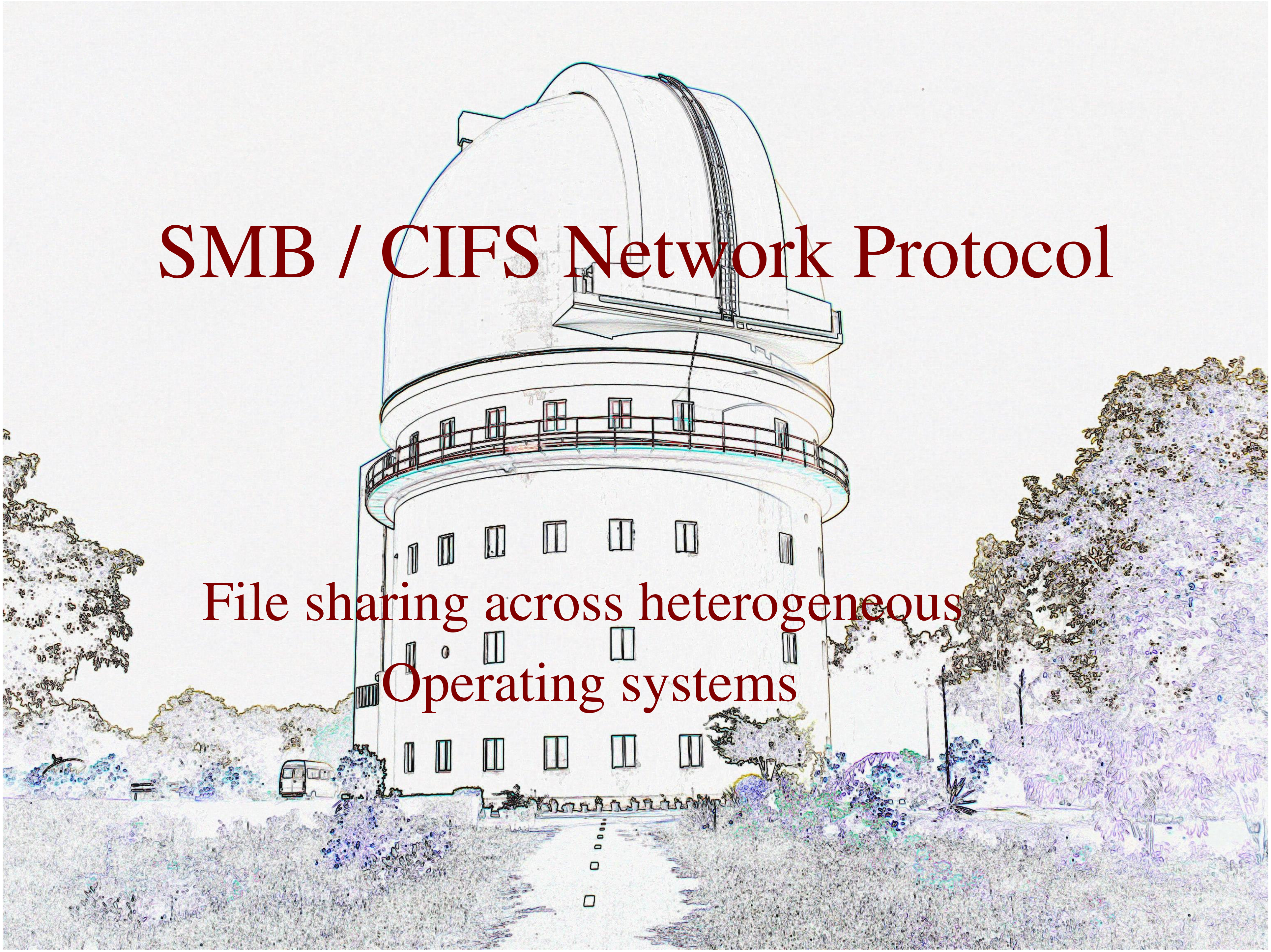
SMB / CIFS Network Protocol

File sharing across heterogeneous

Operating systems



Introduction

SMB - Server Message Block

CIFS - Common Internet File system

- Network Protocol used in windows OS since Windows for Workgroups
- used for file sharing on a LAN (using Network Neighborhood or My Network Places icons)
- Operations such as read,write,create,delete,rename etc can be done on files located on a remote server
- High level protocol – Application / Presentation layer in OSI model

Features

- Client – Server architecture
  - Remote File operations(Mapping network drives)
  - Browsing (Network Neighborhood)
  - Authentication (Windows NT, 2000)

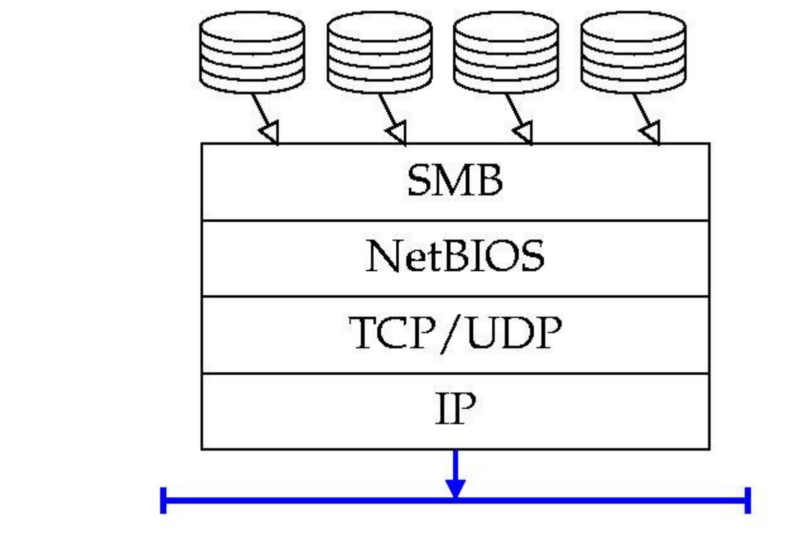
- Other Operating systems
  - Unix, Linux – Samba
  - Apple computers
  - OS/2

History

- In 1984 IBM (Barry Feigenbaum) created SMB protocol
- IBMdeveloped an API for network communication between hosts on a subnet NetBIOS API.
- Combined with a transport protocol to be called NetBEUI – NetBIOS Enhanced User Interface(PC-Network).Other transport protocols used were DECnet, IPX/SPX, TCP/IP
- In 1988 Microsoft & Intel modified the protocol titled “Core Protocol” using NetBIOS API for delivery of upper layer CIFS packets
- In 1996 SMB was renamed as CIFS with new features
- SMB / CIFS –NetBIOS over TCP used by Microsoft till windows 2000.
- Windows support 6 different variations of the CIFS protocol. Nearly 100 different operations supported
- Various versions of CIFS packets
- Internet Engineering Task Force and Storage Networking Industry Association CIFS 1.0 specification …History table
- Samba

NetBIOS over TCP

- NBT (acronym) specifications were documented in 1987 in RFC 1001 & 1002
- Three services are essential for CIFS implementation
  - Name Service
  - Session Service
  - Datagram Service



The NetBIOS layer is sandwiched between the Server Message Block (SMB) filesharing protocol and the underlying network transport layer.

