利要求2说明一种用于提供与平台无关的流技术的方法作为进一步的创造性的设计方案，所述流技术被一次编程并且能被移植到任意的远程通信终端设备上，在所述方法中，各个应用程序(Applikation)(例如视频游戏或实际应用程序)的流处理经由WAN(广域网)来实现，使得：

[0030] a)借助远程通信终端设备(小应用程序)来实施与会话服务器的通信；

[0031] b)为确定的终端用户，针对所涉及的应用程序(例如游戏)的在地理上与远程通信终端设备最近的流服务器，实施确定的会话；

[0032] c)通过所涉及的会话服务器将会话信息通知给远程通信终端设备和流服务器；

[0033] d)在远程通信终端设备与所涉及的应用程序(例如视频游戏)的流服务器之间实施直接连接；

[0034] e)在远程通信终端设备与所涉及的流服务器之间建立直接连接时引入下列步骤：

[0035] i.经由所涉及的流服务器记录运行中的应用程序(例如游戏)的音频/视频数据，所述游戏在所述流服务器上运行。

[0036] ii.通过高质量的硬件编码器压缩所述音频/视频数据；

[0037] iii.经由WAN传输所压缩的音频/视频数据；

[0038] iv.在远程通信终端设备侧接收所述音频/视频数据；

[0039] v.解压缩所述音频/视频数据；

[0040] vi.在远程通信终端设备(小)上可视化所述音频/视频数据；

[0041] vii.在远程通信终端设备(小)上记录远程通信终端设的用户(例如游戏者)的动作(输入)；

[0042] viii.将所述输入高效地往回传输给游戏的所涉及的流服务器以及

[0043] ix.在所述流服务器上再现所传输的输入。

[0044] 该方法能够实现:在软件未知的环境上播放非本地编程的应用程序,更确切地说无需满足未知平台的硬件特定的前提条件、例如在计算性能和图形处理性能方面的前提条件,并且无需满足未知平台的软件特定的前提条件、例如仅在确定的操作系统上运行的应用程序。与例如US 2014/0073428 A1相比,本发明使用专门为所述应用程序构建的客户端。该客户端能够在每个任意的平台上被使用,以便确保几乎无延迟地再现h.254压缩的流。为了传递帧而使用h.254代码。H.264/MPEG-4 AVC是用于高效率视频压缩的H.标准。在2003年通过了该标准。在此,H.264是ITU(国际电信联盟)名称。在ISO/IEC MPEG中,所述标准的名称为MPEG-4/AVC(高阶视讯编码)并且是MPEG-4标准的第十部分(MPEG-4/第十部分,ISO/IEC 14496-10)。此外,在按照本发明的方法中使用资源处理,所述资源处理将负载分布到各个流处理器上,以便一方面节省资源、亦或另一方面节省容量/投入。这能够实现使得所述系统比例如在WO 2012/37170 A1中的类似系统更节省成本地工作。这也提供了如下可能性:在正在进行的运行中切断流服务器,以便例如实施维修工作。一般已知的是,在几乎所有情况下、例如在WO 2010/141522 A1中,总是必须在应用程序的代码中启动所谓的钩子(Hook),以便能够实现使流服务器对应用程序进行流处理。这导致必须改变应用程序代码,这一方面可能导致额外耗费、而另一方面可能导致与应用程序的初始开发者的显著问题。按照本发明的方法使得钩子变得多余并且能够实现自动化所述方法。

[0045] 客户端应用程序(Client-Application)原则上包括三个部分(解码线程(decode thread)、呈现线程(render thread)和交互层)并且被保留在clientnetwork.so(共享库(shared library))中。这些部分被划分到各个模块中。

[0046] 客户端会话(Client-Session)管理模块负责管理(开始/结束)会话并且用于管理由用户开始的会话。经由该模块也能够进行在延迟优化方面的调节。

[0047] 网络模块(Network Module)承担网络通信并且管理与流服务器的通信。

[0048] 控制器模块(Controller Module)捕捉应用程序的用户输入并且将该用户输入传送给游戏流服务器。

[0049] 解码呈现(Decoder-Render)音频模块包括两个部分:解码模块承担对h.264流的解码。视频播放器发出声音。

[0050] 评价器模块(Evaluator Modul)将报告传送给流服务器。

[0051] 复原模块(Recovery Modul)承担对用于问题帧的策略的处理。

[0052] 客户端UI模块(Client UI Modul)被结合到交互层中并且负责应用程序的UI。

[0053] 交互层能够实现在处于其下的渲染线程上可视化信息的附加视觉显示,以便例如显示共有特性/支持或广告。它处于呈现线程之上并且能够特定地由用户适配。

[0054] 对于交互层,为每个平台提供一个预定义的用户界面。然而,用户可以经由所谓的层脚本(Scripting)在确定的框架条件下自行创建相应的用户界面。层脚本为用户提供专门开发的脚本环境,该脚本环境能够实现将确定的功能绑定在预定义的按钮上。因此,用户能够将其UI特定地与其需要适配。

[0055] 流服务器基本上包括三个模块(网络线程、GPU线程和会话处理程序(session handler))并且被保留在servernetwork.dll(共享库)中。分别给每个在流服务器上运行的应用程序指派一个GPU和一个网络线程。该自动的过程由会话处理程序来管理。

[0056] 网络线程负责提供被编码的音频和视频文件。

[0057] GPU线程负责对应用程序的音频和视频帧进行硬件解码、承担经由UDP/TCP的报文缓冲(Buffering)并且承担时间戳(timingstamping)和压缩。

[0058] 会话处理程序负责开始/结束和管理GPU和网络线程。会话处理程序协调可用资源与游戏流服务器并且与会话管理服务器通信。在会话处理程序背后的构思是自动管理资源以便能够节省成本。

[0059] 会话管理服务器包括四个模块:认证模块;网络模块;会话管理模块;评价器模块。

[0060] 接入服务器承担对客户端的认证,以便一方面保存用于流服务器的客户端规格、以便检查客户端是否有权利调用所要求的应用程序。所述认证也可以利用第三方系统相对地起作用,从而也能够连接未知系统。

[0061] 网络模块负责负载平衡(Loadbalancing)、质量安全和管理。负载平衡理解为负载在网络内均匀分布。在质量安全的范围内监控每个单独的流并且与性能相关地对其进行优化(例如通过确定的路由选择)。所述管理应该使管理员能够实现查阅实际负载和路由选择,以便进行确定的配置。

