IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 12, December 2017

VR Based Smart Gaming using Virtual Network Computing

Kaustubh Joshi[1], Saurabh Panse[2], Tejas Kharche[3], Akshay Chaudhari[4], D.O. Shamkuwar[5]

Student, Computer Department, SITS, Narhe,Pune, Maharashtra [1-4]

Assistant Professor, Computer Department, SITS,Narhe, Pune, Maharashtra[5]

Abstract: Today's anyone cannot buy some expensive devices for gaming. Virtual reality gaming is where a person can experience being in a three-dimensional (3D) environment and interact with that environment during a game. Creating fake 3D effect is an important part of the game. Anyone cannot buy expensive hardware but we already have the hardware in your pocket. This application uses the display and sensors of your existing android phone to transform it into a portal to your PC games, so that we can play our PC games in virtual reality on android mobile. This VR system is basically about mirroring your Windows display to your Android Smartphone in the split-screen VR mode. It also makes sure to create a fake 3D effect on games.

Keywords:VR (Virtual Reality), Three-Dimensional Environment, Remote Procedure Call (RPC).Joystick Events, Screen mirroring.

I. INTRODUCTION

VR is basically mirrored for Windows display for Android Smartphone in the split-screen format. It also makes sure to create a fake 3D effect on games. So,more and more people are interested in VR games. To answer these demands, the industries quickly responses by making many different VR games. The various general input devices for computers are the mouse, keyboard.We are giving alternative input devices for VR games i.e. joystick events are sent to the game via joystick.

II. RELATED WORK

[1] In this paper work-in-progressfor the usability study of VR basic controller, an HTC vibe controller and a handmade data glove in the context of VR games is given. We compared the effectiveness of those two-input hardware as hand gesture inputs in VR game.

[2]In This paper, the propose Remote Desktop Client Platform (RDCP) solution allow PC applications to be easily adapted to Smart TV application environment. We have used Virtual Network Computing (VNC) as a core of RDCP. Additionally, we have divided the big screen into two sub-screens, to get extra functionalities and comfortable usage. This method retains original functionalities of Smart TV. Implementation of this solution has been done using Smart TV SDK that contains cutting-edge web technologies with the last updates of HTML5. By using proposed RDCP solution, the VNC ported Smart TV could extend its usage into education and entertainment field. In addition, we have implemented assistive tools to control Smart TV remotely.

[3]In this paper, a hybrid encoder model to cast Windows screen to Android. Compared to the classic way of solving this problem, our method is more efficient in screencasting in that a heuristic decision is made to determine whether traditional VNC mode or streaming mode is activated. The experiments between the server using our hybrid encoder model and classic VNC server show that our method takes less network bandwidth and renders more frames when sending multimedia over a wireless network.

[4] In this paper, implementationof a prototype system for mobile VNC, and several works are done for improving screen update rate. At first, a number of video encoders are integrated into a prototype system, and wehave investigated what is the most suitable codec for mobile VNC. The existing RFB protocol is extended straightforwardly to integrate video codecs. Next, the overall system architecture is modified from serial operation to parallel. Finally, this paper proposes a modified region of coding to further reduce the encoding time of screen images. The proposed methods are implemented into our prototype mobile VNC system, and practical performances are widely evaluated. JPEG is the most suitable for mobile VNC in terms of complexity and compression ratio. In addition,the proposed modified region of coding can decrease encoding time, and consequently increase screen update rate.

[5] In this paper, thedescription of a modified Remote Frame Buffer protocol intended to be used by LAN ultra-thin clients is given. The prototype implementation is based on the ATMEL AVR microcontroller family and on the

DOI 10.17148/IJARCCE.2017.61210

Microchip's Ethernet integrated circuit ENC28J60, showing that a low-cost microcontroller can implement the network functions. The main differences consist of the use of UDP as the transport protocol instead of TCP and in additional fields to the RFB header.

[6]In This paper, description of an extension of VNC for effective real-time collaboration through the internet is given. For effective collaboration, we provide three kinds of access authorities: administrator, worker, and spectator. Theadministrator controls access authorities of the users. Theworkers can access the remote resources provided by thecollaboration server, while the spectators cannot handlethe remote resources, only monitoring the remote desktopof the collaboration server. To support those accessauthorities, the UltraVNC server and the JavaViewer areextended.

III. PROPOSED SYSTEM

The architecture diagram of the proposed system is as shown in below figure. For playing the game first we have to Set PC and Android configuration. The game that you want to play on VR needs to be started on PC. PC client continues to capture images from the pc and compresses it and sends it to VR mounted android phone. Pc screen images are shown in half-duplex mode on phone mounted on VR headset. Accelerometer data will be captured on android phone and corresponding desktop movements will be calculated to show the 3D effect on VR. all controls of the game will be simulated through joystick to be separately installed. The joystick will be connected to be Pc and game should also support a joystick. Remote OS screen is continuously shared with the mobile device. All GUI changes or applications are directly shown over the phone. The user can also perform events on the remote machine such as joystick events.

