iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 3 **Issue:** II **Month of publication:** February 2015

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

Implementing Mobile VNC System: A Literature Survey

Dhawal Solanki [#1], Nishant Toraskar [#2], Priyanka Nevrekar [#3], Ajeet Ghodeswar [#4]
[#1, #2, #3] *Department of Computer Engineering, Atharva College of Engineering, Maharashtra, India*
[#4] *Assistant Professor, Department of Computer Engineering, Atharva College of Engineering, Maharashtra, India*

Abstract-- Mobile market, which is one of the fastest growing technology markets nowadays, has been developed not only by phone manufactures but also by software companies and individual developers. More complex applications are requested by cell phone users, who expect comparable functionality to traditional PC desktop and applications. Developers should be aware that complex application means more potential bugs. In mobile VNC systems, it has been challenging to increase screen update rate by fast screen image encoding. Next, the overall system architecture is modified from serial operation to parallel.
Keywords—VNC, RFB

I. INTRODUCTION

Cellular phones have shown a dramatic improvement in their functionality to a point where it is now possible to have cellular phones execute Java programs. As a result, cellular users throughout Japan are now able to read and write e-mail, browse Web pages, and play Java games using their cellular phones. This trend has prompted us to propose the use of a cellular phone as a device for remotely controlling computers. For example, if a cellular user is able to remotely access computers (such as workstations in offices and personal computers (PCs) in homes) or other networked digital appliances, it would provide the user with the following capabilities:

A. To see the contents of a file placed on the desktop of a remote computer.
B. To reboot a remote server as an administrator.

While it is not very difficult to develop a specific system to satisfy each of the above operations separately, it lacks the generality needed for performing several such functions with one device. This paper presents a virtual network computing (VNC) based architecture for accessing the desktop of various remote systems (such as MS Windows, Macintosh, and UNIX systems) from a cellular phone. It is assumed that the remote computer system is running a VNC server and that it is attached to a network. The cellular user can see and manipulate the desktop on the cellular phone.

II. BACKGROUND

A. Why Remote Frame Buffer?

It is a simple protocol for remote access to graphical user interfaces. Because it works at the frame buffer level it is applicable to all windowing systems and applications. The remote endpoint where the user sits (i.e. the display plus keyboard and/or pointer) is called the RFB client. The endpoint where changes to the frame buffer originate (i.e. the windowing system and applications) is known as the RFB server. RFB is truly a “thin client” protocol. The emphasis in the design of the RFB protocol is to make very few requirements of the client. In this way, clients can run on the widest range of hardware, and the task of implementing a client is made as simple as possible. [3]

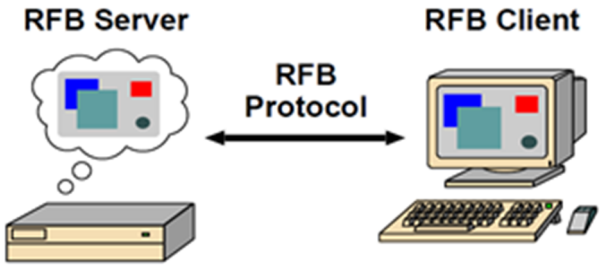


Fig. 1 Remote Frame Buffer

VNC Session is divided into three main stages. The first one is "Handshaking", then "Initialization" and after them normal process occurs. In first phase remote machine establishes a connection and negotiate any special functionality of a protocol used during communication. Negotiations begin with a decision of a protocol version supported by both side. Then is an authentication stage and after that, other features are set during an initialization. During normal VNC session client sends only information of key or pointer events and asks to send him back a graphic buffer of a server desktop (screen). Server is responsible for handling all messages from client and generates a pixel buffer enclosed desktop area. [3]

