SMB 3.0 Protocol Feature Map

Tom Talpey

Microsoft

May 8, 2012

Agenda

- SMB 3.0 status update and protocol changes
- SMB 3.0 “features”
- Mapping SMB features to protocol
  - Phased/focused approaches

SMB 3.0 Feature Set

- Remote File Storage for Server Apps
  - Hyper-V, SQL Server fully supported
- Features
  - SMB Transparent Failover
  - SMB Scale Out
  - SMB Direct
  - SMB Multichannel
  - VSS for SMB File Shares
  - SMB Directory Leasing
  - SMB Encryption
  - Branch Cache V2 support

“SMB 3.0”

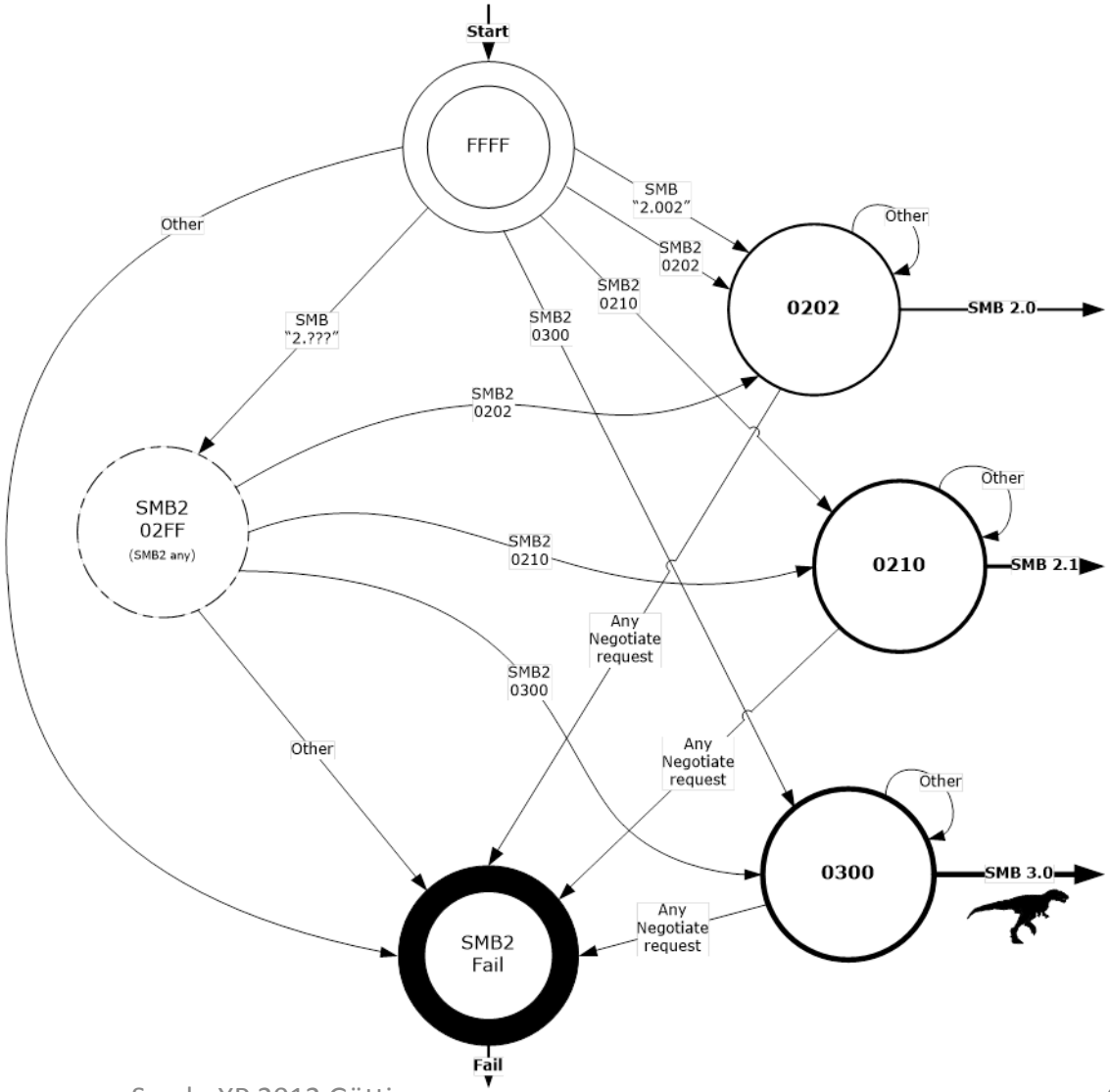
- SMB 2.2 is now SMB 3.0
  - http://blogs.technet.com/b/windowsserver/archive/2012/04/19/smb-2-2-is-now-smb-3-0.aspx
- A rebranding of SMB 2.2
  - Because it’s truly a new major release
  - To specifically highlight the new server application support
    - Hyper-V, SQL Server, …
- No significant protocol change from SMB 2.2
- Preview available:
  - http://download.microsoft.com/download/C/6/C/C6C3C6F1-E84A-44EF-82A9-49BD3AAD8F58/[MS-SMB2-Preview-Windows8].pdf

