SMB3 Multi-Channel in Samba

sambaXP 2015

Michael Adam

Samba Team / Red Hat

May 20, 2015

SMB3 features in Samba

1. SMB 3.0 (Win8 / 2012):
   - new crypto (sign/encrypt) [4.0]
   - secure negotiation [4.0]
   - durable handles v2 [4.0]
   - persistent file handles [WIP/tracer]
   - **Multi-Channel** [WIP+]
   - SMB direct [designing/starting]
   - cluster features [designing]
     - witness [WIP+]
   - storage features [WIP]
2. SMB 3.0.2 (Win8.1 / 2012R2): [master]
3. SMB 3.1.1 (Win10 / 2014):
   - negotiate contexts, preauth: [master]

SMB 3.1

implemented

- 3.1.1 dialect
- negotiate contexts
- preauthentication integrity
- encryption improvements (choose cipher)
  AES-128-CCM $\rightarrow$ AES-128-GCM

not implemented

- cluster dialect fencing
- cluster client failover v2 (client)

Multi-Channel

Multi-Channel - General

- bind multiple transport connections to one session
- increase throughput and fault tolerance

Multi-Channel - Windows/Protocol

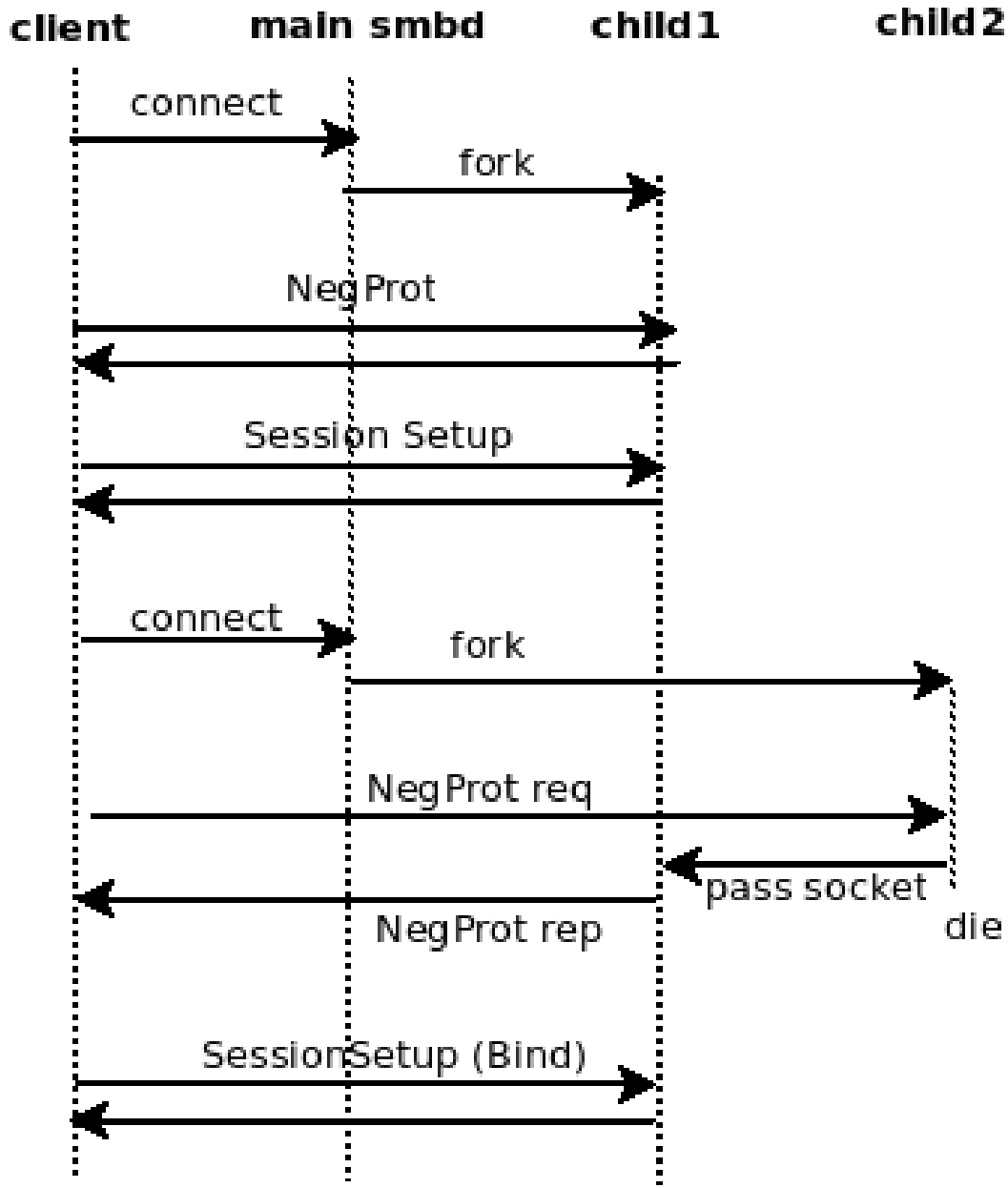
1. establish initial session on TCP connection
2. find interfaces with interface discovery: `FSCTL_QUERY_NETWORK_INTERFACE_INFO`
3. bind additional TCP (or RDMA) connection (channel) to established SMB3 session (session bind)
4. windows: uses connections of same (and best quality)
5. windows: binds only to a single node
6. replay / retry mechanisms, epoch numbers

