Technical Report

SMB 2—Next-Generation CIFS Protocol in Data ONTAP

Bingxue Cai, Reena Gupta, NetApp
September 2011 | TR-3740

SMB 2.0 IN DATA ONTAP 8.1 7G/7M AND SMB 2.1 IN DATA ONTAP 8.1 OPERATING IN CLUSTER-MODE

Server Message Block (SMB) 2 is the next version of the Common Internet File System (CIFS)/SMB protocol. This document describes SMB 2 features, configuration details, and its implementation in Data ONTAP® 8.1 7G/7M and Data ONTAP 8.1 Cluster-Mode. This document also describes SMB 2 benefits over CIFS/SMB 1.0, deployment use cases, best practices, and tools to diagnose and capture SMB 2 information. Note that CIFS and SMB refer to the same protocol version: these two terms are used interchangeably as CIFS (SMB 1.0) in this document.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 INTRODUCTION

Server Message Block (SMB) is a remote file-sharing protocol used by Microsoft® Windows® clients and servers starting in the mid-1980s. The original SMB 1.0 (also known as Common Internet File System or CIFS) was designed and implemented to support file-serving solutions based on the assumptions existing at that time. For the past decade or so, some minor changes and tweaks have been made to the protocol to support some new functionality such as network resiliency, scalability, and so on. SMB 2.0, introduced with Windows Vista®, was the first major redesign that considered the needs of the next generation of file servers and clients. These needs included a redesign for modern networking environments such as wide area networks (WANs), possible high-loss networks, time-outs, high latency, and so on. SMB 2.1 was a new revision built on top of SMB 2.0, with additional features. SMB 2.1 is inclusive of SMB 2.0; when a CIFS server or client is said to support SMB 2.1, this server/client also supports SMB 2.0.

Microsoft's ongoing efforts to evolve SMB 2 have positioned the protocol as the next generation of the previous CIFS (SMB 1.0) protocol. It was first introduced with SMB 2.0 in Windows Vista in 2007 and updated with the release of Windows Server® 2008 and Windows Vista SP1 in 2008. SMB 2.1 was then introduced in Windows 7 and Windows Server 2008 R2. Microsoft plans to support SMB 2 as the file system protocol of choice on all future releases of Microsoft operating systems.

NetApp implemented SMB 2.0 over CIFS (SMB 1.0), starting in Data ONTAP 7.3.1. NetApp® network-attached storage (NAS) platforms with Data ONTAP 7.3.1 up to Data ONTAP 8.1 7G/7M are now able to serve Windows XP or other legacy clients, Windows Vista, and Windows 7 clients simultaneously.

NetApp implemented SMB 2.1 over CIFS (SMB 1.0) starting in Data ONTAP 8.1 Cluster-Mode. NetApp NAS storage platforms with Data ONTAP 8.1 Cluster-Mode can also now serve Windows XP or other legacy clients, Windows Vista clients, and Windows 7 clients simultaneously.

2 AUDIENCES

This document is targeted at technical audiences such as system administrators, architects, system engineers, and application vendors who would like to explore and unleash SMB 2 in their file-sharing environments. This document requires prior knowledge of file sharing in Microsoft Windows networks and familiarity with Microsoft terminology.

3 HISTORY OF CIFS (SMB)

When it was first introduced to the public, the remote file-sharing protocol was called Server Message Block (SMB). SMB was used by Microsoft LAN Manager in 1987 and by Windows for Workgroups in 1992. Later, a draft specification was submitted to the Internet Engineering Task Force (IETF) under the name Common Internet File System (CIFS). The CIFS specification is a description of the protocol as it was implemented in 1996 as part of Microsoft Windows NT® 4.0. A preliminary draft of the IETF CIFS 1.0 specification was published in 1997. Later extensions addressed other Microsoft features such as domains, Kerberos, shadow copy, server-to-server copy, and SMB signing. Windows 2000 incorporated those extensions. At that time, some people went back to calling the protocol SMB. CIFS (SMB 1.0) has also been implemented on UNIX®, Linux®, and many other operating systems (either as part of the operating system or as a server suite such as Samba[1]). Occasionally those UNIX and Linux communities also extended the CIFS (SMB 1.0) protocol to address their unique requirements.

[1] Samba is an open source/free software suite that provides file and print services to CIFS (SMB 1.0) clients.

The CIFS (SMB 1.0) protocol had several limitations:

- The protocol was not created with WAN or high-latency networks in mind. Specifically, CIFS is “chatty.” Chattiness is taking a series of round trips to accomplish many of the most common tasks, such as opening a file, reading data from that file, and so on.
- The field values in the SMB header for the number of open files, shares, and users were limited.
- There were numerous commands and subcommands (over 100) in the protocol design, making it difficult to extend, maintain, and secure.
- There were no allowances made for temporary network connection loss.

4 BENEFITS

The SMB 2 protocol is much more resilient to network interruptions. It is designed to scale and perform better, as well as to provide more security than the CIFS protocol.

