Windows Interoperability Workshop

Christopher R. Hertel
Samba Team

SNIA Legal Notice

- The material contained in this tutorial is copyrighted by the SNIA unless otherwise noted.
- Member companies and individual members may use this material in presentations and literature under the following conditions:
  - Any slide or slides used must be reproduced in their entirety without modification
  - The SNIA must be acknowledged as the source of any material used in the body of any document containing material from these presentations.
- This presentation is a project of the SNIA Education Committee.
- Neither the author nor the presenter is an attorney and nothing in this presentation is intended to be, or should be construed as legal advice or an opinion of counsel. If you need legal advice or a legal opinion please contact your attorney.
- The information presented herein represents the author's personal opinion and current understanding of the relevant issues involved. The author, the presenter, and the SNIA do not assume any responsibility or liability for damages arising out of any reliance on or use of this information.

NO WARRANTIES, EXPRESS OR IMPLIED. USE AT YOUR OWN RISK.

Abstract

Windows Interoperability for Storage Developers

This tutorial is designed to provide Team Leaders, Storage Developers, Protocol Specialists, the Lost and the Curious with an introduction to the Windows File Networking Ecosystem.

- This tutorial provides an overview of the SMB3 Network File Protocol—the heart and soul of Windows Interoperability—and describes some of the unique and powerful features of SMB3.
- Related protocols (such as Witness, Peerdist, and RSVD) will also be described, along with the mechanisms by which they are integrated with SMB3.
- This tutorial will also explain the general structure of Windows Protocols documentation, the best guide for those lost in the Windows Interoperability Jungle.

Objectives

- Our goal is to provide the fundamental information needed to understand Windows Interoperability in a Network Storage environment.
  - Get to know SMB3, and its Protocol Entourage.
  - Gain familiarity with the Windows Interoperability Ecosystem.
  - Learn how to leverage the Windows Protocols documentation.
  - Mind the Gap: POSIX vs. Windows semantics.

Assumptions

- We assume that this topic is of interest to you

  Maybe you want...
  - ...to improve a product
  - ...to maintain someone else's code
  - ...to stop repeatedly banging your head against the same weird problems
  - ...to explore strange new worlds

- We assume some developer know-how

  Geek Creds:
  - Basic client/server
  - Protocol Fundamentals
    - Byte-order
    - [Un]Marshalling
    - Statefulness
  - Familiarity with POSIX file system calls

Overview

- SMB
  - History, Terminology, and Background
  - Structure and Utility
- SMB as a Transport
  - RPC
  - RSVD (and beyond)
  - PeerDist (BranchCache)
- SMB as a Partner
  - Witness
- The POSIX Gap
  - Leave your Expectations at the Door
- Windows Protocols Documentation
  - Form and function
- The Development Community
  - Your Friendly Neighborhood Protocol Geeks

SMB

History • Terminology • Structure

SMB: Why Start Here?

- SMB is at the core of Windows Interoperability
  - SMB provides an authenticated connection to the server
  - SMB provides access to entities in the Object store: Files ● Directories ● Named Pipes
  - SMB3 supports Multi-channel and RDMA transport
- SMB is often the "Interoperability" users expect
  - Access to shares, files, etc.
  - A common access mechanism for a variety of storage platforms
- Features and Services are built on top of SMB
  - SMB is used as an authenticated transport layer for a variety of useful and interesting protocols

SMB provides a reliable, scalable, authenticated Network Transport.