Supporting SMB 3.0

- SMB 3.0 is a superset of SMB 2.x
- New behaviors are optional:
  - client-initiated (optionally)
  - server-acknowledged (optionally)
- Everything negotiated:
  - (well, almost everything, see encryption type)
- Clients initiate use of these features
  - i.e. server applications
- Decision to support individual features is therefore an application question
  - Must first answer does the application require it? Prefer it?

SMB Dialect Negotiation

- From Preview doc
- 3 dialects:
  - 2.002: Vista/Server 2008
  - 2.1: Win7/Server 2008R2
  - 3.0: Win8/Server 2012



SMB3 protocol changes from 2.1

- NOTE this is only a summary. The document is always the official source, unless of course there's a bug, in which case file one! ☺

SMB Negotiation changes

- New dialect number 0x0300
- Dialect array now (typically) contains 3 entries not 2
- Client capabilities added to negotiate request:
  - SMB2_GLOBAL_CAP_DFS
  - SMB2_GLOBAL_CAP_LEASING
  - SMB2_GLOBAL_CAP_LARGE_MTU
  - SMB2_GLOBAL_CAP_MULTI_CHANNEL
  - SMB2_GLOBAL_CAP_PERSISTENT_HANDLES
  - SMB2_GLOBAL_CAP_DIRECTORY_LEASING
  - SMB2_GLOBAL_CAP_ENCRYPTION
- Server capabilities added to negotiate response:
  - SMB2_GLOBAL_CAP_MULTI_CHANNEL
  - SMB2_GLOBAL_CAP_PERSISTENT_HANDLES
  - SMB2_GLOBAL_CAP_DIRECTORY_LEASING
  - SMB2_GLOBAL_CAP_ENCRYPTION

Negotiation - request

Capabilities (4 bytes): If the client implements the SMB 3.0 dialect, the **Capabilities** field MUST be constructed using the following values. Otherwise, this field MUST be set to 0 and the server MUST ignore it on receipt.

Value	Meaning
SMB2_GLOBAL_CAP_DFS 0x00000001	When set, indicates that the client supports the Distributed File System (DFS).
SMB2_GLOBAL_CAP_LEASING 0x00000002	When set, indicates that the client supports leasing.
SMB2_GLOBAL_CAP_LARGE_MTU 0x00000004	When set, indicates that the client supports multi-credit operations.
SMB2_GLOBAL_CAP_MULTI_CHANNEL 0x00000008	When set, indicates that the client supports establishing multiple channels for a single session.
SMB2_GLOBAL_CAP_PERSISTENT_HANDLES 0x00000010	When set, indicates that the client supports persistent handles.
SMB2_GLOBAL_CAP_DIRECTORY_LEASING 0x00000020	When set, indicates that the client supports directory leasing.
SMB2_GLOBAL_CAP_ENCRYPTION 0x00000040	When set, indicates that the client supports encryption.

Dialects (variable): An array of one or more 16-bit integers specifying the supported dialect revision numbers. The array MUST contain at least one of the following values.<7>

Value	Meaning
0x0202	SMB 2.002 dialect revision number.<8>
0x0210	SMB 2.1 dialect revision number.<9>
0x0300	SMB 3.0 dialect revision number. <10>

Negotiation - response

Capabilities (4 bytes): The Capabilities field specifies protocol capabilities for the server. This field MUST be constructed using the following values.

