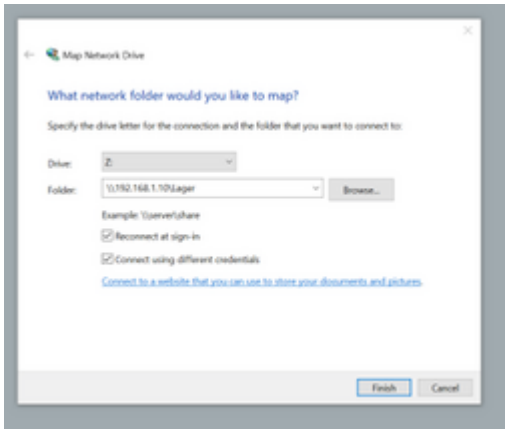
Server Message Block

By Contributors to Wikimedia projects

Published: 2003-10-26 · Archived: 2026-04-05 17:40:15 UTC



Map Network Drive dialog in Windows 10, connecting to a local SMB network drive

Server Message Block (**SMB**) is a communication protocol[1] used to share files, printers, serial ports, and miscellaneous communications between nodes on a network. On Windows, the SMB implementation consists of two vaguely named Windows services: "Server" (ID: `LanmanServer`) and "Workstation" (ID: `LanmanWorkstation`).[2] It uses NTLM or Kerberos protocols for user authentication. It also provides an authenticated inter-process communication (IPC) mechanism.

SMB was originally developed in 1983 by Barry A. Feigenbaum at IBM[3] to share access to files and printers across a network of systems running IBM PC DOS. In 1987, Microsoft and 3Com implemented SMB in LAN Manager for OS/2, at which time SMB used the NetBIOS service atop the NetBIOS Frames protocol as its underlying transport. Later, Microsoft implemented SMB in Windows NT 3.1 and has been updating it ever since, adapting it to work with newer underlying transports: TCP/IP and NetBT. SMB over QUIC was introduced in Windows Server 2022.

In 1996, Microsoft published a version of SMB 1.0[4] with minor modifications under the **Common Internet File System** (**CIFS**) moniker. CIFS was compatible with even the earliest incarnation of SMB, including LAN Manager's.[4] It supports symbolic links, hard links, and larger file size, but none of the features of SMB 2.0 and later.[4][5] Microsoft's proposal, however, remained an Internet Draft and never achieved standard status.[6] Microsoft has since discontinued the CIFS moniker but continues developing SMB and publishing subsequent specifications. Samba is a free software reimplementation of the SMB protocol and the Microsoft extensions to it.

Server Message Block (SMB) enables file sharing, printer sharing, network browsing, and inter-process communication (through named pipes) over a computer network. SMB serves as the basis for Microsoft's Distributed File System implementation.

SMB relies on the TCP and IP protocols for transport. This combination allows file sharing over complex, interconnected networks, including the public Internet. The SMB server component uses TCP port 445. SMB originally operated on NetBIOS over IEEE 802.2 - NetBIOS Frames or NBF - and over IPX/SPX, and later on NetBIOS over TCP/IP (NetBT), but Microsoft has since deprecated these protocols. On NetBT, the server component uses three TCP or UDP ports: 137 (NETBIOS Name Service), 138 (NETBIOS Datagram Service), and 139 (NETBIOS Session Service).

In Microsoft Windows, two Windows services implement SMB. The "Server" service (ID: `LanmanServer`) is in charge of serving shared resources. The "Workstation" service (ID: `LanmanWorkstation`) maintains the computer name and helps access shared resources on other computers.[2] SMB uses the Kerberos protocol to authenticate users against Active Directory on Windows domain networks. On simpler, peer-to-peer networks, SMB uses the NTLM protocol.

Windows NT 4.0 SP3 and later can digitally sign SMB messages to prevent some man-in-the-middle attacks.[7][8][9] SMB signing may be configured individually for incoming SMB connections (by the "LanmanServer" service) and outgoing SMB connections (by the "LanmanWorkstation" service). The default setting for Windows domain controllers running Windows Server 2003 and later is to not allow unsigned incoming connections.[10] As such, earlier versions of Windows that do not support SMB signing from the get-go (including Windows 9x) cannot connect to a Windows Server 2003 domain controller.[8]

SMB supports opportunistic locking (see below) on files in order to improve performance. Opportunistic locking support has changed with each Windows Server release.

Opportunistic locking

[edit]

In the SMB protocol, opportunistic locking is a mechanism designed to improve performance by controlling caching of network files by the client.[11] Unlike traditional locks, opportunistic lock (OpLocks) are not strictly file locking or used to provide mutual exclusion.

There are four types of opportunistic locks.

