SDC 19

September 23-26, 2019
Santa Clara, CA

Surfing the Worldwide File: SMB3 improvements for safe and efficient internet access

Frank Li

Andy Ruiz Cabrera
Microsoft Corp

First, Some Updates

SDC 2018 Coverage

- 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
- SMB3 Signing, Switch from AES-CCM to AES-GCM based.
- Negotiable SMB Traffic Compression.
  - The supported compression algorithms are XPRESS (also known as LZ77), XPRESS Huffman (LZ77+Huffman) and LZNT1 (as defined in [MS-XCA]).
- Negotiable SMB netname context for multitenant servers.

SMB3 Developments in Progress

- Pattern Scanned Compression algorithm.
- Storage RDMA Push Mode to Persistent Memory. (Talk in SDC 2019)
- New Signing/Encryption algorithms AES-256-GCM and AES-256-CCM.
- Watch out for updates to [MS-SMB2].
- Encryption of RDMA payload for direct placement SMB2 reads/writes

Topics

- Why?
- QUIC Overview
- SMB over QUIC Overview
- Integration
- Certificate Management
- Performance
- Demo

Why

Problem

- SMB cannot be easily used over the internet
  - Cannot be used to access cloud providers
  - Need for VPNs for remote traffic

QUIC Overview

High Level Overview
QUIC

- New secure networking transport on top of UDP. Basis for HTTP/3.
- Faster connection set up
  - 1 Round Trip(1-RTT) for initial connections.
  - 0-RTT for resumed connections.
- Better Security
  - Provides mutual auth, TLS 1.3
  - Mitigates MITM attacks as most of the packet is encrypted

QUIC Overview Continue

- Improvements over HTTP2 and TCP
  - No head-of-line blocking
  - Better transitions between networks
  - Loss recovery
    - Up to 256 SACK blocks vs up to 4
    - No retransmission ambiguity
    - Latest congestion control
  - Portability and agility (UDP and user-mode)
- Being standardized by IETF