Value	Meaning
SMB2_GLOBAL_CAP_DFS 0x00000001	When set, indicates that the server supports the Distributed File System (DFS).
SMB2_GLOBAL_CAP_LEASING 0x00000002	When set, indicates that the server supports leasing. This flag is not valid for the SMB 2.002 dialect.
SMB2_GLOBAL_CAP_LARGE_MTU 0x00000004	When set, indicates that the server supports multi-credit operations. This flag is not valid for the SMB 2.002 dialect.
SMB2_GLOBAL_CAP_MULTI_CHANNEL 0x00000008	When set, indicates that the server supports establishing multiple channels for a single session. This flag is only valid for the SMB 3.0 dialect.
SMB2_GLOBAL_CAP_PERSISTENT_HANDLES 0x00000010	When set, indicates that the server supports persistent handles. This flag is only valid for the SMB 3.0 dialect.
SMB2_GLOBAL_CAP_DIRECTORY_LEASING 0x00000020	When set, indicates that the server supports directory leasing. This flag is only valid for the SMB 3.0 dialect.
SMB2_GLOBAL_CAP_ENCRYPTION 0x00000040	When set, indicates that the server supports encryption. This flag is only valid for the SMB 3.0 dialect.

Session setup changes

- Session setup includes new flag:
  - SESSION_FLAG_BINDING
    - In support of multichannel
    - Request must be signed
- Session setup response includes new flag:
  - SESSION_FLAG_ENCRYPT_DATA

Session setup - request

Request:

Flags (1 byte): If the client implements the SMB 3.0 dialect, this field MUST be set to a combination of zero or more of the following values. Otherwise it MUST be set to 0.

Value	Meaning
SMB2_SESSION_FLAG_BINDING 0x01	When set, indicates that the request is to bind an existing session to a new connection.

Response:

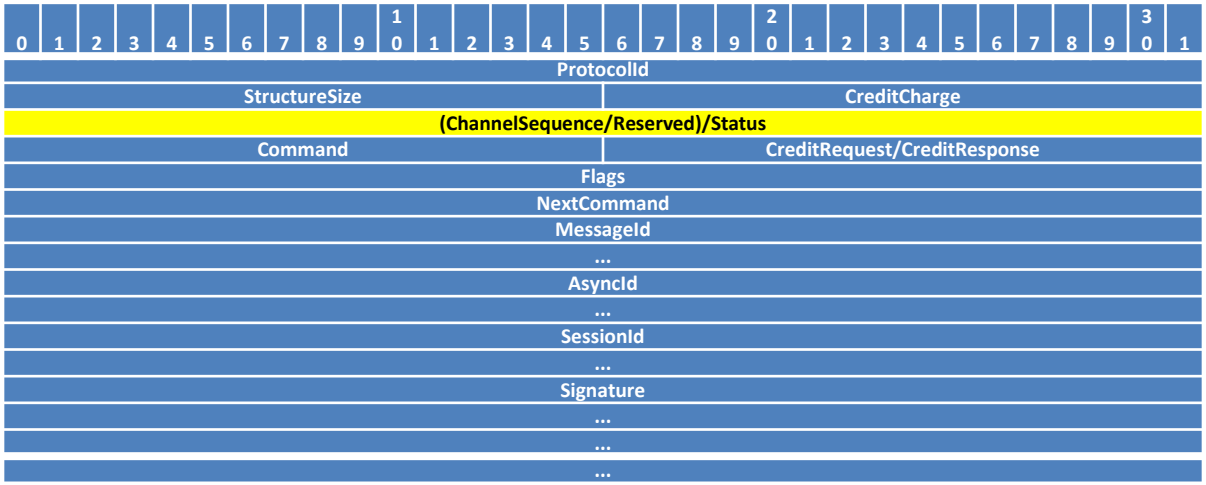
SessionFlags (2 bytes): A flags field that indicates additional information about the session. This field MUST contain either 0 or one of the following values:

Value	Meaning
…	
SMB2_SESSION_FLAG_ENCRYPT_DATA 0x0004	If set, the server supports encryption for messages on this session. This flag is only valid for the SMB 3.0 dialect.

SMB2 header changes

- New ChannelSequence in sync/async request header
  - Set if multichannel or persistent handles in use
  - Fenced/checked on replay operations
- New REPLAY_OPERATION flag in header and CreateGUID identifier in open
  - Plus various per-open state tracking for replay
  - STATUS_FILE_NOT_AVAILABLE replay indication

SMB2 Header (async shown)



Tree Connect

- Tree connect response adds share attributes:
  - ENABLE_HASH_v2
  - SHAREFLAG_ENCRYPT_DATA
- Tree connect response adds server capabilities
  - SHARE_CAP_CONTINUOUS_AVAILABILITY
  - SHARE_CAP_SCALEOUT
  - SHARE_CAP_CLUSTER

Tree Connect response

ShareFlags (4 bytes): This field contains properties for this share….