Batch Locks
: Batch OpLocks were created originally to support a particular behavior of DOS batch file execution operation in which the file is opened and closed many times in a short period, which is a performance problem. To solve this, a client may ask for an OpLock of type "batch". In this case, the client delays sending the close request and if a subsequent open request is given, the two requests cancel each other.[12]

Level-1 OpLocks / Exclusive Locks
: When an application opens in "shared mode" a file hosted on an SMB server which is not opened by any other process (or other clients) the client receives an **exclusive OpLock** from the server. This means that the client may now assume that it is the only process with access to this particular file, and the client may now cache all changes to the file before committing it to the server. This is a performance improvement, since fewer round-trips are required in order to read and write to the file. If another client/process tries to

open the same file, the server sends a message to the client (called a *break* or *revocation*) which invalidates the exclusive lock previously given to the client. The client then flushes all changes to the file.

Level-2 OpLocks

If an exclusive OpLock is held by a client and a locked file is opened by a third party, the client has to relinquish its exclusive OpLock to allow the other client's write/read access. A client may then receive a "Level 2 OpLock" from the server. A Level 2 OpLock allows the caching of read requests but excludes write caching.

Filter OpLocks

Added in Windows NT 4.0, Filter Oplocks are similar to Level 2 OpLocks but prevent sharing-mode violations between file open and lock reception. Microsoft advises use of Filter OpLocks only where it is important to allow multiple readers and Level 2 OpLocks in other circumstances. Clients holding an OpLock do not really hold a lock on the file, instead they are notified via a *break* when another client wants to access the file in a way inconsistent with their lock. The other client's request is held up while the break is being processed.

Breaks

In contrast with the SMB protocol's "standard" behavior, a break request may be sent *from* server *to* client. It informs the client that an OpLock is no longer valid. This happens, for example, when another client wishes to open a file in a way that invalidates the OpLock. The first client is then sent an OpLock break and required to send all its local changes (in case of batch or exclusive OpLocks), if any, and acknowledge the OpLock break. Upon this acknowledgment the server can reply to the second client in a consistent manner.

The use of the SMB protocol has often correlated with a significant increase in broadcast traffic on a network. However the SMB itself does not use broadcasts—the broadcast problems commonly associated with SMB actually originate with the NetBIOS service location protocol.[*clarification needed*] By default, a Microsoft Windows NT 4.0 server used NetBIOS to advertise and locate services. NetBIOS functions by broadcasting services available on a particular host at regular intervals. While this usually makes for an acceptable default in a network with a smaller number of hosts, increased broadcast traffic can cause problems as the number of hosts on the network increases. The implementation of name resolution infrastructure in the form of Windows Internet Naming Service (WINS) or Domain Name System (DNS) resolves this problem. WINS was a proprietary implementation used with Windows NT 4.0 networks, but brought about its own issues and complexities in the design and maintenance of a Microsoft network.

Since the release of Windows 2000, the use of WINS for name resolution has been deprecated by Microsoft, with hierarchical Dynamic DNS now configured as the default name resolution protocol for all Windows operating systems. Resolution of (short) NetBIOS names by DNS requires that a DNS client expand short names, usually by appending a connection-specific DNS suffix to its DNS lookup queries. WINS can still be configured on clients as a secondary name resolution protocol for interoperability with legacy Windows environments and applications. Further, Microsoft DNS servers can forward name resolution requests to legacy WINS servers in order to support name resolution integration with legacy (pre-Windows 2000) environments that do not support DNS.

Network designers have found that latency has a significant impact on the performance of the SMB 1.0 protocol, that it performs more poorly than other protocols like FTP. Monitoring reveals a high degree of "chattiness" and a

disregard of network latency between hosts.[13] For example, a VPN connection over the Internet will often introduce network latency. Microsoft has explained that performance issues come about primarily because SMB 1.0 is a block-level rather than a streaming protocol, that was originally designed for small LANs; it has a block size that is limited to 64K, SMB signing creates an additional overhead and the TCP window size is not optimized for WAN links.[14] Solutions to this problem include the updated SMB 2.0 protocol,[15] Offline Files, TCP window scaling and WAN optimization devices from various network vendors that cache and optimize SMB 1.0[16] and 2.0.[17]

Barry Feigenbaum originally designed SMB at IBM in early 1983 with the aim of turning DOS INT 21h local file access into a networked file system.[3] Microsoft made considerable modifications to the most commonly used version and included SMB support in the LAN Manager operating system it had started developing for OS/2 with 3Com around 1990.[18][19][20] Microsoft continued to add features to the protocol in Windows for Workgroups (c. 1992) and in later versions of Windows. LAN Manager authentication was implemented based on the original legacy SMB specification's requirement to use IBM "LAN Manager" passwords, but implemented DES in a flawed manner that allowed passwords to be cracked.[21] Later, Kerberos authentication was also added. The Windows domain logon protocols initially used 40-bit encryption outside of the United States, because of export restrictions on stronger 128-bit encryption[22] (subsequently lifted in 1996 when President Bill Clinton signed Executive Order 13026[23]).