SMB: History

- 1980's: Server Message Block
  - Created by IBM for PC-DOS and (later) OS/2
    - Productized as “LAN Manager”
  - SMB is a File (not Block) Protocol
    - SMB presents Files and Directories, and other FS Metadata
- 1990's: SMB becomes CIFS
  - Ported to Windows NT as “NT LAN Manager”
  - Dubbed Common Internet File System; competes with WebNFS
  - CIFS Drafts submitted to IETF, updates planned for W2000
  - US and EU Legal Actions
- 2000's: SMB2
  - SMB2, a new protocol, is released (with Windows Vista)
  - Legal Actions Completed
    - “CIFS” is redefined to refer to the “NT LAN Manager” dialect

SMB: History

- 2010's: SMB3 and the decline of CIFS
  - SMB2.2 is released
    - Renumbered SMB3
    - Supports Clustering, Multipath, RDMA...
    - Can be used as a Block Protocol Transport (RSVD)
  - The “CIFS” name is decommissioned
    - Remove this term from your marketing materials
- 2010's: SMB1 (CIFS) is deprecated
  - CIFS is re-renamed “SMB1”
    - Older Core & LAN Manager dialects are summarily ignored
  - SMB1 can now be disabled in Windows
    - Say goodbye, and close the door
    - SMB1 will likely linger in home and SoHo use

SMB: Terminology

- SMB: Server Message Block
  - Any and all flavors
- SMB1: NT LAN Manager
  - Legacy NT SMB; formerly known as “CIFS”
  - Still supported in current Windows versions
- SMB2: A New Protocol, Intended to Supersede SMB1
  - Simpler than SMB1
  - Uses larger integers for better scalability
- SMB3: SMB2 with Features
  - Same protocol, newer dialects
  - Current leading edge

Our focus is on SMB3.

SMB: Structure

- Traditional Client/Server
  - The client sends request messages, and
  - The server sends response messages
- SMB3 supports Multi-channel & SMB Direct
  - **Multi-channel:** A single session is multiplexed over multiple transport connections
  - **SMB Direct:** SMB3 I/O requests can be sent over RDMA
- SMB3 provides basic file I/O and directory access
  - Open, Close, Read, Write
  - Query Directory, Query Info, Set Info
- SMB3 is an authenticated transport for—or co-dependent with—a variety of other protocols
  - MSRPC, PeerDist, RSVD, IOCTL, FSCTL

SMB: Structure

SMB can be viewed as *either* a network protocol, *or* a set of function calls thrown onto the wire.

Both views are useful in understanding the structure of SMB.

Check out SNIA Tutorial:

SMB remote file protocol (including SMB 3.0)

Jose Barreto, Microsoft

SMB as an RPC Transport

- Remote Procedure Call (RPC)
  - RPC uses Named Pipes as a transport mechanism
  - Named Pipes are File System Objects
  - RPC over SMB provides authenticated access
- RPC used for:
  - Management and maintenance tasks
  - Access to various types of Metadata
  - A whole bunch of stuff
- RPC is now considered “legacy”
  - ...but new protocols are still being built on it
  - [See: Witness]

SMB as a Block I/O Engine

- Consider: Mounting a File as a Disk
  - This is done all the time in the Linux and BSD worlds
  - Block I/O is performed on a file, rather than a block device
  - (Fairly basic stuff, really.)
- What if that virtual disk file is on a network share?
  - Easily accessed from one of many clients
  - Sharing and locking controls apply
- What if that file is a HyperV Virtual Disk?
  - Shared access
  - Shadow copy for snapshots
- RSVD: Remote Shared Virtual Disk protocol
  - ...over SMB3, of course

SMB3 and PeerDist

- Distributed Caching System
  - Known as “BranchCache”
  - Works with SMB2.1 and SMB3
  - Uses an FSCTL (IOCTL) call over SMB to access fingerprints
- SMB2 leverages BranchCache to accelerate performance over wide area networks
  - Access via the local cache is faster than over the WAN
  - Cache data is verified using the fingerprints
  - Stale cache data is automatically detected
- BranchCache also works with BITS and HTTP servers
  - BITS is used by Windows Update

SMB3 and Witness

- Cluster Watcher: Service Witness Protocol [SWP]
  - Improve cluster recovery time in the event of a network partition or server crash
  - “Watch” for failures, and guide clients to working cluster nodes
  - Pro-actively re-direct clients to prepare for a planned outage
- Witness is an RPC-based protocol
  - Yeah...good old RPC
  - ...but Witness is fairly simple (~5 commands)
  - Witness supports SMB3 / SMB3 supports Witness

Mind the (POSIX) Gap

Windows Ain't POSIX

Forget what you learned in your CSci class.
Windows behaves differently...