- Opcode complexity:
  - SMB 1.0 has over 100 command opcodes; SMB 2.x has just 19.
  - Extension mechanism: create context and variable offsets, for example.
- Symbolic links
- More flexible compounding:
  - Parallel or chained
  - Response for every element in the chain
- Durable handles:
  - Reconnect on loss of connection
- Better oplock implementation (SMB 2.1):
  - More extensible lease mechanism
- Crediting mechanism for the number of outstanding operations:
  - Resource control for clients
  - Fill underlying pipe
- Increased scalability for large servers:
  - fileID 64-bit
  - sessionID 64-bit
  - treeID 32-bit
  - shareid 32-bit
- Async operations handled separately
- Improved signing security:
  - HMAC SHA-256 versus MD5
- NAT friendliness:
  - VC count is gone

Table 1 describes the benefits of the SMB 2 protocol compared with the CIFS (SMB 1.0) protocol.

Table 1) Benefits of SMB 2 versus CIFS (SMB 1.0).

Benefits	Features	What It Means to Customers
Enhanced performance	• Compounding operations • Larger buffer size • Crediting (QoS) • Lease (SMB 2.1)	• Larger reads and writes in fewer round trips (64KB) • Improved WAN performance • Server can do some load balancing with credit granting • Lease in SMB 2.1 adds more flexibility for controlling client-side caching, thus resulting in significant performance improvement in high-latency networks
Increased server scalability	• Extended Session ID and TreeID fields • Extended UID and filehandle identifier (FID) namespace	Up to 128K user sessions and tree connections per TCP connection
Network resiliency and increased reliability	• Asynchronous messages • Durable handles	• Fewer timeouts on the CIFS sessions • Avoids data loss on the client side
Enhanced security	SMB signing using SHA256	More robust secured signing algorithm

5 PROTOCOL OVERVIEW

This protocol overview describes how SMB 2.0 interacts with respect to the Internet Protocol (IP) layers, which command sets are available, and what's new in the extended identifier fields.

5.1 SIMPLIFIED COMMAND SETS

SMB 2.0 reduces the complexity in the protocol by reducing the number of command sets. There are only 19 opcodes or commands (compared to over 100 commands in the CIFS protocol) used in the message exchanges between the client and the server. These can be grouped into the three following categories:

- Protocol negotiation, user authentication, and share access:
  - SMB2 NEGOTIATE 0x0000
  - SMB2 SESSION_SETUP 0x0001
  - SMB2 LOGOFF 0x0002
  - SMB2_TREE_CONNECT 0x0003
  - SMB2 TREE_DISCONNECT 0x0004
- File, directory, and volume access:
  - SMB2 CREATE 0x0005
  - SMB2 CLOSE 0x0006
  - SMB2 FLUSH 0x0007
  - SMB2 READ 0x0008
  - SMB2 WRITE 0x0009
  - SMB2 LOCK 0x000A
  - SMB2 IOCTL 0x000B

  - SMB2 CANCEL 0x000c
  - SMB2 QUERY_DIRECTORY 0x000E
  - SMB2 CHANGE_NOTIFY 0x000F
  - SMB2 QUERY_INFO 0x0010
  - SMB2 SET_INFO 0x0011
- Others:
  - SMB2 ECHO 0x000D
  - SMB2 OPLOCK_BREAK 0x0012

5.2 CIFS (SMB 1.0) VERSUS SMB 2: OVER-THE-WIRE COMPARISON

SMB 2 runs over port 445 only and uses TCP as its underlying transport protocol. The packet header formats in SMB 2 are different from those in CIFS (SMB 1.0), as shown in Table 2 and Table 3.

Table 2) CIFS (SMB 1.0) packet header.

Field	Size (Bytes)	Description
Protocol	4	Protocol identifier. The value must be `0xFF`, 'SMB'.
Command	1	Command code, from `0x00` to `0xFF`.
Status	4	32-bit error status code. A server returns error information to the client in the Status field.
Flags	2	Flags characterizing the CIFS request/response. If bit 7 (SMB_FLAGS_SERVER_TO_REDIR) is set, this packet is a server response.
Flags2	2	A 16-bit flag field defining the capabilities of the client/server transaction.
TID	2	Tree identifier; a unique ID for a resource in use by the client.
PID	2	Caller process ID.
UID	2	User identifier.
MID	2	Multiplex identifier; used to route requests inside a process.
WordCount	1	Count of parameter words defining the data portion of the packet.
ParameterWords [WordCount]	1	Parameter words defining the data portion of the packet.
ByteCount	1	Size of the data portion of the packet.
Buffer[ByteCount]	Variable	Data portion of the packet. The format of the data portion depends on the command code. Fields in the data portion consist of an identifier byte followed by the data.

The SMB2 packet header has two variants: ASYNC and SYNC. If the SMB2_FLAGS_ASYNC_COMMAND bit is set in the "Flags" field, then the header takes the ASYNC form, as shown in Table 3.

Table 3) SMB2 ASYNC packet header.

