Comparison of File transfer protocols and using Android to access files on Desktop Wirelessly

Aditya Kothari, Tejal Gath, Mitali Gadam, Shubhay Kadam

MITCOE, Pune, Department of Computer Science & Engineering

Abstract- The popularity of wireless services is increasing in the recent years, especially accessing contents on another machine remotely. There are many applications for the purpose. Application like ES file explorer uses smb protocol for file transfer which is less secure and has performance issues. We propose to use an enhanced protocol for file sharing considering security, speed of transfer, performance and few other parameters. In this project, we propose a software that will help us access the files on Linux applications and that will also control various desktop services wirelessly. With this the need to actually change file sharing permissions every time is also minimized by using wireless execution of sharing commands.

Index Terms- Android application, Client, Server, Sharing, Files

I. Introduction

Initially phones were used only for calling and performing basic activities like text messaging. However, today the view has changed. Today, smartphones are evolving. Everything including bank transactions, social connectivity etc we are using smartphones. A lot of data is also being generated, be it personal images or office documents. All these are usually stored on one's personal computer or a hard-drive Our project aims on providing the user with the ability to gain access to these files from anywhere using a smartphones without actually using a wired connector to transfer files. Also file transfer can take place between the mobile and the computer. Problems associated with a user being away from his office or personal computer are also considered. The application will work only if the Android phone is above version 2.3.

To manage and control the data of the computers while in office is simple. But, while you are away from office/home, how will you control/access the data on your systems drive? Instead of depending on third party cloud services, one can always have your cell phone do the job. The basic idea behind our project is to provide the user with an Android application that helps one to connect to the computers while he/she is away from his desk and also get access to files on the system with easy. The computers are connected using their IP Addresses either over the internet or on the same LAN. All the systems which are on the network are scanned and respective system is connected using appropriate login details. In case of remote access, IP address and MAC details need to be saved by the user beforehand so that the system can be pinged to wake up and files can be accessed.

The user's request will be sent to the system via LAN or the internet and the system then responds accordingly and the resulting files are sent to the Android phone. Through our application certain functions are implemented like wake on LAN (WOL) through which we can switch on the computer.

II. LITERATURE SURVEY

While researching as to on which platform or rather operating system the idea has to be modeled ,we selected android due to following factors: Android is an open source platform. Supports multifunction Consists of a lot of usable libraries.. Through the survey we came across various papers which provided remote access to a single computer. Also, presently in the android market we have applications which can access the PC using the phone as if we were actually using the PC i.e. PC mirroring. Some examples include, TeamViewer, LogMeIn, Chrome Remote Desktop etc. While going through all the similar applications, we felt that no application in the market could provide the user to wake up the computer and al so provide file access services all in one application. This was one of the drawbacks of the applications that are present in the android market. Keeping this drawback in consideration, we decided to develop an application which can help the user to remotely control access to all the files on all the computers that he has registered through his android phone. Also we focus WOL(Wake On LAN) which can be done using certain operating system calls. What the previously developed applications lack, we focus on those areas to that will give a UI rich and easy to use application.

Protocols that can be used we studied and a chart based on studies was made.

Three important points to be considered are:

- Encryption - Is it necessary that the protocol is encrypted or can we trust LAN? (FTP is not encrypted, SSH/SFTP/SCP is)
- File transfer Size. (SCP is really slow compared to SMB in Gigabit networks! Google benchmarks if you are interested.)
- Necessary to keep the file permissions of the files transferred?[2].

Speed and Performance Tests

First factor to be considered is Speed of Transfer and Load on the Systems. Tests were done using the following configuration,

- Linux system: Ubuntu / Intel 2.8Ghz, 2GB Memory / Configured with NFS to export a folder.

- Windows: XP Pro / Intel 2.8Ghz, 2GB Memory / Configured with shared folder.
- Network: Computers on 100baseT, OSD connected to 802.11G wifi.
- FTP from the linux server utilized curl FTP (w/FUSE) to mount ftp to the OSD running bftpd.
- SSH from the linux server utilized SSHFS (w/FUSE) to mount sftp to the OSD running drop bear.

READ (From OSD)			
Protocol	Load Avg	Minutes	Xfr Rate
ssh	4.15	6:38	250 KB/s
ftp	3.97	3:38	360 KB/s
nfs	5.9	2:51	580 KB/s
samba	4.4	3:34	460 KB/s
cifs	4.6	3:37	460 KB/s

Fig 1: Read Speed Comparisons

WRITE (To OSD)			
Load Avg	Minutes	Xfr Rate	Transfer To/From
3.7	7:40	210 KB/s	Linux <> OSD
4.14	3:55	420 KB/s	Linux <> OSD
5	3:17	500 KB/s	OSD <> Linux
5.01	4:22	380 KB/s	OSD <> Linux
5.05	4:39	360 KB/s	OSD <> Windows

Fig 2: Write Speed Comparisons

A 100Mb file was written to/from the OSD in all cases.