[0062] 会话管理模块负责游戏流服务器的负载优化和监管。该单元将收到的客户端询问与游戏流服务器上的空闲空间相关联并且然后在客户端与流服务器之间建立直接连接。用于关联的决定性准则为:在流服务器与应用客户端之间的延时和可用的资源。目标是,利用该单元创建节省资源的方法,以便能够切断未被利用的性能。

[0063] 评价器模块。该评价器模块承担统计数字的生成和承担管理并且用于进一步的优化。

[0064] 内容服务器(Content Server)承担在相应的客户端至适配的游戏的交互层上播放广告。广告可以以多种形式显示。要么在应用程序内进行永久的放置,要么预定义确定的时间点,一旦触发所述时间点,则所述时间点设置相应的触发器以便播放广告。

[0065] UDP(用户数据报协议)对于实时数据传输是简单的、更少耗费的并且更高效的。然而,伴随UDP的问题在于:不存在用于处理在网络中丢失的数据包的机制。因此,当游戏在云端中运行时,出现屏幕缺失、抖动和闪烁。

[0066] 我们已经确定了四种智能地修正包丢失状况的策略。

[0067] 锁定(Blockieren):在用户侧的策略,当进行故障排除时在该策略中显示静态图像。这对于用户来说能够实现与屏幕缺失、抖动和闪烁相比改善的用户体验。因此,该方法确保图像在包缺失的情况下是没有缺陷的。

[0068] 不锁定(Nicht-Blockieren):在用户侧的策略,当在服务器中要求传输所丢失的包时,在该策略中不产生静态图像。该重新传输不能与TCP传输相比,因为该重新传输处于我们自己的监管之下并且我们仅在需要该重新传输时才高效地要求该重新传输。

[0069] 内部更新:该策略在用户侧实现,其在运行时间内与(服务器侧的)视频编码器对话。在丢失包的情况下,该策略请求编码器进行图像更新。因此,一旦图像由于丢失图像包而被中断,则在数毫秒内将图像更新应用到该图像上,肉眼都不曾觉察到所述图像更新。

[0070] 图像检查:该策略考虑帧速率,从服务器侧以该帧速率发送图像。该策略在帧速率波动的情况下确保图像包以恒定的帧速率被发送。这在此有助于保证均匀的图像体验。

[0071] 特别有利的方法通过权利要求3表示。这特征在于:具有不同操作系统的终端设

备、例如具有Android软件的终端设备和/或具有iOS软件的终端设备在一个以及同一个远程通信系统中以应用程序工作，其中，在一个终端设备上已加载的应用程序在变换至以另外的软件、例如Android软件工作的另一个终端设备时在新终端设备上不变地被继续使用，之前在较早终端设备上已加载的应用程序由新终端设备在没有重新获得所述应用程序的情况下经由以另外的软件工作的较早的终端设备的识别方法在新终端设备上不变地被继续使用。迄今为止，终端设备用户、例如移动无线电设备用户在变换操作系统、例如从iOS操作系统变换为Android操作系统时必须重新购买应用程序。现在，通过本发明说明一种如何能够避免这点的途径。此后，在变换操作系统时不再需要重新购买先前所购买的应用程序。更确切地说，按照本发明的方法能够实现在新的操作系统上继续使用先前所购买的并且已使用的应用程序。此外，根据不同的操作系统(例如iOS操作系统和/或Android操作系统)工作的终端设备用户能够在任意商店中购买。为此，权利要求4也说明另一种有利的方法。

[0072] 涉及远程通信网络的任务的解决方案

[0073] 该任务按照权利要求5通过一种用于经由确定的远程通信系统来流处理和再现应用程序(APP)的远程通信网络来解决，在所述远程通信网络中，一个流服务器或多个能通过远程通信相互连接的流服务器实施所涉及的应用程序，并且所述流服务器与相应的远程通信终端设备位置靠近地连接，其中，所涉及的远程通信终端设备从位置靠近的服务器调用所期望的应用程序(Applikation)，该位置靠近的服务器为所涉及的应用程序的呈现和编码提供计算性能。

[0074] 按照权利要求6所述的远程通信网络用于在对应用程序陌生的系统环境上再现应用程序，所述系统环境要么以硬件组件要么以软件组件区分，其中，流服务器承担对不同应用程序的处理以及对应用程序及其音频和视频信号的呈现/编码，其中，将数据传输给相应的远程通信终端设备——移动无线电设备、平板、手提电脑、个人电脑、电视并且借助修改的h.254协议来实施所述传输，使用WAN作为用于音频/视频包的经由UDP/TCP的传输方式并且由所涉及的流服务器承担全部的计算性能，其中，仅在远程通信终端设备中对所打包的数据进行解码。

[0075] 按照权利要求7所述的远程通信网络设有电话终端设备、用于启动能在触摸屏上显示的流客户端的处理器和另外的通信设备、例如WLAN接收机或无线电接收机以及具有本地的存储模块，所述电话终端设备具有带有符号的触摸屏，所述触摸屏用于输入和再现信号，所述处理器负责客户端的呈现，所述另外的通信设备用于云端与终端设备之间的通信，其中，远程通信终端设备经由所谓的接口用于两个或更多个设备之间的通信以及用于播放云端中的应用程序或者用于播放本地存储的存储器模块中的交易数据。

[0076] 在权利要求8中要求保护一种优选的远程通信网络。这能够实现在不同的操作系统之间变换时继续使用先前所获得的应用程序。该远程通信网络的特征在于，具有不同操作系统的终端设备、例如具有Android软件的终端设备和/或具有iOS软件的终端设备在一个以及同一个远程通信系统中以应用程序工作，其中，在一个终端设备上已加载的应用程序在变换至以另外的软件、例如Android软件工作的另一个终端设备时在新终端设备上不变地被继续使用，之前在较早终端设备上已加载的应用程序由新终端设备在没有重新获得所述应用程序的情况下经由以另外的软件工作的较早的终端设备的识别方法在新终端设备上不变地被继续使用。

[0077] 另一种有利的实施方式在权利要求9中说明，该实施方式的特征在于，以不同操作系统工作的终端设备——例如iOS终端设备和/或Android终端设备——在使用认证方法的情况下使用相同的应用商店、例如iOS商店和/或Google-Play商店。

