SDC 18
September 24-27, 2018
Santa Clara, CA
www.storagedeveloper.org

SMB3 directions and landscape

Mathew George, Wen Xin
Microsoft Corporation

SMB3 in use today

- Software Defined Datacenter (SDDC)
  - Storage Spaces Direct (Block over SMB3)
  - Scaleout File Server for Application workloads
- Containers
  - Container to host file access
- Cloud Scale Storage
  - Azure Files

What's Coming ?

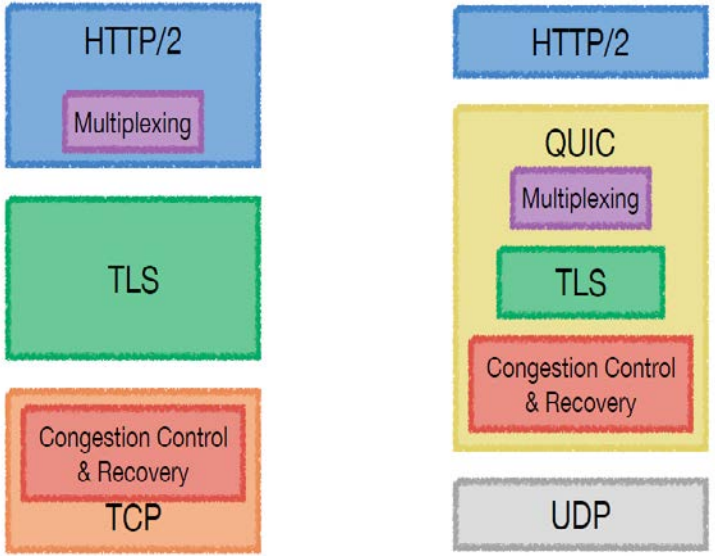
- SMB over QUIC
  - Prototype under development
- New transforms and signing
  - Compression
  - AES-GMAC signing.
  - Signing and RDMA
- RDMA direct access to persistent storage.

Updates to the Microsoft SMB3 stack

- Normalized Name query added to protocol
  - Native support for FileNormalizedNameInformation
- Directory Caching Enhancements
  - Windows clients 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 (on SMB client) for low latency access.

QUIC:UDP based secure stream transport

- Low-latency connection setup
  - 1-RTT for initial connections
  - 0-RTT for repeat connections.
- Secure and Encrypted (TLS 1.3+)
- Improvements over HTTP/2 (“H2”) and TCP
  - Multiple Stream Support
  - ALPN for better multiplexing
  - Support for connection migration across
  - Better congestion control & loss recovery
  - UDP based library implementation
- IETF draft stage.



QUIC - Unknowns

- Still experimental
  - Evidence (Google) shows that it is firewall/NAT friendly – 93%
- Initial implementations are software only
  - Will it catch up with TCP offload ?
  - RDMA over QUIC ?
- Still in development
  - Interoperability concerns

SMB Bindings for QUIC

- QUIC connections can share same 4-tuple.
  - Can multiplex using an ALPN identifier.
  - Can share same port with HTTPS traffic
- Use QUIC as a single channel TCP replacement
  - SMB multichannel will use separate QUIC connections.
- Can QUIC be hooked up to Azure Files ?
  - No more port 445 blocking !

SMB3 Signing – Enabling AES-GMAC

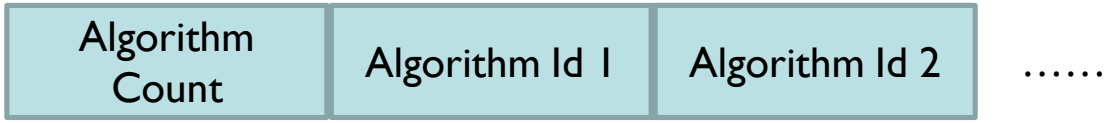
- Switch from AES-CCM to AES-GCM cipher.
  - AES-GCM based SMB3 encryption performs significantly better than AES-CCM based signing.
  - Most modern processors have optimized instructions for AES-GCM computations.
- Can we use the transform headers for signing ?
  - Better layering and unification of encryption and signing.
  - Extra cost of buffering entire message and copy.

