SMB3 Protocol Document 19H1 Changes

Tom Talpey
Windows and Windows Server “19H1” release

- A.k.a. Windows 10 version 1903
- May 22, 2019

Updated doc March 13

- Corrections/updates April 30
- [https://docs.microsoft.com/en-us/openspecs/windows_protocols/ms-smb2/5606ad47-5ee0-437a-817e-70c366052962](https://docs.microsoft.com/en-us/openspecs/windows_protocols/ms-smb2/5606ad47-5ee0-437a-817e-70c366052962)

Also covering 18H2 / Windows Server 2019

- Since it’s a year since we met here!
- Largely maintenance – no protocol changes
SMB3 Changes

New SMB3 features (negotiate contexts)
- Compression
- Server netname

No dialect change
- No dialect bump foreseen
- Since SMB2/3 now has forward-compatible contexts in
  - Negotiate
  - Tree Connect
- Any unrecognized contexts are ignored
  - Not errored, and not returned in response
Compression

New negotiate context SMB2_COMPRESSION_CAPABILITIES
- MS-SMB2 section 2.2.3.1.3 (request) and 2.2.4.1.3 (response)
- ID 0x0003

New SMB2_COMPRESSION_TRANSFORM_HEADER
- New transform specifically for compression
- MS-SMB2 section 2.2.42

Also SMB2_READFLAG_REQUEST_COMPRESSED
- New flag in SMB2_READ request
- MS-SMB2 section 2.2.19
## Negotiable SMB Traffic Compression

Client optionally negotiates compression by appending negotiation context (ID = 0x0003)

<table>
<thead>
<tr>
<th>Algorithm Count</th>
<th>Algorithm Id 1</th>
<th>Algorithm Id 2</th>
<th>Algorithm Id 3</th>
<th>......</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 Byte</td>
<td>2 Byte</td>
<td>2 Byte</td>
<td>2 Byte</td>
<td></td>
</tr>
</tbody>
</table>

Supporting server selects subset of compression algorithms, if any, and responds with:

<table>
<thead>
<tr>
<th>n</th>
<th>Selected Algorithm Id 1</th>
<th>......</th>
<th>Selected Algorithm Id n</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 Byte</td>
<td>2 Byte</td>
<td></td>
<td>2 Byte</td>
</tr>
</tbody>
</table>

Compression algorithms defined in MS-XCA:

- XPRESS (also known as LZ77)
- XPRESS Huffman (LZ77+Huffman)
- LZNT1
Compression + Signing/Encryption Interop

New, compact transform header for SMB Compression (16B)

<table>
<thead>
<tr>
<th>Protocol ID</th>
<th>Original Segment Size</th>
<th>SMB2 HEADER and other payload ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithm</td>
<td>Reserved</td>
<td>Compression Offset</td>
</tr>
</tbody>
</table>

When compression and signing or encryption are needed, transform headers are nested. Compress always first: regular transform header always the *outer* transform header.

| SMB Transform Header | SMB Compression Transform Header | SMB2 HEADER and other payload ... |
Compression processing

MS-SMB2 section 3.1.4.4
Choice of compression types by sender, on each operation
  • As appropriate to type of data, performance, etc

Compress Writes, and requesting compress Reads for client
  • As appropriate to type of data, performance, etc
  • Client may further hint with SMB2_READFLAG_REQUEST_COMPRESSED

CompressAllRequests override for client
Not over RDMA (for now)
Decompression processing

MS-SMB2 sections 3.2.5.1.10 / 3.3.5.2.1.2
Drops connection on fail (size mismatch)
Inevitably drops connection on garbage
Compression commentary

It’s optional!
- Doesn’t compress if payload not smaller
- Only compresses “large” “data-bearing” operations
- Separate decision on both client and server, on each operation sent

Compress before encrypt
- Encrypted data compresses badly
- Note, some encryptions also compress – implementation consideration

Optional to compress SMB headers
- Offset field may point into “middle” of payload
- Windows compresses data-only at ~4KB+
SMB Compression performance under 100Mbps network with EXPRESS using Intel Xeon W3520
Compression Performance Scaling

SMB Compression performance under 200Mbps network with EXPRESS using Intel Xeon W3520

- Patterned Data: 200 (No Compression), 544 (With Compression)
- Random Data: 200 (No Compression), 232 (With Compression)
Compression Use Cases

Reads and Write
  • Not metadata and IOCTL/FSCTL, but possible on any operation

Bulk data on long-haul

Specialized local transfers
  • File copy, migration, etc

Client opt-in
  • Used only in scenarios which might benefit

Server opt-in
  • Used only in responses which do benefit

Future change possible (implementation choice)
Compression future

Alternative compression algorithms
- Hyper-V / VHDX optimized?
  - RLL type algorithm for all-zero blocks is perhaps appealing
  - Still a per-operation and per-payload decision

Interaction with encryption, transport, etc
- Compression when encryption implements
  - Cf. not signing when using authenticated encryption
- Compression over RDMA may have different goals
  - RDMA transport changes the benefit equation
Netname Negotiate Context

Client provides target servername by appending negotiation context (ID = 0x0005)

<table>
<thead>
<tr>
<th>Name length</th>
<th>Unicode null-terminated name</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 Byte</td>
<td>Variable</td>
</tr>
</tbody>
</table>

Provides servername
- Advisory, available prior to session and treeconnect processing

Ignored by Server processing (perhaps surprisingly?)
- May be inspected by load balancers, connection managers, failover, etc
- Server generates no context response
Netname Negotiate Context

SMB2_NETNAME_NEGOTIATE_CONTEXT_ID
- MS-SMB2 Section 2.3.1.4 (request only)
- 0x0005

Included with SMB2_NEGOTIATE by default
- MS-SMB2 section 3.2.4.2.2
- No server processing (no document 3.3.x section)
Updates to the Microsoft SMB3 client

**FileNormalizedNameInformation**
- Normalized Name query added to protocol

**FileIdInformation**
- Omitted in 3.x [oops!] (3.3.5.20.1)
  - Issued by Win8+ clients, but error ignored

**Directory Caching Enhancements**
- Can now cache much larger directories ~ 500K entries.
- Will attempt directory queries with 1 MB buffers to reduce round trips and improve performance

**Accelerated IO path for low latency access**
Other MS-SMB2 Document Changes

MS-XCA normative reference added
  • For compression

Numerous clarity and language tweaks
  • FSCTL input and output counts
  • Transform processing order, invalid protocol id’s
    • New section reorg in April 30 update see 3.2.5.1.1 / 3.3.5.2.1 and subsections
  • Oplock/Lease break client processing
    • Previously omitted
  • Tree connect and redirect
  • Durable reconnect v2 (3.3.5.9.12)
  • Compound processing (18H2 document)
Questions?
Thank you.