[0078] 涉及应用的任务的解决方案

[0079] 该任务按照权利要求10通过一种用于经由确定的远程通信系统来流处理和再现应用程序(APP)的远程通信网络来解决，在所述远程通信网络中，一个流服务器或多个能通过远程通信相互连接的流服务器实施所涉及的应用程序，并且所述流服务器与相应的远程通信终端设备位置靠近地连接，其中，所涉及的远程通信终端设备从位置靠近的服务器调用所期望的应用程序(Applikation)，该位置靠近的服务器为所涉及的应用程序的呈现和编码提供计算性能。

[0080] 权利要求11说明用于在对应用程序陌生的系统环境上再现应用程序的远程通信网络的应用，所述系统环境要么以不同硬件组件要么以不同的软件组件区分，其中，流服务器承担对不同应用程序的处理以及对应用程序及其各个图像(帧)音频和视频信号的呈现和/或编码，其中，将数据传输给相应的远程通信终端设备——移动无线电设备、平板、手提电脑、个人电脑、电视并且借助修改的h.254协议来实施所述传输，使用WAN作为用于音频/视频包的经由UDP/TCP的传输方式并且由所涉及的流服务器承担全部的计算性能，其中，仅在远程通信终端设备中对所打包的数据进行解码。

[0081] 按照权利要求12表示用于提供与平台无关的流技术的远程通信系统的应用，所述流技术被一次编程并且能被移植到任意的远程通信终端设备上，在所述远程通信终端设备中经由WAN实现对各个应用程序(Applikationen)、例如视频游戏的流处理，使得：

[0082] a)借助远程通信终端设备(小应用程序)来实施与会话服务器的通信；

[0083] b)为确定的终端用户，针对所涉及的应用程序(例如游戏)的在地理上与远程通信终端设备最近的流服务器，实施确定的会话；

[0084] c)通过所涉及的会话服务器将会话信息告知给远程通信终端设备和流服务器；

[0085] d)在远程通信终端设备与所涉及的应用程序(例如视频游戏)的流服务器之间实施直接连接；

[0086] e)在远程通信终端设备与所涉及的流服务器之间建立直接连接时引入下列步骤：

[0087] i.经由所涉及的流服务器记录运行的应用程序(例如游戏)的音频/视频数据，所述游戏在所述流服务器上运行。

[0088] ii.通过高质量的硬件编码器压缩所述音频/视频数据；

[0089] iii.经由WAN传输所压缩的音频/视频数据；

[0090] iv.在远程通信终端设备侧接收所述音频/视频数据；

[0091] v.解压缩所述音频/视频数据；

[0092] vi.在远程通信终端设备(小)上可视化所述音频/视频数据；

[0093] vii.在远程通信终端设备(小)上记录远程通信终端设的用户(例如游戏者)的动作(输入)；

[0094] viii.将所述输入高效地往回传输给游戏的所涉及的流服务器，以及

[0095] ix.在流服务器上再现所传输的用于应用程序的输入。

[0096] 权利要求13表示远程通信网络用于与客户端(用户、终端设备)通信的应用，具有

下列源代码：

[0097] /**********AddPortAsynchronisation.java**********

[0098] Responsible for adding relevant ports to network devices to make ensure smooth communication,technology targets a technique that can run independent of network hardware[负责将相关的端口添加至网络设备(例如路由器)以便这样确保流畅的通信。该技术能够实现与用户的网络硬件无关的普遍使用。]

[0099] **********/

[0100] package org.cloundgaming4u.client.portforwarding;

[0101] import java.io.lOException;

[0102] import net.sbbi.upnp.messages.UPNPResponseException;

[0103] import android.content.Context;

[0104] import android.os.AsyncTask;

[0105] import android.util.Log;

[0106] public class AddPortAsync extends AsyncTask<Void,Void,Void>{

[0107] private Context context;

[0108] private UPnPPortMapper uPnPPortMapper;

[0109] private String externallP;

[0110] private String internallP;

[0111] private int externalPort;

[0112] private int internalPort;

[0113] public AddPortAsync(Context context,UPnPPortMapper uPnPPortMapper, String externallP,String internallP,

[0114] int externalPort,int internalPort){

[0115] this.context＝context;

[0116] this.uPnPPortMapper＝uPnPPortMapper;

[0117] this.externallP＝externallP;

[0118] this.internallP＝internallP;

[0119] this.externalPort＝externalPort;

[0120] this.internalPort＝internalPort;

[0121] }

[0122] @Override

[0123] protected void onPreExecute(){

[0124] super.onPreExecute();

[0125] if(uPnPPortMapper＝＝null)

[0126] uPnPPortMapper＝new UPnPPortMapper();

[0127] }

[0128] @Override

[0129] protected Void dolnBackground(Void...params){

[0130] if(uPnPPortMapper!＝null)

[0131] {

[0132] try

[0133] {

[0134] Log.d("cg4u_log","Contacting Router for setting network configurations");

[0135] if(uPnPPortMapper.openRouterPort(externallP,

[0136] externalPort,internallP,internalPort,"CG4UGames"))

[0137] {

[0138] Log.d("cg4u_log",String.format("Setting network configurations successful IP:%s Port:%d",externallP,externalPort));

[0139] Log.d("cg4u_log",String.format("Setting network configurations successful

[0140] IP:%s Port:%d",internallP,intemalPort));

[0141] }

[0142] }

[0143] catch(lOException e)

[0144] {

[0145] e.printStackTrace();

[0146] }

[0147] catch(UPNPResponseException e)

[0148] {

[0149] e.printStackTrace();

[0150] }

[0151] }

[0152] return null;

[0153] }

[0154] @Override

[0155] protected void onPostExecute(Void result){

[0156] super.onPostExecute(result);

[0157] //Send broadcast for update in the main activity

[0158] //Intent i=new lntent(ApplicationConstants.APPLICATION_ENCODING_TEXT);

[0159] //context.sendBroadcast(i);

[0160] }

[0161] }

[0162] /*************UniversalPortMapper.java**********************Responsible for making sure that random port generated by server is dynamically[负责服务器的一般端口指派]

[0163] mapped at

[0164] dient end

[0165] **********/package org.cloundgaming4u.client.portforwarding;

[0166] import net.sbbi.upnp.impls.InternetGatewayDevice;

[0167] import net.sbbi.upnp.messages.UPNPResponseException;

[0168] import java.io.lOException;

[0169] public class UPnPPortMapper{

[0170] private lnternetGatewayDevice[]internetGatewayDevices;

[0171] private InternetGatewayDevice foundGatewayDevice;

