Server Message Block Protocol (SMB)

There are very many systems which can use the NetBIOS / NetBEUI interface or make use of the NetBIOS Frames Protocol, but perhaps one of the most important is the Server Message Block Protocol (SMB). The Server Message Block Protocol (SMB), is an application level protocol used by networking systems and operating systems such as Microsoft's Windows for Workgroups, Windows 95 / 98 / ME, LAN Manager, Windows NT, Windows 2000 and IBM's OS/2 and LAN Server, NetWare 6 and the SAMBA implementation and as such deserves special attention.

SMB is described in *Protocols for X/Open PC Interworking: SMB, Version 2* .

SMB History

In 1987 Microsoft announced the LAN Manager program and in 1988 IBM announced the OS/2 LAN Server, both use versions of the Server Message Block Protocol. Enhancements and changes to the protocol have been made and a history can be found at:

"http://samba.anu.edu.au/cifs/docs/smb-history.html" History of SMB

Dan.Shearer@unisa.edu.au

Microsoft and a number of other companies, are proposing an updated version of SMB as an internet standard The Common Internet File System (CIFS).

SMB Overview

The Server Message Block Protocol (SMB), is an application level protocol see OSI diagram .

SMB is used to implement network session control, network file and print sharing and messaging. SMB is used to provide broadly analogous functionality as the AppleTalk Session Protocol, AppleTalk Filing Protocol, Printer Access Protocol etc in the AppleTalk suite of protocols. SMB is also broadly analogous with Novell's NetWare Core Protocol (NCP). It is difficult to find a non-proprietary protocol or protocols with in the TCP/IP suite which can be compared to SMB; file sharing via FTP or NFS and network printing via LPR are examples of similar functionality.

SMB requires a transport /session protocol and the early versions of IBM's implementation were closely linked with NetBIOS. In general SMB runs either over the NetBIOS Frames Protocol (NBF), NetBIOS over TCP/IP, or NetBIOS over IPX; the most recent versions of CIFS can run directly over TCP/IP.

SMB

Server Message Block (SMB) / CIFS

/		/		\		\
NetBIOS Frames Protocol (NBF) i.e. NetBEUI i.e. NetBIOS	or	NetBIOS over TCP/IP RFC 1001 RFC 1002	or	NetBIOS over IPX	or	directly over TCP/IP

See: OSI diagram for details of the relationship between the various protocols.

SMB has inherited some of the advantages and disadvantages of NetBIOS, in particular, prior to the latest versions of CIFS it was directly linked with the NetBIOS addressing scheme.

SMB Dialects

The SMB protocol has been developed and enhanced since it was first introduced. The original version is known as the "core protocol" and is understood by systems implementing later versions which are supersets of the original. Systems using SMB negotiate which version i.e. dialect they will support.

The function SMBnegprot 0x72 is used at the beginning of a session to establish the dialect to be used. (See SMB Command Codes below.)

When packets are being sent to negotiate the dialect, a string is used to indicate which dialects are supported. So just as the use of the string "SMB" within SMB packets makes identifying such packets easier, the use of readable strings makes understanding which dialects are used easier. Below is a table giving some of the strings used to identify dialects and the terms commonly used to refer to the given dialect.

SMB dialects

string identifying dialect	Reference
PC NETWORK PROGRAM 1.0	core protocol
MICROSOFT NETWORKS 1.03	core plus dialect
MICROSOFT NETWORKS 3.0	extended 1.0 protocol
LANMAN1.0	extended 1.0 protocol, first version of full LANMAN 1.0 protocol
Windows for Workgroups 3.1a	
LM1.2X002	extended 2.0 protocol
LANMAN2.1	
NT LM 0.12	

SMB Addressing

Prior to the latest versions of CIFS, SMB uses network names which are strings of 16 bytes. In general these names are mapped directly on to NetBIOS names (see NetBIOS names above). The traditional SMB names of systems can be up to 15 characters long and are padded with blanks if necessary. The 16th byte is used to indicate whether the name refers to a server or another function.

