PART II

SMB: The Server Message Block Protocol

8

A Little Background on SMB

Your mileage may vary.

— Advertiser's disclaimer

Email

```
From: Steven French, Senior Software Engineer, IBM
  To: Chris Hertel

Chris,

Hope things are going well in the cold north...

I thought the following info would be interesting to you. I met the
original "inventor" of SMB a few years ago - Dr. Barry Feigenbaum -
who back in the early 80's was working on network software
architecture for the infant IBM PCs, working for IBM in the Boca
Raton plant in Florida. He mentioned that it was first called the
"BAF" protocol (after his initials) but he later changed it to SMB.
In the early DOS years IBM and Microsoft (with some input from Intel
and 3Com) contributed to it but by the time of the first OS/2 server
version (LANMAN1.0 dialect and later) Microsoft did much of the work
(for "LAN Manager" and its relatives).
```

Like NetBIOS, the Server Message Block protocol originated a long time ago at IBM. Microsoft embraced it, extended it, and in 1996 gave it a marketing upgrade by renaming it "CIFS."

Over the years there have been several attempts to document and standardize the SMB/CIFS protocol:

- Microsoft keeps an archive of documentation covering older versions of SMB/CIFS. The collection spans a period of roughly ten years, starting at about 1988 with the SMB Core Protocol. The collection is housed, it seems, on a dusty FTP server in a forgotten corner of a machine room somewhere in the Pacific Northwest. The URL for the CIFS archive is `ftp://ftp.microsoft.com/developr/drg/CIFS/`.
- In 1992, X/Open (now known as The Open Group) published an SMB specification titled *Protocols for X/Open PC Interworking: SMB, Version 2*. The book is now many years out of date and SMB has evolved a bit since its publication, yet it is still considered one of the best references available.[1] The Open Group is a standards body so the outdated version of SMB described in the X/Open book is, after all, a standard protocol.
- A few years later, Microsoft submitted a set of CIFS Internet Drafts to the IETF (**I**nternet **E**ngineering **T**ask **F**orce), but those drafts were somewhat incomplete and inaccurate; they were allowed to expire. Microsoft's more recent attempts at documenting CIFS (starting in March, 2002) have been rendered useless by awkward licensing restrictions, and from all accounts contain no new information.[2] The expired IETF Internet Drafts (by Paul Leach and Dilip Naik) are still available from the Microsoft FTP server described above and other sources around the web.
- The CIFS Working Group of the **S**torage **N**etwork **I**ndustry **A**ssociation (SNIA) has published a *CIFS Technical Reference* based on the earlier IETF drafts. The SNIA document is neither a specification nor a standard, but it is freely available from the SNIA website.

---

1. The X/Open SMB documentation is out of print, but electronic copies are now available online (free registration required). See `http://www.opengroup.org/products/publications/catalog/`, and look for documents #C195 and #C209.

2. I must rely on anecdotal evidence to support this claim. Due to the licensing restrictions, I have not read these documents, which were released in March of 2002.

Without a current and authoritative protocol specification, there is no external reference against which to measure the "correctness" of an implementation, and no way to hold anyone accountable. Since Microsoft is the market leader, with a proven monopoly on the desktop, the behavior of their clients and servers is the standard against which all other implementations are measured.

Jeremy Allison, the Samba Team's First Officer,[3] has stated that "The level of detail required to interoperate successfully is simply not documentable." One reason that this is true is that Microsoft can "enhance" SMB behavior at will. Combined with the dearth of authoritative references, this means that the only criteria for a well-behaved SMB implementation is that it works with Microsoft products. As a result, subtle inconsistencies and variations have crept into the protocol. They are discovered in much the same way that a dog-owner discovers poop in the yard in springtime when the snow melts.[4]

Many people dread spring chores, but spring also brings the flowers. The children play, the dog chases a butterfly, the birds sing... and it all seems suddenly worthwhile. It's the same with the work we have ahead. Things are not really too bad, once you've gotten started.