[0172] /**

[0173] *Search for IGD External Address

[0174] *@return String

[0175] */

[0176] public String findExternallPAddress()throws lOException, UPNPResponseException

[0177] {

[0178] /**Upnp devices router

[0179] search*/

[0180] if(internetGatewayDevices==null)

[0181] {

[0182] internetGatewayDevices=

[0183] InternetGatewayDevice.getDevices(ApplicationConstants.SCAN_TIM EOUT);

[0184] }

[0185] if(internetGatewayDevices!=null)

[0186] {

[0187] for(InternetGatewayDevice IGD:internetGatewayDevices)

[0188] {

[0189] foundGatewayDevice=IGD;

[0190] return IGD.getExternallPAddress().toString();

[0191] }

[0192] }

[0193] return null;

[0194] }

[0195] /**

[0196] *Search Found Internet Gateway Device Friendly Name

[0197] *@return

[0198] */

[0199] public String findRouterName(){

[0200] if(foundGatewayDevice!=null){

```
[0201]  return
[0202]  foundGatewayDevice.getIGDRootDevice().getFriendlyName().toString
[0203]  ();
[0204]  }
[0205]  return"null";
[0206]  }
[0207]  /**
[0208]  *Open Router Port
[0209]  *IGD==Internet Gateway Device
[0210]  *
[0211]  *@param internalIP
[0212]  *@param internalPort
[0213]  *@param externalRouterIP
[0214]  *@param externalRouterPort
[0215]  *@param description
[0216]  *@return
[0217]  *@throws IOException
[0218]  *@throws UPNPResponseException
[0219]  */
[0220]  public boolean openRouterPort(String externalRouterIP,int
externalRouterPort,
[0221]  String internalIP,int internalPort,
[0222]  String description)
[0223]  throws IOException,UPNPResponseException{
[0224]  /**Upnp devices router
[0225]  search*/
[0226]  if(internetGatewayDevices==null){
[0227]  internetGatewayDevices=
[0228]  InternetGatewayDevice.getDevices(ApplicationConstants.SCAN_TIM
EOUT);}
[0229]  if(internetGatewayDevices!=null){
[0230]  for(InternetGatewayDevice addIGD:internetGatewayDevices){
[0231]  /**Open port for TCP protocol and also for UDP protocol
[0232]  *Both protocols must be open this is a MUST*/
[0233]  //addIGD.addPortMapping(description,externalRouterIP,internalPort,
externalRouterPort,internalIP,0,ApplicationConstants.TCP_PROTOCOL);
[0234]  addIGD.addPortMapping(description,externalRouterIP,internalPort,
externalRouterPort,internalIP,0,ApplicationConstants.UDP_PROTOCOL);
[0235]  }
```

[0236] return true;

[0237] }else{

[0238] Return false;

[0239] }

[0240] }

[0241] public boolean removePort(String extemallP.int port)throws lOException,

[0242] UPNPResponseException{

[0243] /**Upnp devices router

[0244] search*/

[0245] if(internetGatewayDevices==null){

[0246] internetGatewayDevices=lnternetGatewayDevice.getDevices(5000);

[0247] }

[0248] /**Remote port mapping for all routers*/

[0249] if(internetGatewayDevices!=null){

[0250] for(InternetGatewayDevice removelGD:internetGatewayDevices){// removelGD.deletePortMapping(externallP,port,ApplicationConstants.TCP_PROTOCOL);

[0251] removelGD.deletePortMapping(externallP,port,"UDP");

[0252] }

[0253] return true;

[0254] }else{

[0255] return false;

[0256] }

[0257] }

[0258] }

[0259] **********

[0260] End of ClientNetworkCommunication

[0261] **********

[0262] 在权利要求14中说明另一种优选的方法,该方法用于经由确定的远程通信系统来流处理和再现应用程序,在所述方法中,具有不同操作系统的终端设备、例如具有Android软件的终端设备或具有iOS软件的终端设备在一个以及同一个远程通信系统中以应用程序工作,其中,在一个终端设备上已加载的应用程序在变换至以另外的软件、例如Android软件工作的另一个终端设备时在新终端设备上不变地被继续使用,之前在较早终端设备上已加载的应用程序由新终端设备在没有重新获得所述应用程序的情况下经由以另外的软件工作的较早的终端设备的识别方法在新终端设备上不变地被继续使用。

[0263] 根据权利要求15所述的应用方式的特征在于,以不同操作系统工作的终端设备——例如iOS终端设备和/或Android终端设备——在使用认证方法的情况下使用相同的应用商店、例如iOS商店和/或Google-Play商店。

[0264] 在附图中，借助实施例——部分示意地——阐明所述方法。

[0265] 在不同的配备有触摸屏的移动无线电终端设备之间的通信经由云端和流服务器来实现。

[0266] 用附图标记1表示移动无线电终端设备的触摸屏的总共12个视图，其中，在触摸屏上示出不同的符号，移动无线电设备用户能够经由所述符号从与流服务器3通信的云端2调用应用程序。流服务器3具有需要用于与移动无线电终端设备1通信的全部数据，从而在移动无线电终端设备中的计算性能可以非常小。处理器可以具有最小功率，其中，不同应用程序通过符号反映在触摸屏上。在这些符号后面隐藏有经由云端2由流服务器3调用的数据量。

[0267] 用4表示来自移动无线电终端设备1的控制信号的路径并且用5表示从流服务器3至移动无线电终端设备1的视频信号的路径。

[0268] 在图2中，用附图标记6和7表示远程通信终端设备、例如移动无线电设备或平板，如在根据图1的实施方式中那样，在所述远程通信终端设备上显示未示出的视图——符号，经由所述符号能够通过远程通信终端设备6和7的相应触摸屏调用应用程序。取代移动无线电装置也可以使用平板、手提电脑等。

[0269] 终端设备6和7在需要时经由信号路径8或9与云端10在信号技术或无线电技术上连接，从所述云端中能够调用应用程序。亦即，终端设备6和7具有小的计算性能。云端10如在根据图1的实施方式中那样与流服务器11通信，该流服务器具有需要用于与远程通信终端设备6和7通信的全部数据，由此，在远程通信终端设备6和7中的计算性能能够小地保持。当然，所示出的两个远程通信终端设备6和7仅用于示例说明。在实践中，根据多少用户想要与云端10连接和用户想要调用多少应用程序，可以设有更少或非常多终端设备。同样不必具有仅一个云端2或10。根据需要也可以设置多个或大量这样的云端，多个或许多远程通信终端设备6和7等与所述云端通信并且所述云端包含用于应用程序的相应数据。