Field	Size (Bytes)	Description
Protocol	4	The protocol identifier. The value MUST be (in network order) 0xFE, 'S', 'M', and 'B'.
StructureSize	2	MUST be set to 64, which is the size, in bytes, of the SMB 2 header structure.
CreditCharge	2	In the SMB 2.002 dialect, this field MUST NOT be used and MUST be reserved. The sender MUST set this to 0, and the receiver MUST ignore it. In the SMB 2.1 dialect, this field indicates the number of credits that this request consumes.
Status	4	The status code for a response. For a request, the client MUST set this field to 0, and the server MUST ignore it on receipt. For a response, this field may be set to any value.
Command	2	The command code. This field MUST contain one of the SMB 2 valid command OpCodes.
CreditRequest/Response	2	On a request, this field indicates the number of credits the client is requesting. On a response, it indicates the number of credits granted to the client. If a client does not want more credits, it MUST set this field to 1.
Flags	4	A flags field, which indicates how to process the operation.
NextCommand	4	For a compounded request, this field MUST be set to the offset, in bytes, from the beginning of this SMB 2 header to the start of the subsequent 8-byte aligned SMB 2 header. If this is not a compounded request, or this is the last header in a compounded request, this value MUST be 0.
MessageId	8	A value that identifies a message request and response uniquely across all messages that are sent on the same SMB 2 Protocol transport connection.
AsyncId	8	A unique identification number that is created by the server to handle operations asynchronously.
SessionId	8	Uniquely identifies the established session for the command.
Signature	16	The 16-byte signature of the message, if SMB2_FLAGS_SIGNED is set in the Flags field of the SMB 2 header. If the message is not signed, this field MUST be 0.

If the SMB2_FLAGS_ASYNC_COMMAND bit is not set in "Flags," the header takes the SYNC form as shown in Table 4.

Table 4) SMB2 SYNC packet header.

Field	Size (Bytes)	Description
Protocol	4	The protocol identifier. The value MUST be (in network order) 0xFE, 'S', 'M', and 'B'.
StructureSize	2	MUST be set to 64, which is the size, in bytes, of the SMB 2 header structure.

Field	Size (Bytes)	Description
CreditCharge	2	In the SMB 2.002 dialect, this field MUST NOT be used and MUST be reserved. The sender MUST set this to 0, and the receiver MUST ignore it. In the SMB 2.1 dialect, this field indicates the number of credits that this request consumes.
Status	4	The status code for a response. For a request, the client MUST set this field to 0, and the server MUST ignore it on receipt. For a response, this field may be set to any value.
Command	2	The command code. This field MUST contain one of the SMB 2 valid command OpCodes.
CreditRequest/Response	2	On a request, this field indicates the number of credits the client is requesting. On a response, it indicates the number of credits granted to the client. If a client does not want more credits, it MUST set this field to 1.
Flags	4	A flags field, which indicates how to process the operation.
NextCommand	4	For a compounded request, this field MUST be set to the offset, in bytes, from the beginning of this SMB 2 header to the start of the subsequent 8-byte aligned SMB 2 header. If this is not a compounded request, or this is the last header in a compounded request, this value MUST be 0.
MessageId	8	A value that identifies a message request and response uniquely across all messages that are sent on the same SMB 2 Protocol transport connection.
ProcessId	4	The client-side identification of the process that issued the request. The client MUST set this field to 0xFEFF. The server MUST set this field to the ProcessId value received in the corresponding request, if any, or to 0 otherwise. The client MUST ignore this field on receipt.
TreeId	4	Uniquely identifies the tree connect for the command. This MUST be 0 for the SMB2 TREE_CONNECT Request. The TreeId MAY be any unsigned 32-bit integer that is received from a previous SMB2 TREE_CONNECT Response. The following SMB 2 Protocol commands do not require the TreeId to be set to a nonzero value received from a previous SMB2 TREE_CONNECT Response.
SessionId	8	Uniquely identifies the established session for the command.
Signature	16	The 16-byte signature of the message, if SMB2_FLAGS_SIGNED is set in the Flags field of the SMB 2 header. If the message is not signed, this field MUST be 0.

5.3 8-BYTE ALIGNED BUFFERS

A nonzero value for the NextCommand field in the SMB 2 header signals a compound request. This field contains the data offset, which is the number of bytes from the beginning of the header in consideration to the start of the SMB 2 header of the subsequent request. Unlike in CIFS, compounded requests must be aligned on 8-byte boundaries in SMB 2. This would be handled by the SMB redirector, and the file server would be transparent to the application or the client.

5.4 EXTENDED ID FIELDS

Most of the ID fields have been extended in SMB 2 for increased scalability for large servers.

- File IDs are 64-bit (persistent) + 64-bit (volatile) versus 16-bit.
- Message IDs are 64-bit versus 16-bit.
- Tree IDs are 32-bit versus 16-bit.
- Security signature is 16 bytes versus 8 bytes.

5.5 ONLY UNICODE AND NTSTATUS CODES USED

All the text fields used in SMB 2 packets are in Unicode[2] format. Also, SMB 2 uses only NTSTATUS error codes and does not use DOS error codes.

6 SMB PROTOCOL NEGOTIATION/COMPATIBILITY

The process to agree upon, or negotiate, a common level of SMBs that each host can understand is referred to as SMB protocol negotiation. The SMB protocol version used for the file-sharing operation is determined during this negotiation. Table 5 explains how the SMB protocol version is negotiated between different operating systems.

Table 5) Default protocol used in Data ONTAP.