SMB 1.0 (or SMB1) was originally designed to run on NetBIOS Frames (NetBIOS over IEEE 802.2). Since then, it has been adapted to NetBIOS over IPX/SPX (NBX), and NetBIOS over TCP/IP (NetBT). Also, since Windows 2000, SMB runs on TCP using TCP port 445, a feature known as "direct host SMB".[24] There is still a thin layer (similar to the Session Message packet of NetBT's Session Service) between SMB and TCP.[24] Windows Server 2003, and legacy NAS devices use SMB1 natively.

SMB1 is an extremely chatty protocol, which is not such an issue on a local area network (LAN) with low latency. It becomes very slow on wide area networks (WAN) as the back and forth handshake of the protocol magnifies the inherent high latency of such a network. Later versions of the protocol reduced the high number of handshake exchanges. One approach to mitigating the inefficiencies in the protocol is to use WAN optimization products such as those provided by Riverbed, Silver Peak, or Cisco. A better approach is to upgrade to a later version of SMB. This includes upgrading both NAS devices as well as Windows Server 2003. The most effective method to identify SMB1 traffic is with a network analyzer tool, such as Wireshark. Microsoft also provides an auditing tool in Windows Server 2016 to track down devices that use SMB1.[25]

Microsoft marked SMB1 as deprecated in June 2013.[26] Windows Server 2016 and Windows 10 version 1709 do not have SMB1 installed by default.[27]

In 1996, when Sun Microsystems announced WebNFS,[28] Microsoft launched an initiative to rename SMB to Common Internet File System (CIFS)[3] and added more features, including support for symbolic links, hard links, larger file sizes, and an initial attempt at supporting direct connections over TCP port 445 without requiring NetBIOS as a transport (a largely experimental effort that required further refinement). Microsoft submitted some partial specifications as Internet Drafts to the IETF.[6] These submissions have since expired.

Microsoft introduced a new version of the protocol (SMB 2.0 or SMB2) in 2006 with Windows Vista and Windows Server 2008.[29] Although the protocol is proprietary, its specification has been published to allow other systems to interoperate with Microsoft operating systems that use the new protocol.[30]

SMB2 reduces the 'chattiness' of the SMB 1.0 protocol by reducing the number of commands and subcommands from over a hundred to just nineteen.[13] It has mechanisms for pipelining, that is, sending additional requests before the response to a previous request arrives, thereby improving performance over high-latency links. It adds the ability to compound multiple actions into a single request, which significantly reduces the number of round-trips the client needs to make to the server, improving performance as a result.[13] SMB1 also has a compounding mechanism—known as AndX—to compound multiple actions, but Microsoft clients rarely use AndX.[*citation needed*] It also introduces the notion of "durable file handles": these allow a connection to an SMB server to survive brief network outages, as are typical in a wireless network, without having to incur the overhead of re-negotiating a new session.

SMB2 includes support for symbolic links. Other improvements include caching of file properties, improved message signing with HMAC SHA-256 hashing algorithm and better scalability by increasing the number of users, shares and open files per server among others.[13] The SMB1 protocol uses 16-bit data sizes, which amongst other things, limits the maximum block size to 64K. SMB2 uses 32- or 64-bit wide storage fields, and 128 bits in the case of file-handles, thereby removing previous constraints on block sizes, which improves performance with large file transfers over fast networks.[13]

Windows Vista/Server 2008 and later operating systems use SMB2 when communicating with other machines also capable of using SMB2. SMB1 continues in use for connections with older versions of Windows, as well various vendors' NAS solutions. Samba 3.5 also includes experimental support for SMB2.[31] Samba 3.6 fully supports SMB2, except the modification of user quotas using the Windows quota management tools.[32]

When SMB2 was introduced it brought a number of benefits over SMB1 for third party implementers of SMB protocols. SMB1, originally designed by IBM, was reverse engineered, and later became part of a wide variety of non-Windows operating systems such as Xenix, OS/2 and VMS (Pathworks). X/Open standardized it partially; Microsoft had submitted Internet-Drafts describing SMB2 to the IETF, partly in response to formal IETF standardization of version 4 of the Network File System in December 2000 as IETF RFC 3010;[33] however, those SMB-related Internet-Drafts expired without achieving any IETF standards-track approval or any other IETF endorsement. (See http://ubiqx.org/cifs/Intro.html for historical detail.) SMB2 is also a relatively clean break with the past. Microsoft's SMB1 code has to work with a large variety of SMB clients and servers. SMB1 features many versions of information for commands (selecting what structure to return for a particular request) because features such as Unicode support were retro-fitted at a later date. SMB2 involves significantly reduced compatibility-testing for implementers of the protocol. SMB2 code has considerably less complexity since far less variability exists (for example, non-Unicode code paths become redundant as SMB2 requires Unicode support).

