Mobile VNC System Based Remote Control of Mobile Devices in an Android Environment

Rashmi A. Kalje
PG Student, Department of Computer Engg,
Smt. Kashibai Navale College of Engg
Pune, Maharashtra, India

S. P. Kosbatwar
Asst. Prof. Department of Computer Engg,
Smt. Kashibai Navale College of Engg
Pune, Maharashtra, India

***Abstract*— Remote control systems are very needful systems to monitor and control the devices more easily. VNC (Virtual Network Computing) is most widely used tool for sharing android applications, remotely access the graphic displays. VNC is thin client system which is a display protocol and is platform independent. In this paper, we implement a proposed system for mobile VNC. A number of video encoders are integrated into a prototype system, and we investigate what is the most suitable codec for mobile VNC. The existing RFB protocol is extended to integrate video codec's. The overall proposed architecture is updated from serial operation to parallel. And now finally, we propose a modified region coding to further reduce the encoding time of screen images. In addition, the proposed modified region coding can increase screen update rate, decrease encoding time consequently.**

Keywords*— Android, Decoding, Encoding, Remote control, RFB, Virtualization, VNC

I. Introduction

The project idea includes presenting android based remote control of mobile devices through VNC. This project proposes and analyzes different architectural approaches for the implementation of remote control systems of mobile devices using the Android software stack. In this work, we propose a fast screen sharing method to improve screen update rate in mobile VNC systems. In case of mobile devices, high complexity video compression techniques cannot be employed due to their strict computation limit. However, the bandwidth limitation requires a certain level of compression ratio. Thus, there exists a trade-off between encoder complexity and compression ratio for fast mobile VNC systems. We first integrate various video encoders into our prototype system, and explore their suitability for mobile We additionally propose a new modified region coding method which transmits only modified regions between current and previous screen images. It can further reduce encoder computation and resultantly increase screen update rate.

In recent years, there have been popularly released a variety of multimedia mobile devices such as Smartphone and tablet PC. The service that the today's mobile devices provide to users is almost comparable to desktops or similar devices. This trend requires the necessity of collaboration among these systems such as sharing multimedia contents and applications. VNC has been used as a tool for a multi-platform application suite allowing users to access graphic displays remotely. It is based on the thin-client architecture and uses the RFB (remote frame buffer) protocol for sharing a screen between distinct devices. The client software runs on the local user's machine while the server part operates at the target host. In this way, VNC brings remote display to the local user's machine. Successful application of VNC to mobile devices should guarantee rapid screen image transfer from server to client.

In figure 1, it shows the remote connection between android enabled mobile device and local PC or laptop. While other remote computing solutions and VNC are mature and very effective solutions to provide remote access to a Smartphone from a PC. Remote access from a PC is spontaneously derived from natural tendency to a user since the local PC provides a user interface that is equivalent to that of the remote Smartphone. The overall screen display is shown on the desktop/ laptop and the user controls the remote Smartphone using a mouse and a keyboard at the local desktop. In such a scenario, the overall user experience for operating remote computing is close nearly to that of using a local PC. Mobile VNC poses new encoding schemes while the information can be accessed anywhere and anytime, hence VNC is required maintain the remote control between the devices, but along with the VNC, different encoding schemes are also necessary to provide screen update rate. While performing the remote visualization, numbers of alternatives are available, selecting one as best among them is the important issue. And, for that screen image coding is used to select the best alternative when numbers of alternatives are available and their consequences cannot be forecast with certainty..