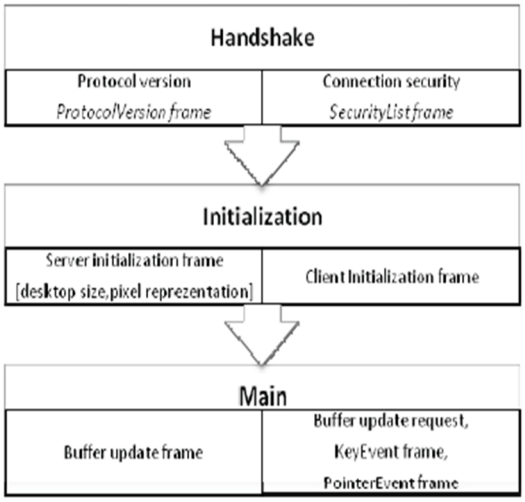


Fig. 2 Flow of the System

B. Modified Region Coding

There may be large regions which have no change between consecutive screen images, depending on applications to be shared. Motivated by this observation, we propose a modified region coding, which encodes the modified region only as illustrated in Fig. 6. Note that modified region coding is allowed for MJPEG only, and it is not applicable to typical video coding standards such as MPEG and H.264 due to their inter-frame coding.

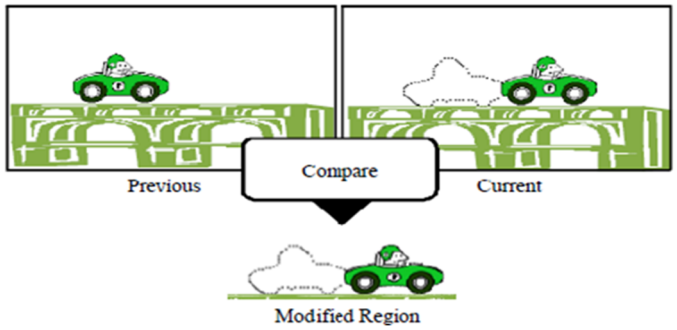


Fig. 3 Modified Region Coding

In the first step of modified region coding, a screen image is segmented into unit rectangles which are fixed size blocks. Then, difference detection between current and previous screen images is performed for each unit rectangle. If all pixel values are identical, the unit rectangle is regarded as a skip block, which does not need to be transmitted. If any difference is detected, the unit rectangle is encoded and is transmitted to the client as usual.[1]

Following Techniques for modified region encoding are to be used in our prototype.

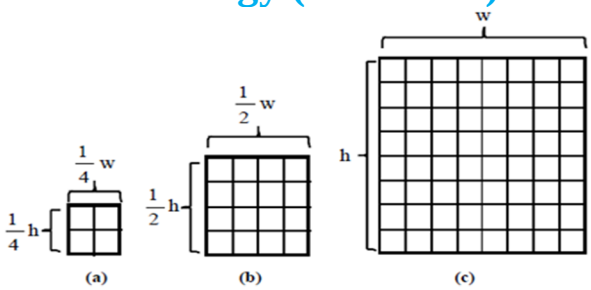


Fig. 4 Hierarchical Region Detection Algorithm

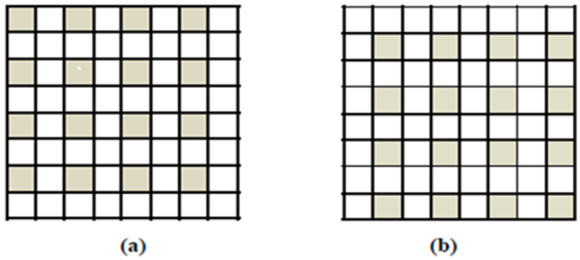


Fig. 5 Dual region detection algorithm

C. Protocol Improvement

RFB protocol operation is changed from serial to parallel in order to remove idle time redundancy. That modification is actually implemented into our prototype system, and the idle time is measured. As shown in Table, we can significantly reduce the redundant idle time, and it can contribute to boost screen update rate. [1]

TABLE
Protocol Improvement

Algorithms	Idle time(ms)
Conventional	163.85
Proposed	9.61

Table 1 *Protocol Improvement*

D. Compression Ratio

Following Table compares the compressed data size and average encoding time per screen image. Note that we use the same QP (quantization parameter) which guarantees sufficient image quality. As expected, MPEG2 and MPEG4 achieve higher compression efficiency, but it takes longer time to encode a screen image. On the other hand, MJPEG shows much faster encoding with reasonable compression ratio. [1]

TABLE
Performance of video encoders

Encoding algorithm	Compressed data size(bytes)	Encoding time(ms)	Quantization parameter
Zlib	157960	361.25	-
MPEG4	3557	193.93	8
MPEG2	4224	190.98	8
MJPEG	19579	138.73	8