Apple migrated to SMB2 (from their own Apple Filing Protocol, now legacy) starting with OS X 10.9 "Mavericks".[34] This transition was fraught with compatibility problems though.[35][36] Non-default support for SMB2 appeared in fact in OS X 10.7, when Apple abandoned Samba in favor of its own SMB implementation called SMBX[34] after Samba adopted GPLv3.[37][38]

The Linux kernel's CIFS client file system has SMB2 support since version 3.7.[39]

SMB 2.1, introduced with Windows 7 and Server 2008 R2, introduced minor performance enhancements with a new opportunistic locking mechanism.[40]

SMB 3.0 (previously named SMB 2.2)[41] was introduced with Windows 8[41] and Windows Server 2012.[41] It brought several significant changes that are intended to add functionality and improve SMB2 performance,[42] notably in virtualized data centers:

- the SMB Direct Protocol (SMB over remote direct memory access [RDMA])
- SMB Multichannel (multiple connections per SMB session),[43][44]
- SMB Transparent Failover[45][46]

It also introduces several security enhancements, such as end-to-end encryption and a new AES based signing algorithm.[47][48]

SMB 3.0.2 (known as 3.02 at the time) was introduced with Windows 8.1 and Windows Server 2012 R2;[49][50] in those and later releases, the earlier SMB version 1 can be optionally disabled to increase security.[51][52]

SMB 3.1.1 was introduced with Windows 10 and Windows Server 2016.[53] This version supports AES-128 GCM encryption in addition to AES-128 CCM encryption added in SMB3, and implements pre-authentication integrity check using SHA-512 hash. SMB 3.1.1 also makes secure negotiation mandatory when connecting to clients using SMB versions that support it.[54]

The specifications for the SMB are proprietary and were initially closed, thereby forcing other vendors and projects to reverse-engineer the protocol to interoperate with it. The SMB 1.0 protocol was eventually published some time after it was reverse engineered, whereas the SMB 2.0 protocol was made available from Microsoft's Open Specifications Developer Center from the outset.[55]

Third-party implementations

[edit]

This section needs to be **updated**. Please help update this article to reflect recent events or newly available information. *(April 2016)*

In 1991, Andrew Tridgell started the development of Samba, a free-software re-implementation (using reverse engineering) of the SMB/CIFS networking protocol for Unix-like systems, initially to implement an SMB server to allow PC clients running the DEC Pathworks client to access files on SunOS machines.[3][56] Because of the importance of the SMB protocol in interacting with the widespread Microsoft Windows platform, Samba became a popular free software implementation of a compatible SMB client and server to allow non-Windows operating systems, such as Unix-like operating systems, to interoperate with Windows.

As of version 3 (2003), Samba provides file and print services for Microsoft Windows clients and can integrate with a Windows NT 4.0 server domain, either as a Primary Domain Controller (PDC) or as a domain member. Samba4 installations can act as an Active Directory domain controller or member server, at Windows 2008 domain and forest functional levels.[57]

Package managers in Linux distributions can search for the *cifs-utils* package. The package is from the Samba maintainers.

NSMB (Netsmb and SMBFS) is a family of in-kernel SMB client implementations in BSD operating systems. It was first contributed to FreeBSD 4.4 by Boris Popov, and is now found in a wide range of other BSD systems including NetBSD and macOS.[58] The implementations have diverged significantly ever since.[59]

The macOS version of NSMB is notable for its now-common scheme of representing symlinks. This "Minshall-French" format shows symlinks as textual files with a `.symlink` extension and a `Xsym\n` magic number, always 1067 bytes long. This format is also used for storing symlinks on native SMB servers or unsupported filesystems. Samba supports this format with an `mfsymlink` option.[60] Docker on Windows also seems to use it.[*citation needed*]

NQ is a family of portable SMB client and server implementations developed by Visuality Systems.[61] The NQ family comprises an embedded SMB stack (written in C), a Pure Java SMB Client, and a storage SMB server implementation. All versions support the SMB 3.1.1 dialect. They support Linux, Windows CE, iOS, Android, VxWorks and other operating systems.

MoSMB is a user space SMB implementation for Linux. It supports SMB 2.x and SMB 3.x. Key features include Cloud-scale Active-Active Scale-out Clusters, SMB Direct (RDMA), SMB Multichannel, Transparent Failover and Continuous Availability. MoSMB also supports Amazon S3 object storage as storage backend in addition to POSIX file systems such as ext4, ZFS, Lustre, Ceph, etc.[62]

Fusion File Share by Tuxera is a proprietary SMB server implementation developed by Tuxera that can be run either in kernel or user space.[63] It supports SMB 3.1.1 and all previous versions, additionally advanced SMB features like continuous availability (persistent handles) scale-out, RDMA (SMB Direct), SMB multichannel, transparent compression, shadow copy.