- Access Controls
- Available Metadata
- Byte-range and File Locks
- File Deletion Behavior
- Opportunistic Locks and Leases
- Quota Handling
- Shadow Copy
- Timestamps

Windows Ain't POSIX

Forget what you learned in your CSci class.
Windows behaves differently...

Most of your time and effort will be spent trying to get your system to emulate Windows semantics in a safe and consistent way.

(Under SMB1, Windows still needs to emulate DOS and OS/2 semantics.)

Windows Ain't POSIX

- Consider: Byte Range Locking
  - Windows: Mandatory locks
  - POSIX: Advisory locks
- Consider: Access Control Lists
  - POSIX: No “official” standard, just a draft
  - Windows: ACLs complicated by allow/deny semantics, inheritance, and SACLs vs. DACLs
  - NFSv4: Almost, but not quite, similar to Windows ACLs...sort of
- Consider: Timestamps
  - Different clock resolution and epoch

Windows Ain't POSIX

- Consider: NFS isn't purely POSIX either
  - NFS and SMB are common business partners
  - NFS is *closer* to POSIX, but it's not 100%
- Samba, as an example, has a three-way battle for semantic consistency
  - Samba provides SMB (Windows) semantics
  - The host system provides POSIX semantics (with variations)
  - NFS provides NFS semantics
  - All three need to be coordinated somehow...and it's not easy!

Windows Protocols Documentation

Protocols Documentation

- At last count: Thousands of docs
  - Find a starting point, and dig from there
- Protocol documentation is arranged in sections:
  1. Introduction
  2. **Message structures (with field definitions)**
  3. **Protocol Details (a detailed state machine)**
  4. Protocol Examples
  5. Security
  6. **Behavior (Windows Behavior Notes)**
  7. Change Tracking
  8. Index
- The structure varies for different documentation types

The Docs: Key Sections

- Message Structures
  - Protocol Data Units (PDUs): the wire format
  - Field descriptions and basic syntax
  - Most developers will be drawn here first
- Protocol Details
  - Provides a comprehensive protocol state machine
  - Difficult to understand at first, but critical to correct semantics
  - References to other documentation that provides further detail
  - The State Machine is a third perspective on SMB3 internals
- Behavior
  - Where behavior is optional, this section tells you what Windows does, by version, and under different situations

The Docs: Getting Help

- The Windows Protocol documentation is supported:
  - dochelp@microsoft.com

The Development Community

Community Support

- Conferences and Plugfests
  - The SNIA Storage Developers Conference
    - Includes an SMB3 Plugfest and updates from Microsoft
    - A key event for SMB and Windows Interoperability
  - Samba/XP
    - Annual Samba conference; updates from the Samba Team
  - Microsoft Plugfests
    - Focused interoperability workshops and presentations
    - Emphasis on testing
- Online
  - Samba Technical Mailing List and IRC Chat
    - Focused on Samba internals
    - Members can point you toward other resources

Appendix

Windows Protocol Documentation

Selected References:

- [MS-PCCRC]: Peer Content Caching and Retrieval: Content Identification
  https://msdn.microsoft.com/en-us/library/dd303704.aspx
- [MS-RSVD]: Remote Shared Virtual Disk Protocol
  https://msdn.microsoft.com/en-us/library/dn393384.aspx
- [MS-SMB2]: Server Message Block (SMB) Protocol Version 2 and 3
  https://msdn.microsoft.com/en-us/library/cc246482.aspx
- [MS-SWN]: Service Witness Protocol
  https://msdn.microsoft.com/en-us/library/hh536748.aspx

Attribution & Feedback

The SNIA Education Committee thanks the following Individuals for their contributions to this Tutorial.

Authorship History

Christopher R. Hertel
September, 2015

Updates:
crh / 7-Sep-2015

Additional Contributors

*Please send any questions or comments regarding this SNIA Tutorial to **tracktutorials@snia.org***