Windows Client Type	Data ONTAP Version	Protocol Used
Windows 7, Windows 2008 R2	NetApp system Data ONTAP 8.1 Cluster-Mode	SMB 2.1
Vista, Windows 2008	NetApp system Data ONTAP 8.1 Cluster-Mode	SMB 2.0
Windows 7, Vista, Windows 2008, Windows 2008 R2	NetApp system Data ONTAP 7.3.1 to Data ONTAP 8.1 7G/7M with default disabled SMB 2 feature: `options cifs.smb2.enable off`	SMB 1.0
Windows 7, Vista, Windows 2008, Windows 2008 R2	NetApp system Data ONTAP 7.3.1 to Data ONTAP 8.1 7G/7M with enabled SMB 2 feature: `options cifs.smb2.enable on`	SMB 2.0
Windows clients prior to Vista	NetApp system Data ONTAP 7.3.1 to Data ONTAP 8.1 7G/7M NetApp system Data ONTAP 8.0 and 8.1 Cluster-Mode	SMB 1.0
All Windows clients	NetApp system Data ONTAP 7.3 or prior	SMB 1.0

Note: SMB 2.0 and CIFS (SMB 1.0) can coexist in Data ONTAP 7.3.1 to Data ONTAP 8.1 7G/7M, and in Data ONTAP 8.1 Cluster-Mode.
There is no need to upgrade all the clients to Windows Vista to work with Data ONTAP 7.3.1. Any legacy Windows clients such as XP, Windows 2000, and Windows 2003 can still work with Data ONTAP 7.3.1 over SMB 1.0.

[2] Unicode is a comprehensive way of defining characters electronically for compatibility with most of the spoken languages in the world.

7 CONFIGURATION

7.1 WINDOWS VISTA/WINDOWS 2008

SMB 2.0 is enabled by default in the Windows Vista and Windows Server 2008 operating systems.

7.2 WINDOWS 7/WINDOWS 2008 R2

SMB 2.1 is enabled by default in the Windows 7 and Windows Server 2008 R2 operating systems.

7.3 DATA ONTAP 7G/7M SYSTEMS

SMB 2.0 is disabled by default in Data ONTAP 7.3.1 up to Data ONTAP 8.1. It should be enabled for any Vista, Windows 7, Windows 2008, or Windows 2008 R2 clients in order to connect to NetApp systems over SMB 2.0. Otherwise those clients continue to connect over SMB 1.0. These are the different commands and options to manage SMB 2.0 in Data ONTAP.

- Options:
  - **cifs.smb2.enable**
    This option enables or disables SMB 2.0 in Data ONTAP. When this option is disabled, the NetApp system does not accept any new SMB 2.0 sessions, but the existing sessions are not terminated (default is "off").
  - **cifs.smb2.signing.required**
    This option enforces SMB signing on all SMB 2.0 sessions (default "off").
  - **cifs.smb2.client.enable**
    This option enables or disables SMB 2.0 client capability on the storage system. When this option is enabled, NetApp system-initiated connections to Windows domain controllers attempt to use the SMB 2.0 protocol. In case the Windows domain controller does not support the SMB 2.0 protocol, then the NetApp system reverts to using CIFS (SMB 1.0). If a session is established over SMB 2.0 and later this option is disabled, existing sessions are not terminated; the NetApp system continues to use SMB 2.0 for the existing sessions, but no new sessions will attempt to use SMB 2.0 (default is "off").
  - **cifs.smb2.durable_handle.enable**
    This option enables or disables the durable handle functionality for SMB 2.0 clients. If this option is enabled, the open files from a client are preserved when the client gets disconnected from the NetApp system. These open files can be reclaimed when the client reconnects to the NetApp system (default is "on").
  - **cifs.smb2.durable_handle.timeout**
    This option configures the duration in seconds for which the NetApp system preserves the durable handle after a temporary network failure. This timer has a default value of 16 minutes, but its value could be changed by system policy to anywhere in the range between 5 seconds and 2147483647 seconds. It can also be configured for infinite value as "-1."
- Command:
  - **cifs sessions –p [smb|smb2]**
    This command has a new option, "-p." This option filters the sessions on the basis of protocol version used. When the -p option is used with "smb" as the argument, only SMB 1.0 sessions are displayed. When the -p option is used with "smb2" as the argument, only SMB 2.0 sessions are displayed. When the -p option is not used, both SMB 1.0 and SMB 2.0 sessions are displayed. The -p option can be used along with -c and -s options, which provide information about open directories and the number of active ChangeNotify requests, as well as security information for a specified connected user.