Using top, the average load average was monitored, and the time (in minutes) was determined using the time command. The transfer rate was determined by the files size divided by the total time (in seconds).

Conclusion of the tests:

- SSH is fine as an alternative for telnet, but transferring files through it is comparitively slow.
- For transferring files to the OSD from a computer, ftp appears to be the quickest.
- The fastest transfer rates from the OSD to/from a computer are using NFS to a Linux system, but it does take a lot more of OSD resource[3].

Comparison of Protocols

SSH is fairly tolerant of security devices like firewalls and handling NAT. FTP is famously tricky to firewall, and generally requires one end-point to have a real IP address (i.e. no NAT).SSH is better at handling NAT. In fact, both end points can be using NAT which is generally not possible without a lot of workaround with FTP.FTP is faster protocol because of its very nature, though the right versions of SFTP can match that level of speed. FTP support is built into most of the browsers, where none have SSH. However there are plugins for this. SSH is vastly more secure, which also allows user authentication. FTP does this but over plain text.

NFS is perhaps best for more 'permanent' network mounted shares such as /home directories or usually accessed shared resources. If you want a network file share that anonymous users can easily connect to, Samba is always considered first. This is because utilities exist more easily across old and proprietary operating systems to temporarily mount and detach from Samba shares. However Samba is not at all a secure protocol. The data isn't encrypted over the transmission [4].

When comparing FTPs and SFTP, both available with strong authentication options. However since SFTP is much easier to get through firewalls, and there is an increasing percentage of trading partners adopting SFTP, SFTP is more suited for our architecture.

Table 1. Protocol Comparison on certain parameters

Protocol	**Authentication**	**Data Encryption**	**Username and password Encryption**	**Implementation**	**Platform**	**Firewall friendly**
NFS	User id and password	No	No	Need kernel Support on android or and root.	Windows + Unix	Less
SCP	User id and password	yes	yes	Installable as command line Utility on Linux	Available default on Linux OS	Less
FTP	User id and password	No	yes	Installable as command line Utility on Linux	Windows + Unix	Less
SFTP	*User id and password. Can use Public and Private Key as well*	*yes*	*yes*	*Default on Linux Machines along with SSH. Allows directory listing also*	*Default on Linux OS*	*Yes*
FTPs	User id and password. Can use Public and Private Key as well	yes	yes	Installable as command line Utility on Linux	Default with SSH	Less
SMB/CIFS	User id and password	No	No	Easy	Default on Windows	Less

Since, considering speed NFS is the fastest followed by FTP, and there are implementation complexities with NFS we consider using FTP.

SFTP gives additional security features over FTP including encryption of Payload as well as login details and also comes shipped in default with the Linux OS, SFTP seems the best option for current needs.

Wake-On-LAN

Wake on LAN (WOL) is a standard that helps you to turn on a computer from another location over a network connection or Internet .The only things you should know are IP address of remote computer and it's MAC address[5].Wake-on-LAN is a standard to set up and use on your local network, and with a little input details you can set it up so you can wake your computer away from any part of the world using a smartphone or another internet enabled system.[6]. This utility allows you to conveniently turn on one or more computers remotely by sending Wake-on-LAN (WOL) magic packet to the remote computers. Once turned on, Wake-On-Lan allows you to scan your network, and gather the MAC addresses of all your computers, and save the machine information into a separate file. Later, when your computers are put to sleep or in standby mode, you can use the file to easily choose the computer you want to turn on, and then turn on all these computers with a single push. Wake-On-Lan also allows you to switch on a system from command-line, by specifying the computer user name, IP address, or the MAC address of the remote network card[7]
The Wake-on-LAN feature can work in a couple of ways as-

- It can boot your system from a completely shut down state.[5]
- Wake-on-LAN can wake your computer from a hibernated or sleeping state. If you're working in Windows, you may need to make some changes to your BIOS before you start using the feature. Once your BIOS is updated, you need to find and adjust a few OS settings to allow you to wake up your computer using the Wake-on-LAN feature.[6]

Wake-On-Lan doesn't require any installation or extra system dll files. All you need to do is enable Wake up on magic packet in your Network card’s settings accessible through device manager setting in Windows. In order to start using it, simple run the exe or use some specified sites.