SMB over QUIC Overview

Advantages

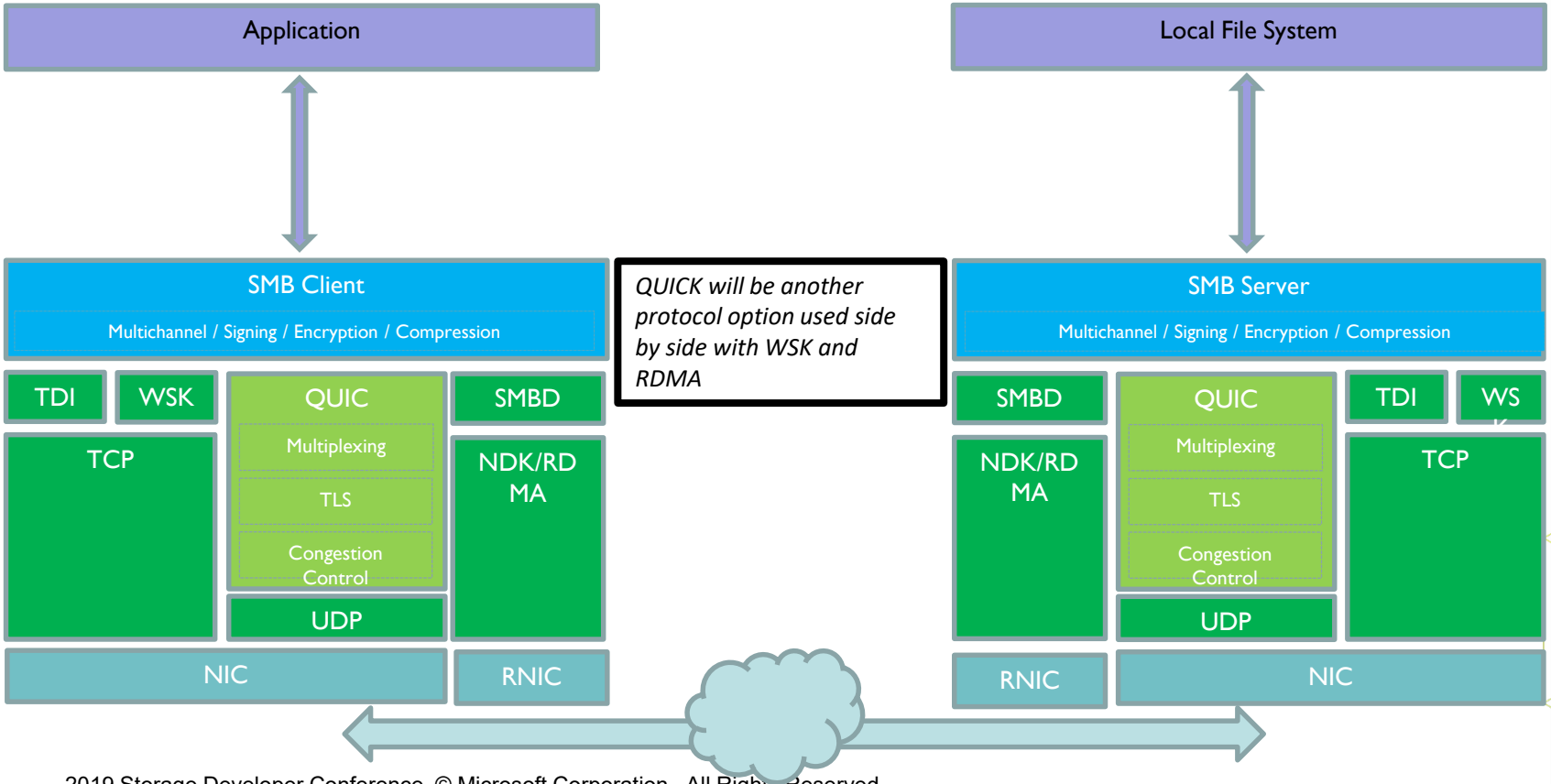
- Security
  - Prevents Server Spoofing
  - QUIC connection is always protected by TLS 1.3 encryption.
- Portability
  - Cases where SMB Port: 445 is blocked, SMB over QUIC (UDP:443) will work
  - Google experiments show a 93% connection success rate when connecting to port 443.
  - Windows kernel QUIC implementation allows multiplexing of multiple protocols on UDP port 443 using ALPN.

Downsides

- Poorer perf than SMB-encrypted
  - No hardware offload support
- Kerberos cannot be used without means to reach KDC from the client so it defaults to NTLM
  - NTLM exchanges are tunneled inside TLS 1.3 connection.
- TLS Encryption is machine to machine rather than user to machine.

Implementation/Integration

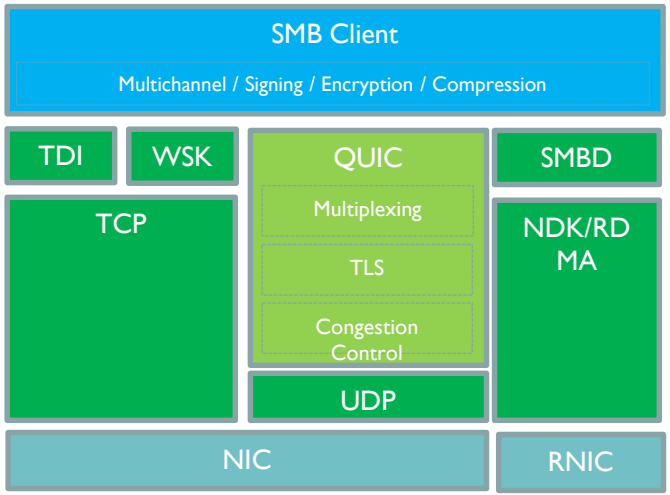
SMB/QUIC: Components



SMB Protocol Stack

- SMB will be layered on top of the QUIC stack.
  - No difference for multichannel
  - No SMB signing/encryption by default
  - SMB over QUIC will use the server certificate to make sure there is no server spoofing attack.
  - No changes to SMB authentication
  - QUIC multisession is not used on the server.
  - Negotiable SMB Connection Setting context for secure connections
    - Client must append the negotiation context ID=0x0006 to learn if the transport layer security is accepted.