Signing & Encryption with RDMA

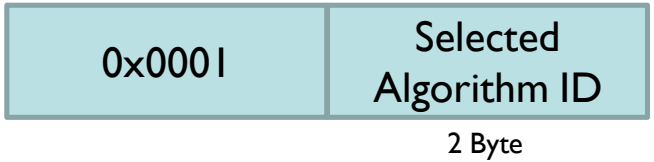
- Why would someone want (or do) this ?
  - Security trumps everything else.
  - Customers don't often understand the impact.
- What happens when you turn on security features?
  - Performance is worse than using TCP (w/ offload)
  - RDMA direct placement is disabled.
  - Fragmentation/reassembly done in software.
- Can we retain direct placement by separately signing / encrypting the RDMA payload from the SMB message ?

Negotiable SMB Signing with New Algorithm

- Negotiable
  - Client will be able to negotiate for using the AES128-GMAC algorithm for signing in SMB 3.1.1. Client must append negotiation context (ID = 0x0008) specifying the algorithm count and algorithm IDs:



  - Supporting server will select 1 signing algorithm, if possible, and respond with:



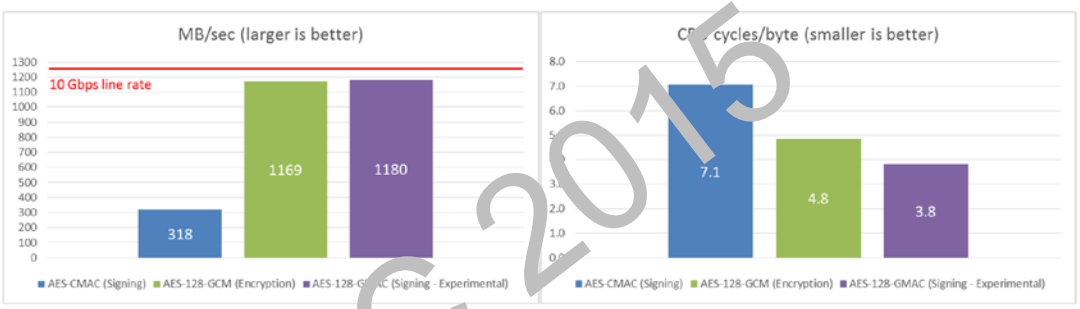
  - The only supported algorithm is AES128-GMAC, more to be added as we adopt any other algorithms.

Negotiable SMB Signing with New Algorithm

- AES128-GMAC and some optimization
  - In SMB 3.1.1 we introduced support for AES128-CMAC signing, we have now prototyped support for AES128-GMAC.
    - Significant performance gain over AES128-CMAC.
    - We include a nonce with each packet to better prevent replay attacks.
  - Signature validation logic on both client and server has been moved to lower layers for faster signature rejection using less resources (DOS attacks, etc).
  - New signing logic prepends the SMB TRANSFORM_HEADER (see [MS-SMB2]) to the payload, with the Flags/Encryption Algorithm field set to 0x0002.