Name Service

- Includes Name registration and Name query
- NetBIOS names are human readable computer names
- Just as DNS system in TCP/IP world these names should be registered and translated to IP address for transport of packets
- DNS names and IP are statically held in a server, whereas with NetBIOS the names are registered dynamically when the computer boots.
- Done by broadcasting or by using NetBIOS name server(NBNS or WINS)
- Computers are configured to use:
  - Broadcast only(b-node)
  - NBNS only (p-node)
  - Broadcast first and NBNS next if no response (m-node)
  - NBNS and Broadcast if server is unresponsive (n-mode)

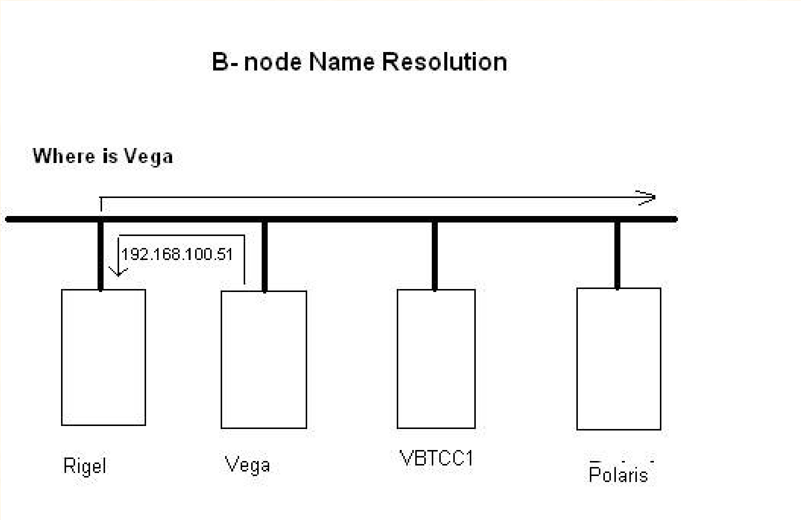
Name registration

- B – node
- Builds a NetBIOS name registration Packet and broadcasts over subnet using UDP protocol on port 137
- Contains the desired name and IP address
- Repeats three times with 250 milliseconds interval
- Any computer having the same name sends a defense packet back.
- If no defense packet is received the computer has successfully registered its name.

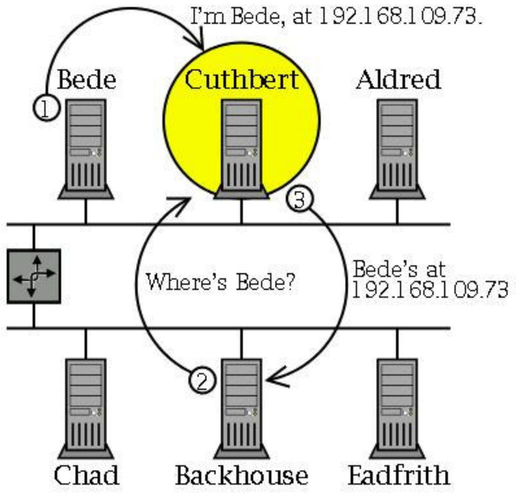
- P – node
- Builds a NetBIOS name registration packet and unicasts to the NBNS using UDP protocol on port 137
- NBNS searches it database
- If an entry with the same name is present, a negative name reg. Packet is sent. Otherwise a positive response packet is sent.

Name query

- B – node
- IP address of machines are required for transport of CIFS packets
- The name query request is broadcast over the subnet via UDP on port 137.Request contains the name
- Repeats 3 times with 5 seconds interval
- Either receives a positive name query response containing IP address or nothing



- P – node
- Name query request packet containing NetBIOS name is unicast to NBNS via UDP on port 137
- NBNS searches its data base and responds with a positive response with IP address, if a match is found. Otherwise it sends a negative response



1. Node Bede registers its name with Cuthbert, the NBNS (WINS server).
2. Node Backhouse sends a query to Cuthbert when looking for Bede.
3. Cuthbert provides the IP address of Bede to Backhouse.

Session Service

- Session is a reliable and sequential message exchange between a pair of NetBIOS applications
- TCP on port 139 is used to emulate session service functionality
- CIFS uses this service to send all upper layer commands like file,printer operations
- The following functions that are mapped into TCP :
  - CALL – initiate a NetBIOS session.Mapped into TCP as initiating and creating a full duplex TCP connection.Send a call packet containing client and server names
  - LISTEN – wait for NetBIOS call.Mapped into TCP as server waiting on port 139 for session request
  - HANG UP - end a NetBIOS session.Initiates A TCP teardown sequence

- SEND - send a message. Mapped into TCP by encapsulating the data with a small header that contains message size and then sending the data over TCP
- RECEIVE – receive e message.Mapped into TCP as receiving from TCP stream till the entire message has arrived.
- SESSION STATUS –obtain information about the requester sessions

Datagram Service

- CIFS implementations need only session and name service,but they include this service for browsing to find CIFS servers on the network
- Browsing is not part of CIFS protocol
- Datagram service is unreliable,nonsequenced,connectionless service
- UDP protocol on port 138 used to implement NetBIOS datagram service
- NetBIOS datagram packets have a header which contains the name of the sender and if the datagram is framented

- CIFS could be run over TCP without NetBIOS, DNS and domain names providing name service,session service running directly over TCP,datagram service directly over UDP

CIFS Properties

- client send requests and server respond to request
- Multiple simultaneous requests outstanding
- Each request has a unique Multiplex id (MID).when server responds to this request,it contains the MID.Client can identify for which request the reply has come.

- Command based: Each CIFS packet has 1 byte command field. Function of the packet is based on this command.The reply to the client also has the same command code.
- Protocol negotiation: There are many versions of the protocol.Each version is called a *dialect* and is assigned a unique string eg "PC NETWORK PROGRAM 1.0" or "NT LM 0.12". The first packet from client to server is the dialect negotiate packet.The client lists the dialects it understands.In the response packet the server indicates the dialect that it would communicate or it understands none.

User/share level security: The server which allows either files or printer to be shared by clients can restrict the access in two ways:-

*User level security:*The client should provide username and password to access the share.Implemented in windows NT and 2000

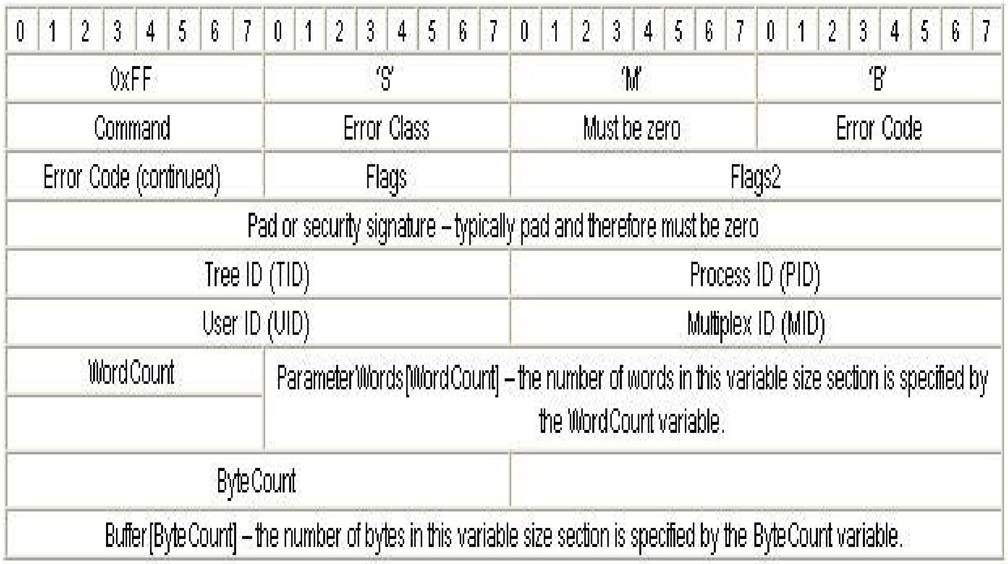
*Share level security:*The share requires only a password, implemented in Windows 95 and 98

Encryption: Both the security use encryption for the password, NT style or LAN Manager style, challenge-response authentication.The server sends a random string and client replies both random string and password.