SMB/QUIC: Client.



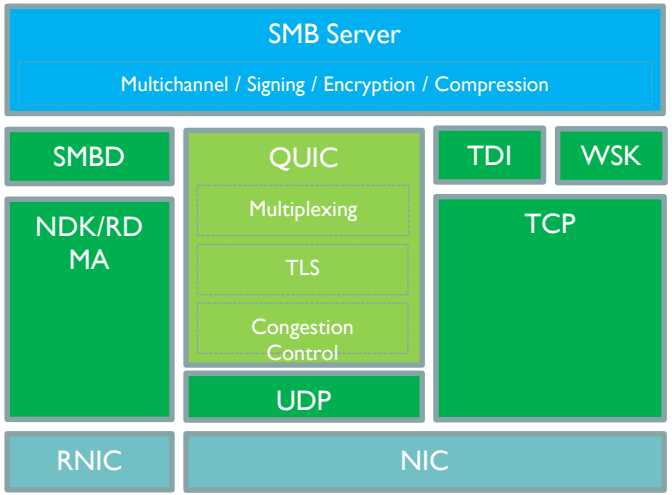
1. *Client opens \\ServerName\Share\foo.tst*
2. *Client resolves ServerName using DNS*
3. *Client fires parallel connect attempts using TCP/IP and QUIC*
4. *Client will start using whatever channel was connected first*
5. *Client's multichannel will negotiate interfaces with server and will select most optimal protocols*
6. *Client send streams/SMB messages*

- Client does not know if server supports QUIC at all or supports only TCP or only QUIC so it have to attempt both.
- By default TCP/IP is given a head start to establish a connection.
- Mount option

SMB/QUIC: Client

```
Status = MsQuic->SessionOpen( Registration,
                              ALPN_SMB_OVER_QUIC,
                              NULL,
                              &VcEndpoint->QUIC.Session );
 Status = MsQuic->ConnectionOpen( VcEndpoint->QUIC.Session,
                             SmbQuicClientConnectionCallback,
                             VcEndpoint,
                             &VcEndpoint->QUIC.Connection );
Status = MsQuic->StreamOpen( VcEndpoint->QUIC.Connection,
                         QUIC_STREAM_OPEN_FLAG_NONE,
                         SmbQuicClientStreamCallback,
                         VcEndpoint,
                         &VcEndpoint->QUIC.Stream );
Status = MsQuic->StreamStart( VcEndpoint->QUIC.Stream,
                              QUIC_STREAM_START_FLAG_NONE );
```

SMB/QUIC: Server



1. *Server opens endpoints listening on UDP 443*
2. *Server receives new QUIC connection requests*
3. *Server finds the certificate for the new QUIC connection*
4. *Server accept the connection*
5. *Server receives QUIC streams/SMB messages*

- Server starts both TCP/IP and QUIC listeners by default.
- Server can selectively start TCP/IP or QUIC listeners or both.

SMB/QUIC: SERVER

```
Status = MsQuic->ListenerOpen( SingleSession,
                               SrvNetQuicServerListenerCallback,
                               Endpoint,
                               &Endpoint->QUIC.Listener );
Status = MsQuic->ListenerStart( Endpoint->QUIC.Listener,
                                &addr );

 MsQuic->SetCallbackHandler(
       Event->NEW_CONNECTION.Connection,
       SrvNetQuicServerConnectionCallback,
       Connection );

 MsQuic->SetCallbackHandler( Event->PEER_STREAM_STARTED.Stream,
                               SrvNetQuicServerStreamCallback,
                               Context );
```