Value	Meaning
…	
SHI1005_FLAGS_ENABLE_HASH_V2 0x00004000	The share supports v2 hash generation for branch cache retrieval of data. For more information, see section 2.2.31.2. This value is only supported for the SMB 3.0 dialect.
SMB2_SHAREFLAG_ENCRYPT_DATA 0x00008000	If set, the server requires encrypted messages for accessing this share. This flag is only valid for the SMB 3.0 dialect.

Capabilities (4 bytes): Indicates various capabilities for this share. …

Value	Meaning
…	
SMB2_SHARE_CAP_CONTINUOUS_AVAILABILITY 0x00000010	The specified share is continuously available. This flag is only valid for the SMB 3.0 dialect.
SMB2_SHARE_CAP_SCALEOUT 0x00000020	The specified share is present on a server configuration which facilitates faster recovery of durable handles. This flag is only valid for the SMB 3.0 dialect.
SMB2_SHARE_CAP_CLUSTER 0x00000040	The specified share is present on a server configuration which provides monitoring of the availability of share through the Witness service specified in [MS-SWN]. This flag is only valid for the SMB 3.0 dialect.

Create

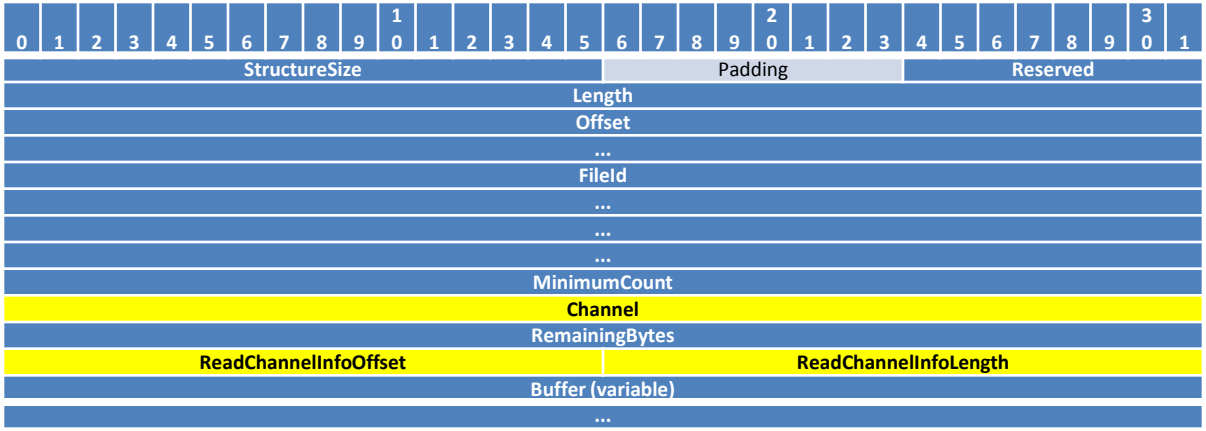
- New optional Create request contexts:
  - SMB2_CREATE_REQUEST_LEASE_V2
  - SMB2_CREATE_DURABLE_HANDLE_REQUEST_V2
    - Supports persistence and timeout
  - SMB2_CREATE_DURABLE_HANDLE_RECONNECT_V2
  - SMB2_CREATE_APP_INSTANCE_ID
    - Supports faster application failover
- Create - can request a lease on directories (if dir leasing supported)
- Create response:
  - SMB2_CREATE_FLAG_REPARSEPOINT flag
- Create response contexts:
  - SMB2_CREATE_RESPONSE_LEASE_V2
  - SMB2_CREATE_DURABLE_HANDLE_RESPONSE_V2

Read and Write

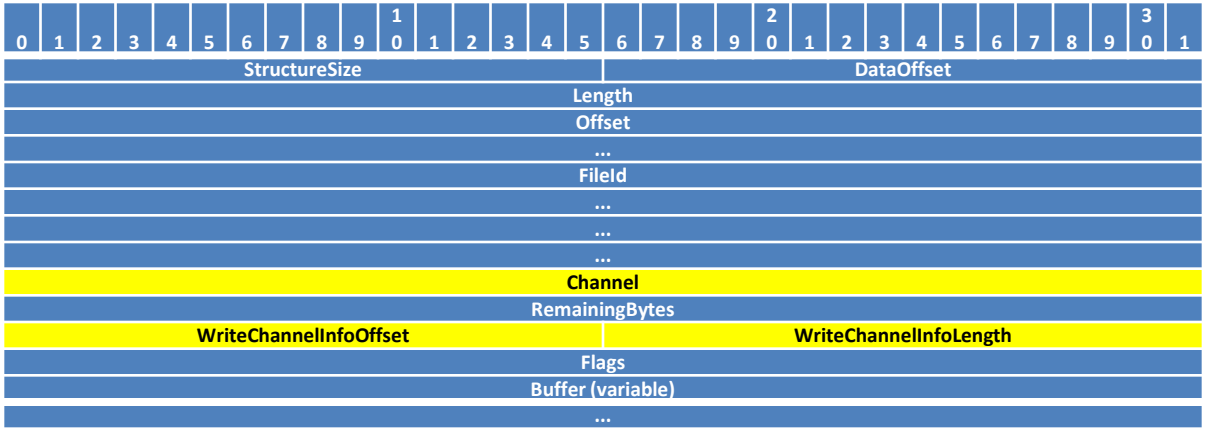
- Read/write requests:
  - Channel info indicates RDMA-enabled payload
    - Previously reserved field
- Read/write responses:
  - No change
- SMB Direct header carries transport context
  - Prepended to SMB2 header