In Microsoft networks with NT 3.x and NT 4.0 systems some names are used with NT 3.x and NT 4.0 Domains as well as for computer names. Some examples of names and use of the 16th byte are given below:

SMB Names

SMB Name	Purpose
Computername[0x00]	Workstation service
Computername[0x20]	Server service
Domainname[0x00]	Register computer in domain
Domainname[0x1C]	Domain controller

Unique NetBIOS names will map to SMB individual system names, and NetBIOS group names will map to workgroup or domain names.

Like NetBIOS names, traditional SMB names are non hierarchical and constitute a flat non-routable name space which does not scale well.

SMB on NBF

SMB on NBF datagram frames

Datagram frames
(Octets in order transmitted.)

		Data frame	Data frame
Field Name	**Length**	**DATAGRAM**	**SMB**
Length	2	0x2C	
		0x00	
Deliminator	2	0xFF	
		0xEF	
Command	1	0x08	
Data 1	1	Reserved	
Data 2	2	Reserved	
		Reserved	
XMIT Cor	2	Reserved	
		Reserved	
RSP Cor	2	Reserved	
		Reserved	
Destination Name	16	Name of receiver	
Source Name	16	Name of sender	
Optional		Datagram	SMB frame

Datagram frames
(Octets in order transmitted.)

		Data frame	Data frame
Field Name	**Length**	**DATAGRAM BROADCAST**	**SMB**
Length	2	0x2C	
		0x00	
Deliminator	2	0xFF	
		0xEF	
Command	1	0x09	
Data 1	1	Reserved	
Data 2	2	Reserved	
		Reserved	
XMIT Cor	2	Reserved	
		Reserved	
RSP Cor	2	Reserved	
		Reserved	
Destination Name	16	Reserved	
Source Name	16	Name of sender	
Optional		Datagram	SMB frame

SMB on NBF session frames

Session Data Transfer frames
(Octets in order transmitted.)

	Data frame	Data frame

		Re-synch indicator	
XMIT Cor	2	nnnn	
		nnnn	
RSP Cor	2	nnnn	
		nnnn	
Dest Num	1	Remote session num	
Source Num	1	Local session num	
Optional data		USER DATA Message from send	SMB frame

Session Data Transfer frames
(Octets in order transmitted.)

		Data frame	**Data frame**
Field Name	**Length**	**DATA ONLY LAST**	**SMB**
Length	2	0x0E	
		0x00	
Deliminator	2	0xFF	
		0xEF	
Command	1	0x16	
Data1	1	Brrrxryz	
Data2	2	Re-synch indicator	
		Re-synch indicator	
XMIT Cor	2	nnnn	
		nnnn	
RSP Cor	2	nnnn	
		nnnn	
Dest Num	1	Remote session num	
Source Num	1	Local session num	
Optional data		USER DATA Message from send	SMB frame

SMB frame header

The general format of SMB frame headers is given below:

SMB frames
(Octets in order transmitted.)

Field Name	Length	SMB
Deliminator	1	0xFF
ID	3	0x53 "S"
		0x4d "M"
		0x42 "B"
Command	1	0xNN
Error class	1	0xNN
Reserved	1	reserved
Error code	2	0xNN
		0xNN
Flags	1	0xNN
Flags 2 / Reserved	2	0xNN
		0xNN
Reserved? 12?	12	0xNN
		0xNN
		0xNN
		0xNN
		0xNN
		0xNN
		0xNN
		0xNN
		0xNN
		0xNN
		0xNN
		0xNN
authenticated resource identifier / Tree ID	2	0xNN
		0xNN
caller's Process ID	2	0xNN
		0xNN
unathenticated User ID	2	0xNN
		0xNN
Multiplex ID	2	0xNN
		0xNN
count of 16-bit fields Word count	1	0xNN
variable no of 16-bit fields byte count	2	0xNN
		0xNN
count of 8-bit fields that follow	2	0xNN
		0xNN
variable number of 8-bit fields	2	0xNN
		0xNN

SMB is very analogous to the NetWare Core Protocol (NCF); there are numerous functions available for accomplishing various tasks. There are very many SMB frames depending upon the function, all share the same header format; the second field, command, determines the function and possibly the format of the rest of the frame following the header.