7.4 DATA ONTAP CLUSTER-MODE SYSTEMS

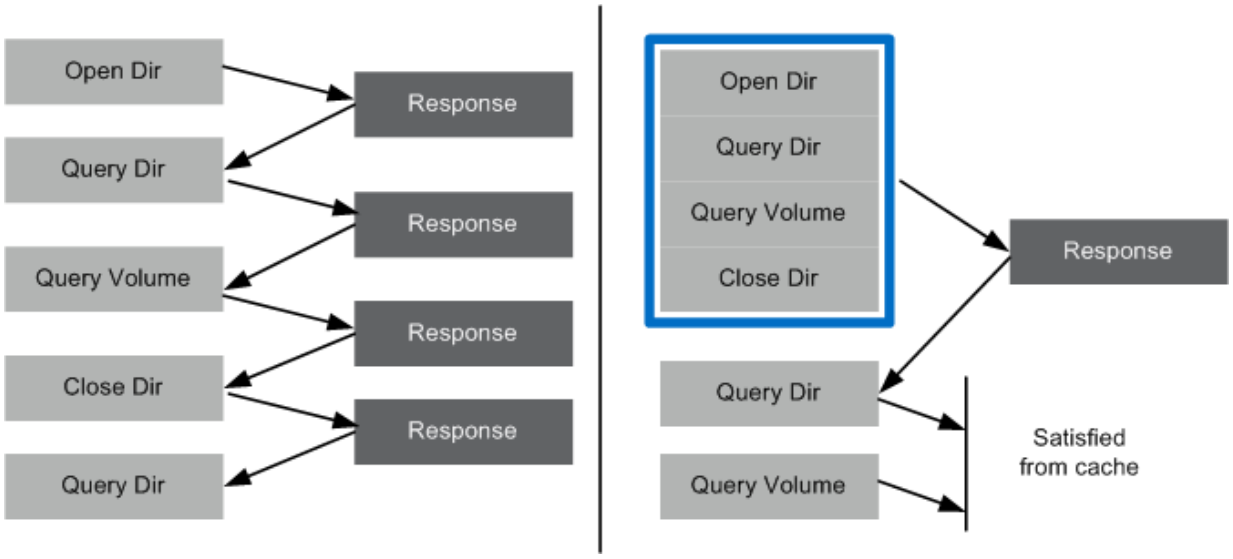
SMB 2.1 is enabled by default starting in Data ONTAP 8.1 Cluster-Mode systems. There is no SMB 2 support in Data ONTAP 8.0 Cluster-Mode systems.

8 SMB 2 FEATURES IN DATA ONTAP

8.1 COMPOUNDED OPERATIONS

SMB 2 provides a method of combining multiple SMB 2 messages and commands into a single transmission request for submission to the underlying transport. This compounding reduces the round trips between the client and the server, thus reducing the CIFS protocol “chattiness” and improving the protocol's performance. A common example of compounding commands is putting "OpenDir," "QueryDir," "QueryVolume," and "CloseDir" together into a single SMB packet for requesting a directory browsing operation.

Figure 1) Compounded operations.



There are two types of compounded messages.

- **Related compounded messages**
  For a related compounded request, the NetApp system processes compounded requests in a sequential manner. The processing continues until all the requests in the compounded request are processed, even if one of the requests fails. If a subsequent command succeeds, even if a previous command failed, it is marked successful. Respective responses are compounded together and sent back to the client.

  Note: If one of these compounded requests becomes an asynchronous type of request (as defined in section 8.4), all subsequent ones go async as well and have an interim response from the server for each of the async requests.

- **Unrelated compounded messages**
  For an unrelated compounded request, all the requests are processed independently irrespective of the result of processing other requests, and the responses are sent independently.

The NetApp system supports both related and unrelated compounded requests. Data ONTAP 7G/7M systems compound the response for related compounded requests. The Data ONTAP 8.1 Cluster-Mode

system does NOT compound the requests, and instead always sends individual responses. For unrelated compounded requests, the NetApp system does not compound the responses.

8.2 DURABLE HANDLES

Durable handles are the file handles that persist across SMB 2 sessions. They are designed to prevent data loss caused by short network outages by absorbing writes cached on the client on a different SMB 2 session. When a client opens a file, it specifies whether it needs the file handle to be durable. If the current connection goes away, the client would try to use the durable handle on a new connection if it is still valid on the server. The server for its part issues a durable handle only if it supports the functionality. The server keeps track of open files even after the connection drops. Upon session disconnection, the server makes the handles available for reclaim by the same authenticated user context on a different connection. Any pending cached writes during the disconnection can be flushed to the disk upon the reconnection. It brings resiliency to the network by avoiding data loss despite connections being dropped, especially on WANs.

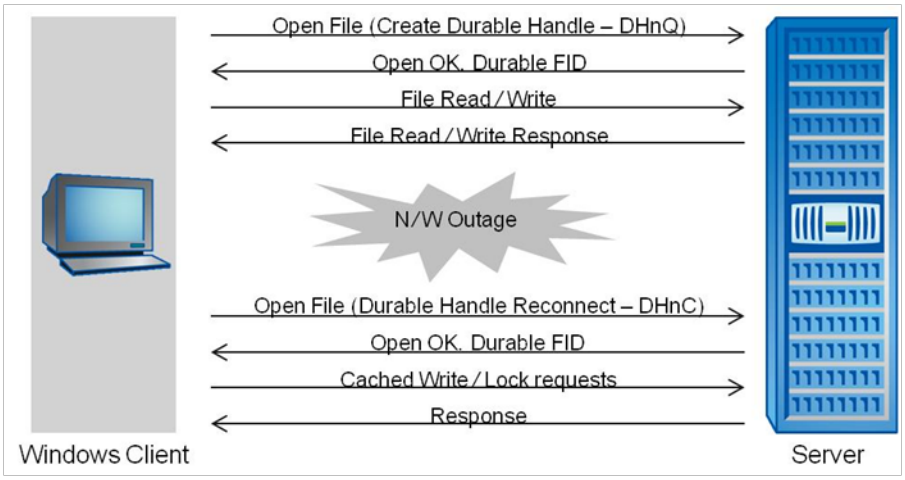
The applications make use of durable handles through the redirector only, not by any client application programming interface (API). The redirector silently requests (and uses) durable handles on every file it opens with a batch opportunistic lock (oplock) without requiring the application to do anything differently.