Command batching: Many CIFS packets are capable of piggybacking other CIFS packets to reduce response latency and better network bandwidth utilization.

Opportunistic locking: This blocks multiple users modifying a same file at the same time.The server provides this oplock when a client opens a file.

CIFS Packet



0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7
0xFF	'S'	'M'	'B'
Command	Error Class	Must be zero	Error Code
Error Code (continued)	Flags	Flags2	
Pad or security signature – typically pad and therefore must be zero			
Tree ID (TID)		Process ID (PID)	
User ID (UID)		Multiplex ID (MID)	
WordCount	ParameterWords[WordCount] – the number of words in this variable size section is specified by the WordCount variable.		
ByteCount			
Buffer[ByteCount] – the number of bytes in this variable size section is specified by the ByteCount variable.			

CIFS Packet

- Header – 4 byte
- Command – 1 byte
- Error class & code – 1 & 2
- Flags & Flags2 – various options
- Tree ID – identifies the resource.TID given by server after receiving the name from client
- Process ID –identifies the process on the client which issued the request
- User ID – after verifying username and password,server issues an UID for a session
- Multiplex ID – allows multiple outstanding client requests to exist without any confusion.Server reply has MID to correlate.

CIFS Packet(contd.)

- Word Count and parameter words: These fields hold command specific data. The parameter words (various packet options)can be of variable length which is specified by the word count.
- Byte Count and buffer: Buffer hold a variable amount of raw data which is specified by the byte count.

Example Packet sequence

Example one: to connect to a server resource

Packet #1,request from client to establish a NetBIOS session

First client establishes a full duplex TCP connection with server on port 139.Then it builds and send a NetBIOS session request packet which contains client name and server name and the command for session setup.

Packet#2,response from server

The server sends a session established acknowledgement (or an error code)

Packet#3,client request for dialect negotiation

SMB_COM_NEGOTIATE command,a unique MID and a list of dialect it understands in the buffer

Packet#4,response from server of dialect chosen

MID and command are same as above,dialect is in parameter words and a eight byte random string in buffer for encryption.

Packet#5,request from client for user login

SMB_COM_SESSION_SETUP_ANDX command,username,password and other strings that identify the operating system

Packet#6,response from server

Error code if authentication fails or UID which client would send in further packets.Buffer contains details about server OS and LAN manager

Example one contd.

Packet#7, request from client to connect to a shared resource

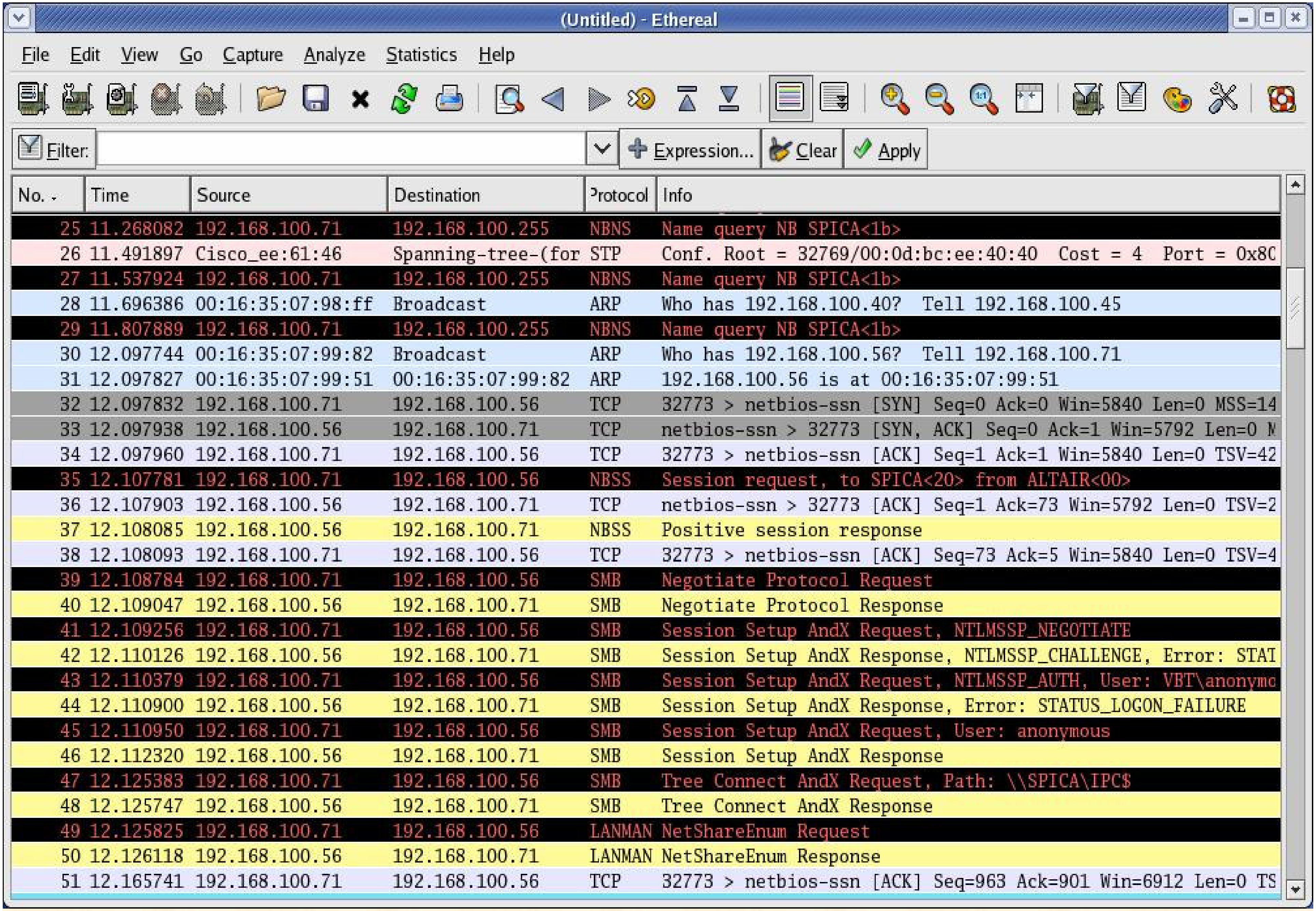
SMB_COM_TREE_CONNECT_ANDX command, share name in UNC in the buffer,UID that server gave

Packet#8,response from server indication Tree ID

TID if share exists and required permission is there or error code and class,file system type an device type in the buffer