[0270] 云端10在信号技术上经由信号路径17和18与不同的商店12或13连接，其中，从附图可见数量的商店12、13同样仅是示例说明。可以设置仅一个这样的商店或多个商店，从所述商店中能调用应用程序。例如，商店12可以是iOS商店，而商店13是Google-Play商店。

[0271] 远程通信终端设备6例如可以是iPAD/移动无线电设备，而终端设备7是手提电脑或根据Android标准工作的移动无线电终端设备。

[0272] 迄今为止是这样的：在远程通信终端设备(例如移动无线电设备)上的应用程序的用户在其设备变换成另外的操作系统、例如从具有iOS的终端设备变换成根据Android标准工作的终端设备时必须再购买一次已经在先前的终端设备上已加载的应用程序。现在，这按照本发明不再是必须的，因为不同的远程通信终端设备6和7能够经由云端10访问任意的商店12或13，无论其根据何种标准工作。如果远程通信终端设备6例如是根据iOS标准工作的iPAD/移动无线电设备，则其经由云端10在调用应用程序时经由信号路径14与流服务器11通信，所述流服务器检查远程通信终端设备6的用户的权利。然后，经由信号路径15通过云端10实现经由信号路径8到远程通信终端设备6上的释放。如果远程通信终端设备6的用户例如从iOS标准变换为例如根据Android标准工作的远程通信终端设备7时，他可以在不重新购买应用程序的情况下在其新的远程通信终端设备7上继续使用之前在终端设备6上已使用的应用程序。经由信号路径9和云端10以及信号路径14，流服务器11借助认证方法检

查远程通信终端设备7的用户的权利，他是否有权利使用所述应用程序，则经由信号路径14和云端10以及信号路径9释放用于终端设备7的用户的所涉及的应用程序，该用户则能够访问商店12和13之一。同样的情况根据图示也适用于远程通信终端设备6的用户，从而不仅Android用户7能够使用iOS商店12，而且根据iOS标准工作的远程通信终端设备6的用户例如可以访问商店13、例如Google-Play商店。

[0273] 在权利要求书中并且在说明书中说明的特征以及由附图可见的特征不仅可以单独地、而且可以以任意组合的方式对于实现本发明是重要的。

[0274] 附图标记列表

[0275] 1 移动无线电终端设备的视图/应用程序

[0276] 2 云端

[0277] 3 流服务器

[0278] 4 控制信号、信号路径

[0279] 5 视频信号、信号路径

[0280] 6 远程通信终端设备

[0281] 7 远程通信终端设备

[0282] 8 信号路径

[0283] 9 信号路径

[0284] 10 云端

[0285] 11 流服务器

[0286] 12 商店

[0287] 13 商店

[0288] 14 信号路径

[0289] 15 信号路径

[0290] 16 信号路径

[0291] 17 信号路径

[0292] 18 信号路径

[0293] 术语解释

[0294]	Application Layer	应用层
[0295]	Applications-Code	应用程序代码
[0296]	buffering	缓冲
[0297]	Client	客户端(德语为“Kunde”，即客户侧
[0298]		的应用程序或客户端应用程序)表示在
[0299]		网络的终端设备上实施的并且与中央
[0300]		服务器通信的计算机程序。
[0301]	Client Ul Module	客户端用户界面模块
[0302]	Client-Application	客户端应用程序
[0303]	Client-Session	客户端会话
[0304]	Cloud	云端-多个服务器在网络中的结合
[0305]	codec	编码器-解码器

[0306]	Content Layers	内容层
[0307]	Content Servers	内容服务器
[0308]	Content Streaming	内容流
[0309]	Controller	控制器
[0310]	Controller Module	控制器模块
[0311]	decode thread	解码线程
[0312]	Decoder-Render	解码-呈现
[0313]	Evaluator	评价器
[0314]	Evaluator Modul	评价器模块
[0315]	Frame Validation	帧检验
[0316]	interactive layer	交互层
[0317]	Intrarefresh	内部更新
[0318]	Loadbalancing	负载平衡
[0319]	Network Module	网络模块
[0320]	Not Blocking	未锁定
[0321]	Overlay	覆盖
[0322]	packaging	打包
[0323]	Recovery Module	复原模块
[0324]	Recovery Strategies	复原策略
[0325]	Render Thread	可视化实施,负责呈现【可视化】应用
[0326]		程序
[0327]	Scripting	脚本
[0328]	Session handler	会话处理程序
[0329]	shared library	共享库
[0330]	Streaming	流
[0331]	Streaming-Server	流处理器
[0332]	Streaming-Technology	流技术
[0333]	Timestamping	表示将日期指派给数据包
[0334]	UDP	用户数据包协议
[0335]	Ul Layer	用户界面层
[0336]	WAN	广域网
[0337]	Web-Apps	网页应用程序
[0338]	Webcontent Management	网页内容管理