Likewise developed a CIFS/SMB implementation (versions 1.0, 2.0, 2.1 and SMB 3.0) in 2009 that provided a multiprotocol, identity-aware platform for network access to files used in OEM storage products built on Linux/Unix based devices. The platform could be used for traditional NAS, Cloud Gateway, and Cloud Caching devices for providing secure access to files across a network. Likewise was purchased by EMC Isilon in 2012.

KSMBD is an open source in-kernel CIFS/SMB server implementation for the Linux kernel. Compared to user-space implementations, it provides better performance and makes it easier to implement some features such as SMB Direct. It supports SMB 3.1.1 and previous versions.

Over the years, there have been many security vulnerabilities in Microsoft's implementation of the protocol or components on which it directly relies.[64][65] Other vendors' security vulnerabilities lie primarily in a lack of support for newer authentication protocols like NTLMv2 and Kerberos in favor of protocols like NTLMv1,

LanMan, or plaintext passwords. Real-time attack tracking[66] shows that SMB is one of the primary attack vectors for intrusion attempts,[67] for example the 2014 Sony Pictures attack,[68] and the WannaCry ransomware attack of 2017.[69] In 2020, two SMB high-severity vulnerabilities were disclosed and dubbed as SMBGhost (CVE-2020-0796) and SMBleed (CVE-2020-1206), which when chained together can provide RCE (Remote Code Execution) privilege to the attacker.[70]

- List of products that support SMB
- Administrative share
- Shared file access
- AppleTalk
- Network File System (protocol)
- Remote File System
- WebDAV
- Uniform Naming Convention
- Network Neighborhood

1. ^ *"Microsoft SMB Protocol and CIFS Protocol Overview"*. *Microsoft*. *October 22, 2009. Archived from the original on August 2, 2016. Retrieved April 10, 2019.*
2. ^ Jump up to: **a** **b** *"Lan Manager Networking Concepts"*. *Support. Microsoft. Archived from the original on December 30, 2012. Retrieved September 18, 2014.*
3. ^ Jump up to: **a** **b** **c** **d** *Tridgell, Andrew. "Myths About Samba". Archived from the original on October 20, 2017. Retrieved January 3, 2016.*
4. ^ Jump up to: **a** **b** **c** *"Common Internet File System". Windows 2000 Web and Application Services Technical Overview. Microsoft. 18 July 2012. Archived from the original on 30 January 2022. Retrieved 30 January 2022 – via Microsoft Docs.*
5. ^ *Coulter, David; Satran, Michael; Batchelor, Drew (8 January 2021). "Microsoft SMB Protocol and CIFS Protocol Overview". Windows App Development. Microsoft. Archived from the original on 28 January 2022. Retrieved 30 January 2022 – via Microsoft Docs.*
6. ^ Jump up to: **a** **b** See:
    - *Heizer, I.; Leach, P.; Perry, D. (June 13, 1996). "Common Internet File System Protocol (CIFS/1.0)". Archived from the original on August 8, 2019.*
    - *Leach, Paul J.; Naik, Dilip C. (January 3, 1997). "CIFS Logon and Pass Through Authentication". Archived from the original on May 31, 2024. Retrieved December 10, 2017.*
    - *Leach, Paul J.; Naik, Dilip C. (January 10, 1997). "CIFS/E Browser Protocol". Archived from the original on May 31, 2024. Retrieved December 10, 2017.*
    - *Leach, Paul J.; Naik, Dilip C. (January 31, 1997). "CIFS Printing Specification". Archived from the original on May 31, 2024. Retrieved December 10, 2017.*
    - *Leach, Paul J.; Naik, Dilip C. (February 26, 1997). "CIFS Remote Administration Protocol". Archived from the original on May 31, 2024. Retrieved December 10, 2017.*
    - *Leach, Paul J.; Naik, Dilip C. (December 19, 1997). "A Common Internet File System (CIFS/1.0) Protocol". Archived from the original on May 31, 2024. Retrieved December 10, 2017.*