Table 2 Compression Ratio

III. RELATED WORK

Ha-Young Ko, Jae-Hyeok Lee, Jong-Ok Kim describe that 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 we investigate what is the most suitable codec for mobile VNC. To integrate video codecs into our VNC system, the existing RFB protocol is extended, preserving backward compatibility. 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. In particular, modified region detection is significantly effective for gaming video contents with small texture regions. [1]

Rashmi A. Kalje, Prof. S. P. Kosbatwar describe that Remote control systems are a very needful element to control and monitor devices quickly. Although the part of remote control, in case of mobile devices, has been slight explored, it may provide important advantages for testing software and hardware developments in several real devices. 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. 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. 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 [2].

Tristan Richardson, Quentin Stafford-Fraser, Kenneth R. Wood and Andy Hopper describe that technology underlying the VNC system is a simple protocol for remote access to graphical user interfaces. It works at the frame buffer level and therefore applies to all operating systems, windowing systems, and applications—indeed to any device with some form of communications link. The protocol will operate over any reliable transport such as TCP/IP. The endpoint with which the user interacts (that is, the display and/or input devices) is called the *VNC client* or *viewer*. The endpoint where changes to the frame buffer originate (that is, the windowing system and applications) is known as the *VNC server*. VNC is truly a “thin-client” system. Its design makes very few requirements of the client, and therefore simplifies the task of creating clients to run on a wide range of hardware [3].

Kaja Masthan, K. Sharath Kumar, V. Hari Prasad describe that Through the use of software VNC, acronym for virtual network computing, makes it possible to interact with a computer from any computer or mobile device on the Internet. VNC software provides cross-platform support allowing remote control between different types of computers. To use VNC you must have a network TCP/IP connection, a VNC server and a VNC viewer to connect to the computer running the VNC server. The open source version of VNC has been freely available since 1998, and more than 20 million copies of the software have been downloaded. The existing RFB protocol is extended straightforwardly to integrate video codecs. Next, the overall system architecture is modified from serial operation to parallel. Finally, we propose a modified region 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. We report that JPEG is the most suitable for mobile VNC in terms of both complexity and compression ratio. In addition, the proposed modified region coding can decrease encoding time and consequently increase screen update rate**.** [4]

Ajit Kotkar, Alok Nalawade, Siddhesh Gawas, Aniket Patwardhan, Snehal Mangale describe that Android based remote desktop client is a remote control system which allows you to view and interact with one computer (known as “server”) to another computer or cellular phones (Android OS) anywhere on the intranet. A proxy is used to send the image of the desktop to the cellular phone, to convert different devices, to suppress network traffics, and to support recovery from an unscheduled disconnection. A prototype of the proposed system been implemented using Android and will be tested on a Android Virtual Device emulator. In the proposed system we allow access to user which is registered on the network, so there is less constraint on security. Server we designed is java based and can be installed on any operating system which runs java. The only thing needed is android operating system Smartphone.[5]

Jaya Bharathi chintalapati1, Srinivasan Rao T.Y.S2 describe that image of the desktop is compressed before it is transmitted to the cellular phone. There are several functions provided so as to ease the viewing on cell-phones. There is shortcut function that can be used to quickly access the frequently used area. Current key assignments can be viewed using guidance function. A user can view two areas simultaneously using a twin view function. The prototype is already implemented using java and tested on a java based cellular phone. The VNC architecture will be used for implementation of the system. Due to wide use of android devices, this

system will be developed for tablets and other handheld devices. This system will provide mobility for users for controlling their computer desktops over internet.[6]

Md. Sanaullah Baig, Rajasekar M, and Balaji P describe that cloud computing means using multiple server computers via a digital network as though they were one computer. Often, the services available is considered part of cloud computing. Cloud computing broadens the range of applications offered to mobile end-users with demanding applications in terms of graphical hardware, such as 3D virtual environments, or storage capacity, such as 3D medical imaging applications. As the cloud infrastructure is shared among multiple users, these hardware resources can be provided in a cost-effective way. Mobile cloud computing can give mobile device users a number of advantages. Company users are able to share resources and applications without a high level of capital expenditure on hardware and software resources. the principle of mobile cloud computing allows to access even the most demanding applications in the cloud from intrinsically resource-constrained mobile devices. In this article, we have surveyed contemporary remote display optimization techniques specifically tailored to the short mobile device battery lifetime, the varying and limited bandwidth availability on wireless links and the interaction latency [7].

