SMB3 Multichannel Update

Günther Deschner
<gd@samba.org>

Sachin Prabhu
<sprabhu@redhat.com>

Agenda

- Samba/CTDB Clustering with GlusterFS
- SMB3 Multichannel recap
- Oplock/Lease break failures and Multichannel
- Multichannel and testing
- IP failover with Multichannel and CTDB
- Further reading & Q/A

Samba/CTDB clustering with GlusterFS

Red Hat Gluster Storage (RHGS)

- "Red Hat Gluster Storage provides an open, software-defined storage solution across physical, virtual, and cloud resources."
- SMB storage on top of GlusterFS (currently) using Samba
- CTDB for clustering
- `vfs_glusterfs` module for Samba
  - (uses libgfapi for storage I/O)
- Upcoming: `vfs_glusterfs_fuse` module
  - Leverages fusemounted glusterfs
  - Implements `VFS_GET_REAL_FILENAME`
- Current Release:
  - RHGS 3.4.4 with Samba 4.8.5 offering SMB3 features
- SMB Multichannel only as "Tech Preview"

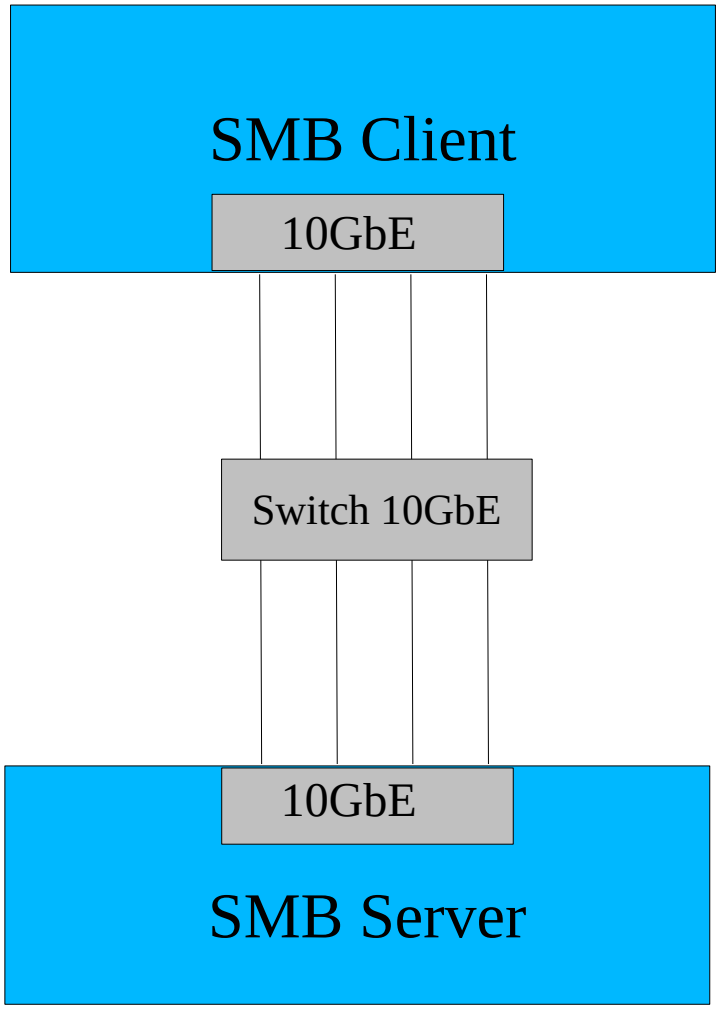
What was SMB Multichannel again?

SMB Multichannel

- **SMB3 performance and reliability feature**
- **Available since Windows 2012**
- **Maximize throughput**
  - Multiple TCP transport connections aggregated in one session
  - Multiple NICs (NIC teaming, RDMA)
  - Multiple CPU Cores with RSS (Receive Side Scaling)
- **Increase fault tolerance**
  - Multichannel setups compensate TCP failures on channels
- **Automatic configuration**
  - Feature is automatically and transparently enabled when prerequisites are met:
  - Client and Server support SMB3
  - Automatic detection of matching interfaces