The durable handles FID field is 128 bits long and has two parts:

- Persistent (64 bits long): stays valid as long as the file is open on the storage system. That means it can be valid across multiple client sessions. A persistent ID doesn't identify a file uniquely on the server, so if a file is closed and opened again, there is no assurance that the persistent ID will be the same.
- Volatile (64 bits long): stays valid for one SMB session.

During a network outage, if some other client requests the same file, it must send an oplock break request for this disconnected durable handle. In this situation, the durable handle would be cleaned up to prevent any sort of access-denied messages, and the client would not be able to claim the file handle.

Figure 2) Durable handles in a network outage.



Durable handles are not designed to survive a server or client reboot or cluster failover, because the durable file handle structure is maintained in the server's memory, not on the disk. For example, in a clustered system, during an OS upgrade of the primary server, the partner server doesn't automatically inherit the durable handle from the primary server. Clients would have to make a fresh connection to the server; hence, it is not a transparent reconnection. If the Windows client panics, the durable handles information on the client is lost. The retry mechanism used by the client is limited to less than five times

and primarily forced by the application above it using the handle rather than being driven by a durable handle timer.

There are two commands in the Data ONTAP 7G/7M system to manage the durable handles, as described in section 7.3:

- options cifs.smb2.durable_handle.enable
- options cifs.smb2.durable_handle.timeout

Durable handle in Data ONTAP Cluster-Mode systems is always turned on along with SMB 2.1. The system keeps the durable handle for 15 minutes upon detection of TCP disconnect.

8.3 CREDIT SYSTEM IN DATA ONTAP 7G/7M SYSTEMS

SMB 2 has a mechanism for the clients to send a number of outstanding requests to a server. This allows the client to build a pipeline of requests instead of waiting for a response before sending the next request. This is especially relevant when using a high-latency network. It also makes sure that a client cannot overload the server with a large number of pending requests, thus providing quality of service (QoS).

SMB 2 uses credit-based flow control, which allows a server to control the number of outstanding requests on a given SMB 2 session. The client is given a certain small number of credits from the server. For every credit it has, it may send one message to the server. As the client sends messages, it continually requests more credits to continue sending traffic. The server can therefore control the amount of traffic (in outstanding SMB operations on a session) by granting more or fewer credits to the client in question; thus some basic QoS is provided by avoiding excess use of resources by one client. This allows the server to throttle back connections when it becomes overburdened and to grant certain clients a higher “priority” than other clients. With this feature, the protocol can keep more data in flight and better use the available bandwidth. This is a key to making a large transfer take much less time in a high-bandwidth, high-latency network.

A Windows Vista client requests 128 credits by default in each message sent to the server and also uses one credit for each message. A NetApp storage server would grant 50 credits by default to a client, and it can load balance dynamically by adjusting the number of credits on the fly for each request. There might be multiple requests on a single SMB session, so the credits are requested and granted per request.

Note: Windows Vista and Windows Server 2008 do not grant credits on interim responses. An interim response for an asynchronously processed SMB 2 CHANGE_NOTIFY request grants credits to keep the transaction from stalling in case the client is out of credits.

Note: The Data ONTAP 8.1 Cluster-Mode system does not support Credit System in its SMB 2.1 implementation.

8.4 ASYNCHRONOUS OPERATIONS

Certain SMB commands from the clients can take a longer time to process on the server, in which case the server sends an interim asynchronous response to the client. Examples for these commands are:

- Oplock break
- Change-notify
- Named-pipe operations on blocking pipes
- Byte-range lock requests that might wait for lock availability
- A create that triggers an oplock break

These asynchronous responses are sent before the final response to the client and are in the form of async headers.

A client cannot request an async header; the server decides based on the type of request whether to send an async response. Async operations don't actively consume credits, but the responses may be used to grant credits. Async processing provides the ability for the server to modify the crediting behavior for clients performing significant async operations.

8.5 LARGER BUFFER SIZE

SMB 2 now has a much larger read and write buffer size; the default has been increased to 64k as compared to the 32k default in CIFS/SMB. Larger reads and writes make better use of faster networks, even with high latency. Any applications that can make use of the larger reads and writes can realize performance improvements.

8.6 INCREASED SCALABILITY

SMB 2 increases the restrictive constants within the protocol design to allow scalability for file sharing. Number of users, open shares, and open files per TCP connection for a server are greatly increased. Table 6 differentiates the field sizes and limits increased in SMB 2 over CIFS/SMB.

Table 6) Field sizes and limits in SMB 2.

Protocol	Type of Identifier	Field Size	MAX Limits	Comments
CIFS/SMB	UID	16 bits	64k (2^{16})	Number of sessions
	TID	16 bits	64k (2^{16})	Open share connections
	FID	16 bits	64k (2^{16})	Open file connections
SMB 2	SessionID (UID)	64 bits	2^{64}	Number of sessions
	TreeID (TID)	32 bits	2^{32}	Open share connections
	FID (FID)	128 bits	2^{64}	Open file connections

8.7 SMB SIGNING

SMB signing is a feature through which all communications using the SMB protocol can be digitally signed at the packet level. Digitally signing the packets enables the recipient of the packets to confirm their point of origination and their authenticity, thus avoiding man-in-the-middle attacks.