7. ^ *"Overview of Server Message Block signing". Windows Server troubleshooting. Microsoft. 24 November 2021. Archived from the original on 29 January 2022. Retrieved 29 January 2022 – via Microsoft Docs.*
8. ^ Jump up to: **a b** *Johansson, Jesper M. (20 May 2005). "How to Shoot Yourself in the Foot with Security, Part 1". Security Guidance. Microsoft. Archived from the original on 19 October 2018. Retrieved 19 October 2018 – via Microsoft Docs. "This article addresses [...] Server Message Block (SMB) message signing."*
9. ^ *Barreto, Jose (1 December 2010). "The Basics of SMB Signing (covering both SMB1 and SMB2)". Jose Barreto's Blog Archive. Microsoft. Archived from the original on 2 December 2012 – via Microsoft Docs. "This security mechanism in the SMB protocol helps avoid issues like tampering of packets and "man in the middle" attacks. [...] SMB signing is available in all currently supported versions of Windows, but it's only enabled by default on Domain Controllers. This is recommended for Domain Controllers because SMB is the protocol used by clients to download Group Policy information. SMB signing provides a way to ensure that the client is receiving genuine Group Policy."*
10. ^ *"MSKB887429: Overview of Server Message Block signing". Microsoft. November 30, 2007. Archived from the original on November 20, 2010. Retrieved October 24, 2012. "By default, SMB signing is required for incoming SMB sessions on Windows Server 2003-based domain controllers."*
11. ^ *"Opportunistic Locks". Microsoft. May 31, 2018. Archived from the original on August 19, 2021. Retrieved August 19, 2021.*
12. ^ *"All About Opportunistic Locking". Sphere IT. 2014. Archived from the original on August 19, 2021. Retrieved August 19, 2021.*
13. ^ Jump up to: **a b c d e** *Jose Barreto (December 9, 2008). "SMB2, a Complete Redesign of the Main Remote File Protocol for Windows". Microsoft Server & Management Blogs. Archived from the original on January 12, 2013. Retrieved November 1, 2009.*
14. ^ *Neil Carpenter (October 26, 2004). "SMB/CIFS Performance Over WAN Links". Microsoft. Archived from the original on February 13, 2020. Retrieved February 13, 2020.*
15. ^ *"What's New in SMB in Windows Server". Microsoft. 31 August 2016. Archived from the original on February 11, 2017. Retrieved 6 February 2017.*
16. ^ *Mark Rabinovich, Igor Gokhman. "CIFS Acceleration Techniques" (PDF). Storage Developer Conference, SNIA, Santa Clara 2009. Archived (PDF) from the original on September 30, 2020. Retrieved July 6, 2020.*
17. ^ *Mark Rabinovich. "Accelerating SMB2" (PDF). Storage Developer Conference, SNIA, Santa Clara 2011. Archived (PDF) from the original on May 31, 2024. Retrieved July 6, 2020.*
18. ^ *Speed, Richard. "Have to use SMB 1.0? Windows 10 April 2018 Update says NO". theregister.com. Archived from the original on 18 February 2023. Retrieved 18 February 2023.*
19. ^ *guenni (15 June 2017). "Microsoft plans to deactivate SMBv1 in Windows 10 V1709". Born's Tech and Windows World. Archived from the original on 18 February 2023. Retrieved 18 February 2023.*
20. ^ *Giret, Laurent (20 April 2022). "Microsoft Gets Ready to Disable SMB1 Protocol on Windows 11". Thurrott.com. Archived from the original on 31 May 2024. Retrieved 18 February 2023.*
21. ^ *Christopher Hertel (1999). "SMB: The Server Message Block Protocol". Archived from the original on March 10, 2010. Retrieved November 1, 2009.*
22. ^ *"Description of Microsoft Windows Encryption Pack 1". Microsoft. November 1, 2006. Archived from the original on October 2, 2009. Retrieved November 1, 2009.*