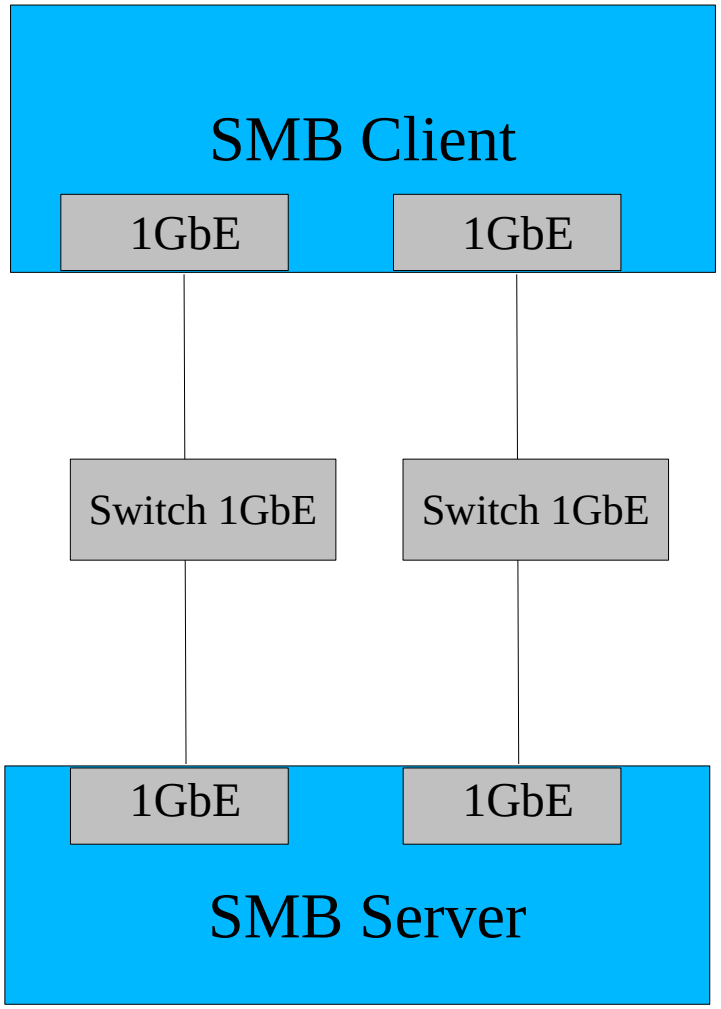
SMB Multichannel

- **Single NIC with RSS**



SMB Multichannel

- **Multiple NICs**



SMB Multichannel

- **Overview ((c) Microsoft)**

	Throughput	Fault Tolerance for SMB	Fault Tolerance for non-SMB	Lower CPU utilization
Single NIC (no RSS)	▲			
Multiple NICs (no RSS)	▲▲	▲		
Multiple NICs (no RSS) + NIC Teaming	▲▲	▲▲	▲	
Single NIC (with RSS)	▲			
Multiple NICs (with RSS)	▲▲	▲		
Multiple NICs (with RSS) + NIC Teaming	▲▲	▲▲	▲	
Single NIC (with RDMA)	▲			▲
Multiple NICs (with RDMA)	▲▲	▲		▲

SMB Multichannel in Samba

- **First implementation in Samba 4.4 (2016)**
  - “server multi channel support = yes”
  - Uses fd-passing so all channels point to one smbd
  - Experimental feature, since not all scenarios are covered
- **Current limitations:**
  - Oplock and lease break not Multichannel aware and does not attempt to retry **DONE**
  - Multichannel not testable in autobuild **IN_PROGRESS**
  - No interaction with CTDB failover ip management **TODO**

Oplock/Lease break failures and Multichannel

Oplock/Lease Break with Multichannel

- Oplock/Lease Break are issued by the Server
- “The SMB2 Oplock Break Notification packet is sent by the server when the underlying object store indicates that an opportunistic lock (oplock) is being broken, representing a change in the oplock level.”
- “The SMB2 Lease Break Notification packet is sent by the server when the underlying object store indicates that a lease is being broken, representing a change in the lease state.”
- Multiple channels can be available for sending break notifications
- Which one is chosen?
- What happens on channel failure?