[0339] 参考文献清单

[0340] WO 2009/073830 A1

[0341] WO 2010/141522 A1

[0342] WO 2012/037170 A1

[0343] US 2014/0073428 A1

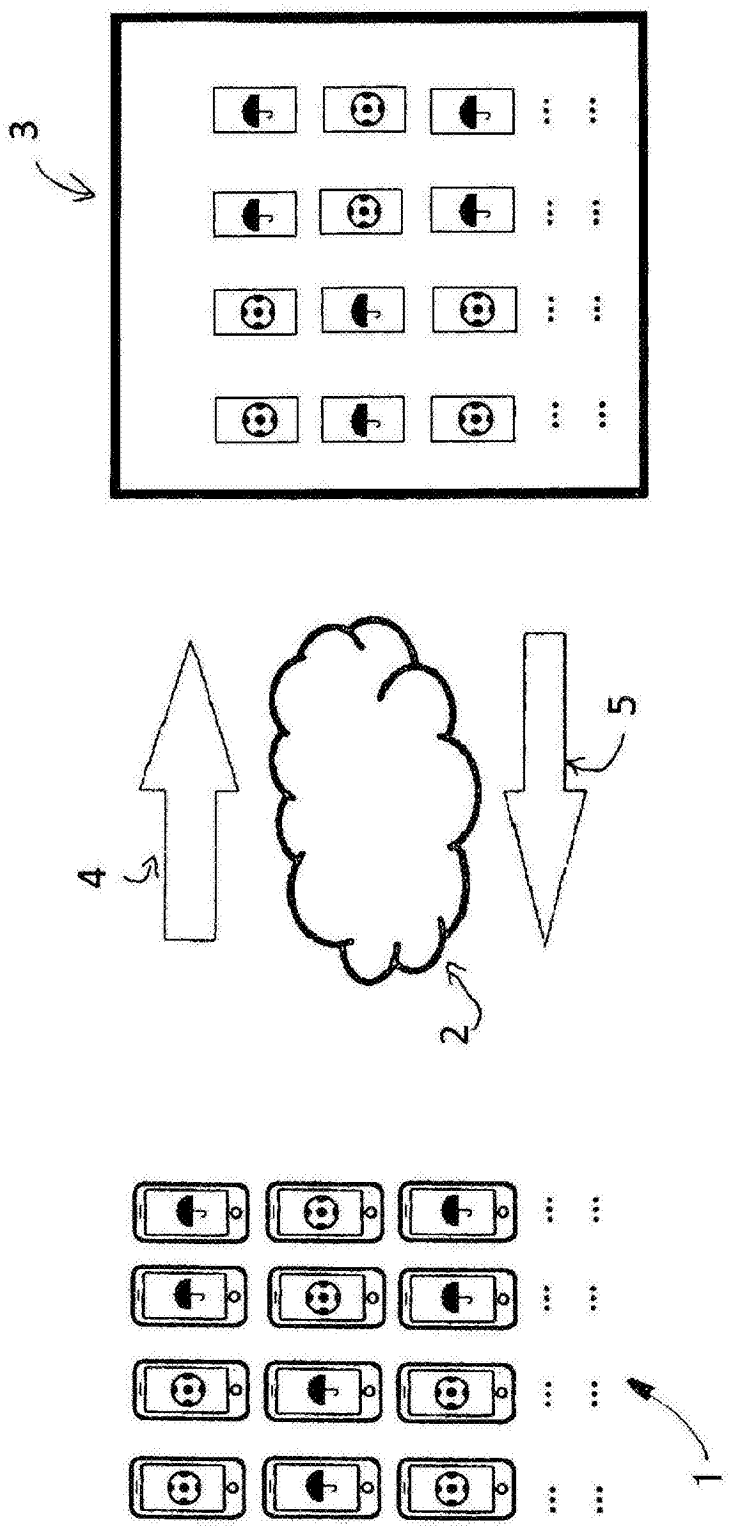


图1

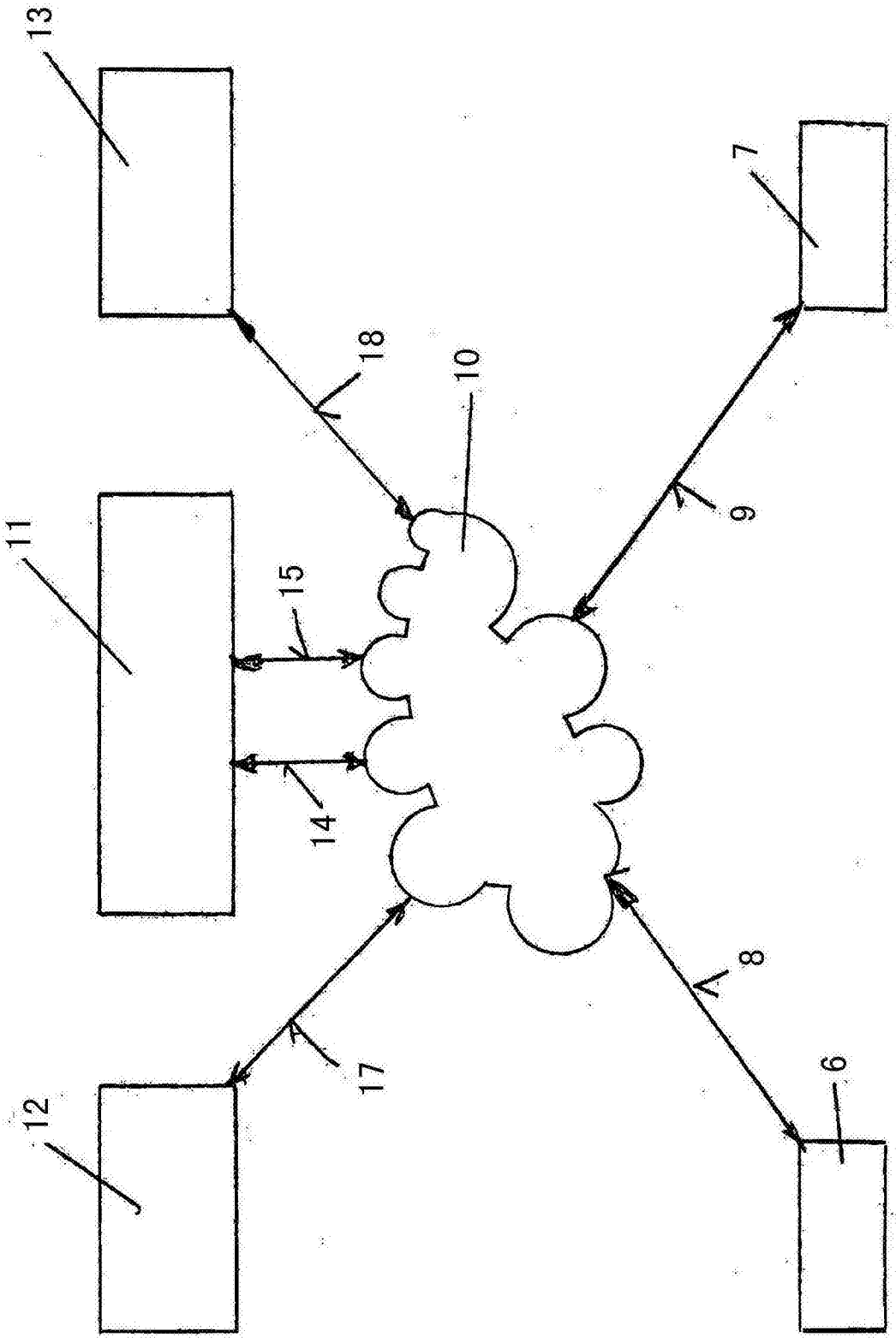


图2

1.一种用于借助配备有触摸屏的移动无线电终端设备来播放云端中的应用程序的方法,所述移动无线电终端设备具有触摸数字转换器和LCT显示器,其中,计算性能连同图形处理性能被放置在所述云端中并且在所述触摸屏上通过触摸/语音控制在所述触摸屏那里显示的符号来调用应用程序,其中,具有不同操作系统的终端设备、例如具有Android软件的终端设备和/或具有iOS软件的终端设备在一个以及同一个远程通信系统中以应用程序工作,其中,在一个终端设备上已加载的应用程序在变换至以另外的软件、例如Android软件工作的另一个终端设备时在新终端设备上不变地被继续使用,之前在较早终端设备上已加载的应用程序由新终端设备在没有重新获得所述应用程序的情况下经由以另外的软件工作的较早的终端设备的识别方法在新终端设备上不变地被继续使用,以不同的操作系统工作的终端设备——例如iOS终端设备和/或Android终端设备——在使用认证方法的情况下使用同样的应用商店、例如iOS商店和/或Google-Play商店,云端的计算性能与流服务器的性能关联,使得其直接经由确定的指示图形是可访问的,其中,流服务器还能够访问内部的服务、例如设备的存储模块,以便远程控制信息和功能。

2.按照权利要求1所述的方法,在应用与平台无关的流技术的情况下,所述流技术被一次编程并且能被移植到任意的远程通信终端设备上,在所述方法中,对各个应用程序(Applikation)、例如视频游戏的流处理经由WAN来实现,使得:

a)借助远程通信终端设备(小应用程序)来实施与会话服务器的通信;

b)为确定的终端用户,针对所涉及的应用程序、例如游戏的在地理上与远程通信终端设备最近的流服务器,实施确定的会话;

c)通过所涉及的会话服务器将会话信息通知给远程通信终端设备和流服务器;

d)在远程通信终端设备与所涉及的应用程序、例如视频游戏的流服务器之间实施直接连接;

e)在远程通信终端设备与所涉及的流服务器之间建立直接连接时引入下列步骤:

i.经由所涉及的流服务器记录运行的应用程序、例如游戏的音频/视频数据,所述游戏在所述流服务器上运行;

ii.通过高质量的硬件编码器压缩所述音频/视频数据;

iii.经由WAN传输所压缩的音频/视频数据;

iv.在远程通信终端设备侧接收所述音频/视频数据;

v.解压缩所述音频/视频数据;

vi.在远程通信终端设备(小)上可视化所述音频/视频数据;

vii.在远程通信终端设备(小)上记录远程通信终端设备的用户、例如游戏者的动作(输入);

viii.将所述输入高效地往回传输给游戏的所涉及的流服务器,以及

ix.在流服务器上再现所传输的输入。