23. ^ *"US Executive Order 13026"* (PDF). *United States Government*. 1996. *Archived* (PDF) from the original on October 10, 2009. Retrieved November 1, 2009.
24. ^ Jump up to: **a** **b** *"Direct hosting of SMB over TCP/IP"*. *Microsoft*. October 11, 2007. *Archived* from the original on March 26, 2011. Retrieved November 1, 2009.
25. ^ Kyttle, Ralph (13 May 2017). *"SMB1 – Audit Active Usage using Message Analyzer"*. Microsoft TechNet. Microsoft. *Archived* from the original on March 28, 2019. Retrieved 28 March 2019.
26. ^ *"The Deprecation of SMB1 – You should be planning to get rid of this old SMB dialect – Jose Barreto's Blog"*. blogs.technet.microsoft.com. 21 April 2015. *Archived* from the original on May 21, 2017. Retrieved 2019-10-09.
27. ^ *"SMBv1 is not installed by default in Windows 10 Fall Creators Update and Windows Server, version 1709 and later versions"*. support.microsoft.com. *Archived* from the original on October 10, 2019. Retrieved 2019-10-09.
28. ^ *"WebNFS - Technical Overview"*. Archived from *the original* on 2007-05-18.
29. ^ Navjot Virk and Prashanth Prahalad (March 10, 2006). *"What's new in SMB in Windows Vista"*. Chk Your Dsks. *Microsoft*. Archived from *the original* on May 5, 2006. Retrieved May 1, 2006.
30. ^ *Server Message Block (SMB) Protocol Versions 2 and 3*. Windows Protocols. Open Specifications (Technical report). *Microsoft Docs*. *Microsoft*. MS-SMB2. Retrieved 2020-11-29.
31. ^ *"Samba 3.5.0 Available for Download"*. *Archived* from the original on July 24, 2011. Retrieved July 8, 2011.
32. ^ *"Samba 3.6.0 Available for Download"*. *Archived* from the original on September 24, 2011. Retrieved August 10, 2011.
33. ^ *NFS version 4 Protocol*. *IETF*. December 2000. *doi:10.17487/RFC3010*. *RFC 3010*.
34. ^ Jump up to: **a** **b** Eran, Daniel (June 11, 2013). *"Apple shifts from AFP file sharing to SMB2 in OS X 10.9 Mavericks"*. Appleinsider.com. *Archived* from the original on February 12, 2017. Retrieved January 12, 2014.
35. ^ Vaughan, Steven J. (October 28, 2013). *"Mavericks' SMB2 problem and fixes"*. ZDNet. *Archived* from the original on January 5, 2014. Retrieved January 12, 2014.
36. ^ MacParc. *"10.9: Switch the SMB stack to use SMB1 as default"*. Mac OS X Hints. macworld.com. *Archived* from the original on January 12, 2014. Retrieved January 12, 2014.
37. ^ Topher Kessler (March 23, 2011). *"Say adios to Samba in OS X"*. CNET. *Archived* from the original on January 15, 2014. Retrieved January 12, 2014.
38. ^ Thom Holwerda (March 26, 2011). *"Apple Ditches SAMBA in Favour of Homegrown Replacement"*. *Archived* from the original on November 2, 2013. Retrieved January 12, 2014.
39. ^ *"Linux 3.7 - Linux Kernel Newbies"*. *Archived* from the original on September 11, 2016. Retrieved September 4, 2016.
40. ^ *"Implementing an End-User Data Centralization Solution"*. *Microsoft*. October 21, 2009. pp. 10–11. *Archived* from the original on September 6, 2010. Retrieved November 2, 2009.
41. ^ Jump up to: **a** **b** **c** Jeffrey Snover (April 19, 2012). *"SMB 2.2 is now SMB 3.0"*. Windows Server Blog. *Archived* from the original on July 8, 2020. Retrieved July 6, 2020.
42. ^ Chelsio Communications (2 April 2013). *"40G SMB Direct"*. *Archived* from the original on September 7, 2013. Retrieved June 18, 2013.

43. ^ *Jose Barreto (October 19, 2012). "SNIA Tutorial on the SMB Protocol" (PDF). Storage Networking Industry Association. Archived (PDF) from the original on June 3, 2013. Retrieved November 28, 2012.*
44. ^ *Thomas Pfenning. "The Future of File Protocols: SMB 2.2 in the Datacenter" (PDF). Archived from the original (PDF) on 2012-07-20.*
45. ^ *Joergensen, Claus (2012-06-07). "SMB Transparent Failover – making file shares continuously available". Microsoft TechNet. Archived from the original on January 11, 2019. Retrieved February 1, 2017.*
46. ^ *Savill, John (2012-08-21). "New Ways to Enable High Availability for File Shares". Windows IT Pro. Archived from the original on November 27, 2016. Retrieved February 1, 2017.*
47. ^ *"SMB Security Enhancements". Microsoft Technet. January 15, 2014. Archived from the original on October 9, 2014. Retrieved June 18, 2014.*
48. ^ *Jose Barreto (May 5, 2013). "Updated Links on Windows Server 2012 File Server and SMB 3.0". Microsoft TechNet Server & Management Blogs. Archived from the original on August 3, 2016. Retrieved August 14, 2016.*
49. ^ *Jose Barreto (July 7, 2014). "Updated Links on Windows Server 2012 R2 File Server and SMB 3.02". Microsoft TechNet Server & Management Blogs. Archived from the original on August 26, 2016. Retrieved August 14, 2016.*
50. ^ *Jose Barreto (December 12, 2013). "Storage Developer Conference – SDC 2013 slides now publicly available. Here are the links to Microsoft slides…". Microsoft TechNet Server & Management Blogs. Archived from the original on August 26, 2016. Retrieved August 14, 2016.*
51. ^ *Eric Geier (December 5, 2013). "WindowsNetworking.com: Improvements in the SMB 3.0 and 3.02 Protocol Updates". WindowsNetworking.com. Archived from the original on April 9, 2015. Retrieved April 6, 2015.*
52. ^ *Jose Barreto (April 30, 2015). "SMB3 Networking Links for Windows Server 2012 R2". Microsoft TechNet Server & Management Blogs. Archived from the original on August 26, 2016. Retrieved August 14, 2016.*
53. ^ *Jose Barreto (May 5, 2015). "What's new in SMB 3.1.1 in the Windows Server 2016 Technical Preview 2". Microsoft TechNet Server & Management Blogs. Archived from the original on October 8, 2016. Retrieved August 14, 2016.*
54. ^ *"SMB security enhancements". Microsoft Learn. Archived from the original on 2023-04-10. Retrieved 2023-04-10.*
55. ^ *"Windows Protocols". Archived from the original on September 26, 2009. Retrieved October 13, 2009.*
56. ^ *Tridgell, Andrew (June 27, 1997). "A bit of history and a bit of fun". Archived from the original on July 17, 2011. Retrieved July 26, 2011.*
57. ^ *"Samba 4 functional levels". February 25, 2011. Archived from the original on July 29, 2014. Retrieved January 12, 2014.*
58. ^ *"netsmb(4)". NetBSD 8.0 manual pages. Archived from the original on 17 November 2022. Retrieved 5 January 2020.*
59. ^ `nsmb.conf(5)` – FreeBSD File Formats Manual.
60. ^ *"UNIX Extensions". SambaWiki. Archived from the original on June 12, 2020. Retrieved March 15, 2020.*
61. ^ https://www.visualitynq.com