Read and Write requests

Read:



Write:

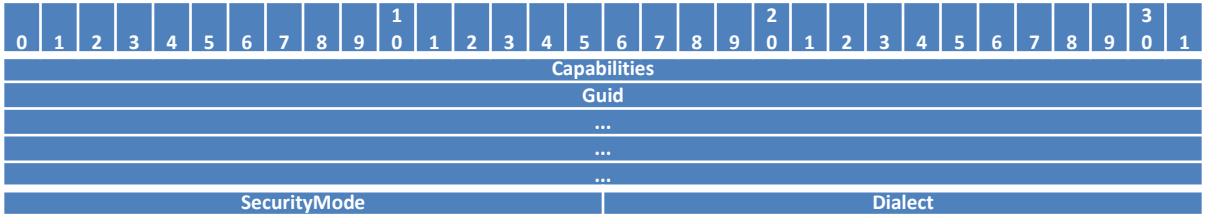


IOCTL/FSCTL

- 2 new SMB-specific ioctls:
  - FSCTL_VALIDATE_NEGOTIATE_INFO
    - Protects against man-in-the-middle negotiation downgrade
      - Repeats the negotiate process on signed/sealed e2e channel
      - Signed response (even error) used to verify e2e result
    - Validates no downgrade of dialect, security mode, capabilities, server id
  - FSCTL_QUERY_NETWORK_INTERFACE_INFO
    - Multichannel discovery, including RDMA
- FSCTL_DFS_GET_REFERRALS_EX
  - DFS SiteName
- Offload read/write and Trim support
  - Passed to storage subsystem

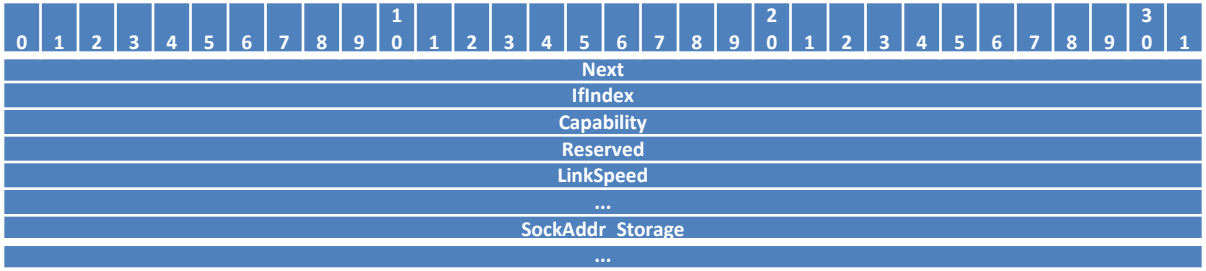
VALIDATE_NEGOTIATE_INFO

The VALIDATE_NEGOTIATE_INFO response is returned to the client by the server in an SMB2 IOCTL response for FSCTL_VALIDATE_NEGOTIATE_INFO request. The response is valid for servers which implement the SMB 3.0 dialect, and optional for others.



NETWORK_INTERFACE_INFO

The NETWORK_INTERFACE_INFO is returned to the client by the server in an SMB2 IOCTL response for FSCTL_QUERY_NETWORK_INTERFACE_INFO request. The interface structure is defined as following.
IfIndex (4 bytes): This field specifies the network interface index.
Capability (4 bytes): This field specifies the capabilities of the network interface.
LinkSpeed (8 bytes): The field specifies the speed of the network interface in bits per second.



Value	Meaning
RSS_CAPABLE 0x00000001	When set, specifies that the interface is RSS-capable.
RDMA_CAPABLE 0x00000002	When set, specifies that the interface is RDMA-capable.