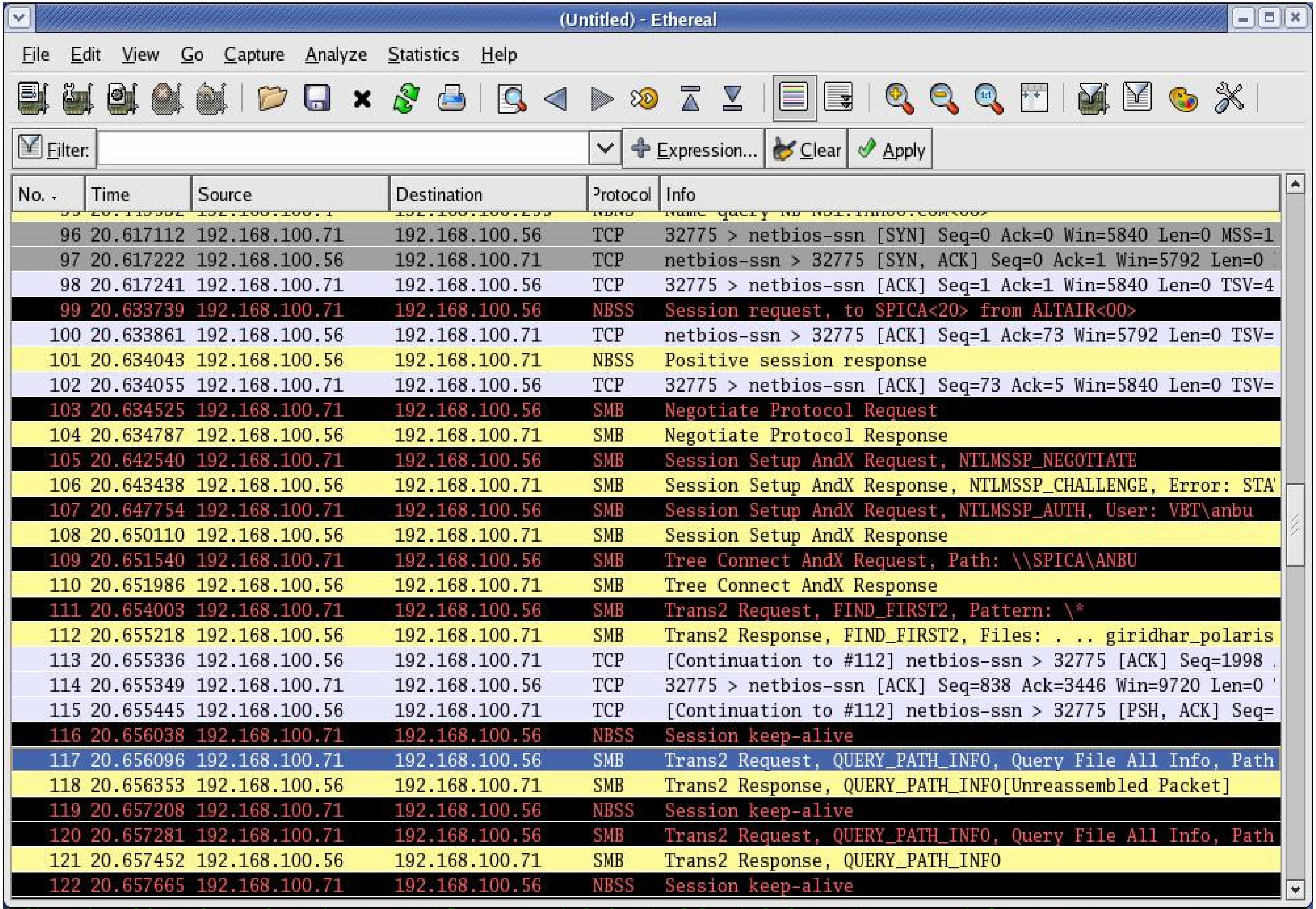
Example for file open and read

- Packet#1, request from client to open a file
  - SMB_COM_OPEN_ANDX command, file name to be opened in the buffer,options in parameter words about the opening mode and share mode
- Packet#2,response from server indicating File ID
  - IF file exists and UID has permission ,FID is sent in the parameter field or error code and class
- Packet#3, request from client for file read
  - SMB_COM_READ_ANDX command, no data in buffer, FID,file offset and 16 bit value which specifies the file offset and the amount of data required for reading
- Packet#4, response from server with file data
  - Buffer holds the file data requested



No.	Time	Source	Destination	Protocol	Info
25	11.268082	192.168.100.71	192.168.100.255	NBNS	Name query NB SPICA<1b>
26	11.491897	Cisco_ee:61:46	Spanning-tree-(for	STP	Conf. Root = 32769/00:0d:bc:ee:40:40 Cost = 4 Port = 0x80
27	11.537924	192.168.100.71	192.168.100.255	NBNS	Name query NB SPICA<1b>
28	11.696386	00:16:35:07:98:ff	Broadcast	ARP	Who has 192.168.100.40? Tell 192.168.100.45
29	11.807889	192.168.100.71	192.168.100.255	NBNS	Name query NB SPICA<1b>
30	12.097744	00:16:35:07:99:82	Broadcast	ARP	Who has 192.168.100.56? Tell 192.168.100.71
31	12.097827	00:16:35:07:99:51	00:16:35:07:99:82	ARP	192.168.100.56 is at 00:16:35:07:99:51
32	12.097832	192.168.100.71	192.168.100.56	TCP	32773 > netbios-ssn [SYN] Seq=0 Ack=0 Win=5840 Len=0 MSS=14
33	12.097938	192.168.100.56	192.168.100.71	TCP	netbios-ssn > 32773 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 M
34	12.097960	192.168.100.71	192.168.100.56	TCP	32773 > netbios-ssn [ACK] Seq=1 Ack=1 Win=5840 Len=0 TSV=42
35	12.107781	192.168.100.71	192.168.100.56	NBSS	Session request, to SPICA<20> from ALTAIR<00>
36	12.107903	192.168.100.56	192.168.100.71	TCP	netbios-ssn > 32773 [ACK] Seq=1 Ack=73 Win=5792 Len=0 TSV=2
37	12.108085	192.168.100.56	192.168.100.71	NBSS	Positive session response
38	12.108093	192.168.100.71	192.168.100.56	TCP	32773 > netbios-ssn [ACK] Seq=73 Ack=5 Win=5840 Len=0 TSV=4
39	12.108784	192.168.100.71	192.168.100.56	SMB	Negotiate Protocol Request
40	12.109047	192.168.100.56	192.168.100.71	SMB	Negotiate Protocol Response
41	12.109256	192.168.100.71	192.168.100.56	SMB	Session Setup AndX Request, NTLMSSP_NEGOTIATE
42	12.110126	192.168.100.56	192.168.100.71	SMB	Session Setup AndX Response, NTLMSSP_CHALLENGE, Error: STAT
43	12.110379	192.168.100.71	192.168.100.56	SMB	Session Setup AndX Request, NTLMSSP_AUTH, User: VBT\anonymo
44	12.110900	192.168.100.56	192.168.100.71	SMB	Session Setup AndX Response, Error: STATUS_LOGON_FAILURE
45	12.110950	192.168.100.71	192.168.100.56	SMB	Session Setup AndX Request, User: anonymous
46	12.112320	192.168.100.56	192.168.100.71	SMB	Session Setup AndX Response
47	12.125383	192.168.100.71	192.168.100.56	SMB	Tree Connect AndX Request, Path: \\SPICA\IPC$
48	12.125747	192.168.100.56	192.168.100.71	SMB	Tree Connect AndX Response
49	12.125825	192.168.100.71	192.168.100.56	LANMAN	NetShareEnum Request
50	12.126118	192.168.100.56	192.168.100.71	LANMAN	NetShareEnum Response
51	12.165741	192.168.100.71	192.168.100.56	TCP	32773 > netbios-ssn [ACK] Seq=963 Ack=901 Win=6912 Len=0 TS

Smb://spica (from 192.168.100.71 to .56)