SMB Command Codes

Below is a table giving some of the Core SMB commands:

Core SMB Commands

Field Name	smb_com	Description
SMBmkdir	0x00	Create directory
SMBrmdir	0x01	Delete directory
SMBopen	0x02	Open file
SMBcreate	0x03	Create file
SMBclose	0x04	Close file
SMBflush	0x05	Commit all files
SMBunlink	0x06	Delete file
SMBmv	0x07	Rename file
SMBgetatr	0x08	Get file attribute
SMBsetatr	0x09	Set file attribute
SMBread	0x0a	Read byte block
SMBwrite	0x0b	Write byte block
SMBlock	0x0c	Lock byte block
SMBunlock	0x0d	Unlock byte block
SMBmknew	0x0f	Create new file
SMBchkpth	0x10	Check directory
SMBexit	0x11	End of process
SMBlseek	0x12	LSEEK
SMBtcon	0x70	Start connection
SMBtdis	0x71	End connection
SMBnegprot	0x72	Verify dialect
SMBbskattr	0x80	Get disk attributes
SMBsearch	0x81	Search multiple files
SMBsplopen	0xc0	Create spool file
SMBsplwr	0xc1	Spool byte block
SMBsplclose	0xc2	Close spool file
SMBsplretq	0xc3	Return print queue
SMBsends	0xd0	Send message
SMBsendb	0xd1	Send broadcast
SMBfwdname	0xd2	Forward user name
SMBcancelf	0xd3	Cancel forward
SMBgetmac	0xd4	Get machine name
SMBsendstrt	0xd5	Start multi-block message
SMBsendend	0xd6	End multi-block message
SMBsendtxt	0xd7	Multi-block message text
Never valid	0xfe	Invalid
Implementation-dependant	0xff	Implementation-dependant

Below is a table giving some of the Core plus commands:

Core plus Commands

Field Name	smb_com	Description
SMBlockreadr	0x13	Lock then read data
SMBwriteunlock	0x14	Write then unlock data
SMBreadBraw	0x1a	Read block raw
SMBwriteBraw	0x1d	Write block raw

Below is a table giving some of the LANMAN 1.0 SMB commands:

LANMAN 1.0 SMB Commands

Field Name	smb_com	Description
SMBreadBmpx	0x1b	Read block multiplexed
SMBreadBs	0x1c	Read block (secondary response)
SMBwriteBmpx	0x1e	Write block multiplexed
SMBwriteBs	0x1f	Write block (secondary response)
SMBwriteC	0x20	Write complete response
SMBsetattrE	0x22	Set file attributes expanded
SMBgetattrE	0x23	Get file attributes expanded
SMBlockingX	0x24	Lock/unlock byte ranges and X
SMBtrans	0x25	Transaction (name, bytes in/out)
SMBtranss	0x26	Transaction (secondary request/response)
SMBioctl	0x27	Passes the IOCTL to the server
SMBioctls	0x28	IOCTL (secondary request/response)
SMBcopy	0x29	Copy
SMBmove	0x2a	Move
SMBecho	0x2b	Echo
SMBwriteclose	0x2c	Write and Close
SMBopenX	0x2d	Open and X
SMBreadX	0x2e	Read and X
SMBwriteX	0x2f	Write and X
SMBsesssetup	0x73	Session Set Up and X (including User Logon)
SMBtconX	0x75	Tree connect and X
SMBffirst	0x82	Find first
SMBfunique	0x83	Find unique
SMBfclose	0x84	Find close
SMBinvalid	0xfe	Invalid command

SMB Error Class

Below is a table giving some of the SMB Error class values:

SMB Error Class

Field Name	Value	Description
SUCCESS	0x00	The request was successful
ERRSRV	0x02	Error generated by the LMX server

SMB Return Codes for Error class 0x00

Below is a table giving some of the SMB Return Code Values when the Error class is 0x00:

SMB Return Code

Field Name	Value	Description
BUFFERED	0x54	The Message was buffered
LOGGED	0x55	The Message was logged
DISPLAYED	0x56	The Message was displayed

SMB Return Codes for Error class 0x02

Below is a table giving some of the SMB Return Code Values when the Error class is 0x02:

SMB Return Code

Field Name	Value	Description
ERRerror	0x01	Non-specific error code
ERRbadpw	0x02	Bad password
ERRbadtype	0x03	Reserved

Further information

Just what is SMB? V1.0 Richard Sharpe

Previous Contents Next