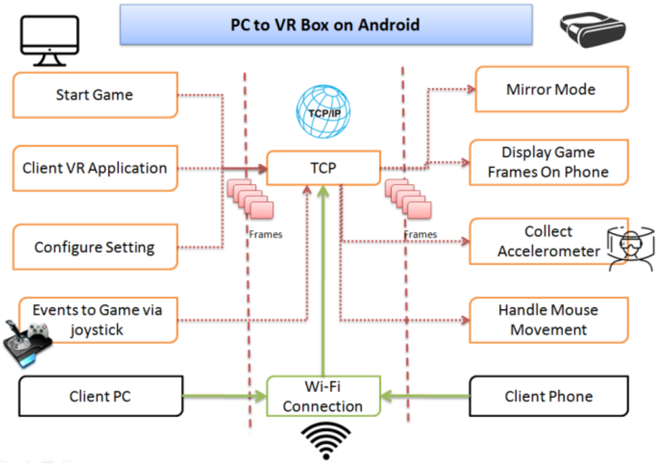

Fig. Architecture Diagram of PC to VR Box on Android

a. Remote Applications Deployment

This technique requires services to be deployed on the remote host. The offloading device invokes the target service on the remote server using well-known mechanisms such as Remote Procedure Call (RPC). Although this approach is well-supported by APIs, service pre-deployment on remote hosts poses a critical restriction to the ad-hoc nature of mobile cloud systems.

IV. PROPOSED ALGORITHM

1. AES Algorithm

AES is based on a design principle known as a substitution-permutation network, and is fast in both software and hardware.[8] Unlike its predecessor DES, AES does not use a Feistel network. AES is a variant of Rijndael which has a fixed block size of 128 bits, and a key size of 128, 192, or 256 bits. By contrast, the Rijndael specification per se is specified with block and key sizes that may be any multiple of 32 bits, both with a minimum of 128 and a maximum of 256 bits.

AES operates on a 4×4 column-major order matrix of bytes, termed the state, although some versions of Rijndael have a larger block size and have additional columns in the state. Most AES calculations are done in a special finite field.
The key size used for an AES cipher specifies the number of repetitions of transformation rounds that convert the input, called the plaintext, into the final output, called the cipher text.
The number of cycles of repetition are as follows:

- 10 cycles of repetition for 128-bit keys.
- 12 cycles of repetition for 192-bit keys.
- 14 cycles of repetition for 256-bit keys.

DOI 10.17148/IJARCCE.2017.61210

Each round consists of several processing steps, each containing four similar but different stages, including one that depends on the encryption key itself. A set of reverse rounds are applied to transform cipher text back into the original plaintext using the same encryption.

2. RFB Algorithm(Remote Frame Buffer)

To implement RFB server-side protocol and demonstrate with a small Java Swing application to transmit the main window over TCP connection to VNC viewers. The idea is to demonstrate basic features of the protocol and possible implementation in Java.
After a VNC viewer (client) establishes a TCP connection to a VNC server (RFB service), the first phase involves the exchange of protocol version:
RFB Service -------“RFB 003.003\n”------->VNC viewer
RFB Service <------“RFB 003.008\n”--------VNC viewer
It’s a simple stream of bytes which can be decoded into ASCII characters, such as “RFB 003.008\n”.
Once that is done, the next step is authentication. VNC server sends an array of bytes to indicate what type of authentications it supports. For example:
RFB Service ---------- 0x01 0x02--------->VNC viewer
RFB Service <---------- 0x02------------ VNC viewer
Here the VNC server sent only 1 possible authentication type (0x02). The first byte 0x01 denotes the number of authentication types available. VNC viewer has to reply with value 0x02, since that’s the only possible type supported by the server.
Next, server will send authentication challenge and the client has to respond with proper challenge response message and wait for the server to confirm the response. Once the client is authenticated, they can continue with the process of session establishment.
The simplest way here is to choose no authentication at all. RFB protocol is insecure anyway, regardless of authentication mechanism. If security is important, the proper way would be to tunnel RFB sessions via VPN or SSH connections. Virtual Network Computing to be a concept that simply allows one computer to delegate its GUI display to another machine for the sake of convenience. People generally use the term "VNC" informally to denote that generic category of software. If user wanted to create own system that allows virtual computing over a network then user might choose to use the existing RFB protocol to do so since it is likely that doing so will make product easier to deploy. RFB is designed specifically for VNC systems but could be used to extend the capabilities of existing products. A full VNC system consists of two parts, a client and a server. RFB is the protocol that governs the format used for passing data between the client and server.
There are many clients available already on the web, many of them free. If user create his own VNC server and adhere to the RFB specification then these pre-existing clients will be compatible with user’s system.

V. CONCLUSION

Here we present a system, Virtual reality gaming on the android phone. The user can experience being in a three-dimensional (3D) environment and interact with that environment. One of the advantages of this system is that the whole system can be implemented in very low cost and provides better speed to send data on VR mounted phone.

REFERENCES

[1]Seokwon Lee, Kihong Park, Junyeop Lee, Kibum Kim" User study of VR basic controller and data glove as hand gesture inputs in VR games"2017 IEEE DOI 10.1109/ISUVR.2017.16,2017 International Symposium on Ubiquitous Virtual Reality.
[2]Baul Kim, Rustam RakhimovIgorevich, PrimovEldorIsmoilovich, Yunjung Park, DugkiMin"Smart TV Usage Extension Method using Screen Division and Virtual Network Computing"2013.
[3]"Casting Windows to Android-Based on Hybrid Encoder Model" Ligang Wu, Wenjun Wu,2013 IEEE.
[4]"Implementation and Evaluation of Fast Mobile VNC Systems", Ha-Young Ko, Jae-Hyeok Lee, Jong-Ok Kim, Member, IEEE Transactions on Consumer Electronics, Vol. 58, No. 4, November 2012.
[5]"Design of a modified RFB protocol and its implementation in an ultra-thin client"2010 IEEE.
[6]"Extending VNC for Effective Collaboration", Tae-Ho Lee, Hong-Chang Lee, Jung-Hyun Kim, Myung-Joon Lee, Information Technologies,2008 IEEE.

DOI 10.17148/IJARCCE.2017.61210