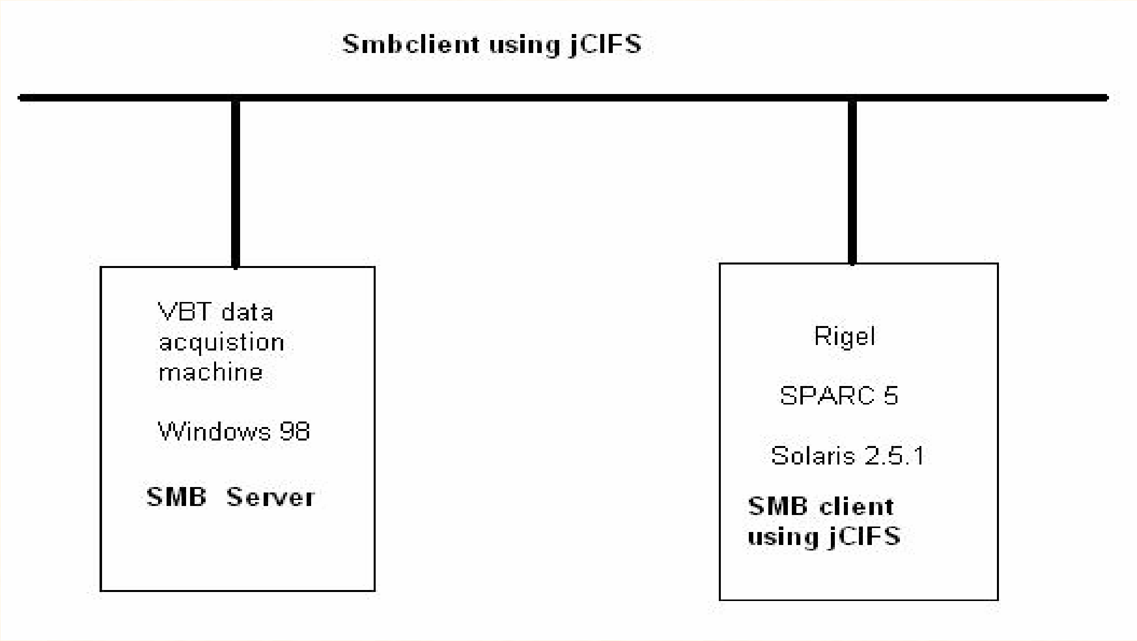
No.	Time	Source	Destination	Protocol	Info
96	20.617112	192.168.100.71	192.168.100.56	TCP	32775 > netbios-ssn [SYN] Seq=0 Ack=0 Win=5840 Len=0 MSS=1
97	20.617222	192.168.100.56	192.168.100.71	TCP	netbios-ssn > 32775 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0
98	20.617241	192.168.100.71	192.168.100.56	TCP	32775 > netbios-ssn [ACK] Seq=1 Ack=1 Win=5840 Len=0 TSV=4
99	20.633739	192.168.100.71	192.168.100.56	NBSS	Session request, to SPICA<20> from ALTAIR<00>
100	20.633861	192.168.100.56	192.168.100.71	TCP	netbios-ssn > 32775 [ACK] Seq=1 Ack=73 Win=5792 Len=0 TSV=
101	20.634043	192.168.100.56	192.168.100.71	NBSS	Positive session response
102	20.634055	192.168.100.71	192.168.100.56	TCP	32775 > netbios-ssn [ACK] Seq=73 Ack=5 Win=5840 Len=0 TSV=
103	20.634525	192.168.100.71	192.168.100.56	SMB	Negotiate Protocol Request
104	20.634787	192.168.100.56	192.168.100.71	SMB	Negotiate Protocol Response
105	20.642540	192.168.100.71	192.168.100.56	SMB	Session Setup AndX Request, NTLMSSP_NEGOTIATE
106	20.643438	192.168.100.56	192.168.100.71	SMB	Session Setup AndX Response, NTLMSSP_CHALLENGE, Error: STA
107	20.647754	192.168.100.71	192.168.100.56	SMB	Session Setup AndX Request, NTLMSSP_AUTH, User: VBT\anbu
108	20.650110	192.168.100.56	192.168.100.71	SMB	Session Setup AndX Response
109	20.651540	192.168.100.71	192.168.100.56	SMB	Tree Connect AndX Request, Path: \\SPICA\ANBU
110	20.651986	192.168.100.56	192.168.100.71	SMB	Tree Connect AndX Response
111	20.654003	192.168.100.71	192.168.100.56	SMB	Trans2 Request, FIND_FIRST2, Pattern: *
112	20.655218	192.168.100.56	192.168.100.71	SMB	Trans2 Response, FIND_FIRST2, Files: . .. giridhar_polaris
113	20.655336	192.168.100.56	192.168.100.71	TCP	[Continuation to #112] netbios-ssn > 32775 [ACK] Seq=1998
114	20.655349	192.168.100.71	192.168.100.56	TCP	32775 > netbios-ssn [ACK] Seq=838 Ack=3446 Win=9720 Len=0
115	20.655445	192.168.100.56	192.168.100.71	TCP	[Continuation to #112] netbios-ssn > 32775 [PSH, ACK] Seq=
116	20.656038	192.168.100.71	192.168.100.56	NBSS	Session keep-alive
117	20.656096	192.168.100.71	192.168.100.56	SMB	Trans2 Request, QUERY_PATH_INFO, Query File All Info, Path
118	20.656353	192.168.100.56	192.168.100.71	SMB	Trans2 Response, QUERY_PATH_INFO[Unreassembled Packet]
119	20.657208	192.168.100.71	192.168.100.56	NBSS	Session keep-alive
120	20.657281	192.168.100.71	192.168.100.56	SMB	Trans2 Request, QUERY_PATH_INFO, Query File All Info, Path
121	20.657452	192.168.100.56	192.168.100.71	SMB	Trans2 Response, QUERY_PATH_INFO
122	20.657665	192.168.100.71	192.168.100.56	NBSS	Session keep-alive

Smb://spica/anbu (from 192.168.100.71 to .56)

Requirements for file sharing at VBO

- CCD Data Acquisition systems ,windows based
- Data reduction in Solaris or Linux systems
  - Photometrics,Pixcellent,IIA Echelle systems
  - ftp used
  - Transfer of bulk data in multiple directories tiresome

Java Smbclient using jCIFS libray (RemoteCopy)