Multi-Channel - Samba

samba/smbd: multi-process

- **Currently:** process $\Leftrightarrow$ TCP connection
- **Idea:** transfer new TPC connection to existing smbd
- **How?** $\Rightarrow$ use fd-passing (sendmsg/recvmsg)
- **When?** as early as possible, based on client GUID
  $\Rightarrow$ per client GUID single process model

Multi-Channel - Samba



Multi-Channel - Samba

1. preparation:
   messaging rewrite using unix dgm sockets with sendmsg [DONE,4.2]
2. add fd-passing to messaging [DONE,4.2]
3. preparations in internal structures [ess.DONE]
4. implement smbd message to pass a tcp connection [ess.DONE]
5. transfer connection already in negprot (ClientGUID) [largely DONE]
6. implement session bind [ess.DONE]
7. implement channel epoch numbers [WIP]
8. implement interface discovery [WIP]
9. implement test case [WIP]

MSG_SMBXSRV_CONNECTION_PASS

from smbXsrv.idl

```
typedef struct {
    NTTIME                  initial_connect_time;
    GUID                    client_guid;
    hyper                   seq_low;
    DATA_BLOB               negotiate_request;
} smbXsrv_connection_pass0;
```

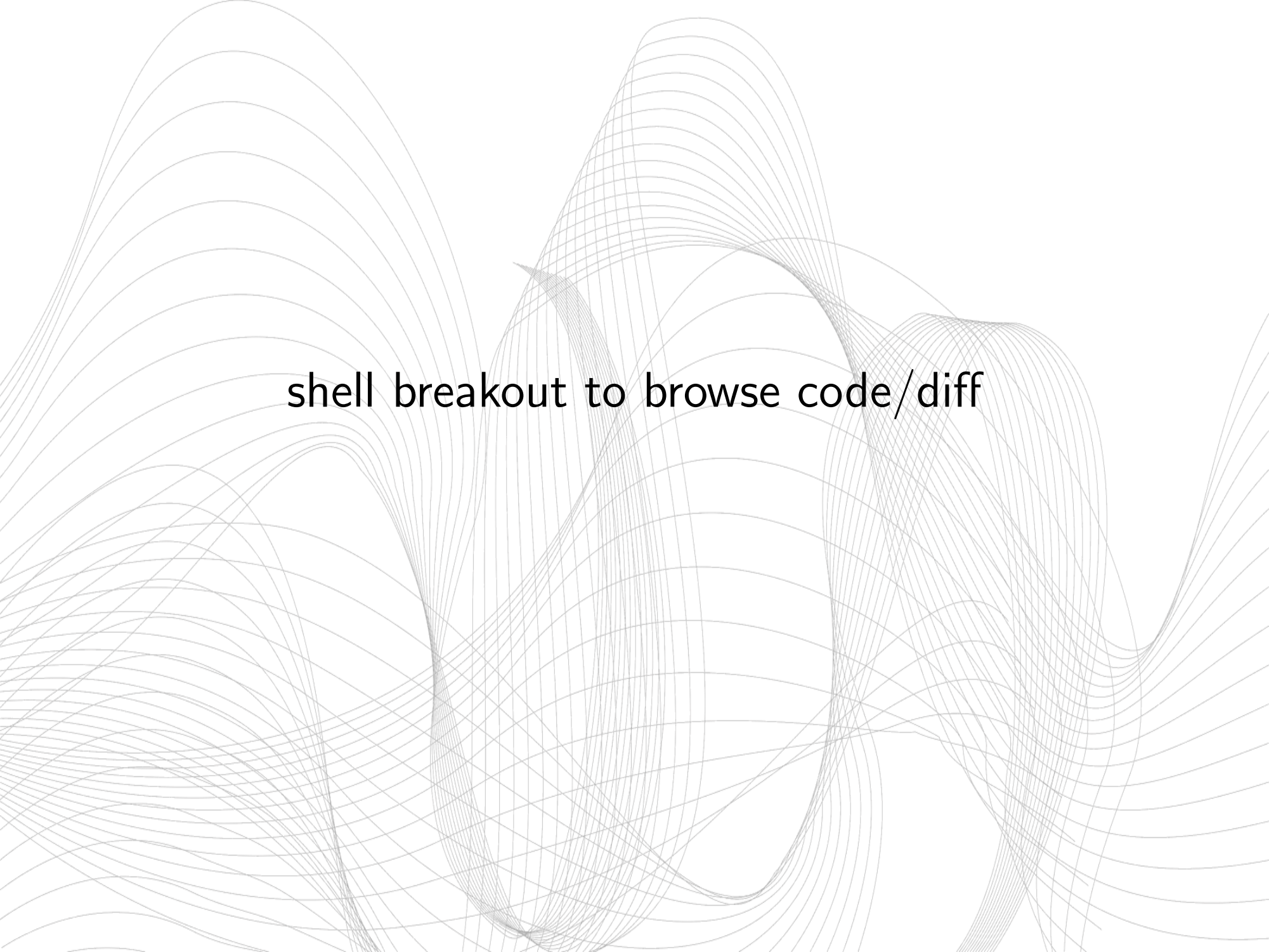
Internal Structures (smbXsrv)