62. ^ *Sunu Engineer. "Building a Highly Scalable and Performant SMB Protocol Server" (PDF). Archived (PDF) from the original on September 27, 2016. Retrieved September 25, 2016.*
63. ^ *"Microsoft and Tuxera strengthen partnership through Tuxera SMB Server". Microsoft. Microsoft News Center. 14 September 2016. Archived from the original on November 17, 2016. Retrieved 6 February 2017.*
64. ^ *"MS02-070: Flaw in SMB Signing May Permit Group Policy to Be Modified". Microsoft. December 1, 2007. Archived from the original on October 8, 2009. Retrieved November 1, 2009.*
65. ^ *"MS09-001: Vulnerabilities in SMB could allow remote code execution". Microsoft. January 13, 2009. Archived from the original on October 5, 2009. Retrieved November 1, 2009.,*
66. ^ *"Sicherheitstacho.eu". Deutsche Telekom. March 7, 2013. Archived from the original on March 8, 2013. Retrieved March 7, 2013.*
67. ^ *"Alert (TA14-353A) Targeted Destructive Malware". US-CERT. Archived from the original on December 20, 2014. Retrieved December 20, 2014.*
68. ^ *"Sony Hackers Used Server Message Block (SMB) Worm Tool". 19 December 2014. Archived from the original on December 20, 2014. Retrieved December 20, 2014.*
69. ^ *"WannaCry Ransomware Attack Hits Victims With Microsoft SMB Exploit". eWeek. Retrieved 13 May 2017.*
70. ^ *"SMBleedingGhost Writeup: Chaining SMBleed (CVE-2020-1206) with SMBGhost". Jamf Blog. 2020-06-09. Retrieved 2020-11-19.*

- *"[MS-CIFS]: Common Internet File System (CIFS) Protocol". Open Specifications. Microsoft. 30 October 2020.*
    - Specifies the Common Internet File System (CIFS) Protocol, a cross-platform, transport-independent protocol that provides a mechanism for client systems to use file and print services made available by server systems over a network
- *"[MS-SMB]: Server Message Block (SMB) Protocol". Open Specifications. Microsoft. 25 June 2021.*
    - Specifies the Server Message Block (SMB) Protocol, which defines extensions to the existing Common Internet File System (CIFS) specification that have been implemented by Microsoft since the publication of the CIFS specification.
- *"[MS-SMB2]: Server Message Block (SMB) Protocol Versions 2 and 3". Open Specifications. Microsoft. 14 December 2021.*
    - Specifies the Server Message Block (SMB) Protocol Versions 2 and 3, which support the sharing of file and print resources between machines and extend the concepts from the Server Message Block Protocol.
- *"[MS-SMBD]: SMB2 Remote Direct Memory Access (RDMA) Transport Protocol". Open Specifications. Microsoft. 25 June 2021.*
    - Specifies the SMB2 Remote Direct Memory Access (RDMA) Transport Protocol, a wrapper for the existing SMB2 protocol that allows SMB2 packets to be delivered over RDMA-capable transports such as iWARP or Infiniband while utilizing the direct data placement (DDP) capabilities of these transports. Benefits include reduced CPU overhead, lower latency, and improved throughput.

- Hertel, Christopher (2003). *Implementing CIFS – The Common Internet FileSystem Archived 2004-02-02 at the Wayback Machine*. Prentice Hall. ISBN 0-13-047116-X. (Text licensed under the Open Publication License, v1.0 or later, available from the link above.)

- Steven M. French, A New Network File System is Born: Comparison of SMB2, CIFS, and NFS Archived 2012-09-07 at the Wayback Machine, Linux Symposium 2007
- Steve French, The Future of File Protocols: SMB2 Meets Linux Archived 2018-08-17 at the Wayback Machine, Linux Collaboration Summit 2012

- DFS section in "Windows Developer" documentation
- the NT LM 0.12 dialect of SMB. In Microsoft Word format