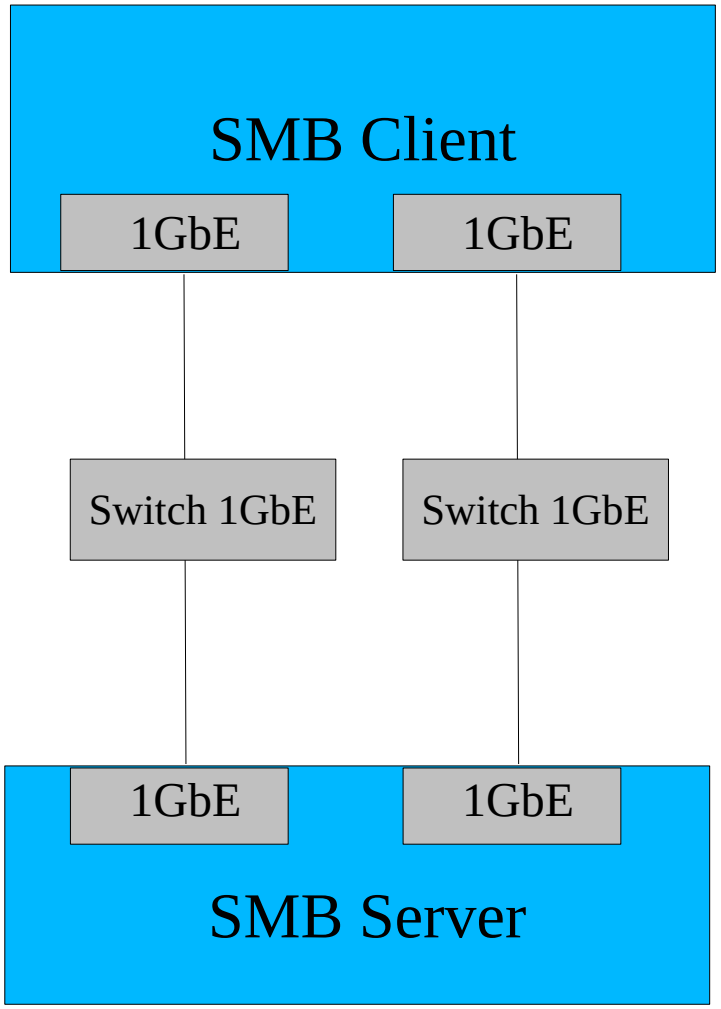
Oplock/Lease Break with Multichannel

- **MS-SMB2 update (2019-04-30) now provides more details:**

  ”If the server implements the SMB 3.x dialect family, SMB2 Oplock Break Notification MUST be sent to the client using the first available connection in Open.Session.ChannelList where Channel.Connection is not NULL. If the server fails to send the notification to the client, the server MUST retry the send using an alternate connection, if available, in Open.Session.ChannelList.”