before

```
smbXsrv_session->smbXsrv_connection
```

after

```
smbXsrv_session->smbXsrv_client->smbXsrv_connection
```


shell breakout to browse code/diff

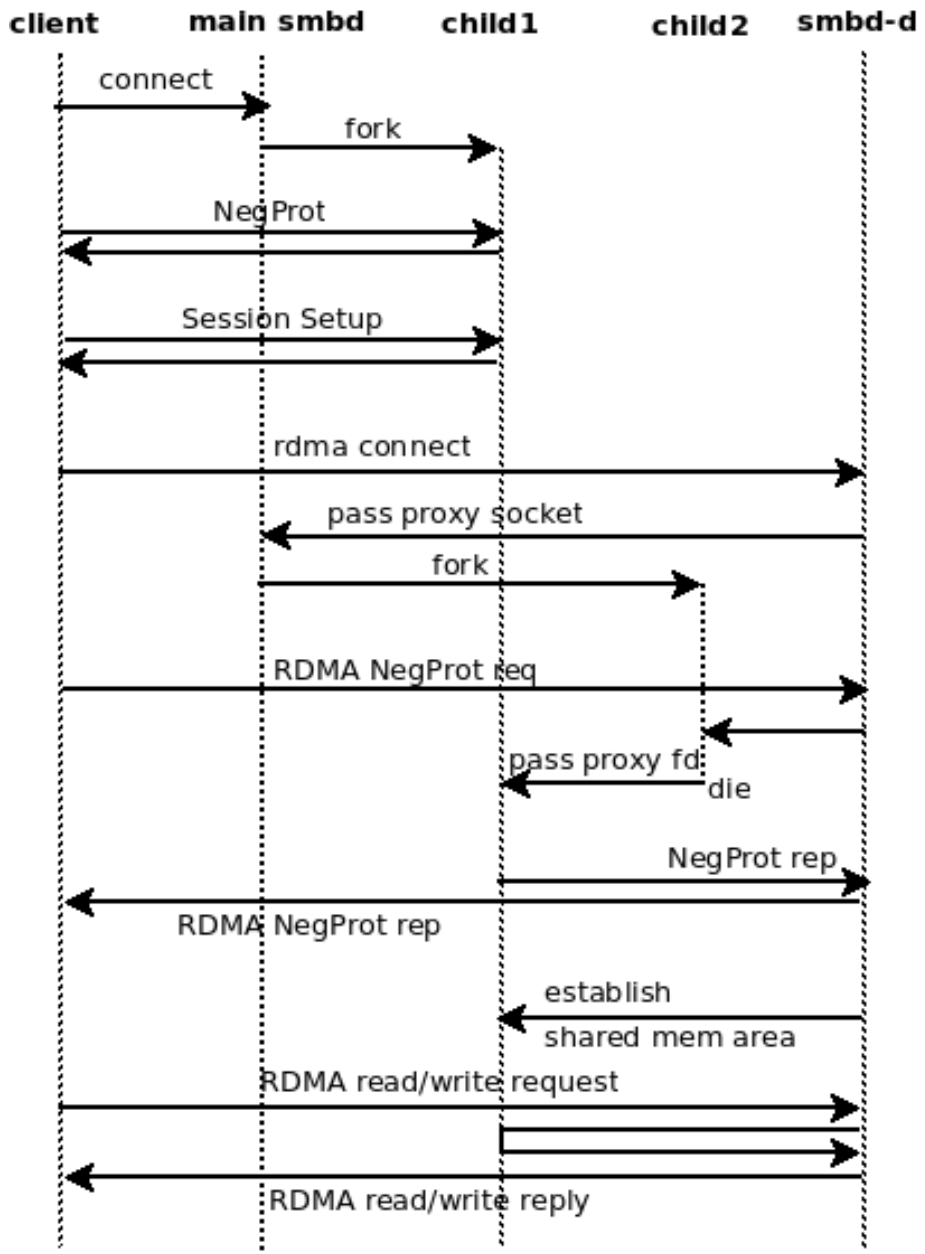
Multi-Channel Demo

Outlook: SMB Direct

SMB Direct (RDMA)

- windows:
  - requires multi-channel
  - start with TCP, bind an RDMA channel
  - reads and writes use RDMA write/read
  - protocol/metadata via send/receive
- wireshark dissector: [DONE]
- samba (TODO):
  - prereq: multi-channel / fd-passing
  - buffer / transport abstractions [TODO]
  - problem: libraries: not fork safe and no fd-passing
    $\Rightarrow$ central daemon (or kernel module) to serve as RDMA "proxy"

SMB Direct (RDMA) - Plan



SMB features in Samba

https://wiki.samba.org/index.php/Samba3/SMB3

Thanks for your attention!

Questions?

obnox@samba.org

madam@redhat.com