3.一种用于经由确定的用于实施根据权利要求1或跟随该权利要求的权利要求之一所述的方法的远程通信系统来流处理和再现应用程序(APP)的远程通信网络,在所述远程通信网络中,一个流服务器或多个能通过远程通信相互连接的流服务器实施所涉及的应用程序,并且所述流服务器与相应的远程通信终端设备位置靠近地连接,其中,所涉及的远程通信终端设备从位置靠近的服务器调用所期望的应用程序(Applikation),该位置靠近的服

务器为所涉及的应用程序的呈现和编码提供计算性能，其中，在对应用程序陌生的系统环境上实现应用，所述系统环境要么以不同的硬件组件要么以不同的软件组件区分，其中，流服务器承担对不同应用程序的处理以及对应用程序及其音频和视频信号的呈现和/或编码，其中，将数据传输给相应的远程通信终端设备——移动无线电设备、平板、手提电脑、个人电脑、电视并且借助修改的h.264协议来实施所述传输，使用WAN作为用于音频/视频包的经由UDP/TCP的传输方式并且由所涉及的流服务器承担全部的计算性能，其中，仅在远程通信终端设备中对所打包的数据进行解码，其中，具有不同操作系统的终端设备、例如具有Android软件的终端设备和/或具有iOS软件的终端设备在一个以及同一个远程通信系统中以应用程序工作，其中，在一个终端设备上已加载的应用程序在变换至以另外的软件、例如Android软件工作的另一个终端设备时在新终端设备上不变地被继续使用，之前在较早终端设备上已加载的应用程序由新终端设备在没有重新获得所述应用程序的情况下经由以另外的软件工作的较早的终端设备的识别方法在新终端设备上不变地被继续使用，以不同操作系统工作的终端设备——例如iOS终端设备和/或Android终端设备——在使用认证方法的情况下使用同样的应用商店、例如iOS商店和/或Google-Play商店。

4.用于实施根据权利要求3所述的方法的远程通信网络，其具有电话终端设备、用于启动能在触摸屏上显示的流客户端的处理器和另外的通信设备、例如WLAN接收机或无线电接收机以及具有本地的存储模块，所述电话终端设备具有用于输入和再现信号的带有符号的触摸屏，所述处理器负责客户端的呈现，所述另外的通信设备用于云端与终端设备之间的通信，其中，远程通信终端设备经由所谓的接口用于两个或更多个设备之间的通信以及用于播放云端中的应用程序或者用于播放本地存储的存储器模块中的交易数据。

5.用于经由确定的远程通信系统来流处理和再现应用程序(APP)的远程通信网络的应用，在所述远程通信网络中，一个流服务器或多个能通过远程通信相互连接的流服务器实施所涉及的应用程序，并且所述流服务器与相应的远程通信终端设备位置靠近地连接，其中，所涉及的远程通信终端设备从位置靠近的服务器调用所期望的应用程序(Applikation)，该位置靠近的服务器为所涉及的应用程序的呈现和编码提供计算性能，流服务器承担对不同应用程序的处理以及对应用程序及其各个图像(帧)的音频和视频信号的呈现和/或编码，其中，将数据传输给相应的远程通信终端设备——移动无线电设备、平板、手提电脑、个人电脑、电视并且借助修改的h.264协议来实施所述传输，使用WAN作为用于音频/视频包的经由UDP/TCP的传输方式并且由所涉及的流服务器承担全部的计算性能，其中，仅在远程通信终端设备中对所打包的数据进行解码，其中经由WAN实现对各个应用程序(Applikation)、例如视频游戏的流处理，使得：