AES-GMAC expected performance

7 – AES-GMAC file copy performance

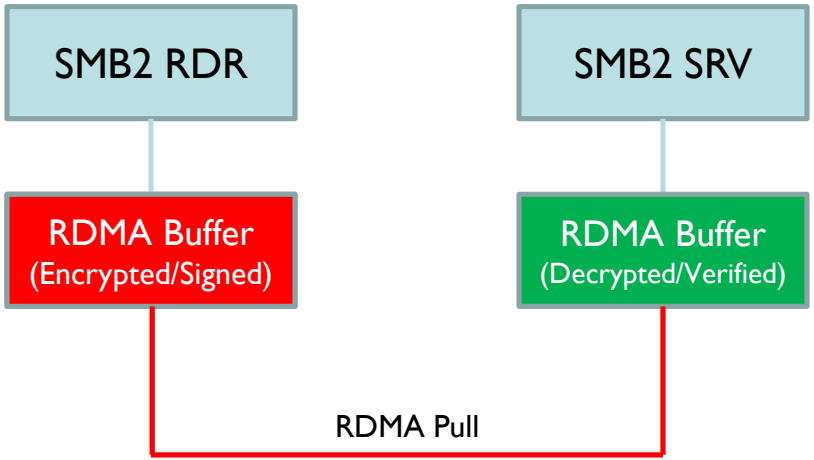


- AES-GMAC results in significant performance improvements!
  - 46% reduction in Cycles/Byte compared to AES-CMAC
  - 21% reduction in Cycles/Byte compared to AES-GCM
- Prototype focused on functional correctness not performance
  - We identified several fairly easy improvements that could be made to further decrease CPU cycles/byte.

Better Signing and Encryption in RDMA

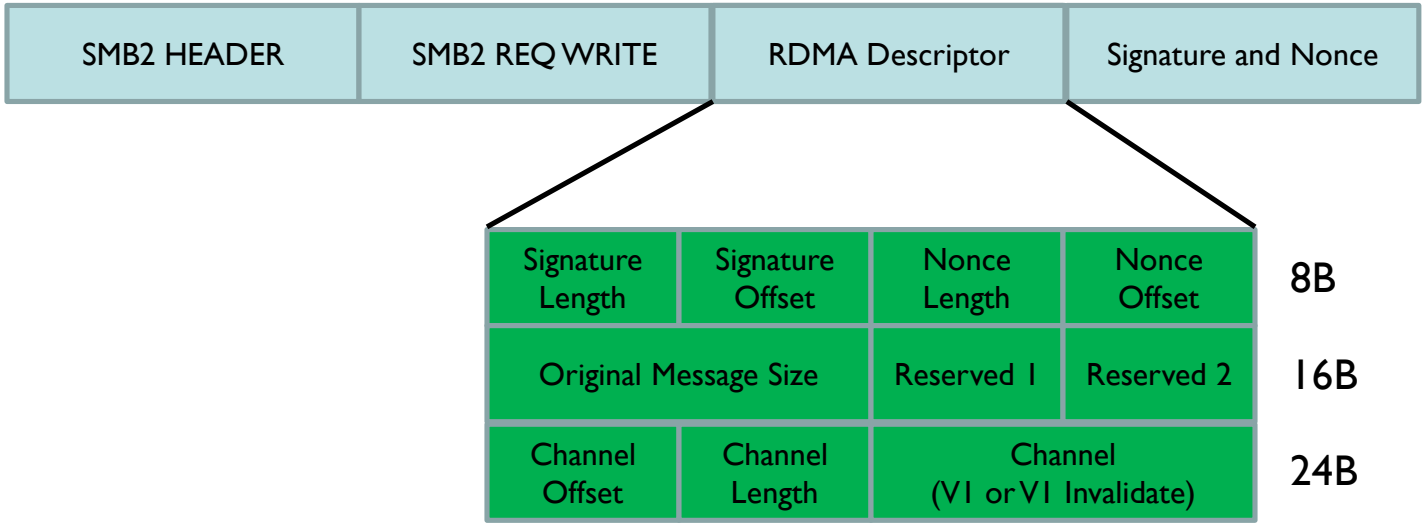
- Signing and Encryption over SMB RDMA.
  - Performance gain over current packet-based authenticated and/or encrypted traffic over SMB RDMA.
  - Supports AES128-GMAC for signing, AES-CCM and AES-GCM for encryption.

E.g. An SMB RDMA write:



