Lab Sheet 3

File Commands

(i) Listing files

Sr. No.	Command	Purpose
1	ls	List a files
2	ls -l	lists your files in 'long format', which contains lots of useful information, e.g. the exact size of the file, who owns the file and who has the right to look at it, and when it was last modified.
3	ls -a	lists all files, including the ones whose filenames begin in a dot, which you do not always want to see.
4	ls -t	List files according to the last date and time of modification
5	ls <file1> <file2> <file3>	Search for a list of files Suppose file2 does not exist, OS will inform that file2 not found.

(ii) Copying file: cp <source file> <destination file>

(iii) Renaming file: mv <source file> <destination name>

(mv actually moves contents of source file to destination file and destroys source file. As an effect do not give the destinations file same name as source file otherwise it will destroy the file.)

(iv) Removing file: rm <file1> <file2> <file3>

(v) Viewing files

Before you start, please copy below text in a new file call ‘abstract’ and ‘introduction’.

Abstract

Despite its popularity, relatively little is known about the traffic characteristics of the Skype VoIP system and how they differ from other P2P systems. We describe an experimental study of Skype VoIP traffic conducted over a one month period, where over 30 million datapoints were collected regarding the population of online clients, the number of supernodes, and their traffic characteristics. The results indicate that although the structure of the Skype system appears

to be similar to other P2P systems, particularly KaZaA, there are several significant differences in traffic. The number of active clients shows diurnal and work-week behavior, correlating with normal working hours regardless of geography.
The population of supernodes in the system tends to be relatively stable; thus node churn, a significant concern in other systems, seems less problematic in Skype. The typical bandwidth load on a supernode is relatively low, even if the supernode is relaying VoIP traffic.
The paper aims to aid further understanding of a significant, successful P2P VoIP system, as well as provide experimental data that may be useful for design and modeling of such systems. These results also imply that the nature of a VoIP P2P system like Skype differs fundamentally from earlier P2P systems that are oriented toward file-sharing, and music and video download applications, and deserves more attention from the research community.

Introduction

Email was the original killer application for the Internet. Today, voice over IP (VoIP) and instant messaging (IM) are fast supplementing email in both enterprise and home networks. Skype is an application that provides these VoIP/IM services in an easy-to-use package that works behind NAT/firewalls; it has attracted a user-base of 50 million users, and is considered valuable enough that eBay Inc. recently acquired it for more than $2.6 billion. In this paper, we conduct a measurement study of the Skype P2P VoIP network.
While measurement studies of both P2P file-sharing networks [24, 25, 2, 12, 19] and “traditional” VoIP systems [13, 16, 3] have been performed in the past, little is known about VoIP systems that are built using a P2P architecture.
One of our key goals in this paper is to understand how P2P VoIP traffic in Skype differs from traffic in P2P file-sharing networks and from traffic in traditional voice-communication networks. Do Skype users leave their client on for days, or start it just to make a call and close it soon afterwards, like file-sharing users [12]? We find that unlike file-sharing users, Skype users regularly run the client during normal working hours and close it in the evening, leading to different network dynamics. Does the fact that Skype calls are free encourage users

to talk for longer than they do on telephones where calls
are charged by the minute? We find evidence to the affirmative.
Does Skype really need the resources of millions
of peers to provide a global VoIP service, or can
a global VoIP service be supported by dedicated infrastructure?
We find that the median network utilization in
Skype peers is very low, but that peak usage can be high.
Overall, our work makes three contributions. First,
in x2, we shed light on some design choices in the proprietary
Skype network and how they affect robustness
and scalability. Second we analyze node dynamics and
churn in Skype's peer-to-peer overlay, and the network
workload generated by Skype users in x3 and x4 respectively.
Third, we provide data on user-behavior that can
be used for design and modeling of peer-to-peer VoIP
networks. Altogether, we find evidence that Skype is
fundamentally different from the peer-to-peer networks
studied in the past.

Sr. No.	Command	Purpose
1	cat <file1>	See the contents of file
2	cat <file1> <file2> <file3>	See the contents of the files together
3	more <file1> <file2> <file3>	See the contents of the files together page wise
	Enter	Display next line
	Space	Display next page
	Q	Exit from more

(vi) Create directory: mkdir <dirname>

(vii) List directories: ls –F

(viii) Remove directory:

Sr. No.	Command	Purpose
1	rmdir <dirname>	Remove empty directory
2	rm –r	Remove directory tree

(ix) Change directory:

Sr. No.	Command	Purpose
1	cd..	Move one directory back
2	cd <dirname>	Change directory
	cd <absolute path starting from/>	
	cd <relative path>	

(x) Permission notations: Read: r , Write: w, Execute: x

With read permission cat, more, vi, cp, lp commands work.

With write permission mv, rm commands work.

With execute permission, a file is declared executable.

Check File permissions: ls-l

Ex. drwxr-xr-x	2	gwl	devel	32	Feb 24 09:46	Address
-rw-r-----	1	gwl	devel	6	Jan 12 19:28	Phone

Explanation:

Character 1: d stands for directory and – for file

Character 2-4: permissions for the owner of the file. In this example owner is gwl.

Character 5-7: permissions for the group. In this example group is devil.

Character 8-10: permissions for user and everyone else.

(xi) Change permissions

Sr. No.	Command	Purpose
1	chmod o+r <filename>	O indicates others, + indicates add permission to existing permission, r indicates read permission
2	chmod o-r <filename>	- Indicates subtract the permission
3	chmod ugo=r <filename>	Set read only permission for everyone

(xii) Try youself and note down the output

1. Use mkdir to create a directory temp. Use ls to list files.
2. Use cd to change directory to temp. use pwd to make sure it works
3. Use mkdir to create another directory test. Use ls to list files.
4. Use cd to change directory to test. Use pwd to make sure it works.
5. In directory test, create a filename ‘password’. Save and exit Vi
6. Use cp to copy the /etc/passwd file to the current directory. Use ls-l to list files.

7. Use chmod to change the file permissions on passwd so that you are allowed to write to it. Use ls –l to list files.
8. Use mv and a relative path name to move passwd up one level
9. Use cd to change directory up one level and use ls to list files.
10. Use pwd to find out where you are located.
11. Use cd to change directory to your home directory. Use pwd to find out where you are located.
12. Use rmdir to remove the temp directory.
13. Use rm –r temp to remove temp directory and all its descendents. Use ls to list files.