FileSystem Control

- New sector info query
  - Storage alignment optimization
- (and the ioctls above)
- Passthrough to filesystem
  - MS-FSCC/MS-FSA

Encryption/signing

- Signing and integrity
  - New signing algorithm: AES-CMAC-128 (RFC4493)
    - More efficient than HMAC-SHA256
    - Compatible with encryption when used
  - Per-session signing key
- Encryption (optional)
  - AES128-CCM
  - Per-session encryption and decryption keys
  - Can be enabled per-share or whole server
- Encrypts individual requests on single connection
  - Interesting packet traces ☺

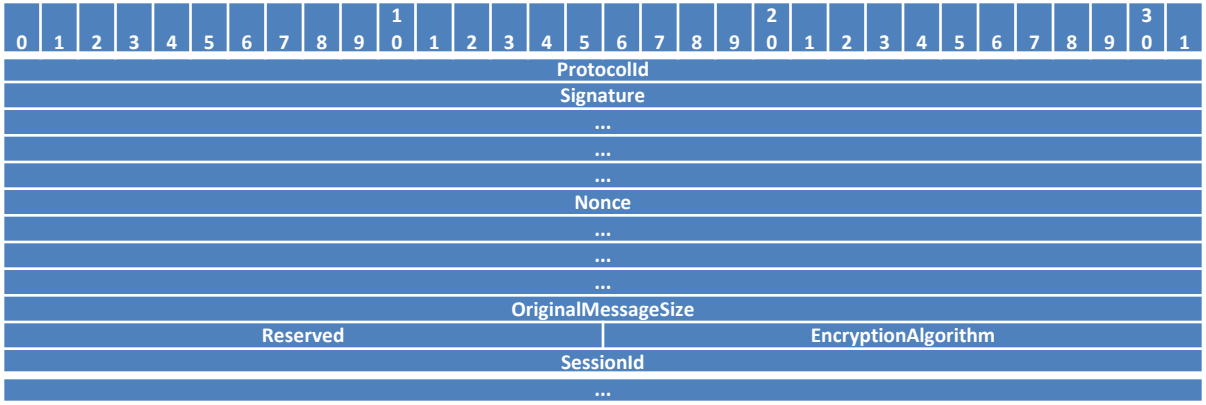
TRANSFORM_HEADER

- New SMB2_TRANSFORM_HEADER prepended to request/response
- Transmitted in clear, followed by encrypted SMB message
- Encryption – extensible to multiple algorithms via token in transform header
  - AES128-CCM only supported currently
  - Not "negotiated"

Transform header

The SMB2 Transform Header is used by the client or server when sending encrypted messages.
The SMB2 TRANSFORM_HEADER is only valid for the SMB 3.0 dialect.

ProtocolId (4 bytes): The protocol identifier. The value MUST be (in network order) **0xFD**, 'S', 'M', and 'B'.



EncryptionAlgorithm (2 bytes): The algorithm used for encrypting the SMB2 message.
This field MUST be set to one of the following values:

Value	Meaning
SMB2_ENCRYPTION_AES128_CCM 0x00000001	The message is encrypted by using the AES128 algorithm.