The SMB 2 signing feature uses the more secure HMAC-SHA256 algorithm instead of MD5 (as in CIFS/SMB) for generating the digital signature. SMB 2 signing is never disabled as seen in CIFS/SMB; the possible configurations are either "required" or "not required."

Table 7) SMB 2 signing.

SMB 2 Signing	Server Required	Server Not Required
Client Required	Signed	Signed
Client Not Required	Signed	Not signed

If signing is negotiated for an SMB 2 session during session setup, the signatures of all the SMB 2 messages received on that session are verified, and the request is rejected if the message is not signed or if the signature is not valid.

Note: Data ONTAP 8.1 Cluster-Mode only supports client-side SMB 2 signing. It does not support server-side SMB 2 signing.

SMB signing with other features:

- Signing with compounded messages: Each individual request is signed.
- Signing with asynchronous responses: Interim responses are also signed.

Note:

  - Windows Servers do not sign the interim responses and oplock breaks.
  - SMB signing has a performance impact.

8.8 SMB 2.1 LEASE

Lease is introduced as a new type of client caching mechanism in SMB 2.1. Compared to the original oplock mechanism, lease offers more flexibility and levels in controlling the client caching. This results in significant performance improvement in high-latency and erratic networks.

The primary types of leases are:

- Read-caching lease: allows caching reads and can be shared by multiple clients.
- Write-caching lease: allows caching writes and is exclusive to only one client.
- Handle-caching lease: allows caching handles and can be shared by multiple clients.

Requests for leases can be a combination of one or more of the lease types above. The valid leases in Windows 7 are:

- R (READ_CACHING): This is similar to Level 2 oplock type.
- RW (READ_CACHING | WRITE_CACHING): This is similar to exclusive oplock type.
- RH (READ_CACHING | HANDLE_CACHING): Compared with Level 2 oplocks, this lease level offers significant improvement for complex I/O-intensive applications.
- RWH (READ_CACHING | WRITE_CACHING|HANDLE_CACHING): This is similar to the batch oplock type.

9 SUPPORTABILITY

SMB 2.0 is supported on NetApp NAS storage platforms beginning with Data ONTAP 7.3.1 and up to Data ONTAP 8.1 7G/7M.

Microsoft supports SMB 2.0 beginning with Windows Vista and Windows Server 2008. Microsoft will continue support for SMB 2.0 on all future releases of Microsoft operating systems.

SMB 2.1 is supported on NetApp NAS storage clustered platforms beginning with Data ONTAP 8.1 Cluster-Mode.

Windows 7 and Windows 2008 R2 both support SMB 2.1 as a new dialect.

Data ONTAP 7.3.1 up to Data ONTAP 8.1 7G/7M support both CIFS (SMB 1.0) and SMB 2.0; therefore customers will still be able to connect to a NetApp storage system from any existing Windows XP, Windows Vista, or Windows 7 client. There is no change in the license requirement; the same CIFS license is applicable for SMB 2.0 as well. Note that SMB 2.0 is not available in workgroup mode, because extended security is not available in workgroup mode.

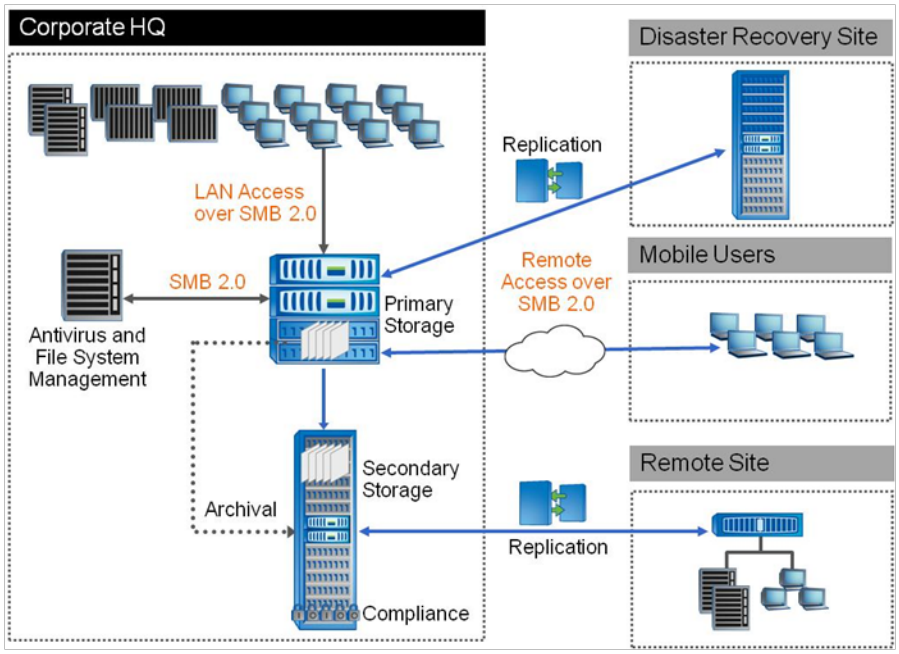
Data ONTAP 8.1 Cluster-Mode supports CIFS (SMB 1.0), SMB 2.0, and SMB 2.1. Customers can still connect to a NetApp storage system from any existing Windows client, such as Windows XP, Windows Vista, Windows 2008, Windows 7, or Windows 2008 R2.