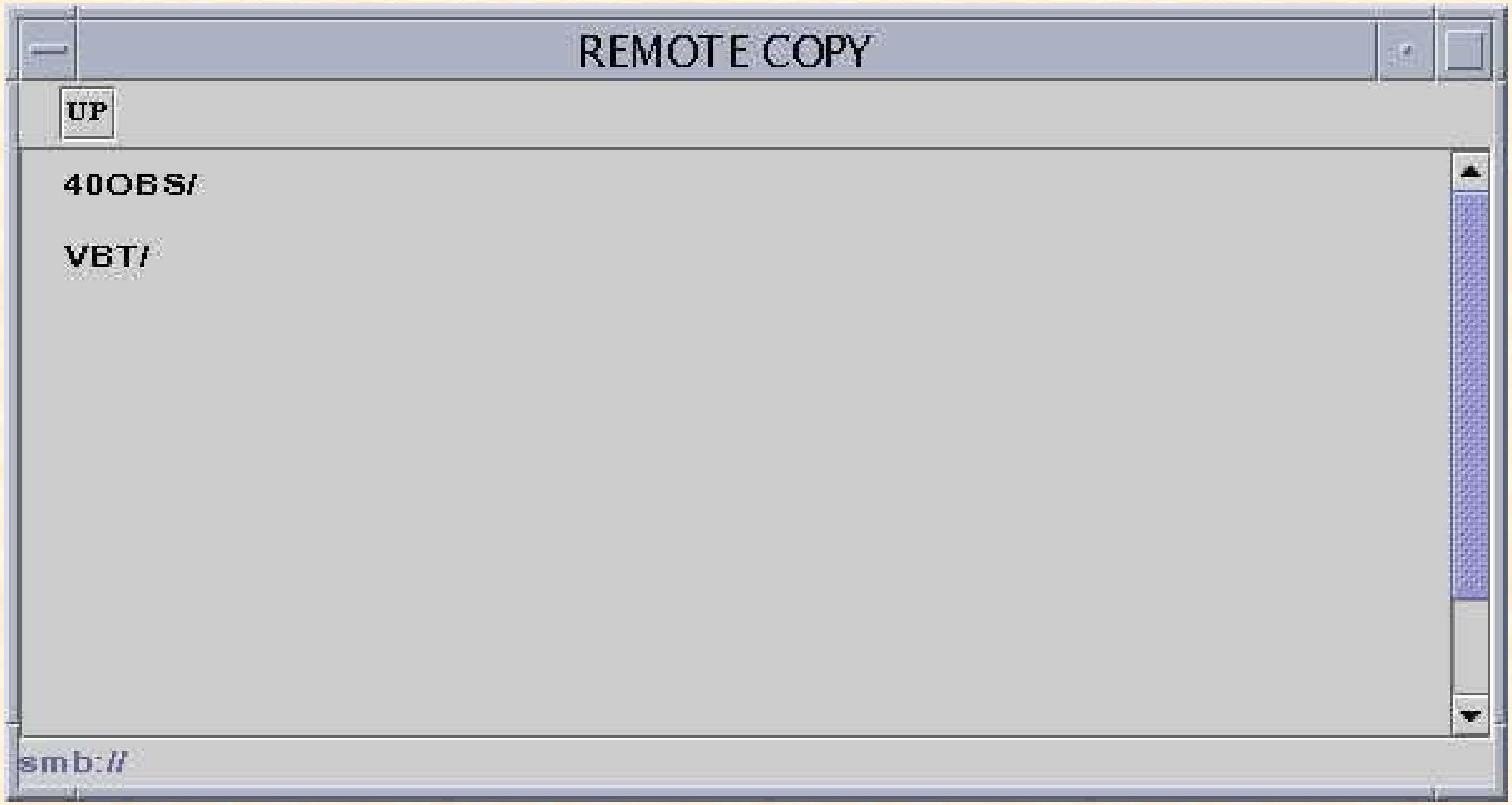
- jCIFS is Open Source client library for SMB protocol implemented in Java by the Samba team
- Created a GUI for smbclient in a Sparc5 workstation hosting Solaris 2.5.1 (with no SMB server) to connect to shares on data acquisition (windows based)
- Copy files from shares to local file system and vice versa

RemoteCopy (pc_copy)

- jCIFS version 0.6.8 used for Java 1.2 for Solaris 2.5.1
- Popup menus added instead of menu bar
- Authentication was added
- Added modules to copy files across SMB Servers

- Installed Samba servers for Solaris 2.5.1, 2.8
- Configured Samba shares in all Unix and Linux systems to facilitate easy data transfer.

Screenshots of RemoteCopy



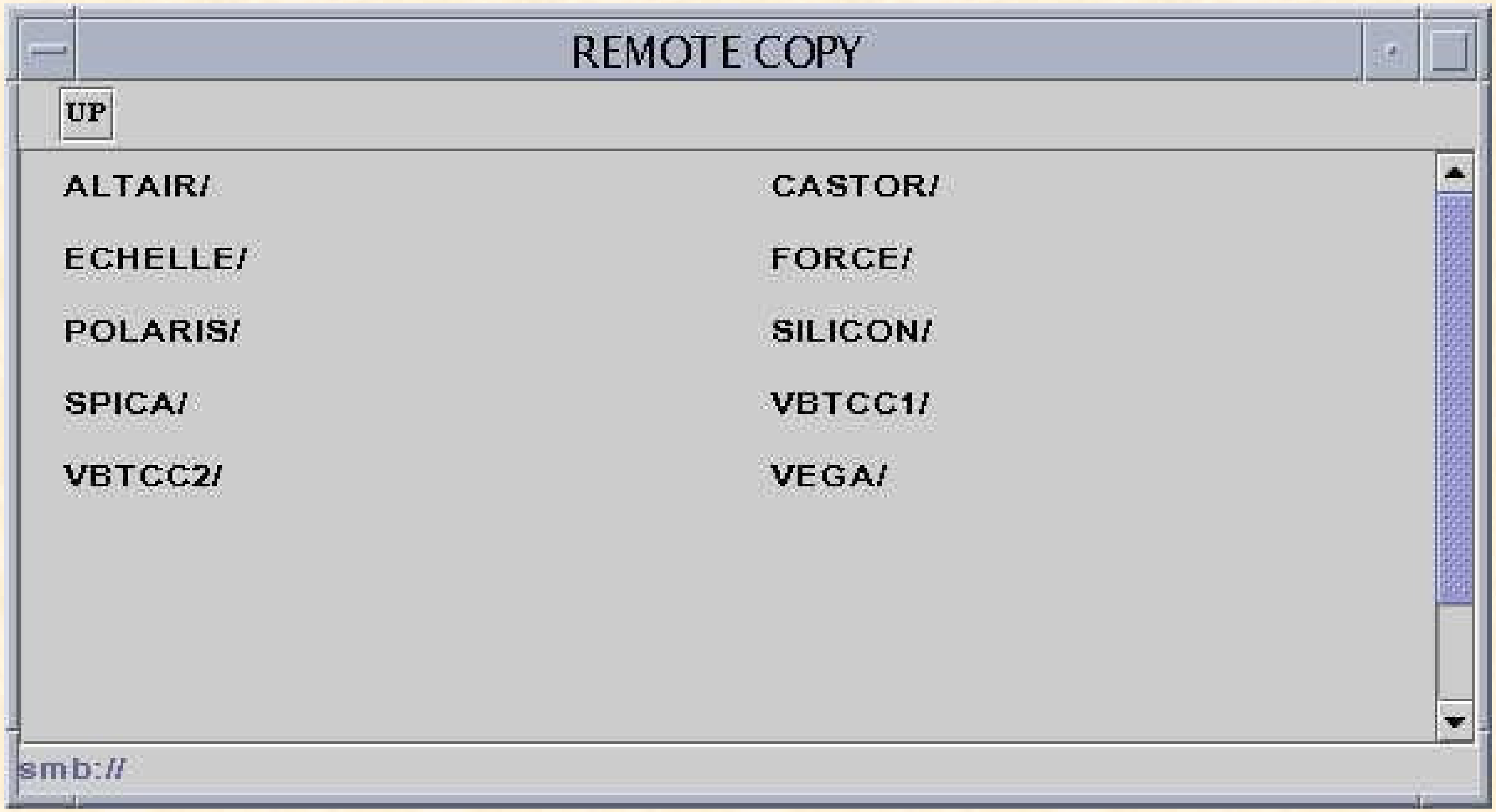
40OBS/

VBT/

smb://

- Java RemoteCopy smb://

Screenshots(contd.)