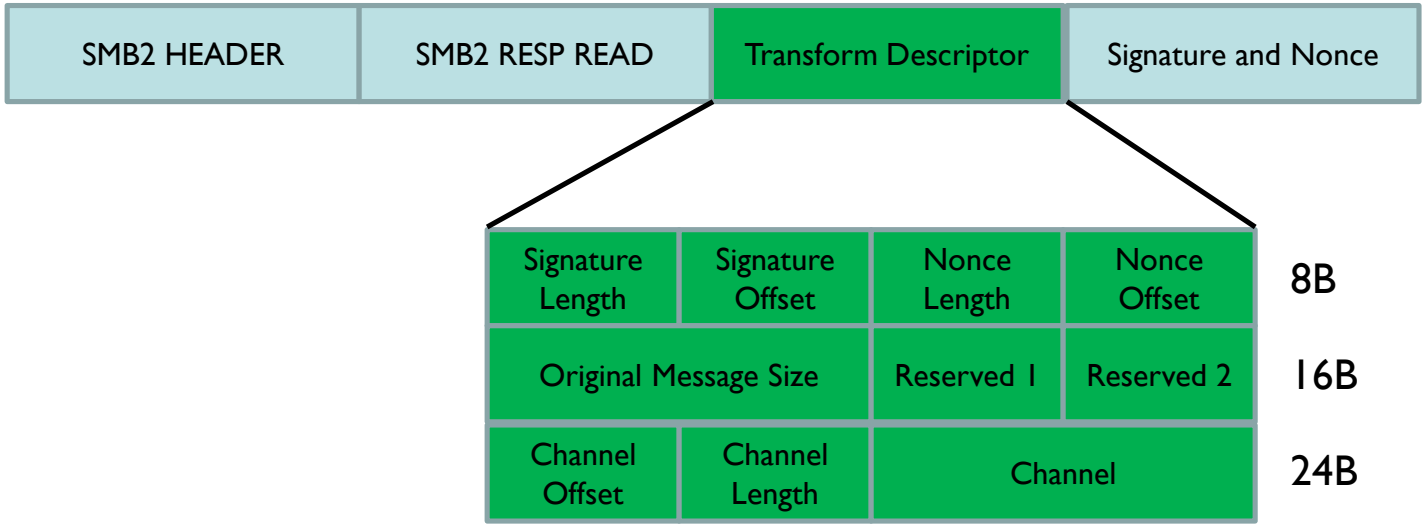
Better Signing and Encryption in RDMA

- How are signature and nonce transmitted?
- Introducing our Transform Descriptor! (Channel Type 0x0003)
- Recall the SMB write request packet for RDMA:



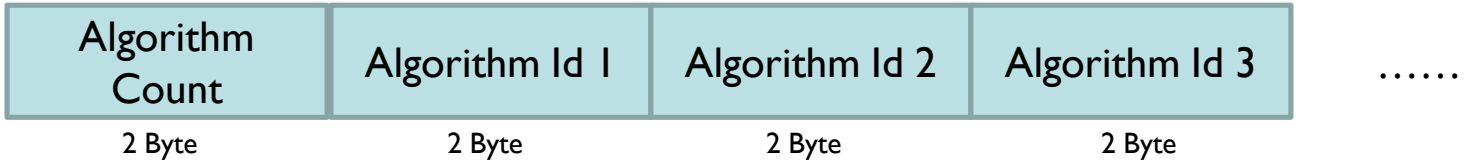
Better Signing and Encryption in RDMA

- Similar transform descriptor should also be used with SMB2 Read Response, the difference is that the RDMA descriptor should not follow.

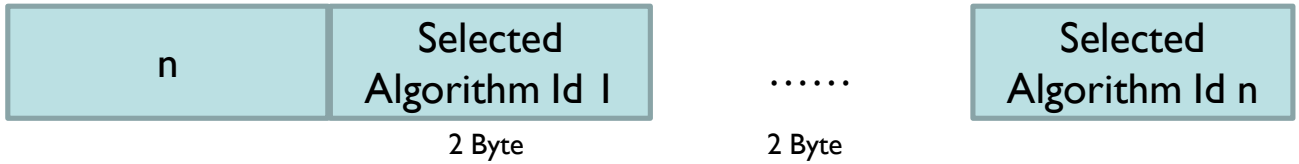


Negotiable SMB Traffic Compression

- Negotiable
  - Client will be able to negotiate for compression. To do so client must append the negotiation context (ID = 0x0004)