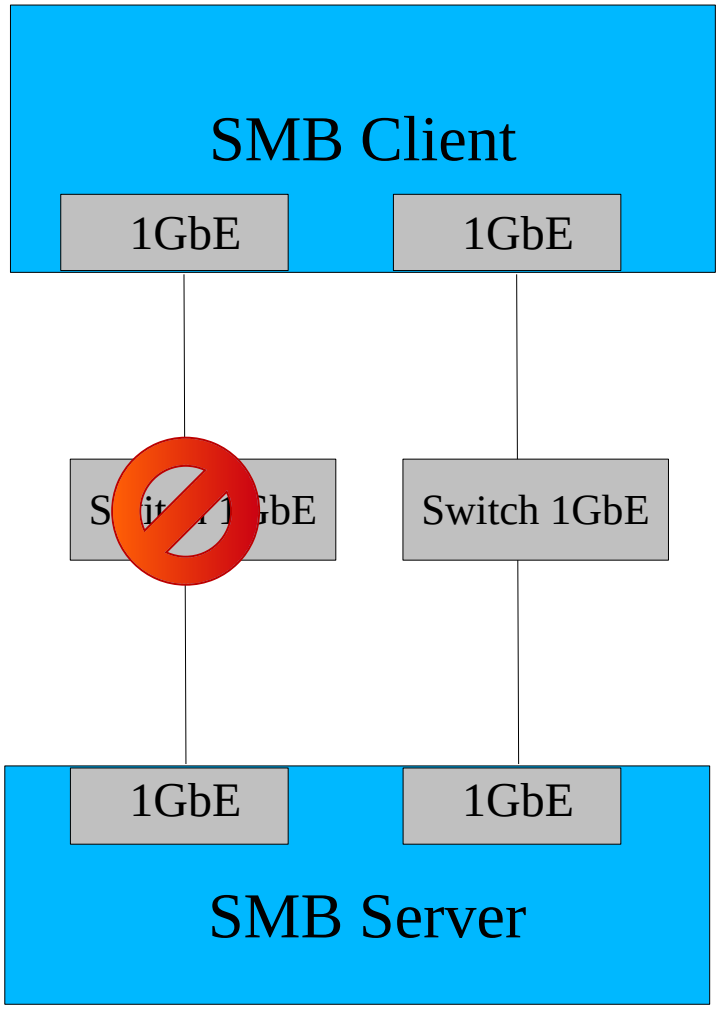
Oplock/Lease Break with Multichannel

- **What happens when a channel fails?**



Oplock/Lease Break with Multichannel

- **What happens when a channel fails?**



Oplock/Lease Break with Multichannel

- **Microsoft Interop Lab 2016/2017 research oplock and lease behavior with multichannel on Windows**
  - smbtorture testsuite
- **How to simulate channel failure?**
  - iptables to drop packets
- **Testing results for oplock break:**
  - smbtorture //server/share -U user%password smb2.multichannel.oplock
  - Oplock break is sent on the last created channel
  - Retry is not attempted at all
- **Testing results for lease break:**
  - smbtorture //server/share -U user%password smb2.multichannel.leases
  - Lease break is sent on the first created channel
  - Retry is attempted on the first connected channel

Oplock/Lease Break with Multichannel

- **TCP properties during retry:**
  - Observation during MS Interop Lab: Windows 2012 and 2016 will start retrying 10 times after inactivity of 10 seconds with a 1 second interval
- **Testsuite (smbtorture) extended to cover more granular cases**
  - Two major subtestsuites: “oplock”, “leases”
  - Additional “generic” tests to cover protocol details
  - All pushed upstream, currently not running against Samba server
- **Testsuite provides implementation specific mechanisms to block oplock/lease break packets to be received**
  - Samba Server: testsuite ignores break requests from server
  - Windows Server: blocking using iptables.
    - smbtorture uses parameter use_iptables=true
    - Limited to running on Linux
    - Needs root privileges.

Oplock/Lease Break with Multichannel

- **Samba does currently not deal with Oplock/Lease Break retries with multichannel**
- **TODO:**
  - ~~Make sure we calculate and verify delivery of break responses (compare send and receive queue packet counters)~~
  - Cleanup disconnected/failed channels
  - TCP settings to speed up discovery of failed channels (just as on Windows)
  - Why are oplock break notifications never retried on Windows?

Oplock/Lease Break with Multichannel

- **Timeout / retry observations and implementation aspects:**
  - On Windows:
    - Timeout changes with number of channels, retry on all channels
    - Tests with 2 channels: timeout ~5sec per channel
    - Tests with 3 channels: timeout ~1sec per channel
  - In Samba:
    - Current implementation timeout after OPLOCK_BREAK_TIMEOUT (30s) minus 5secs
    - retried once on next channel with 5sec timeout
  - Which timeout semantics to apply in final patches?
  - How many retries? On every available channel?

Multichannel and testing

Selftest support for Multichannel

- Every commit in Samba is run through automated testing during autobuild
- For enabling SMB Multichannel by default it must be tested permanently and automatically
- Samba automated testing uses abstraction libraries (cwrap.org):
  - socket_wrapper, nss_wrapper, uid_wrapper, resolv_wrapper, pam_wrapper, etc.
- Fd-passing?
- Present scenario
  - hard coded hack (lets socketwrapper work w/o fd-passing)

Selftest support for Multichannel

- Support for fd-passing worked on by Annop C S, Andreas Schneider and Michael Adam
- Plan for socket_wrapper implementation:
  - Refactoring of internal data structures => done and merged
  - Make socket_wrapper thread-safe => done and merged
  - Switch to mmap-ed file for shared memory among processes (protected with pthread robust mutexes) => done
  - Implement fd-passing => wip
  - Approach: Send socket_info array indexes as the fd array instead of actual fds via pipe and create new fd structure based on the corresponding indexes received at the other end.