SMB/QUIC: Send

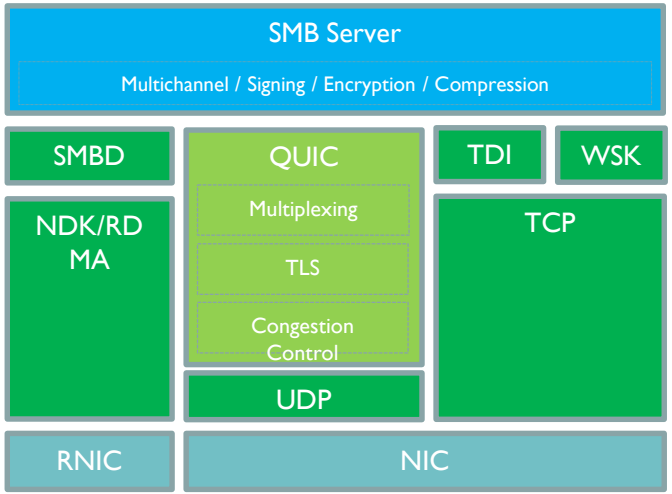
```
Status = MsQuic->StreamSend( AsyncSendContext->VcEndpoint->QUIC.Stream,
                             pQuicBuffer,
                             bufferCount,
                             QUIC_SEND_FLAG_NONE,
                             AsyncSendContext );

Status = MsQuic->StreamSend(
                            Connection->QUIC.QuicStream,
                            QuicBuffer,
                            Count,
                            QUIC_SEND_FLAG_NONE,
                            sendParams );
```

SMB/QUIC: Receive

- The server or client receive the QUIC events through stream call back.
- The client or server process the smb message.

Server Certificate Mapping Management

SMB/QUIC: Server Certificate Mapping



- *Introduced Server Certificate mapping for server.*
- *Server supports multiple Secure Principal Names.*
- *SMB Server creates a listening QUIC socket, but it does not automatically register any certificates so any attempt to connect will fail at TLS.*
- *By default SMB server assumes there are no certificates associated with the principal names.*
- *User can associate a certificate with SPN using new management utility. Certificate will be installed into local machine store, and registered with QUIC. From this point on connections using QUIC to that SPN will pass TLS.*
- *Or customer can disable the WSK listeners or disable TCP:445 on the corporate firewall. This will force QUIC when server is accessed from internet.*
- *Customer can also specify the connection type in client side.*

New SMB CMDLets

- `New-SmbServerCertificateMapping Function SmbShare ...`
- `Remove-SmbServerCertificateMapping Function SmbShare ...`
- `Get-SmbServerCertificateMapping Function SmbShare ...`

Common Scenarios

- The server certificate is in local system store.
- New-SmbServerCertificateMapping –Name serverDNSname –ThumbPrint xxxxxx
- Remove-SmbServerCertificateMapping –Name serverDNSname

Performance

Performance comparison with TCP

Storage : ram (ramdisk)
Two different machines - each with [2 25-Gbps Mellanox links, 208 GB Ram, 2 2.1 GHz Processors (16 core each)]
Filesize : 10GB
Time : 60 sec
RDMA turned off
WSK - SetSMBServerConfiguration -EncryptData [$True for TCP/IP, $False for QUIC]

	TCP/IP					QUIC				
	Avg IO/sec	Avg MB/sec	Med Latency (ms)	Avg Client CPU util	Avg Server CPU util	Avg IO/sec	Avg MB/sec	Med Latency (ms)	Avg Client CPU util	Avg Server CPU util
R-Read Block-Size:4K	131234	512	0.85	20.5	23.77182185	241479	943	0.548	39.3	47.81889045
R-Write Block-Size:4K	209028	816	0.559	29.9	31.67527888	267001	1042	0.448	42	56.8782974
R-Read Block-Size:1M	5487	5487	10.348	31.3	22.62082924	2376	2376	21.35	26.1	26.73773474
R-Write Block-Size:1M	5563	5563	11.853	18.1	20.81962855	2558	2558	19.448	30	22.33869402

For small size IOs, QUIC has better IOPs for both read and write than TCP/IP but consumed more CPUs.
For large size IOs, TCP/IP has better performances and the throughput is about 2x.

Demo

Q&A?