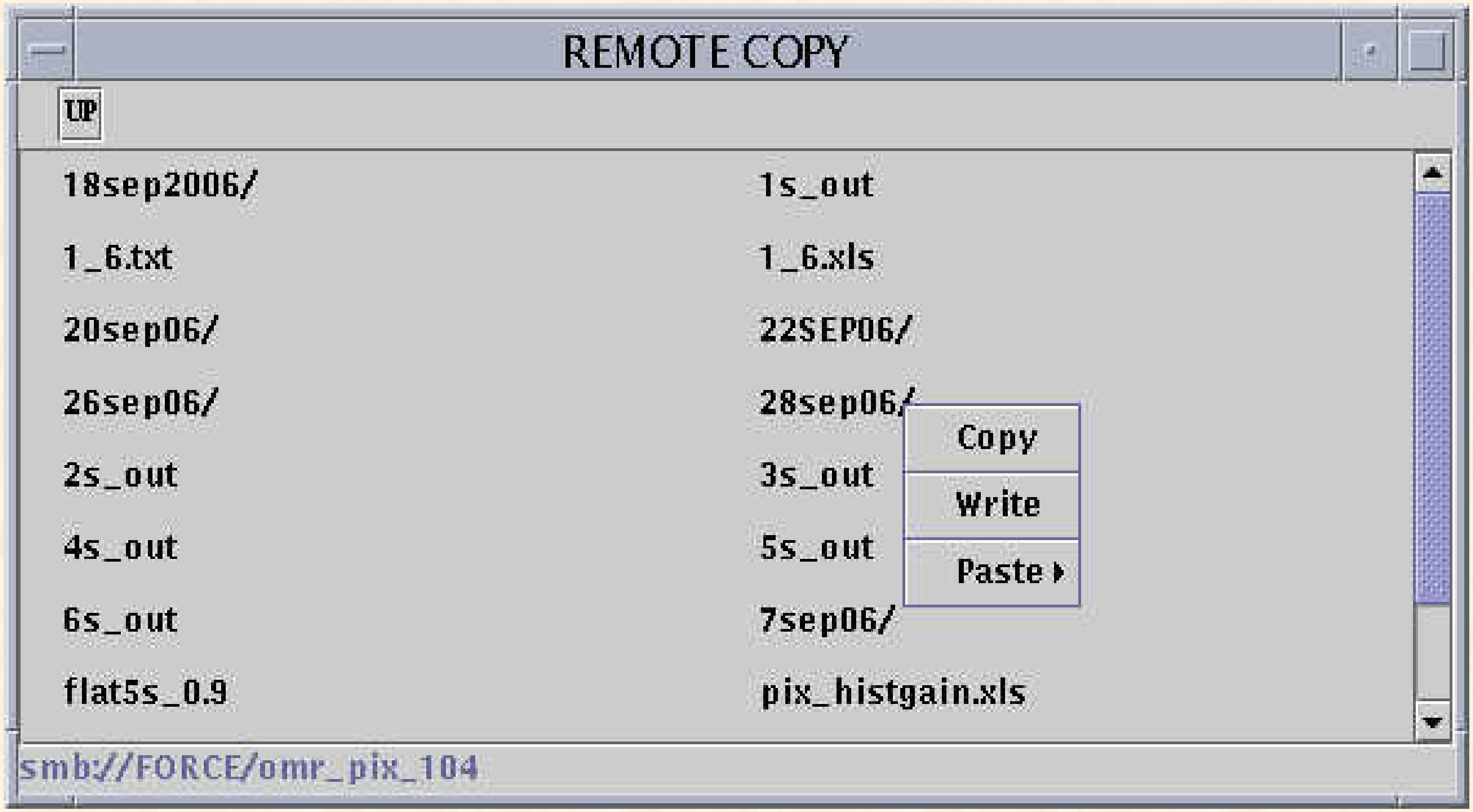


ALTAIR/ CASTOR/
ECHELLE/ FORCE/
POLARIS/ SILICON/
SPICA/ VBTCC1/
VBTCC2/ VEGA/

smb://

- SMB Servers in VBT workgroup

Screenshots(contd.)

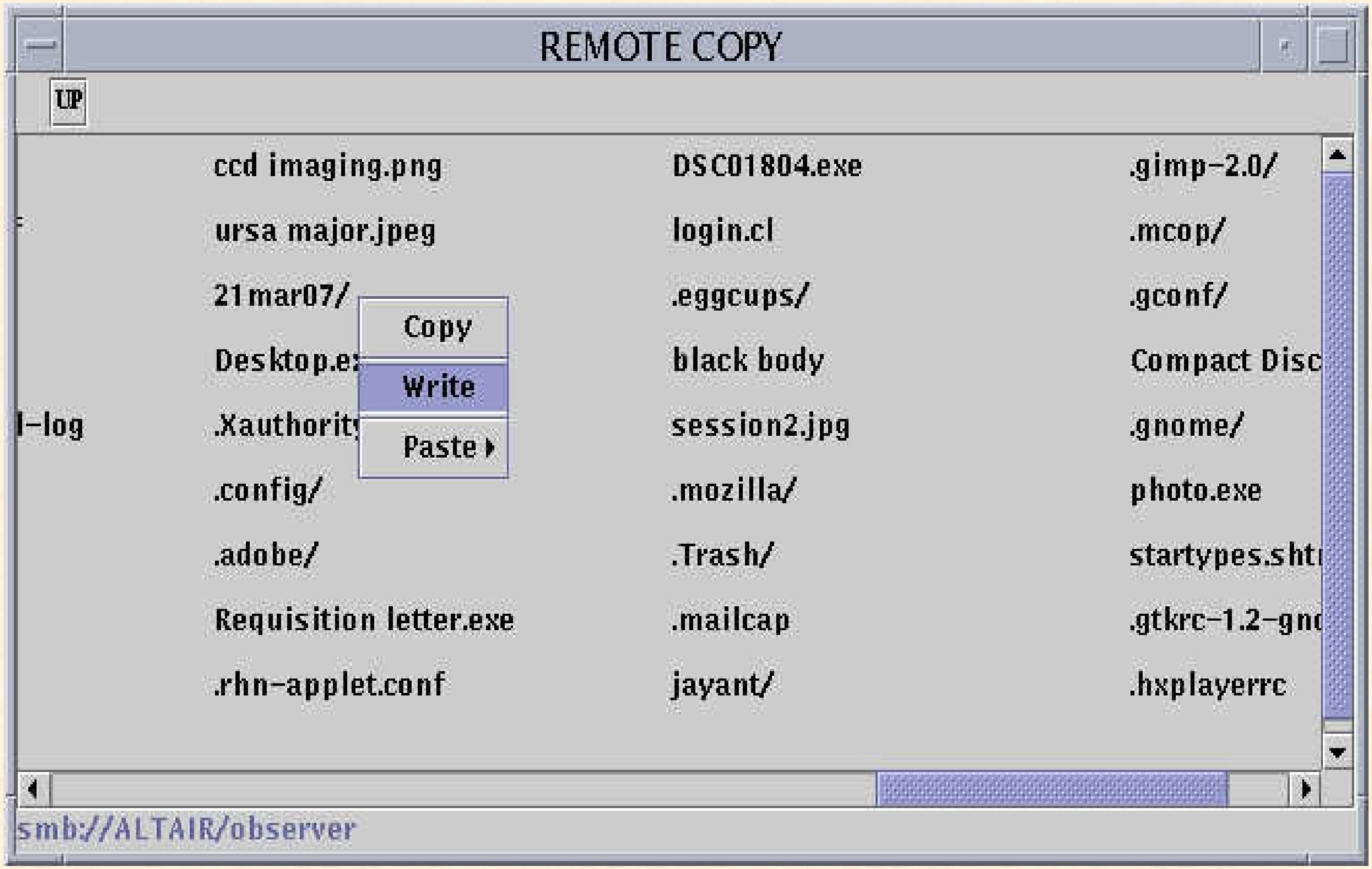


18sep2006/	1s_out
1_6.txt	1_6.xls
20sep06/	22SEP06/
26sep06/	28sep06/
2s_out	3s_out
4s_out	5s_out
6s_out	7sep06/
flat5s_0.9	pix_histgain.xls