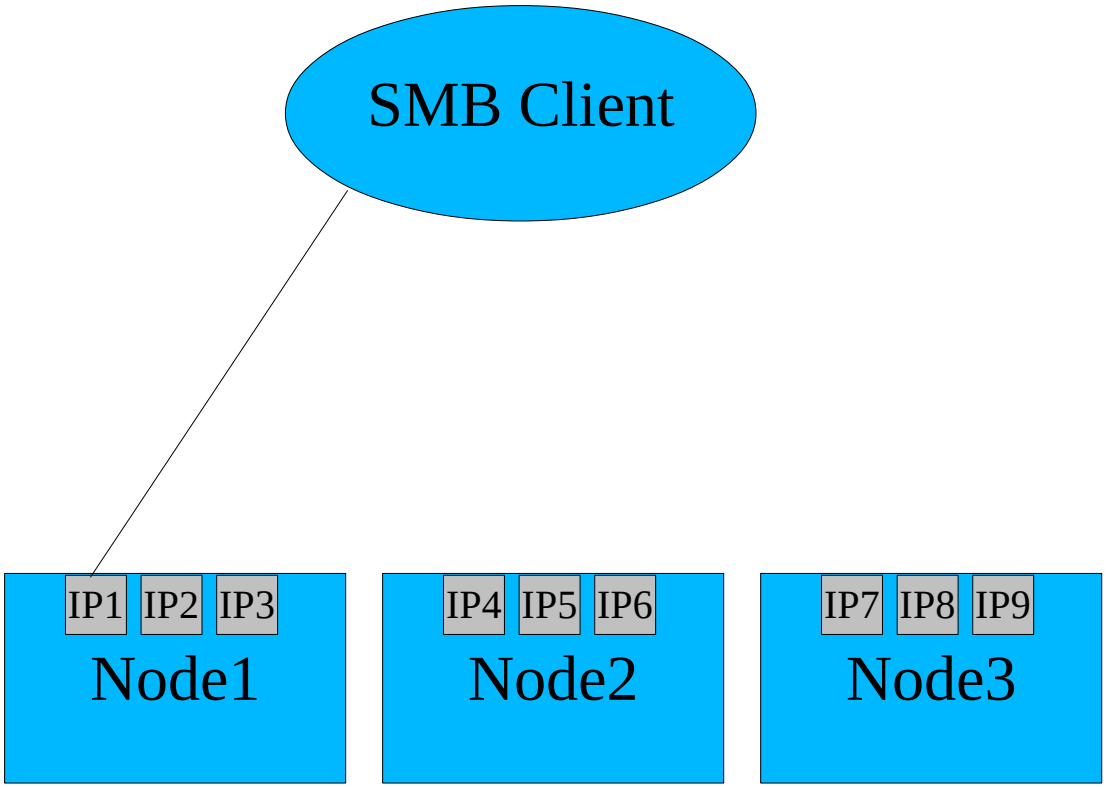
Multichannel and CTDB

Multichannel and CTDB

- Typically: multiple public ip addresses per cluster
- Ip addresses can be moved within the cluster
- Ip addresses can spawn over multiple nodes
- With multichannel and fd-passing all ip addresses of one multichannel session *must* reside on the same node
- Current solutions:
  - No /etc/ctdb/public_addresses file and hard coded ip addresses
  - Individual /etc/ctdb/public_addresses files per node
- Requirement: automatic configuration and transparent failover