Initially you need to scan the network and collect the MAC addresses/computer names/IP addresses on your network. If a computer doesn’t show up in the scan process, you can manually enter the details like IP address and MAC address that helps identify you system on the network. Once system is found, the application just needs to send the magic packet on the receipt of which the network card helps turn-on on your pc without actually pressing the power key.

The necessary conditions for Wake on LAN to work are, that the BIOS settings have been changed and the computer is either put in sleep mode or hibernate state. While certain users may worry about power consumption and wastage of power by putting the system in hibernate mode, here is a comparison of various SLEEP states of a system.

	Shutdown (S5)	**Hibernate (S4)**	**Sleep (S3)**
Power consumption	Off, except for trickle current to devices such as the power button	Off, except for trickle current to devices such as the power button & similar devices	Less consumption. Processor is Off
Software resumption	Boot is required upon awakening.	System restarts from the saved hibernate file	YES
Hardware latency	Long and undefined	Long and undefined	Almost 2 Sec.
System hardware context	None retained.	None retained in hardware. The system writes an image of memory in the hibernate file before powering down.	Only system memory is retained. CPU context, cache contents, and chipset context are lost.
WOL Supported	No	Yes	Yes

Fig.3. Sleep State Comparison

As hibernate has negligible power consumption as compared to shut down, we ultimately choose hibernate state of both as Wake-On-Lan does not work in shut down state.

ARCHITECTURE

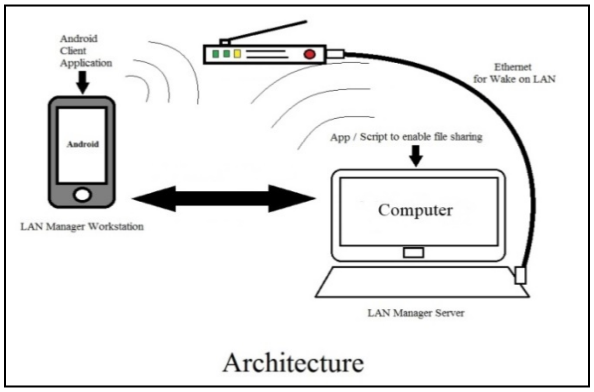

Fig 3. Architecture

The diagram above depicts the complete setup of accessing files on desktop using android. The three main components in this architecture are-

- Android phone
- Personal Computer
- Wireless Access Pont

The android phone here is client from where the actual process starts. It has a application where you need to enter the MAC address and IP address and then you get a list of all available PC's. We just need to select the required PC to get connected.
Once the connection is established you can access the PC and perform required operations.

III. FUTURE SCOPE

The android application provides you with the ease of accessing the files on your desktop, however it can also be extended to access the different services on your PC without physically connecting to it.

IV. CONCLUSION

After studying on certain parameters, it is identified that SFTP is the best suited for the needs of the application. Hence a wireless system to control the files on your Computer from an Android phone is proposed and appropriate protocols have been compared and used.

REFERENCES

[1] Shubham Bidya, Nikhil Sonawane, Nandkishor Shegokar, Prashank Bhosale and Anisaara Nadap, WITNESS ANDROID CONTROLLING REMOTE PC, published in International Journal of Advanced Research in Computer Science and Software Engineering, Volume 4, Issue 2, February 2014.

[2] "Sharing files in LAN through Samba or SSH" available at: http://askubuntu.com/questions/96849/sharing-files-in-lan-through-samba-or-ssh.

[3] "NFS vs SMB vs FTP vs SSH speed test results" available at: http://wdtvforum.com/main/index.php?topic=5393.0

[4] "Cleo protocol comparison guide" available at: https://www.cleo.com/documents/protocol_comparison_guide.pdf

[5] "Wake on Lan" available at: http://wakeonlan.me/

[6] "Access Your Computer Anytime and Save Energy with Wake-on-LAN":http://lifehacker.com/348197/access-your-computer-anytime-and-save-energy-with-wake-on-lan

[7] "Start using wake-on-lan" available at: http://www.nirsoft.net/utils/wake_on_lan.html.

AUTHORS

First Author – Aditya Kothari, MITCOE, Pune, Department of Computer Science & Engineering, kotharriaditya2@gmail.com

Second Author – Tejal Gath, MITCOE, Pune, Department of Computer Science & Engineering, tgtejal@gmail.com
Third Author – Mitali Gadam, MITCOE, Pune, Department of Computer Science & Engineering, mitalihgadam@gmail.com
Fourth Author – Shubhay Kadam, MITCOE, Pune, Department of Computer Science & Engineering, shubhay.kadam7@gmail.com