Miscellaneous

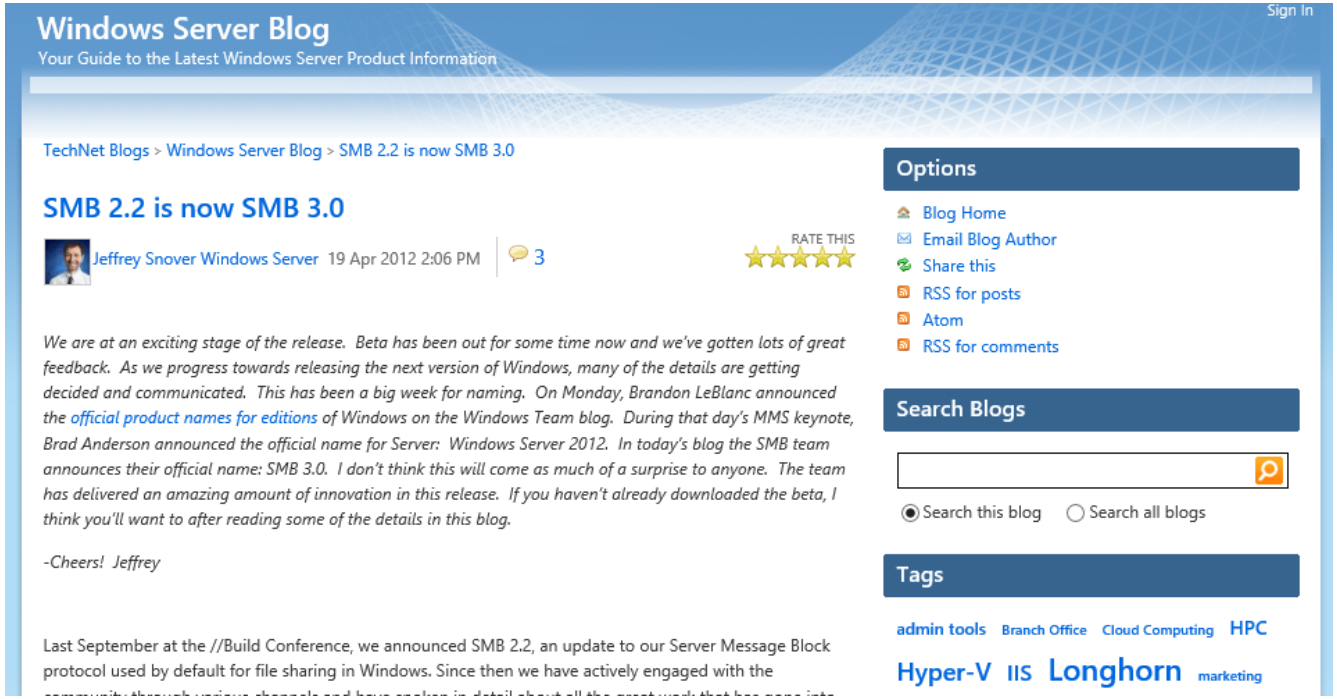
- Lease break v2: adds epoch
- Multichannel behavior:
  - Now a “many-to-many” association - was multiple sessions on a connections, now multiple sessions on a channel and multiple channels on a session

Feature Mapping

- Pivot from protocol changes to SMB3 features
- What high-level features map to these protocol primitives?
- Motivate staged development approaches

High-level SMB 3.0 Features

- From blog:
  - http://blogs.technet.com/b/windowsserver/archive/2012/04/19/smb-2-2-is-now-smb-3-0.aspx



Server Applications

SMB for Server Applications

> "Many of the new SMB features are specifically designed for server applications that store the data on file shares—for example, database applications such as Microsoft SQL Server or virtualization software such as Hyper-V. This allows applications to take advantage of advances in storage management, performance, reliability, and cost efficiency that come with SMB to deliver an application storage solution that rivals traditional Fibre Channel storage solutions in features and capabilities, but remains easier to provision and less expensive to implement."

- Need to answer these requirements individually
  - Application profiles – under consideration

Scale Out

Active file sharing with SMB Scale Out

> "Enables customers to scale share bandwidth by adding cluster nodes, as the maximum share bandwidth is the aggregate bandwidth of all file server nodes and not restricted to the bandwidth of a single cluster node as in previous versions. Scale-out file shares also makes it much easier to manage a file server cluster, as it is no longer necessary to create multiple clustered file servers, each with separate cluster disks, to take advantage of all nodes in a cluster. Further, the administrator can transparently redirect SMB client connections to a different file server cluster node to better balance the cluster load."

- Server Share CONTINUOUS_AVAILABILITY, SCALEOUT and/or CLUSTER attributes
- Client and Server PERSISTENT_HANDLE capability attribute
- Witness (MS-SWN) support

Multichannel

Fast data transfers and network fault tolerance with SMB Multichannel

> "Given that customers can now store server application data on remote SMB file shares, SMB was enhanced to improve network performance and reliability. SMB Multichannel takes advantage of multiple network interfaces to provide both high performance through bandwidth aggregation, and network fault tolerance through the use of multiple network paths to data on an SMB share."

- Server and client MULTICHANNEL capability attribute
- 2 or more equivalent interfaces

SMB Direct (SMB 3.0 over RDMA)

Scalable, fast, and efficient storage access with SMB Direct

> "SMB Direct (SMB over Remote Direct Memory Access (RDMA)) is a new transport protocol for SMB in Windows Server 2012. It enables direct memory-to-memory data transfers between servers, with minimal CPU utilization and low latency, using standard RDMA-capable network adapters (iWARP, InfiniBand, and RoCE). Any application which accesses files over SMB can transparently benefit from SMB Direct. Minimizing the CPU cost of file I/O means application servers can handle larger compute workloads with the saved CPU cycles (for example, Hyper-V can host more virtual machines)."