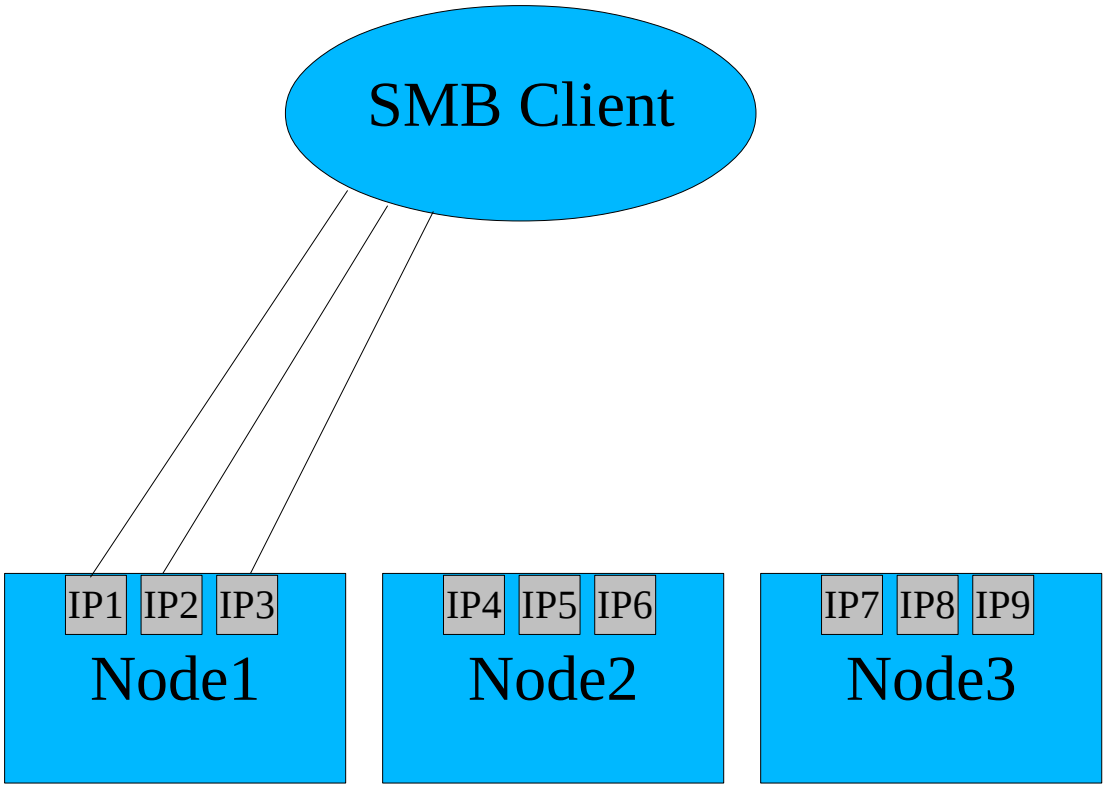
Multichannel and CTDB

- **Query for FSCTL QUERY NETWORK INTERFACE INFO on IP1**



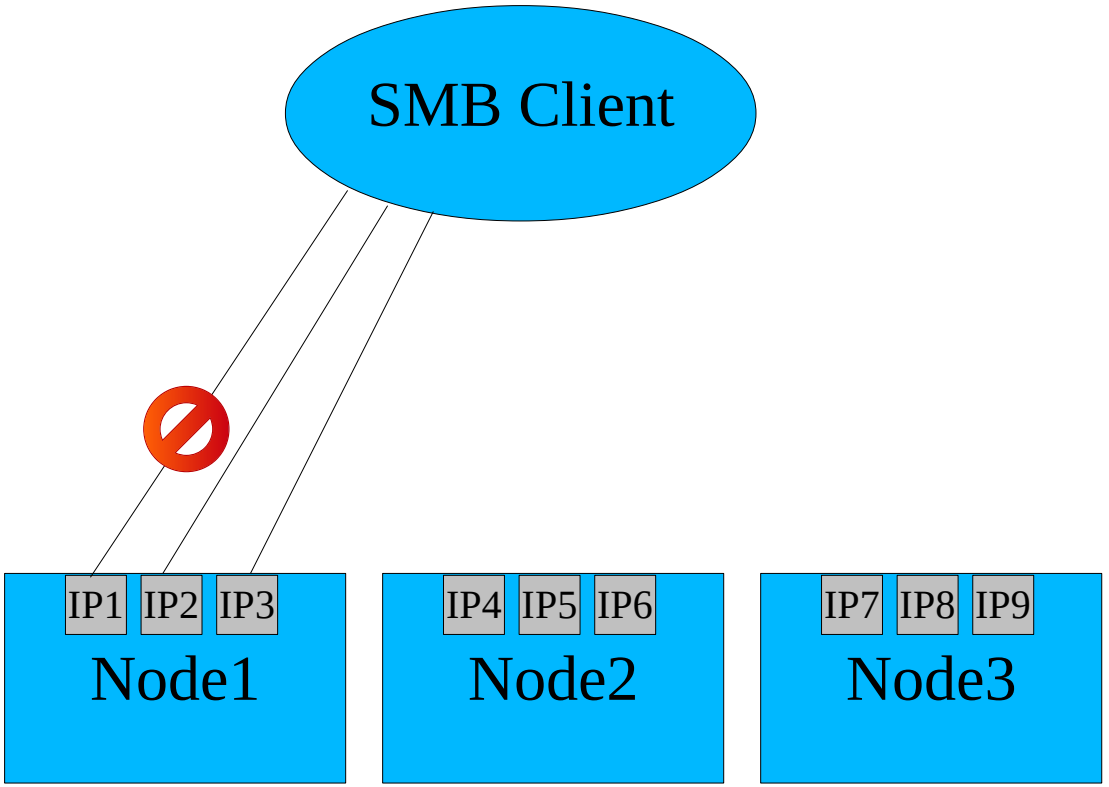
Multichannel and CTDB

- **Multiple channels (IP1, IP2, IP3) bound to same SMB3 session**



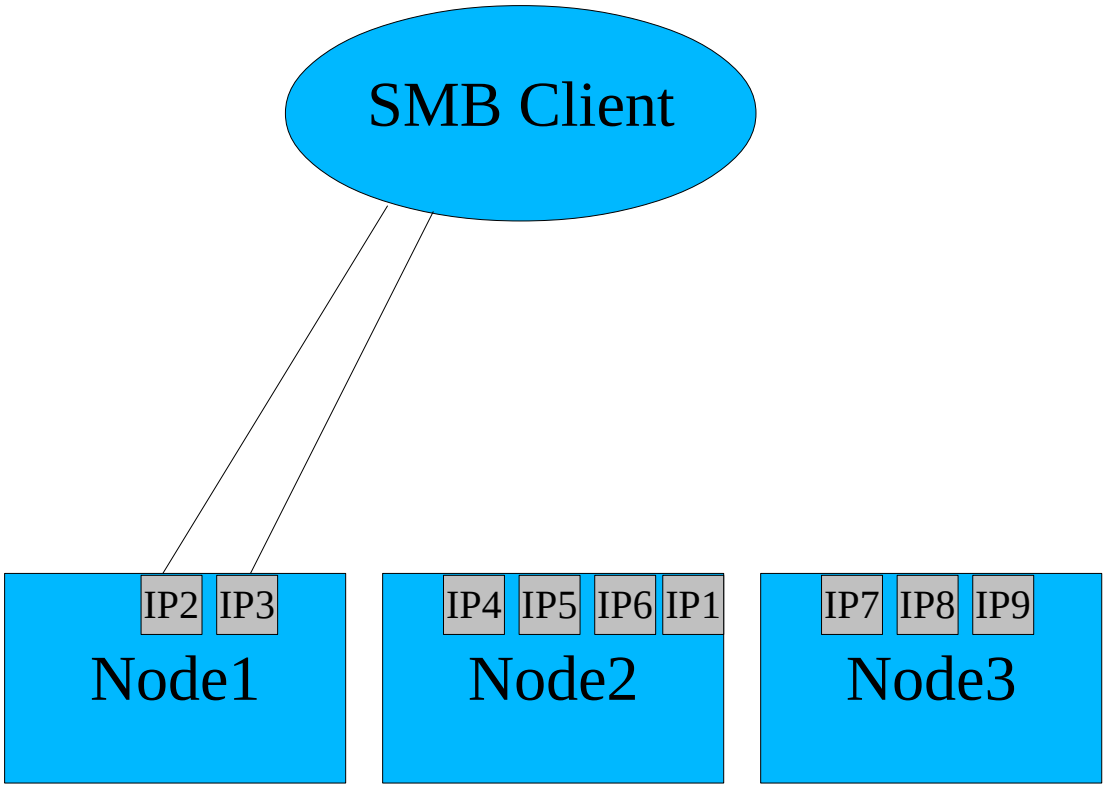
Multichannel and CTDB

- **Interface/Channel failure for IP1**



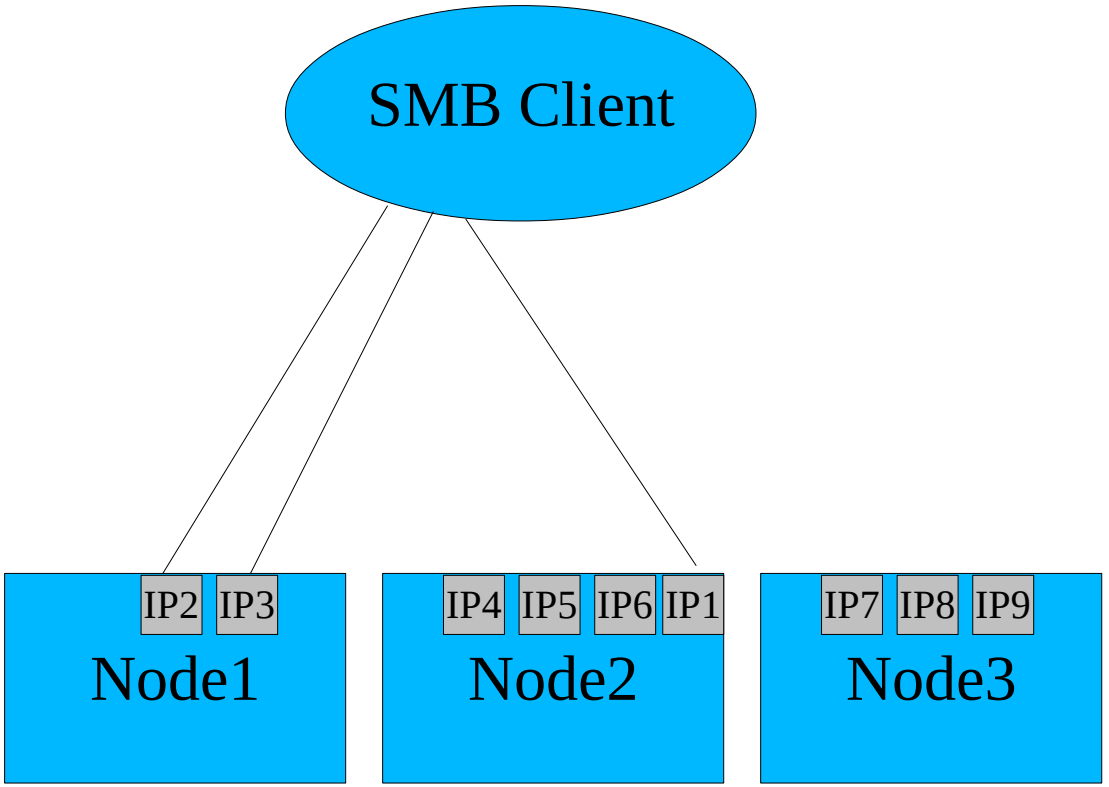
Multichannel and CTDB

- **CTDB would migrate IP1 to another Node...**



Multichannel and CTDB

- .. and failover the client ! Multichannel could not be re-established



Multichannel and CTDB

- Possible future solution for automatic configuration:
  - Monitor all NICs in the cluster using CTDB and move all channels on failure
  - Transparent failover Witness interface (RPC server dependencies)
  - SMB 3.1.1 tree connect context redirection
  - TBD
- multichanneld within CTDB
  - Offload IP management complexity to CTDB
  - Provide API for smbd to query list of enabled interfaces/addresses

Further reading

- **Microsoft Protocol Documentation:**
  - **MS-SMB2, MS-SWN**
- **Various Microsoft Technet articles**

Questions and answers

- **Mail gd@samba.org, sprabhu@redhat.com**
- **#samba-technical on irc.freenode.net**

Thank you for your attention!

www.redhat.com
www.samba.org

<gd@samba.org>