Daniel Thommes, Student Member , IEEE, Ansgar Gerlicher, Member, IEEE, Qi Wang, Member, IEEE, Christos Grecos, Senior Member, IEEE describe that Modern consumer electronic devices can easily integrate into distributed computing environments. By applying network-based user interface systems to control these devices, innovative applications become feasible. However, several requirements like high responsiveness and low bandwidth consumption need be satisfied. Particularly, to be applicable in wireless and mobile local and wide area networks, they must circumvent low available data rates and high network latency. We propose a new solution called Remote UI, which works with abstract user interface (UI) descriptions and their prioritized, incremental remote replication. The system combines different approaches from thin client computing and widget based UI systems and is further optimized in the context of wireless networks and mobile consumer devices. To demonstrate the advantages of our solution, we experimentally compare the proposed Remote UI System with Virtual Network Computing (VNC) implementations and an HTML-based solution [8]

IV. OUR ANALYSIS

In the proposed system we are going to implement VNC server which overcome some disadvantages of existing system like, Slower Screen Update Rate, Low Security. We are going to implement server which has 2 level of security one with Login ID & Password and second level has Random key generation feature. Random key will generated by server and only valid client knows the correct answer for that key. We will also implement client which support these features.

A. Properties Expected In System

1) *Performance* When working together in real-time, there should not be a too long time span between a user action and the response triggered by that action: too high delays increases task completion time and user error rate. Thus, the system should try to keep latency low.

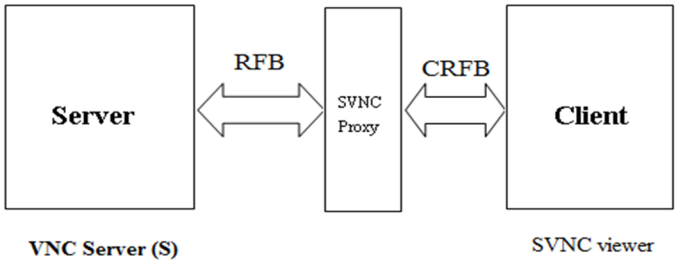


Fig. 6 Proposed Architectural Diagram

2) *Scalability* It can be that just two or well over twenty users take part. Ideally, the used system should deliver the same high level of performance with any number of connected clients. Put short, it should do well in terms of scalability.