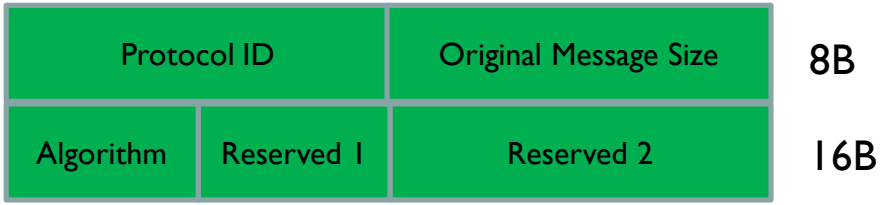
  - Supporting server will select a subset of compression algorithms, if possible, and respond with:



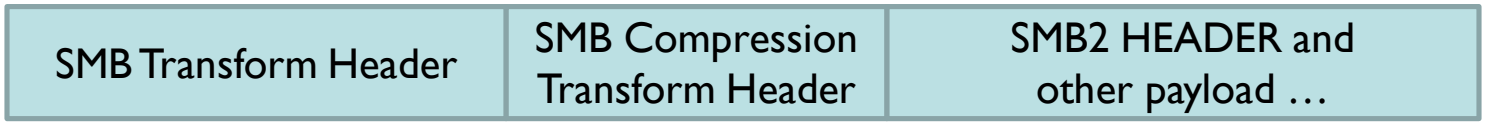
  - The supported compression algorithms are XPRESS (also known as LZ77), XPRESS Huffman (LZ77+Huffman) and LZNT1 (as defined in [MS-XCA]).

Compression + Signing/Encryption Interop

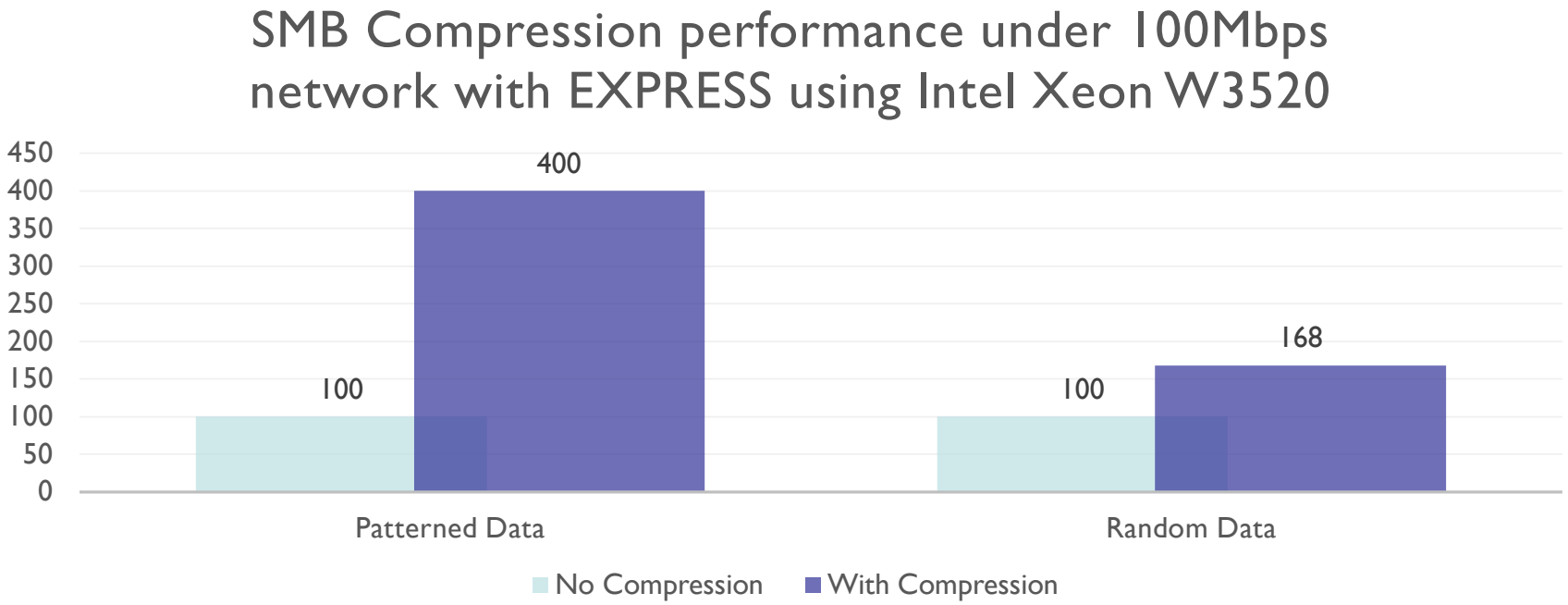
- New, compact transform header for SMB Compression.