10 DEPLOYMENT USE CASES

A typical Windows file-sharing deployment is shown in Figure 3. SMB 2 can be used in all of these typical Windows file-sharing deployments:

- LAN access among all the supported Windows Servers and clients to the NetApp storage systems
- LAN access from any virus scanning and file screening servers
- WAN access from the supported remote Windows systems in the remote offices
- Virtual private network (VPN) access from any mobile users

Figure 3) Typical Windows file-sharing deployment.



11 IMPACT ON APPLICATIONS

There are opportunities for new applications to use the underlying SMB 2 protocol. For example, Petrel (an oil and gas application) is being tested to run over SMB 2. Applications that use the CIFS protocol will benefit further from SMB 2 in terms of the reliability, scalability, and other features it offers.

CIFS features such as auditing, group policy objects, and access-based enumeration have been shown to experience no impact from using SMB 2. Virus scanning would depend on the antivirus (AV) scan server where the AV scan engine is running. If the AV scan server is a Windows 2008 server, then the communication between the NetApp system and the AV scan server would happen over SMB 2 (if it's enabled on the NetApp system). There won't be any effect on any cross-protocol file access between SMB 2 and NFSv3/v4 because SMB 2 continues to use the same NTFS Access Control List (ACL) structure as in CIFS/SMB.

12 PERFORMANCE

SMB 2 offers better performance in certain cases than CIFS/SMB. For example, for NAS access over the WAN, where the SMB 2 protocol has the ability to do compounded operations, its larger buffer sizes and its durable handles can help boost performance. The concept of credits can also help implement QoS for clients, especially in a high-utilization environment.

Currently there is no industry performance benchmark for SMB 2. The performance improvement over the CIFS (SMB 1.0) protocol depends heavily on the customer's environment, including workload and network infrastructure, where workload can mean sequential/random, op mix, and number of clients. Some of our customers using their own test methodology observed higher performance using Data ONTAP 7.3.1 and above and Windows Vista/Windows 7 compared to their previous environment using Windows XP.

13 BEST PRACTICES

Windows Vista clients are mostly autotuned for getting the maximum benefits of the SMB 2.0 protocol. Windows 7 clients are mostly tuned for the SMB 2.1 protocol. Certain best-practice guidelines should be followed in order to achieve the maximum performance benefit for the SMB 2.0/SMB 2.1 protocol.

- Use Windows Vista SP1 or later as a client. Windows Vista SP1 has the full implementation of the SMB 2.0 protocol; Vista RTM had partial implementations of some SMB 2.0 features.
- Use Windows 7 and Windows 2008 R2SP1 as a client to get the full benefits of SMB 2.1.
- If you are using any applications over SMB 2.0/SMB 2.1, then, wherever applicable, tune your applications to:
  - Leverage the larger block size to send data.
  - Send requests for more concurrent blocks.
- Use Gigabit Ethernet or better network speed for high bandwidths.
- Use premium hardware on the client and server side, because a powerful configuration can yield better performance.
- Turn SMB signing off if not needed.

14 DIAGNOSTIC TOOLS

14.1 PACKET TRACE ANALYZER

Microsoft NetMon v3.1 and v3.2 as well as Wireshark can capture and decode SMB 2.0 packets. The protocol identifier is `0xFE 'S' 'M' 'B'`, although NetMon is more complete and better at decoding SMB 2.0 packets.

14.2 DATA ONTAP 7G/7M TOOLS

- **`cifs sessions –p [smb|smb2]`**
  This command filters the sessions on the basis of protocol version used to find out which clients are connected to the NetApp system using SMB 2.0 and which clients are connected using SMB 1.0.
- **`cifs-stat`**
  This command shows the CIFS statistics and also displays the specific SMB 2.0 statistics that are being generated.

14.3 DATA ONTAP 8.1 CLUSTER-MODE TOOLS

- **`statistics show`**
- **`statistics show-periodic`**

15 REFERENCES

- NetApp Streamlines Data Management for Petrotechnical Applications: http://media.netapp.com/documents/netapp-zeus-technology-reprint.pdf
- SMB 2.0 Protocol Specification: http://msdn.microsoft.com/en-us/library/cc212614.aspx
- Network Monitor 3.2 tracing tool: www.microsoft.com/downloads/details.aspx?FamilyID=f4db40af-1e08-4a21-a26b-ec2f4dc4190d&displaylang=en
- Microsoft's SMB 2.0 Performance white paper by Tolly Group: www.microsoft.com/downloads/details.aspx?FamilyID=04cad8b9-9f9f-453a-893a-458d22dbb3c5&DisplayLang=en
- Microsoft's KB article on Explanation of Opportunistic Locking on Windows NT: http://support.microsoft.com/kb/129202

16 CONCLUSION

SMB 2 is a next-generation NAS protocol for Windows that has been redesigned to accommodate next-generation NAS servers' requirements, especially for wireless networks and remote office deployments. It offers enhanced performance, flexibility, reliability, scalability, and security. This provides opportunities for application vendors to use their applications to get maximum benefits from SMB 2 features.

17 REVISIONS

Date	Name	Description
March 2009	Reena Gupta	Creation
September 2011	Bingxue Cai	Addition of SMB 2.1 in Data ONTAP 8.1 Cluster-Mode