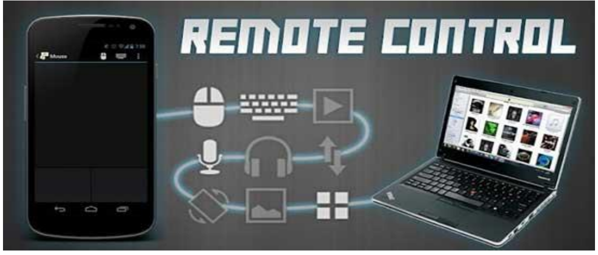

Figure 1. Remote connection between android device and local PC

II. Related Work

Many of previous works have studied efficient screen image encoding. Screen image compression is typically required to transfer screen image data with limited network bandwidth, and an appropriate encoder should be carefully selected in terms of compression ratio and speed. Original VNC uses only six encoding algorithms where all implement lossless compression. It implemented a new encoding scheme called ‘tight’. The ‘tight’ uses data analyzer and a set of data filters as preprocessors to improve adaptation ability in encoding. Tight VNC which developed by Kaplinsky [11] is an updated version of the original VNC. Image blocks are encoded by H.264 and run-length coding which are divided into high and low motion parts, respectively. Shen et al [8] first detects whether image blocks are modified or not, and only modified blocks are encoded by either JPEG or colour index based coding. H. Shen [8] deals with how to efficiently code image blocks while our work concentrates on how fast we code image blocks using modified region detection.)

In [5], the remote computer's desktop is accessed from the normal mobile phone. The remote desktop connection can be made wireless and can be accessed from any part of the world. We can control the remote computer as like our normal local computer by using a java enabled mobile phone.

In [2], it proposed that the user will be able to access and manipulate the desktops of remote computers through a VNC viewer that will be provided on the user's cell-phone.

VNC is a client-pull system where the server sends screen image data to the client only in response to the screen update request of the client. Since it prevents surplus updates, it is probably suitable for thin-client system such as VNC. In high latency environment, however, the update request from the client can be delayed, and it badly affects screen update performance. In C. Taylor and J. Pasquale [4], the message accelerator is employed to mitigate the effects of network latency. The message accelerator requests more frequently updates to the server while forwarding screen update data to the client.

System Model

The entire procedure of the mobile VNC architecture can be described in seven sequential steps. A flow graph of the proposed architecture is described in figure. 2.

1: In this step, for communication between client and server, the client (user) must login by its user name and password in order to request the service from server.

2: In this step, a client must be authenticated by server if the user ID and password are correct.

3: In this step, if client is an authorized user, then the connection between client and server is established.

4: In this step, after connection is established, the user requests the desktop of android enabled mobile device.

5: In this step, server can perform encoding which expands original bit sequence by adding necessary redundancy to check screen errors.

6: In this step, client can also access other processes like file transfer, applications, etc.

7: After finishing all the user requests, server will generate the updated mobile screen and send it to client.

8: Client can perform decoding which decodes the receiving bit sequence to recover original data i.e it receives the updated mobile screen.

9: If client doesn't want any service from server, it will end the connection with the server.

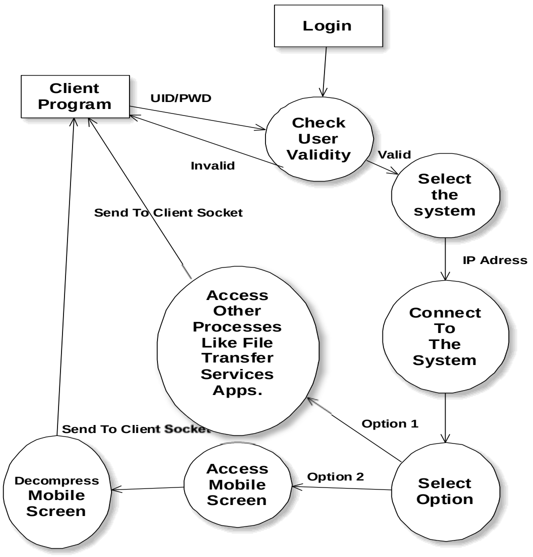

Figure 2. Flow graph of mobile VNC architecture

III. Proposed System

We implement a prototype system to evaluate proposed methods practically. Our proposed system can be used on mobile operating system platform which can be used for android devices or tablet PC's which having installed VNC application. It consists of VNC client and VNC server which are based on open source android platform.

The figure 3 describes the overall mobile VNC architecture. How the server sends the updated screen to the client is described below:

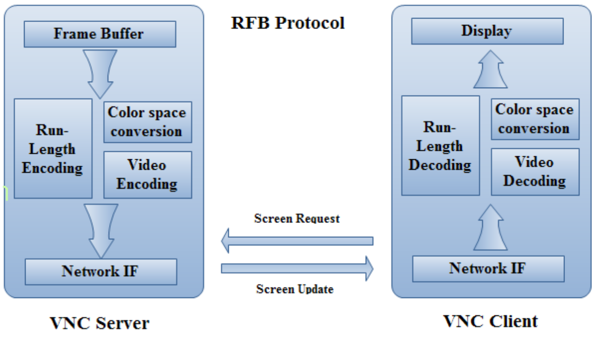

Figure 3. Mobile VNC architecture

VNC server kept the client requests in a separate frame buffer which is called as RFB (Remote Frame buffer) protocol. After receiving the screen update request from client, the server captures the screen image from frame buffer. It can be used for further processing. VNC server uses one of the improved encoding technique that is CoRRE, the colour space of the screen images is then converted into YUV format before compression because most video encoders supports YUV format which is easy to compress.