- Server and client Multichannel with RDMA_CAPABLE interface(s)
- SMB Direct (MS-SMBD) support

I call your attention to…

- This week at Interop:
  - SMB Direct - 5.8GBps (Giga BYTES/s) on single port
  - http://blogs.technet.com/b/josebda/archive/2012/05/06/windows-server-2012-beta-with-smb-3-0-demo-at-interop-shows-smb-direct-at-5-8-gbytes-sec-over-mellanox-connectx-3-network-adapters.aspx

Workload: 512KB IOs, 8 threads, 8 outstanding

Configuration	BW MB/sec	IOPS 512KB IOs/sec	%CPU Privileged
Non-RDMA (Ethernet, 10Gbps)	1,129	2,259	~9.8
RDMA (InfiniBand QDR, 32Gbps)	3,754	7,508	~3.5
RDMA (InfiniBand FDR, 54Gbps)	5,792	11,565	~4.8
Local	5,808	11,616	~6.6

http://smb3.info

Workload: 8KB IOs, 16 threads, 16 outstanding

Configuration	BW MB/sec	IOPS 8KB IOs/sec	%CPU Privileged
Non-RDMA (Ethernet, 10Gbps)	571	73,160	~21.0
RDMA (InfiniBand QDR, 32Gbps)	2,620	335,446	~85.9
RDMA (InfiniBand FDR, 54Gbps)	2,683	343,388	~84.7
Local	4,103	525,225	~90.4

Failover

Transparent Failover and node fault tolerance with SMB

> "Supporting business critical server application workloads requires the connection to the storage back end to be continuously available. The new SMB server and client cooperate to make failover of file server cluster nodes transparent to applications, for all file operations, and for both planned cluster resource moves and unplanned node failures."

- Server Share CONTINUOUS_AVAILABILITY / CLUSTER
- Client and server GLOBAL_CAP_PERSISTENT_HANDLES
- Witness (MS-SWN)
- Client and Server Application ID context

Volume ShadowCopy (Snapshots)

VSS for SMB file shares

> "VSS for SMB file shares extends the Windows Volume ShadowCopy Service infrastructure to enable application-consistent shadow copies of server application data stored on SMB file shares, for backup and restore purposes. In addition, VSS for SMB file shares enables backup applications to read the backup data directly from a shadow copy file share rather than involving the application server in the data transfer. Because this feature leverages the existing VSS infrastructure, it is easy to integrate with existing VSS-aware backup software and VSS-aware applications like Hyper-V."

- Remote VSS protocol (MS-FSRVP)
- No SMB3 requirements (!)
- Coordination from snapshots to sharenames

Encryption

Secure data transfer with SMB encryption

> "SMB Encryption protects data in-flight from eavesdropping and tampering attacks. Deployment is as simple as checking a box, with no additional setup requirements. This becomes more critical as mobile workers access data in centralized remote locations from unsecured networks. SMB Encryption is beneficial even within a secured corporate network if the data being accessed is sensitive."

- Client and server support of new signing and encryption AES algorithms
- Server and client GLOBAL_CAP_ENCRYPTION

Directory Leasing

Faster access to documents over high latency networks with SMB Directory Leasing

> "SMB Directory Leasing reduces the latency seen by branch office users accessing files over high latency WAN networks. This is accomplished by enabling the client to cache directory and file meta-data in a consistent manner for longer periods, thereby reducing the associated round-trips to fetch the metadata from the server. This results in faster application response times for branch office users."

- Filesystem support for directory leasing
- Client and server support of DIRECTORY_LEASING capability

SMB Ecosystem

SMB Ecosystem

> “A critical aspect of Windows Server 2012 development is the partnership we have established with vendors to ship SMB 3.0 capable systems. We have been working closely with several server vendors and open source partners over the past year, by proactively providing extensive protocol documentation and numerous open “plugfest” events provide opportunities for test and feedback. Finally, and most importantly, the SMB ecosystem now reaches all the way to key server applications such as SQL Server and Hyper-V to ensure that SMB 3.0 capabilities are fully leveraged all the way through the stack, and across the multivendor network.”

- Use the documents and join the plugfests!

Also Consider…

- Storage subsystem:
  - Offload read/write, Trim
- FileSystem:
  - Enhanced leasing
- FSCC operations
- BranchCache v1/v2
- DFS:
  - site awareness


SMB 3.0
BIGGER. FASTER.
SCALIER.

Summary

- Relatively simple pieces
- Huge result when taken together
- Can take on in phases
- Consider protocol capabilities and attributes
  - On client and server separately
  - One at a time