a)借助远程通信终端设备(小应用程序)来实施与会话服务器的通信；

b)为确定的终端用户，针对所涉及的应用程序、例如游戏的在地理上与远程通信终端设备最近的流服务器，实施确定的会话；

c)通过所涉及的会话服务器将会话信息通知给远程通信终端设备和流服务器；

d)在远程通信终端设备与所涉及的应用程序、例如视频游戏的流服务器之间实施直接连接；

e)在远程通信终端设备与所涉及的流服务器之间建立直接连接时引入下列步骤：

i.经由所涉及的流服务器记录运行的应用程序、例如游戏的音频/视频数据，所述游戏

在所述流服务器上运行；

ii.通过高质量的硬件编码器压缩所述音频/视频数据；

iii.经由WAN传输所压缩的音频/视频数据；

iv.在远程通信终端设备侧接收所述音频/视频数据；

v.解压缩所述音频/视频数据；

vi.在远程通信终端设备(小)上可视化所述音频/视频数据

vii.在远程通信终端设备(小)上记录远程通信终端设备的用户、例如游戏者的动作(输入)；

viii.将所述输入高效地往回传输给游戏的所涉及的流服务器，以及

ix.在流服务器上再现所传输的用于应用程序的输入。

6.根据权利要求4所述的应用，具有源代码——用于与客户端(用户、终端设备)通信——如下：

```
/**********AddPortAsynchronisation.java**********
```

Responsible for adding relevant ports to network devices to make ensure smooth communication, technology targets a technique that can run independent of network hardware [负责将相关的端口添加至网络设备（例如路由器）以便这样确保流畅的通信。该技术能够实现与用户的网络硬件无关的普遍使用]

```
**********/
package org.cloundgaming4u.client.portforwarding;
import java.io.lOException;
import net.sbbi.upnp.messages.UPNPResponseException;
import android.content.Context;
import android.os.AsyncTask;
import android.util.Log;
public class AddPortAsync extends AsyncTask<Void, Void, Void> {
private Context context;
private UPnPPortMapper uPnPPortMapper;
private String externall P;
private String internall P;
private int externalPort;
private int internalPort;
public AddPortAsync(Context context,UPnPPortMapper
uPnPPortMapper,
String
externallP, String internallP,
int externalPort, int internalPort) {
this.context = context;
this.uPnPPortMapper = uPnPPortMapper;
```

```
this. externallP = externallP;
this.internallP = internallP;
this. externalPort = externalPort;
this.internalPort = internalPort;
}
@Override
protected void onPreExecute() {
super.onPreExecute();
if(uPnPPortMapper == null)
uPnPPortMapper = new UPnPPortMapper();
}
@Override
protected Void dolnBackground(Void... params) {
if(uPnPPortMapper != null)
{
try
{
Log.d("cg4u_log","Contacting Router for setting network configurations");
if(uPnPPortMapper.openRouterPort(externallP, externalPort,internallP,internalPort, "CG4UGames"))
{
Log.d("cg4u_log",String.format("Setting network configurations successful IP:%s Port:%d",externallP,externalPort));
Log.d("cg4u_log",String.format("Setting network configurations successful
IP:%s Port:%d",internallP,intemalPort));
}
}
```

```
catch (lOException e)
{
e.printStackTrace();
}
catch (UPNPResponseException e)
{
e.printStackTrace();
}
}
return null;
}
@Override
protected void onPostExecute(Void result) {
super.onPostExecute(result);
//Send broadcast for update in the main activity
//Intent i = new
lntent(ApplicationConstants.APPLICATION_ENCODING_TEXT);
//context.sendBroadcast(i);
}
}
/*************UniversalPortMapper.java**********************
Responsible for making sure that random port generated by server is
dynamically mapped at client end [负责服务器的一般端口指派]

**********************************************************/
package org.cloundgaming4u.client.portforwarding;
import net.sbbi.upnp.impls.InternetGatewayDevice;
import net.sbbi.upnp.messages.UPNPResponseException;
import java.io.lOException;
```

```
public class UPnPPortMapper {
private lnternetGatewayDevice[] internetGatewayDevices;
private InternetGatewayDevice foundGatewayDevice;
/**
* Search for IGD External Address
* @ return String
*/
public String findExternallPAddress () throws lOException,
UPNPResponseException {
/** Upnp devices router
search*/
if(internetGatewayDevices == null)
{
internetGatewayDevices =
InternetGatewayDevice.getDevices(ApplicationConstants.SCAN_TIM
EOUT);
}
if(internetGatewayDevices != null)
{
for (InternetGatewayDevice IGD : internetGatewayDevices)
{
foundGatewayDevice = IGD;
return IGD.getExternallPAddress().toString();
}
}
return null;
}
/**
* Search Found Internet Gateway Device Friendly Name
```

```
* @return
*/
public String findRouterName(){
if(foundGatewayDevice != null){
return
foundGatewayDevice.getIGDRootDevice().getFriendlyName().toString
();
}
return "null";
}
/**
* Open Router Port
* IGD == Internet Gateway Device
*
* @param internalIP
* @param internalPort
* @param externalRouterIP
* @param externalRouterPort
* @param description
* @return
* @throws IOException
* @throws UPNPResponseException
*/
public boolean openRouterPort(String externalRouterIP,int
externalRouterPort,
String internalIP,int internalPort,
String description)
throws IOException, UPNPResponseException {
/** Upnp devices router
```

```
search*/
if(internetGatewayDevices == null){
internetGatewayDevices =
lnternetGatewayDevice.getDevices(ApplicationConstants.SCAN_TIM
EOUT);
}
if(internetGatewayDevices != null){
for (InternetGatewayDevice addIGD : internetGatewayDevices) {
/** Open port for TCP protocol and also for UDP protocol
* Both protocols must be open this is a MUST*/
//addlGD.addPortMapping(description, externalRouterIP,
internalPort, externalRouterPort, internallP, 0,
ApplicationConstants.TCP_PROTOCOL);
addlGD.addPortMapping(description, externalRouterIP, internalPort,
externalRouterPort, internallP, 0,
ApplicationConstants.UDP_PROTOCOL);
}
return true;
}else{
Return false;
}
}
public boolean removePort(String extemallP,int port) throws
lOException,
UPNPResponseException{
/** Upnp devices router
search*/
if(internetGatewayDevices == null){
internetGatewayDevices = lnternetGatewayDevice.getDevices(5000);
```

```
}
/**Remote port mapping for all routers*/
if(internetGatewayDevices != null){
for (InternetGatewayDevice removelGD : internetGatewayDevices) {
//              removelGD.deletePortMapping(externallP,          port,
ApplicationConstants.TCP_PROTOCOL);
removelGD.deletePortMapping(externallP, port, "UDP");
}
return true;
}else{
return false;
}
}
}
```

****

End of ClientNetworkCommunication

客户端网络通信结束

****