The captured screen image data is encoded and send form server to client with a framebufferUpdate message. The client then decoded and receives the updated bit stream. The procedure continues until the connection between client and server is finished or the connection is lost.

IV. Conclusion

In this paper, we implemented a proposed system for mobile VNC, and described practical performance evaluations. To integrate video codec's into our mobile VNC system, the existing RFB protocol is extended, also, protocol operations are modified to parallel for reducing unnecessary idle time. In addition to the adoption of video codec, we propose a modified region coding to further reduce the encoding time of screen images. Based on the experiments, we found that CoRRE is the most suitable for mobile VNC systems in terms of both complexity and compression ratio.

V. References

[1] http://www.android.com Retrieved March 1st, 2011.

[2] Archana Jadhav, Vipul Oswal, Sagar Madane, Harshal Zope, Vishal Hatmode. vnc architecture based remote desktop access through android mobile phones, Vol. 1, Issue 2**,** 2012.

[3] Cheng-Lin Tsao, Sandeep Kakumanu, and Raghupathy Sivakumar School of Electrical and Computer Engineering Georgia Institute of Technology Atlanta, GA 30332, USA (clt-sao,ksandeep,siva) at ece.gatech.edu, "SmartVNC: An Effective Remote Computing solutions for smart phones, in 2011

[4] C. Taylor and J. Pasquale, “Improving video performance in VNC under high latency conditions,” Proc. of the International Symposium on *Collaborative Technologies and systems* (CTS 2010), pp.26-35, May 2010.

[5] Esteve, Member, IEEE. R. Manikandasay, Remote Desktop Connection Using Mobile Phone, Volume 2, Issue 8, Aug 2013.

[6] Gonzalez Villan, Student Member, IEEE and Josep Jorba Remote Control of Mobile Devices in Android Platform Angel Gonzalez Villan, Student, IEEE and Joseph Jorba Estave, member, IEEE.

[7] Ha-Young Ko, Jae-Hyeok Lee, Jong-Ok Kim, Member, IEEE, Implementation and Evaluation of Fast Mobile VNC Systems, IEEE Transactions on Consumer Electronics, Vol. 58, No. 4, nov 2012.

[8] H. Shen, “A high-performance remote computing platform,” *Proc. of* IEEE International Conference on Pervasive Computing and *Communication (PerCom 2009), pp.1-6, Mar. 2009*

[9] Justin Grover a,b,"Android forensics: Automated data collection and reporting from a mobile device", Digital Investigation 10 (2013) S12-S20, 2013

[10] Kiran Karra, Virginia Polytechnic Institute and State University, "Wireless Distributed Computing on the Android Platform", 2012.

[11] K. V. Kaplinsky, “VNC tight encoder-data compression for VNC,” *Proc.* of the 7th International Scientific and Practical Conference of Students, Post-graduates and Young Scientists (MTT 2001), pp. 155–157, Feb. 2001.

[12] Tristan Richardson RealVNC Ltd (formerly of Olivetti Research Ltd ATandT Labs Cambridge), The RFB Protocol, Version 3.8, 2010.

[13] Tristan Richardson, Quentin Sta_ord-Fraser, Kenneth R.Wood and Andy Hopper, Virtual Network Computing, Volume 2, Number 1, Jan/feb 1998.

[14] Venkata Rakesh Ayitha, V.B.V Suman, K.Venkatesh, Srm University," Remote Display Access Using Remote Frame Buffer and Io streaming", may 2012.

[15] X. Zhang and H. Takahashi, “A hybrid data compression scheme for improved VNC,” Systemics, Cybernetics and Informatics, Vol. 5, No. 2 pp. 1-4, 2, 007.