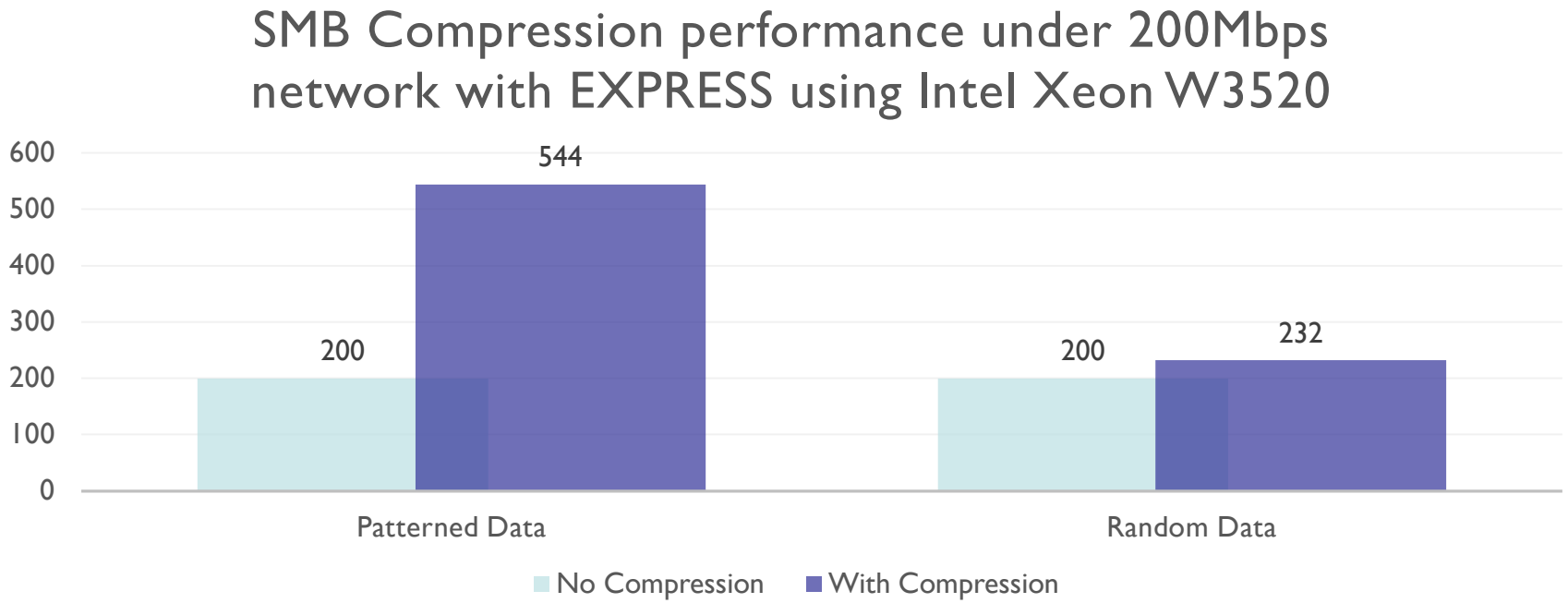
- When compression and signing or encryption are needed, we nest the transform headers. Since we always compress first, the regular transform header will always be the outer transform header.



Compression Performance



Compression Performance



Key Takeaways

- Lot of what we talked is work in progress
  - Things can (and will) change!
  - Watch out for updates to [MS-SMB2].
- SMB3 over the internet ?
  - More cloud implementations of SMB3.
  - Efforts to make the protocol more “internet friendly”.
- Performance and security improvements for SDDC
- SMB1 deprecation continues to make progress.