smb://FORCE/omr_pix_104

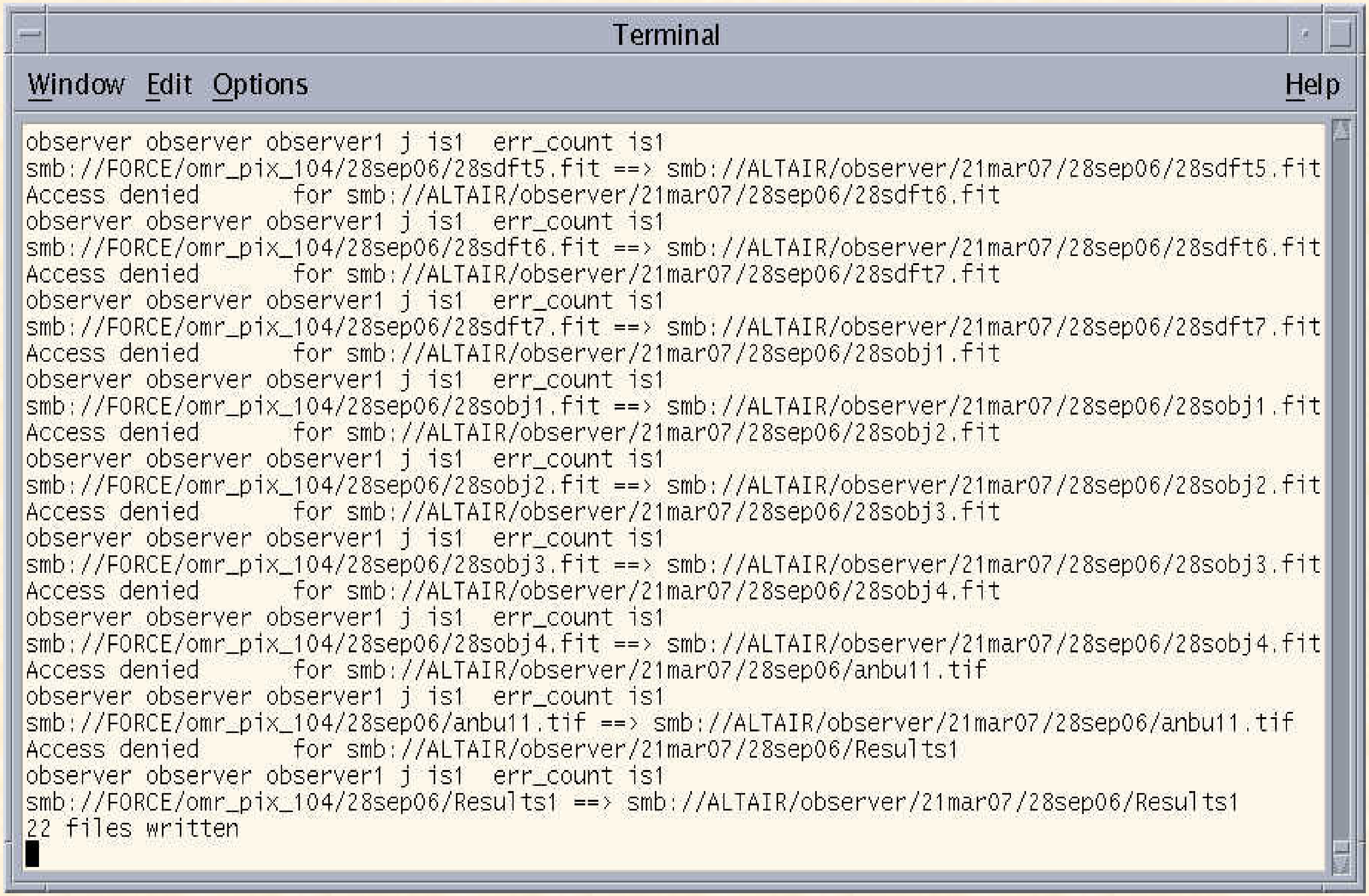
- Source directory to be copied :- smb://FORCE/omr_pix04/28sep06

Screenshots(contd.)



- Destination directory :- smb://ALTAIR/observer/21mar07

Screenshots(contd.)



```
observer observer observer1 j is1  err_count is1
smb://FORCE/omr_pix_104/28sep06/28sdft5.fit ==> smb://ALTAIR/observer/21mar07/28sep06/28sdft5.fit
Access denied       for smb://ALTAIR/observer/21mar07/28sep06/28sdft6.fit
observer observer observer1 j is1  err_count is1
smb://FORCE/omr_pix_104/28sep06/28sdft6.fit ==> smb://ALTAIR/observer/21mar07/28sep06/28sdft6.fit
Access denied       for smb://ALTAIR/observer/21mar07/28sep06/28sdft7.fit
observer observer observer1 j is1  err_count is1
smb://FORCE/omr_pix_104/28sep06/28sdft7.fit ==> smb://ALTAIR/observer/21mar07/28sep06/28sdft7.fit
Access denied       for smb://ALTAIR/observer/21mar07/28sep06/28sobj1.fit
observer observer observer1 j is1  err_count is1
smb://FORCE/omr_pix_104/28sep06/28sobj1.fit ==> smb://ALTAIR/observer/21mar07/28sep06/28sobj1.fit
Access denied       for smb://ALTAIR/observer/21mar07/28sep06/28sobj2.fit
observer observer observer1 j is1  err_count is1
smb://FORCE/omr_pix_104/28sep06/28sobj2.fit ==> smb://ALTAIR/observer/21mar07/28sep06/28sobj2.fit
Access denied       for smb://ALTAIR/observer/21mar07/28sep06/28sobj3.fit
observer observer observer1 j is1  err_count is1
smb://FORCE/omr_pix_104/28sep06/28sobj3.fit ==> smb://ALTAIR/observer/21mar07/28sep06/28sobj3.fit
Access denied       for smb://ALTAIR/observer/21mar07/28sep06/28sobj4.fit
observer observer observer1 j is1  err_count is1
smb://FORCE/omr_pix_104/28sep06/28sobj4.fit ==> smb://ALTAIR/observer/21mar07/28sep06/28sobj4.fit
Access denied       for smb://ALTAIR/observer/21mar07/28sep06/anbu11.tif
observer observer observer1 j is1  err_count is1
smb://FORCE/omr_pix_104/28sep06/anbu11.tif ==> smb://ALTAIR/observer/21mar07/28sep06/anbu11.tif
Access denied       for smb://ALTAIR/observer/21mar07/28sep06/Results1
observer observer observer1 j is1  err_count is1
smb://FORCE/omr_pix_104/28sep06/Results1 ==> smb://ALTAIR/observer/21mar07/28sep06/Results1
22 files written
```

- Console output

Further reading:-

Under Network Neighborhood | Linux Magazine

http://www.linux-mag.com/id/785/

CIFS Explained – white paper by John Kleven

http://www.codefx.com/whitepapers.htm

Implementing CIFS – online book by C.Hertel

http://ubiqx.org/cifs/index.html

Glossary

- CIFS implementations
- Pc_copy details