8.1 Getting Started

This part of the book will cover the basics of SMB, enumerate and describe some of the SMB message types (commands), discuss protocol dialects, give some details on authentication, and provide a few examples. That should be enough to help you develop a working knowledge of the protocol, a working SMB client, and possibly a simple server.

Bear in mind, though, that SMB is more complex and less well defined than NBT. In the NBT section it was possible to describe every message type and provide a comprehensive review of the entire NBT protocol. It is not practical to cover all of SMB in the same way. Instead, the goal here is to explain

---

3. ...and Tactical Officer. He's the one with the prosthetic forehead.

4. I live in Minnesota, where it most definitely snows in winter. I share my home with a Pembrokeshire Welsh Corgi and a Golden Retriever, so the springtime scenario described above is vividly real and meaningful to me. Some of my Australian Samba Team friends have suggested that people in other parts of the world may find it less familiar. Use your imagination.

the basics of SMB, provide details that are missing from other sources, and describe how to go about exploring SMB on your own. In other words, the goal is to develop understanding rather than simply providing knowledge.

The textbook for this class is the latest version of the SNIA *CIFS Technical Reference*. Additional sources are listed in the References section near the end of this book. The most important tool, however, is probably the protocol analyzer. Warm up your copy of Ethereal or NetMon, and get ready to do some packet shoveling.

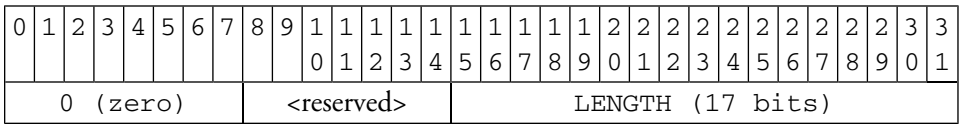
8.2 NBT or Not NBT

Before we actually start, there is one more thing to mention: The SMB protocol is supposed to be "transport independent." That is, SMB *should* work over any reliable transport that meets a few basic criteria. NBT is one such transport, but SMB does not really require the NetBIOS API. It can, for instance, be run directly over TCP/IP.

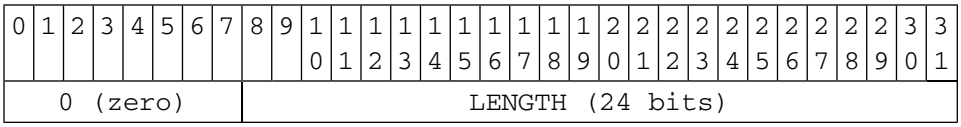
Just for fun, we will refer to SMB over TCP/IP without NBT as "naked" or "raw." When running naked, SMB defaults to using TCP port 445 instead of the NBT Session Service port (TCP/139). Windows 2000, Windows XP, and Samba all support raw transport, but the large number of "legacy" Windows clients still in use suggest that NBT will not go away any time soon.

Other than the new port number, there are only two notable differences between NBT and naked transport. The first is that naked transport does not make use of the NBT `SESSION REQUEST` and `POSITIVE SESSION RESPONSE` messages. The second is that the two transports interpret the `SESSION MESSAGE` header a bit differently.

Recall (from Chapter 6 on page 129) that the NBT Session Service prepends a four-byte header to each `SESSION MESSAGE`, like so:



The LENGTH field, as shown, is 17 bits wide.[5] Raw TCP transport also prepends a four-byte header, but there are no reserved bits so the LENGTH may use three full bytes:



Appendix B of the SNIA *CIFS Technical Reference* is the only source that was found which clearly shows the naked transport LENGTH field as being 24 bits wide. This 24-bit field translates to 16 megabytes, though, and that's a bigbunch — more than is typically practical. Fortunately, the actual maximum message size is something that is negotiated when the client and server establish the session.

When we discuss the SMB messages themselves we will ignore the SESSION MESSAGE headers, since they are part of the transport, not the SMB protocol.

5. There are some old archived conversations on Microsoft's CIFS mailing list which suggest that some implementors were — and possibly still are — only allowing for a 16 bit LENGTH field in the NBT SESSION MESSAGE.