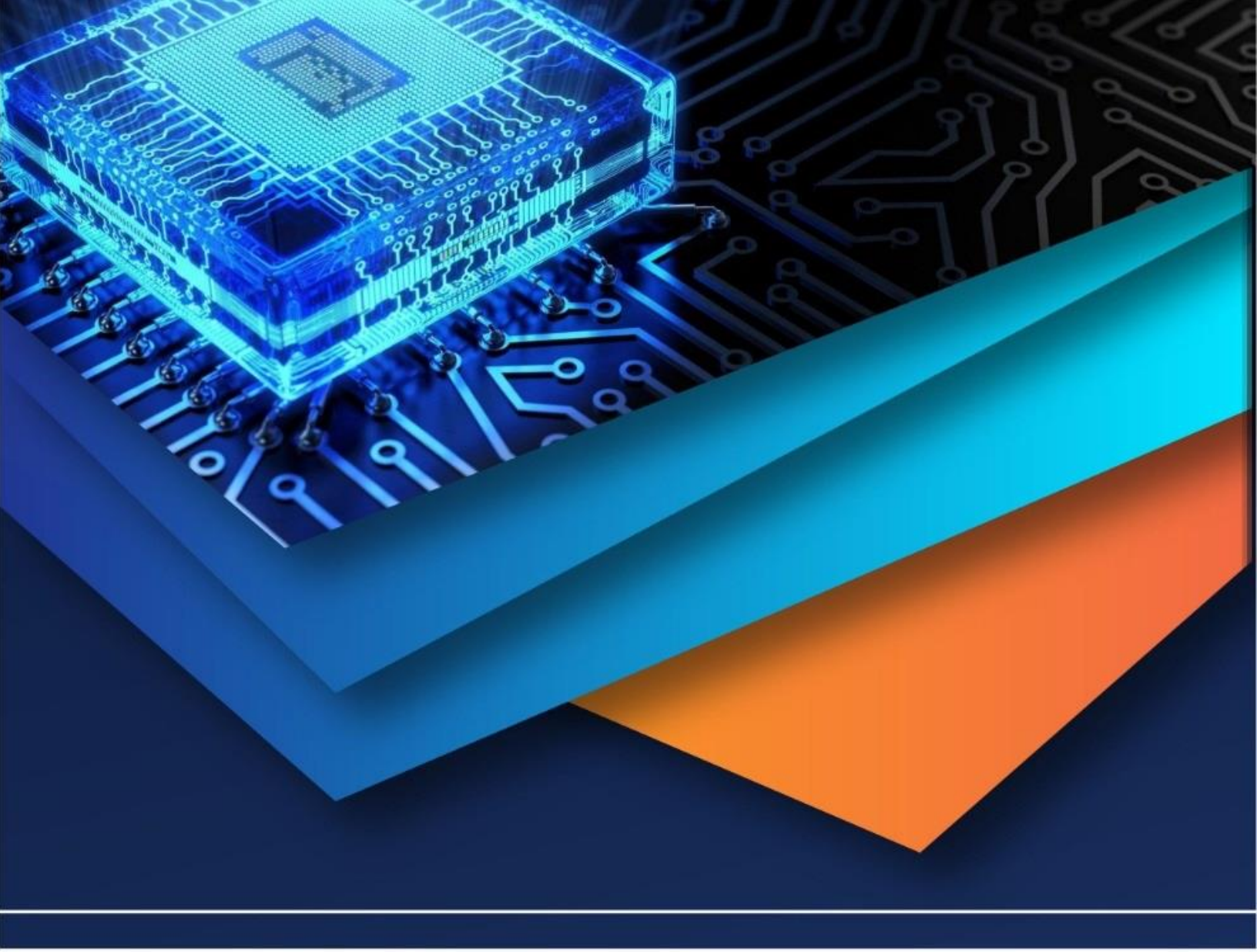
3) *Portability* It is portable as we are going to use mobile phone. We have constrains in case of using different OS. As we are going to use Java, it has problem with cross platform environment.
4) *Security* Finally, security of the employed system is of importance. On the one hand, this includes the basic question of who is allowed to use the system and who is not. On the other hand, security includes guaranteeing confidentiality, availability and integrity of the communication that is taking place.

V. CONCLUSION

We are proposing system for mobile VNC, and reported practical performance evaluations. To integrate video codecs into our VNC system, the existing RFB protocol is extended, preserving backward compatibility. 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 numerous experiments, we found that MJPEG is the most suitable for mobile VNC systems in terms of both complexity and compression ratio. Besides, the proposed modified region coding can further decrease encoding time, and consequently increase screen update rate at the client. Various practical and diverse experimental results demonstrate that the proposed methods guarantee fast screen image encoding without visual quality degradation. In particular, modified region detection is significantly effective for gaming video contents with small texture regions.

REFERENCES

[1] Implementation and Evaluation of Fast Mobile VNC Systems - Ha-Young Ko, Jae-Hyeok Lee, Jong-Ok Kim, Member, IEEE
[2] Rashmi A. Kalje et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (4) , 2014, 5089-5091
[3] T. Richardson et al., “Teleporting in an X Window System Environment,” IEEE Personal Comm., No. 3, 1994, pp. 6-12. Also available as ORL Technical Report 94.4, ORL, Cambridge CB2 1QA, England..
[4] Kaja Masthan et al, International Journal of Computer Science and Mobile Computing Vol.2 Issue. 8, August- 2013, pg. 130-135.
[5] International Journal of Innovative Research in Computer and Communication Engineering Vol. 1, Issue 2, April 2013
[6] IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 5, No 3, September 2012 ISSN (Online): 1694-0814
[7] International Journal of Computer Science and Telecommunications [Volume 3, Issue 5, May 2012]
[8] IEEE Transactions on Consumer Electronics, Vol. 58, No. 3, August 2012.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY