SMB3 and Beyond: Accessing Samba from Linux

Steve French
Principal Systems Engineer – Primary Data

Legal Statement

- This work represents the views of the author(s) and does not necessarily reflect the views of Primary Data Corporation
- Linux is a registered trademark of Linus Torvalds.
- Other company, product, and service names may be trademarks or service marks of others.

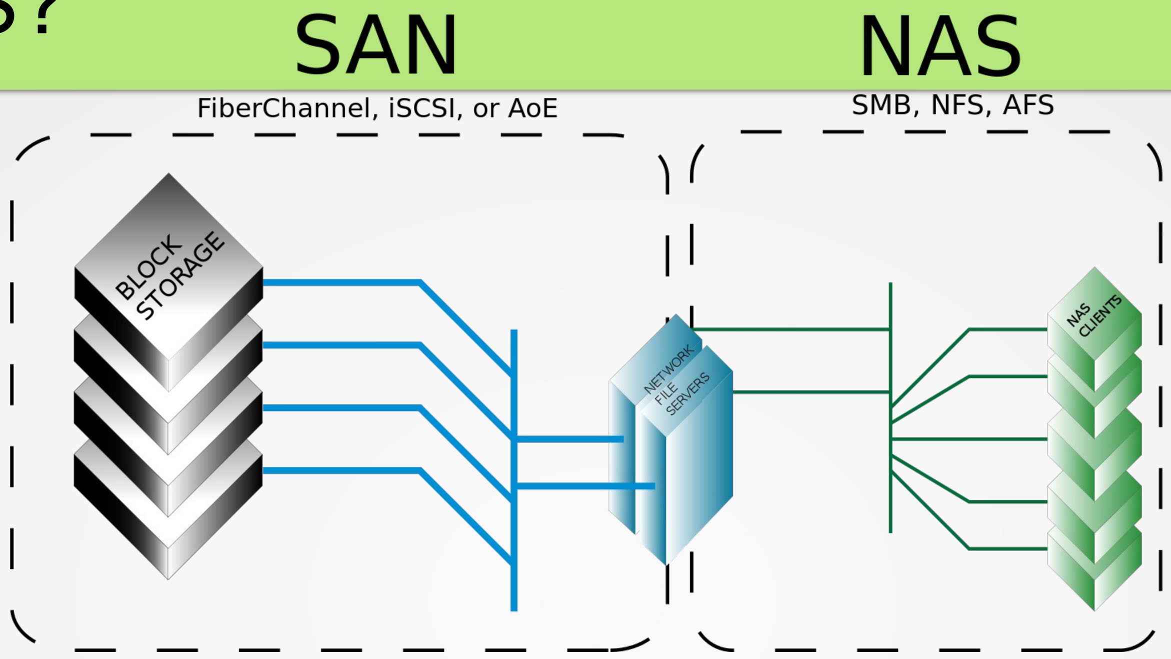
Who am I?

- Steve French   smfrench@gmail.com
- Author and maintainer of Linux cifs vfs (for accessing Samba, Windows and various SMB3/CIFS based NAS appliances)
- Also wrote initial SMB2 kernel client prototype
- Member of the Samba team, coauthor of   SNIA CIFS Technical Reference and former SNIA CIFS Working Group chair
- Principal Systems Engineer, Protocols: Primary Data

Wondering why we care about FS?

- 50 years ago first Hierarchical File System was built, http://www.multicians.org/fjcc4.html , yet more than ever we care how we store our data. Amount of data (largely unstructured) exceeded two Zettabytes by 2012 (IDC estimate), continues to double every two to three years.
- And it is transferred around A LOT
  - “Annual global IP traffic will surpass the zettabyte (1000 exabytes) threshold in 2016.” (CISCO estimate)
- Nearly all workloads depend heavily on file systems.

Why NAS?



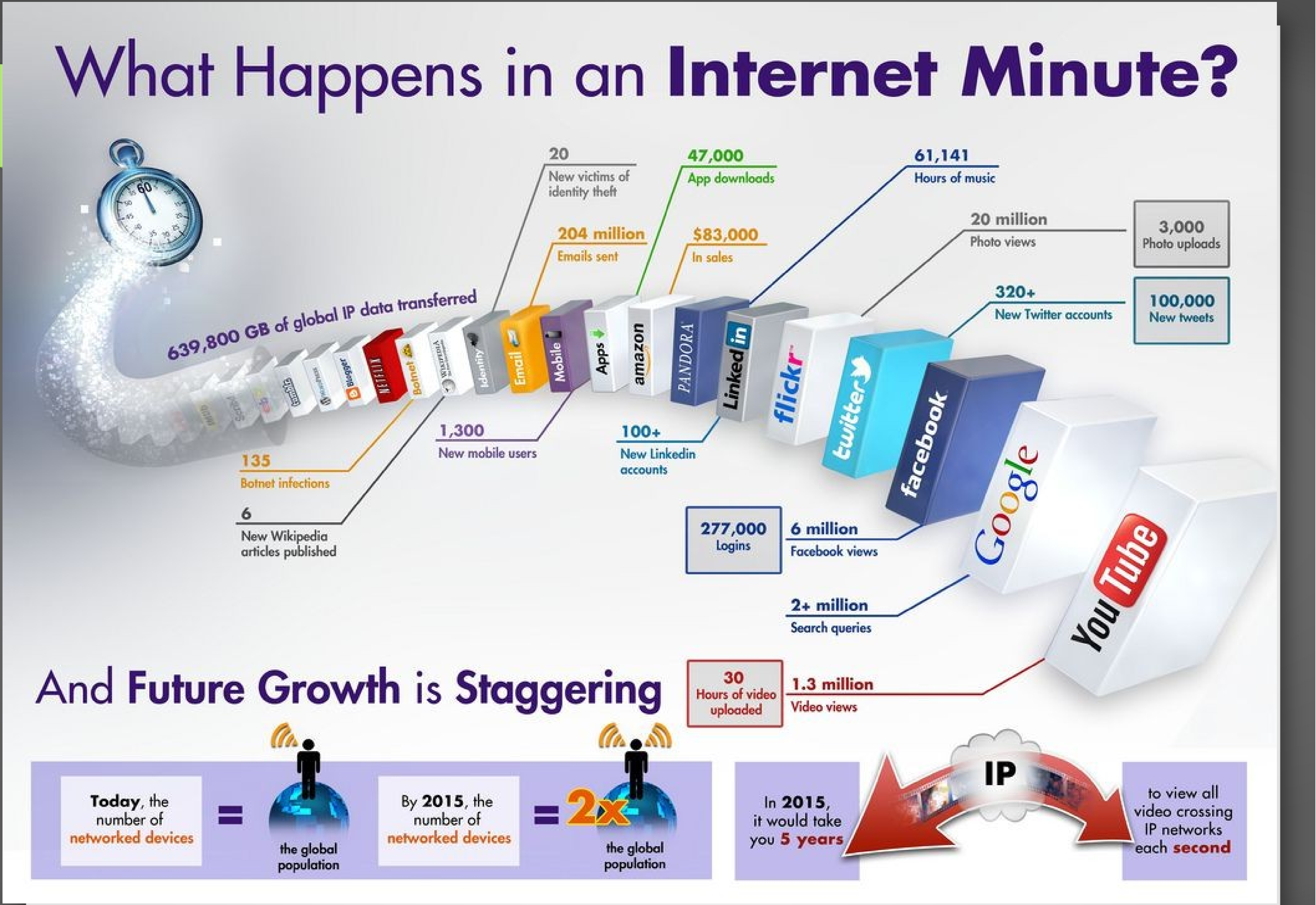
SAN

FiberChannel, iSCSI, or AoE

NAS

SMB, NFS, AFS

- In case you came to the wrong conference and really didn't know ...
  - NAS is superset of block (SAN) and object … but easier to manage
  - NAS (now) can get 90+ of the performance of SAN with lower administrative costs and more flexibility
  - Attributes at the right granularity (file/directory/volume)
  - Ownership information, easier to understand security, easy backup, optimizable with useful info on application access patterns, intuitive archive/encryption/compression policy, quotas, quality of service



What Happens in an **Internet Minute?**

639,800 GB of global IP data transferred

20 New victims of identity theft

204 million Emails sent

47,000 App downloads

$83,000 In sales

61,141 Hours of music

20 million Photo views

3,000 Photo uploads

320+ New Twitter accounts

100,000 New tweets

135 Botnet infections

6 New Wikipedia articles published

1,300 New mobile users

100+ New Linkedin accounts

277,000 Logins

6 million Facebook views

2+ million Search queries

30 Hours of video uploaded

1.3 million Video views

And **Future Growth** is **Staggering**

Today, the number of networked devices = the global population

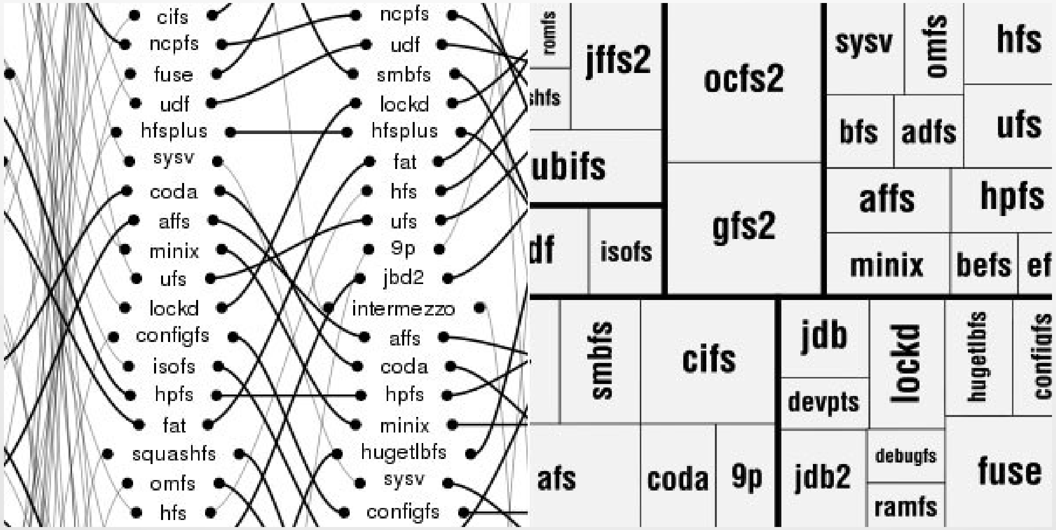
By **2015**, the number of networked devices = 2x the global population

In **2015**, it would take you **5 years** to view all video crossing IP networks each **second**

http://www.intelfreepress.com/news/networks-strain-to-keep-pace-with-data-explosion/191/

And why Linux?

- Large Talented Community. Rate of improvement is unsurpassed. For example in the past year (since 3.15-rc3)
  - More than 77,000 changesets developed, reviewed tested and merged to improve kernel
  - More than 4400 in the file system alone
    - 830,000 lines of (often very terse, and highly optimized) file system code in Linux
  - Changes from over 1200 developers are added to the kernel each release
  - Development never stops – constant incremental improvements and fixes
  - Great processes and pragmatic tools (e.g. “git” distributed source code control and xfstest)
- Broad selection of file systems. More than 50 file systems to choose from not just cifs and ext4!



Linux FS Community is talented
(Picture at 2015 FS Summit in Boston)

Most Active Linux Filesystems (2014-15)

- 4481 kernel filesystem changes in last year (since 3.15-rc3 kernel)!
  - Linux kernel file system activity is continuing to be very strong
  - Lots of improvement in defacto standard Linux xfstest test suite as well
- cifs.ko (cifs/smb3 client) among more active fs
  - Btrfs 684 changesets
  - VFS (overall fs mapping layer and common functions) 581
  - Xfs 429
  - Nfs client 461
  - Ext4 255
  - CIFS/SMB2/SMB3 client 180
  - Nfs server 439 (activity increased dramatically)
- NB: Samba (cifs/smb2/smb3 server) is more active than all those put together since it is broader in scope (by a lot) and also is in user space not in kernel

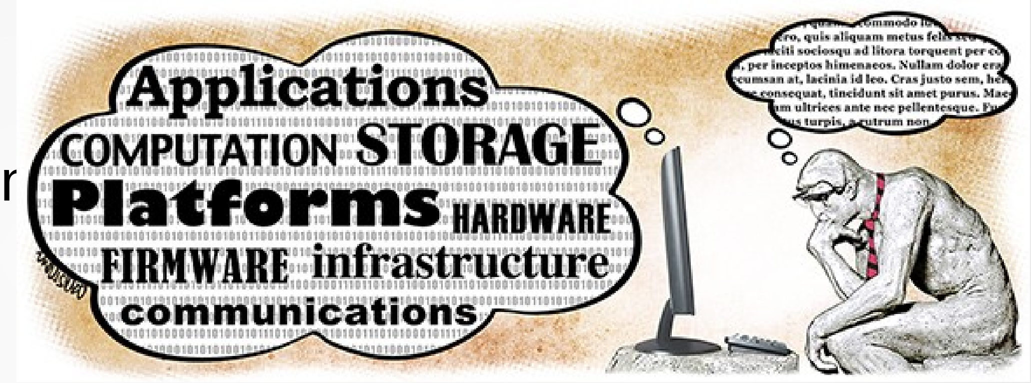
Kernel (including cifs client) improving

- Now we have Linux 4.1-rc3 ie “Hurr Durr I'm a Sheep”

11 months ago we had 3.15 “Shuffling Zombie Juror”

Work In Progress

- Improved xfstest (automated verification test) compatibility (fix a few remaining bugs)
  - Fix fallocate/punch hole bug
- SMB3 (vs. CIFS) implementation gaps
  - CIFS ACLs, KRB5
- Better POSIX emulation/support for SMB3
- Improved ACL support
- Performance improvements



SMB2/SMB3 Optional Feature Status

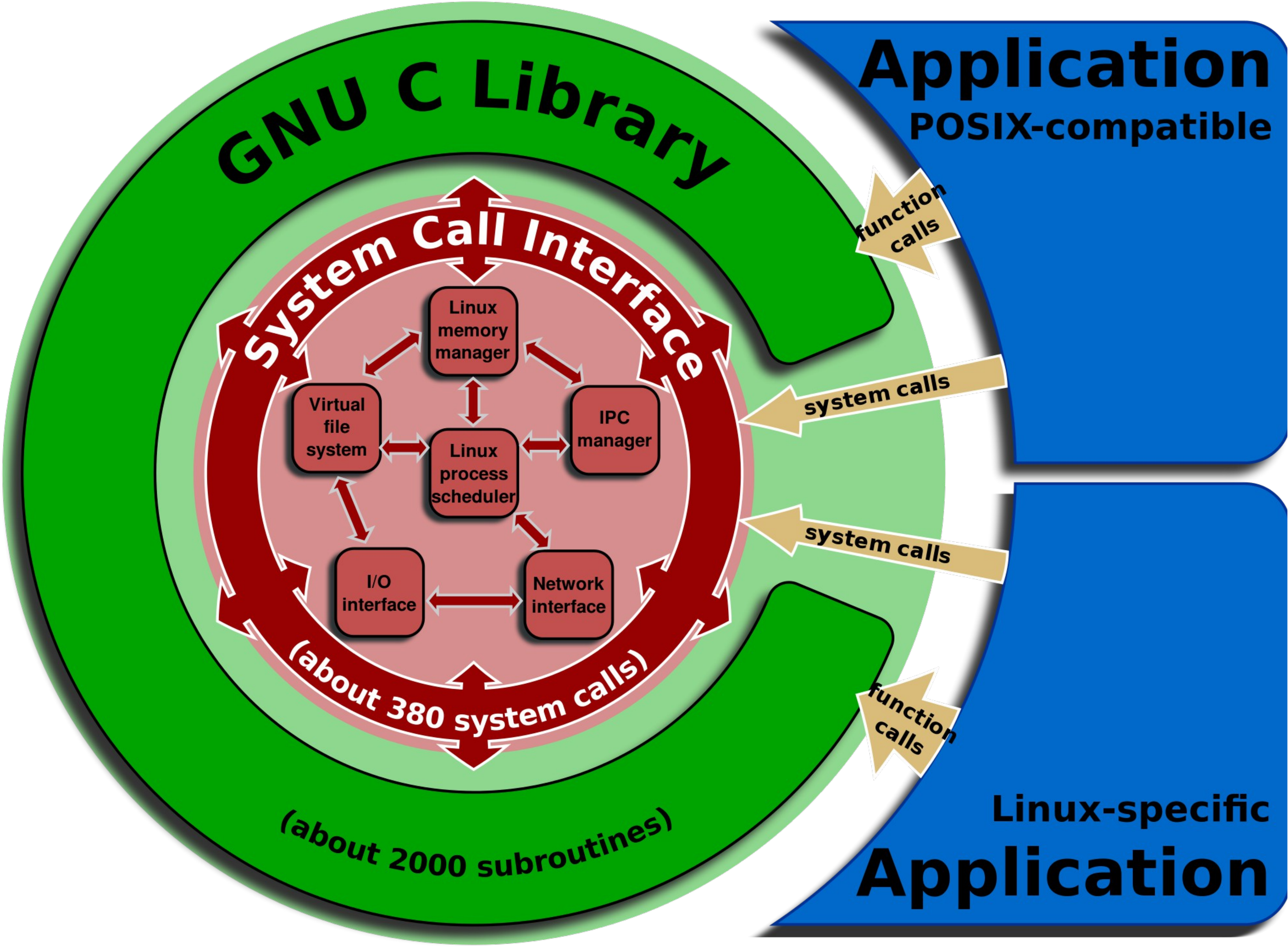
- Security
  - Complete: Downgrade attack protection, SMB2.1 signing
  - In progress: SMB3.11 negotiate contexts
  - Not yet: CBAC (DAC ACLs), per-share encryption
- Data Integrity:
  - Durable Handle Support (complete)
- Performance
  - Complete: multicredit, large I/O
  - Not yet: T10 copy offload, Multichannel, RDMA, directory leases Branch Cache integration, use of compound ops on wire
- Clustering
  - Not yet: Witness protocol integration, Persistent Handles/Continuous Availability
- Other
  - Set/Get Compression and Sparse File support (complete)

POSIX Compatibility

- *The problem:* **SMB/CIFS deprecation** (now that SMB3 is pervasive and more secure and faster and ...). See: http://blogs.technet.com/b/josebda/archive/2015/04/21/the-deprecation-of-smb1-you-should-be-planning-to-get-rid-of-this-old-smb-dialect.aspx
- Specialized POSIX Protocol Extensions that Samba implements are **CIFS only**
- *The Answer:* Move to SMB3 (and later) … BUT …
- *2nd problem:* Full "POSIX" compatibility (actually better to say we need "pragmatic Linux application interoperability") for SMB3 or at least as good CIFS ("good enough")
- *Requirement*:

```
for (all key features)
        SMB3 >= CIFS
```

- Customers don't want SMB3 to be a step back or to break their apps
- Fortunately we are close to solving this and making Linux SMB3 support even better!


GNU C Library
System Call Interface
Linux memory manager
Virtual file system
Linux process scheduler
IPC manager
I/O interface
Network interface
(about 380 system calls)
(about 2000 subroutines)
Application
POSIX-compatible
function calls
system calls
system calls
function calls
Linux-specific
Application

POSIX/Linux Compatibility: Details

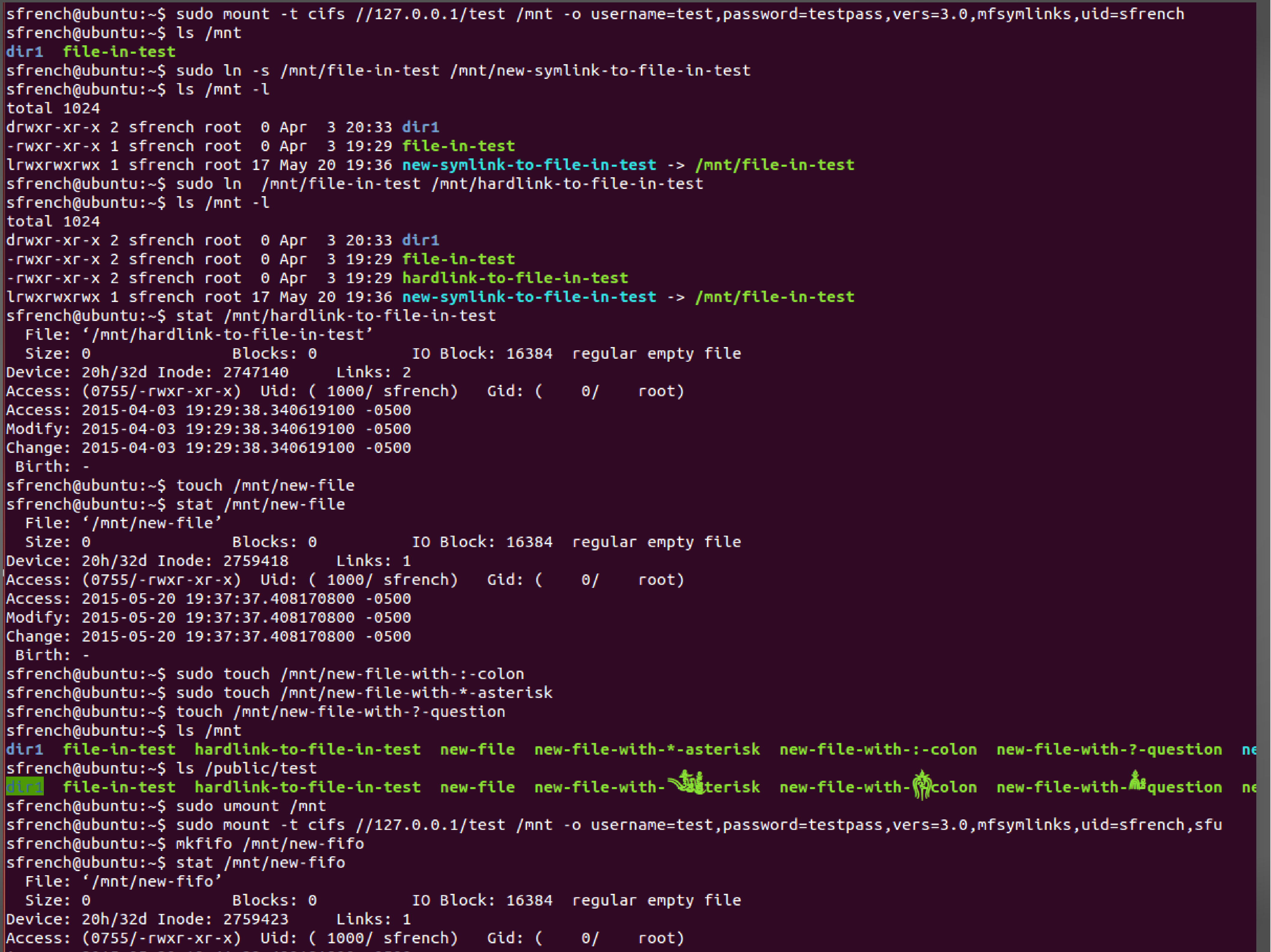
- *Implemented:*
  - *Hardlinks*
- *Emulated: (current cifs.ko SMB3 code)*
  - *POSIX Path Names:* Approximately 7 reserved characters not allowed in SMB3/NTFS etc. *(e.g. ? * \ : !)*
  - *Symlinks* (ala “mfsymlinks” Minshall-French symlinks)
  - *Pseudo-Files:* FIFOs, Pipes, Character Devices (ala “sfu” aka “Microsoft services for unix”)
- *Partial:*
  - *Extended attribute flags* (lsattr/chattr) including compressed flag
  - *POSIX stat and statfs info*
  - *POSIX Byte Range Locks*
- *Not implemented, but emulatable with combination of SMB3 features and/or use of Apple AAPL create context*
  - *Xattrs* (Security/Trusted for SELinux, User xattrs for apps)
  - *POSIX Mode Bits*
  - *POSIX UID/GID ownership information*
  - *Case Sensitivity* in opening paths
- *Not solvable without additional extensions:*
  - *POSIX Delete (unlink) Behavior*

POSIX Compatibility: How to Solve

- *Finish SMB3 ACL support (so we can get mode bits back)*
  - *Allow AAPL create context* so Apple servers and Samba with VFS fruit can return mode bits
- *Detect and recognize case sensitive volumes*
- *Enable cifs uid upcall for SMB3 (to get winbind to map uids/gids for ownership information)*
  - *Only loosely related: Enable krb5 for SMB3* (only works for cifs in current code)
- *Cleanup Microsoft “nfs symlink” code to recognize Windows symlinks*
- *Add extensions (trivial create context flag): enables posix open/unlink/byte-range locking behavior*
- *Improvements to Samba too,* for example:
  - Map of (non-wide-link) mfsymlinks (or equivalent reparse points) to real symlinks on fly

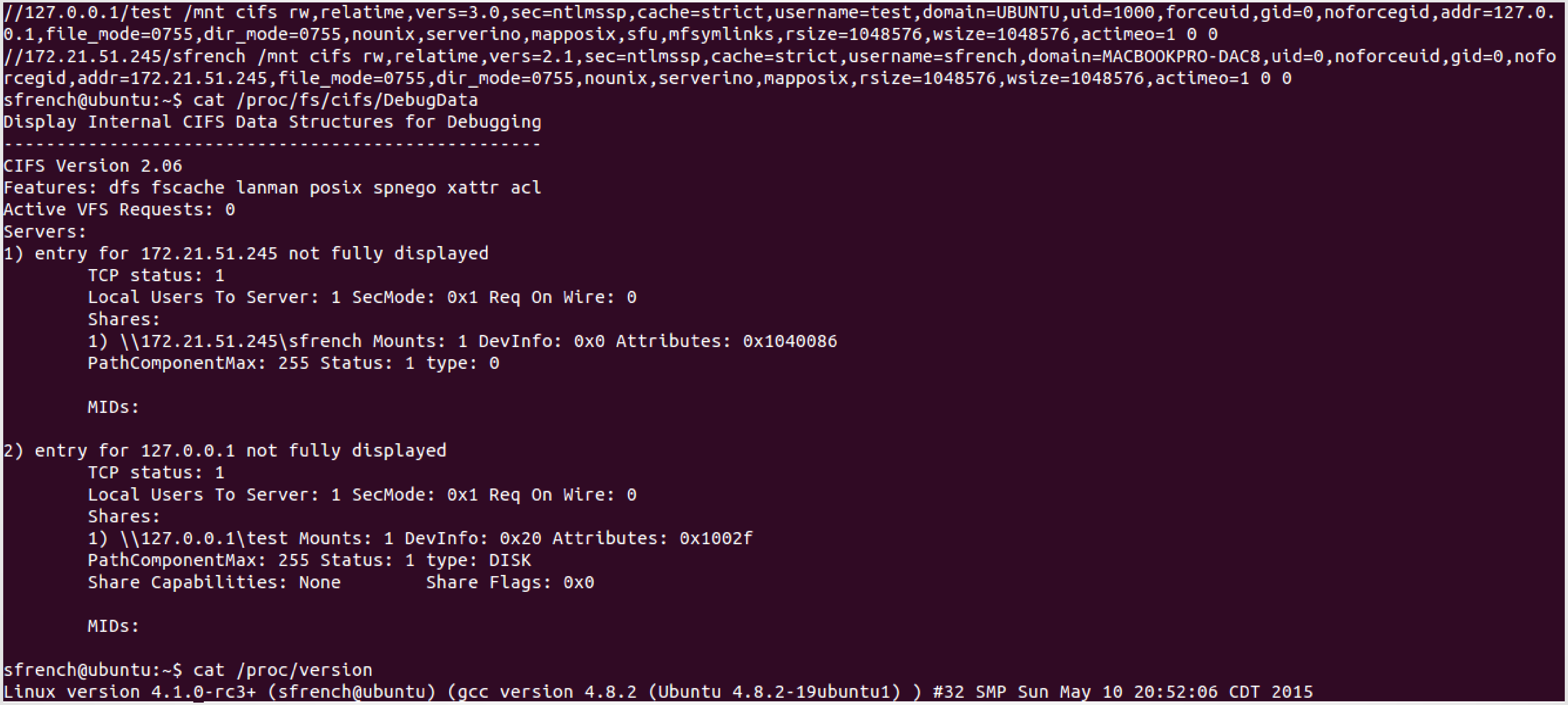
Demo

- Client:
  - Current kernel (4.1-rc) mainline (on an Ubuntu VM in this machine)
- Mounted
  - via SMB3.0 to Samba server version 4.1.6 Ubuntu
  - and Mac … screenshots then copied via SMB2.1 mount to host
- Most features worked
  - Still work to do (returning mode bits from ACL or AAPL e.g.)
  - But also noticed bug in detection of FIFOs
- NB: (Demo does not show “sfu”mount option which was added partway through in another window)



```
sfrench@ubuntu:~$ sudo mount -t cifs //127.0.0.1/test /mnt -o username=test,password=testpass,vers=3.0,mfsymlinks,uid=sfrench
sfrench@ubuntu:~$ ls /mnt
dir1  file-in-test
sfrench@ubuntu:~$ sudo ln -s /mnt/file-in-test /mnt/new-symlink-to-file-in-test
sfrench@ubuntu:~$ ls /mnt -l
total 1024
drwxr-xr-x 2 sfrench root  0 Apr  3 20:33 dir1
-rwxr-xr-x 1 sfrench root  0 Apr  3 19:29 file-in-test
lrwxrwxrwx 1 sfrench root 17 May 20 19:36 new-symlink-to-file-in-test -> /mnt/file-in-test
sfrench@ubuntu:~$ sudo ln  /mnt/file-in-test /mnt/hardlink-to-file-in-test
sfrench@ubuntu:~$ ls /mnt -l
total 1024
drwxr-xr-x 2 sfrench root  0 Apr  3 20:33 dir1
-rwxr-xr-x 2 sfrench root  0 Apr  3 19:29 file-in-test
-rwxr-xr-x 2 sfrench root  0 Apr  3 19:29 hardlink-to-file-in-test
lrwxrwxrwx 1 sfrench root 17 May 20 19:36 new-symlink-to-file-in-test -> /mnt/file-in-test
sfrench@ubuntu:~$ stat /mnt/hardlink-to-file-in-test
  File: ‘/mnt/hardlink-to-file-in-test’
  Size: 0               Blocks: 0          IO Block: 16384  regular empty file
Device: 20h/32d Inode: 2747140     Links: 2
Access: (0755/-rwxr-xr-x)  Uid: ( 1000/ sfrench)   Gid: (    0/    root)
Access: 2015-04-03 19:29:38.340619100 -0500
Modify: 2015-04-03 19:29:38.340619100 -0500
Change: 2015-04-03 19:29:38.340619100 -0500
 Birth: -
sfrench@ubuntu:~$ touch /mnt/new-file
sfrench@ubuntu:~$ stat /mnt/new-file
  File: ‘/mnt/new-file’
  Size: 0               Blocks: 0          IO Block: 16384  regular empty file
Device: 20h/32d Inode: 2759418     Links: 1
Access: (0755/-rwxr-xr-x)  Uid: ( 1000/ sfrench)   Gid: (    0/    root)
Access: 2015-05-20 19:37:37.408170800 -0500
Modify: 2015-05-20 19:37:37.408170800 -0500
Change: 2015-05-20 19:37:37.408170800 -0500
 Birth: -
sfrench@ubuntu:~$ sudo touch /mnt/new-file-with-:-colon
sfrench@ubuntu:~$ sudo touch /mnt/new-file-with-*-asterisk
sfrench@ubuntu:~$ touch /mnt/new-file-with-?-question
sfrench@ubuntu:~$ ls /mnt
dir1  file-in-test  hardlink-to-file-in-test  new-file  new-file-with-*-asterisk  new-file-with-:-colon  new-file-with-?-question  ne
sfrench@ubuntu:~$ ls /public/test
dir1  file-in-test  hardlink-to-file-in-test  new-file  new-file-with-[illegible]terisk  new-file-with-[illegible]colon  new-file-with-[illegible]question  ne
sfrench@ubuntu:~$ sudo umount /mnt
sfrench@ubuntu:~$ sudo mount -t cifs //127.0.0.1/test /mnt -o username=test,password=testpass,vers=3.0,mfsymlinks,uid=sfrench,sfu
sfrench@ubuntu:~$ mkfifo /mnt/new-fifo
sfrench@ubuntu:~$ stat /mnt/new-fifo
  File: ‘/mnt/new-fifo’
  Size: 0               Blocks: 0          IO Block: 16384  regular empty file
Device: 20h/32d Inode: 2759423     Links: 1
Access: (0755/-rwxr-xr-x)  Uid: ( 1000/ sfrench)   Gid: (    0/    root)
```

Detailed information on the mounts



```
//127.0.0.1/test /mnt cifs rw,relatime,vers=3.0,sec=ntlmssp,cache=strict,username=test,domain=UBUNTU,uid=1000,forceuid,gid=0,noforcegid,addr=127.0.
0.1,file_mode=0755,dir_mode=0755,nounix,serverino,mapposix,sfu,mfsymlinks,rsize=1048576,wsize=1048576,actimeo=1 0 0
//172.21.51.245/sfrench /mnt cifs rw,relatime,vers=2.1,sec=ntlmssp,cache=strict,username=sfrench,domain=MACBOOKPRO-DAC8,uid=0,noforceuid,gid=0,nofo
rcegid,addr=172.21.51.245,file_mode=0755,dir_mode=0755,nounix,serverino,mapposix,rsize=1048576,wsize=1048576,actimeo=1 0 0
sfrench@ubuntu:~$ cat /proc/fs/cifs/DebugData
Display Internal CIFS Data Structures for Debugging
---------------------------------------------------
CIFS Version 2.06
Features: dfs fscache lanman posix spnego xattr acl
Active VFS Requests: 0
Servers:
1) entry for 172.21.51.245 not fully displayed
        TCP status: 1
        Local Users To Server: 1 SecMode: 0x1 Req On Wire: 0
        Shares:
        1) \\172.21.51.245\sfrench Mounts: 1 DevInfo: 0x0 Attributes: 0x1040086
        PathComponentMax: 255 Status: 1 type: 0

        MIDs:

2) entry for 127.0.0.1 not fully displayed
        TCP status: 1
        Local Users To Server: 1 SecMode: 0x1 Req On Wire: 0
        Shares:
        1) \\127.0.0.1\test Mounts: 1 DevInfo: 0x20 Attributes: 0x1002f
        PathComponentMax: 255 Status: 1 type: DISK
        Share Capabilities: None        Share Flags: 0x0

        MIDs:

sfrench@ubuntu:~$ cat /proc/version
Linux version 4.1.0-rc3+ (sfrench@ubuntu) (gcc version 4.8.2 (Ubuntu 4.8.2-19ubuntu1) ) #32 SMP Sun May 10 20:52:06 CDT 2015
```

Other Features under investigation

- SMB3 ACL support
- Better streams support (how to list streams, useful for backup e.g.)
- DCE/RPC over SMB3: Pipe reads/write over IPC$ pseudo-mount
- Recovery of pending byte range locks after server failure (we already recover successful locks)
- Investigation into additional copy offload (server side copy) methods
- Full Linux xattr support
  - Empty xattr (name but no value)
  - Case sensitive xattr values
  - Security (SELinux) namespace (and others)

Improvements by release (continued)

- 3.12 40 changes, cifs version 2.02: **SMB3 support much improved**
  - SMB3.02 dialect negotiation added
  - Authentication overhaul
  - SMB3 multiuser signing improvements, (thank you Shirish!) allows per-user signing keys on ses
  - SMB2/3 symlink support (can follow Windows symlinks)
  - Improved data integrity: Lease improvements (thank you Pavel!)
  - debugging improvements
- 3.13 34 changes
  - Add support for setting (and getting) per-file compression (e.g. “chattr +c /mnt/filename”)
  - Add SMB copy offload ioctl (CopyChunk) for very fast server side copy
  - Add secure negotiate support (protect SMB3 mounts against downgrade attacks)
  - Bugfixes (including for setfacl and reparse point/symlink fixes)
  - Allow for O_DIRECT opens on directio (cache=none) mounts. Helps apps that require directio such as newer specsfs benchmark and some databases
  - Server network adapter and disk/alignment/sector info now visible in /proc/fs/cifs/DebugData
- 3.14 27 changes
  - Security fix for make sure we don't send illegal length when passed invalid iovec or one with invalid lengths
  - Bug fixes (SMB3 large write and various stability fixes) and aio write and also fix DFS referrals when mounted with Unix extensions

Improvements by release (continued)

- 3.15 18 changes
  - Various minor bug fixes (include aio/write, append, xattr, and also in metadata caching)
- 3.16 25 changes
  - Allow multiple mounts to same server with different dialects
  - Authentication session establishment rewrite to improve gssapi support
  - Fix mapchars (to allow reserverd characters like : in paths) over smb3 mounts
- 3.17 65 changes (cifs version 2.04 – visible in modinfo)
  - Much faster SMB3 large read/write: including multicredit support (thank you Pavel!)
  - Many SMB3 fixes (found by newly updated automated fs tests: “xfstests”)
  - Directio allowed on cache=strict mounts
  - Fallocate/sparse file support for SMB3
  - Fixed SMB 2.1 mounts to MacOS
- 3.18 (Some highlights of what to expect in next kernel)
  - SMB3 Emulated symlinks: Mfsymlink support for smb2.1/smb3 (complete).
  - SMB3 POSIX Reserved Character mapping: support for reserved characters e.g. * : ? < > etc. (complete)
  - Workaround MacOS problem with CIFS Unix Extensions from Linux

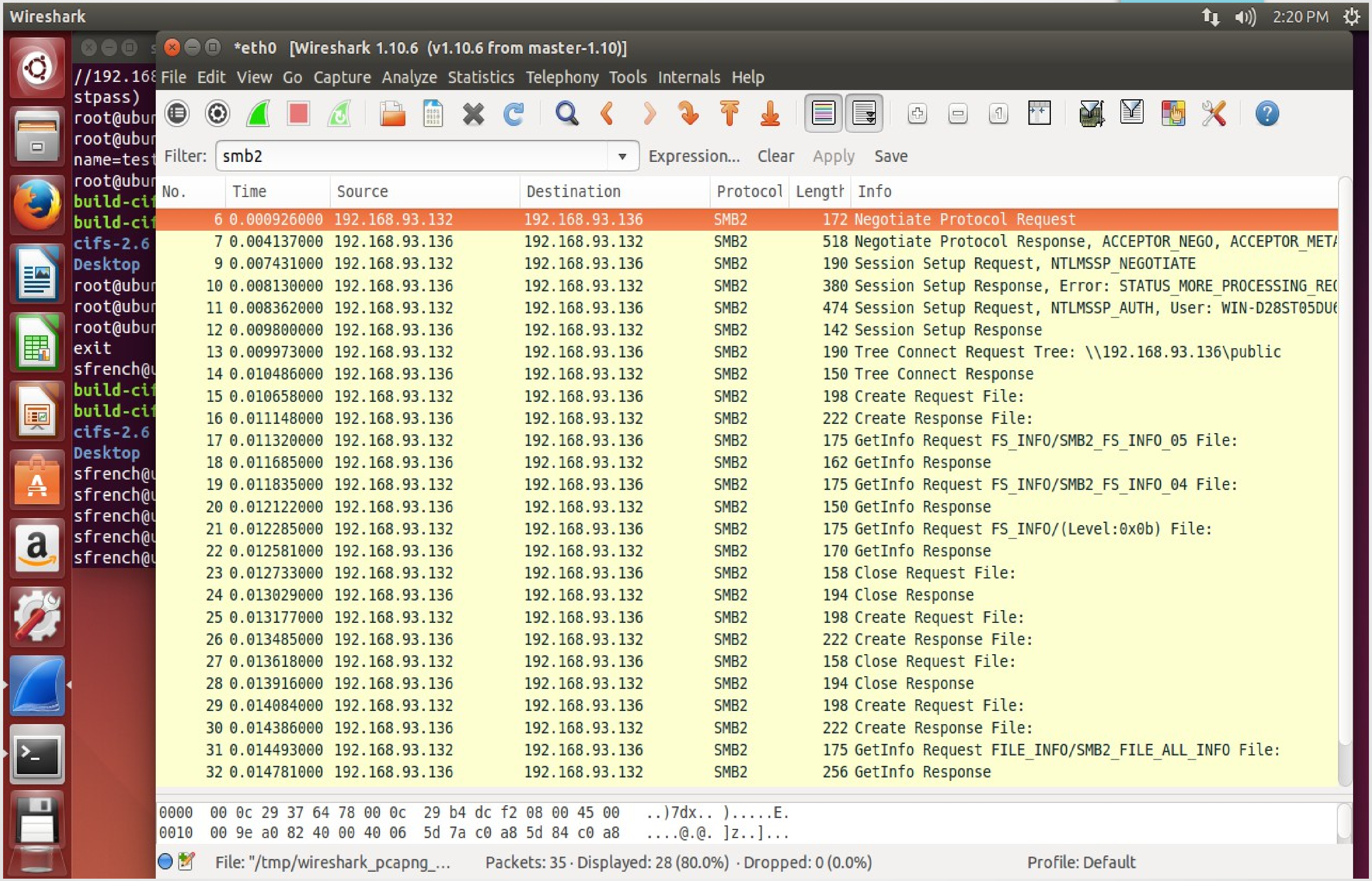
Improvements by release (continued)

- 3.19 26 changesets
  - Fix Oplock bug, inode caching bug and ioctl clone bug
  - Fix conflicts between SecurityFlags (which allowed CONFIG_MUST_LANMAN and CONFIG_MUST_PLNTXT
  - Improve fallocate support
- Linux 4.0 (!) 21 changesets
  - Various minor stability fixes
- Linux 4.1
  - Stability fixes: Mapchars fix, fix to allow Unicode surrogate pairs (improved character conversion for some Asian languages), DFS fix, inode number reuse fix
- Linux 4.2 (expected)
  - SMB 3.11 (Windows 10) dialect support (improved security)
  - And more!!

Cifs-utils

- The userspace utils: mount.cifs, cifs.upcall,set/getcifsacl,cifscreds, idmapwb (idmap plugin),pam_cifscreds
  - thanks to Jeff Layton for maintaining cifs-utils
- 6 changesets over the past year
  - Current version is 6.4.1
  - Minor bugfixes

SMB3.02 Mount to Windows


Wireshark
2:20 PM
*eth0 [Wireshark 1.10.6 (v1.10.6 from master-1.10)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Expression... Clear Apply Save

```
//192.168
stpass)
root@ubun
root@ubun
name=test
root@ubun
build-cif
build-cif
cifs-2.6
Desktop
root@ubun
root@ubun
root@ubun
exit
sfrench@u
build-cif
build-cif
cifs-2.6
Desktop
sfrench@u
sfrench@u
sfrench@u
sfrench@u
sfrench@u
```

Filter: smb2

```
No.  Time         Source          Destination     Protocol Length Info
   6 0.000926000  192.168.93.132  192.168.93.136  SMB2      172 Negotiate Protocol Request
   7 0.004137000  192.168.93.136  192.168.93.132  SMB2      518 Negotiate Protocol Response, ACCEPTOR_NEGO, ACCEPTOR_META
   9 0.007431000  192.168.93.132  192.168.93.136  SMB2      190 Session Setup Request, NTLMSSP_NEGOTIATE
  10 0.008130000  192.168.93.136  192.168.93.132  SMB2      380 Session Setup Response, Error: STATUS_MORE_PROCESSING_REQ
  11 0.008362000  192.168.93.132  192.168.93.136  SMB2      474 Session Setup Request, NTLMSSP_AUTH, User: WIN-D28ST05DU6
  12 0.009800000  192.168.93.136  192.168.93.132  SMB2      142 Session Setup Response
  13 0.009973000  192.168.93.132  192.168.93.136  SMB2      190 Tree Connect Request Tree: \\192.168.93.136\public
  14 0.010486000  192.168.93.136  192.168.93.132  SMB2      150 Tree Connect Response
  15 0.010658000  192.168.93.132  192.168.93.136  SMB2      198 Create Request File:
  16 0.011148000  192.168.93.136  192.168.93.132  SMB2      222 Create Response File:
  17 0.011320000  192.168.93.132  192.168.93.136  SMB2      175 GetInfo Request FS_INFO/SMB2_FS_INFO_05 File:
  18 0.011685000  192.168.93.136  192.168.93.132  SMB2      162 GetInfo Response
  19 0.011835000  192.168.93.132  192.168.93.136  SMB2      175 GetInfo Request FS_INFO/SMB2_FS_INFO_04 File:
  20 0.012122000  192.168.93.136  192.168.93.132  SMB2      150 GetInfo Response
  21 0.012285000  192.168.93.132  192.168.93.136  SMB2      175 GetInfo Request FS_INFO/(Level:0x0b) File:
  22 0.012581000  192.168.93.136  192.168.93.132  SMB2      170 GetInfo Response
  23 0.012733000  192.168.93.132  192.168.93.136  SMB2      158 Close Request File:
  24 0.013029000  192.168.93.136  192.168.93.132  SMB2      194 Close Response
  25 0.013177000  192.168.93.132  192.168.93.136  SMB2      198 Create Request File:
  26 0.013485000  192.168.93.136  192.168.93.132  SMB2      222 Create Response File:
  27 0.013618000  192.168.93.132  192.168.93.136  SMB2      158 Close Request File:
  28 0.013916000  192.168.93.136  192.168.93.132  SMB2      194 Close Response
  29 0.014084000  192.168.93.132  192.168.93.136  SMB2      198 Create Request File:
  30 0.014386000  192.168.93.136  192.168.93.132  SMB2      222 Create Response File:
  31 0.014493000  192.168.93.132  192.168.93.136  SMB2      175 GetInfo Request FILE_INFO/SMB2_FILE_ALL_INFO File:
  32 0.014781000  192.168.93.136  192.168.93.132  SMB2      256 GetInfo Response
```

```
0000  00 0c 29 37 64 78 00 0c  29 b4 dc f2 08 00 45 00   ..)7dx.. ).....E.
0010  00 9e a0 82 40 00 40 06  5d 7a c0 a8 5d 84 c0 a8   ....@.@. ]z..]...
```

File: "/tmp/wireshark_pcapng_... Packets: 35 · Displayed: 28 (80.0%) · Dropped: 0 (0.0%) Profile: Default

Using SMB3

- Practical tips
  - Use -o vers=3.0 to Samba or Windows (or vers=3.02 to latest Windows, consider vers=2.1 to MacOS or 3.0 to most recent Mac)
  - Mount options to consider
    - “mfsymlinks” (3.18 or later kernel)
    - “sfu” option enables creation of FIFOs and char devices
    - Consider experimenting with default rsize/wsize (which is 1MB) to improve large file I/O performance
- Restrictions
  - Case sensitivity
  - POSIX vs. Windows byte range locks, and unlink behavior

SMB3 Kernel Client Status

- SMB3 support is solid (and large file I/O FAST!), but lacks many optional features
  - Metadata performance expected to be slower (need to add open/query compounding)
- Badly need to prototype Apple's SMB2.1/SMB3 “AAPL” create context” to determine if adequately addresses a few remaining POSIX compatibility issues)
- Can mount with SMB2.02, SMB2.1, SMB3, SMB3.02
  - Specify vers=2.0 or vers=2.1 or 3.0 or 3.02 on mount
  - Default is cifs but also mounting with vers=1.0 also forces using smb/cifs protocol
  - Default will change to SMB3 soon (likely with new “mount -t smb3” ie using new “/sbin/mount.smb” and/or “mount.smb3” symlink – to avoid changing “mount -t cifs” behavior for existing users)

SMB3 Performance considerations

- Informal perf results 3.16-rc4 (Ubuntu) client. Server Windows 8.1.   VMs on same host (host disk is fairly fast SSD).
    - Copy to server performance increased about 20% percent (similar with or without conv=fdatasync)
    - dd if=/dev/zero of=/mnt/targetfile bs=80M count=25
    - 1st run copy to empty directory, 2nd run copy over target, (pattern repeated multiple times) averaging results
    - New code (with Pavel's patches)
    - --
    - CIFS 167MB/s
    - SMB3 200MB/s

    - Existing code (without his patches)
    - ---
    - SMB3 166MB/s
    - CIFS 164.5MB/s

More SMB3 Performance

- For large file reading SMB3 performance with Pavel's patches increased 76% over existing SMB3 code
- dd of=/dev/null if=/mnt/targetfile bs=80M count=25 (mounting and unmounting between attempts to avoid caching effects on the client)
- New code (with Pavel's patches)
- --
- CIFS 114MB/s
- SMB3 216MB/s

- Existing code (without his patches)
- --
- SMB3 123MB/s
- CIFS 110MB/s

More SMB3 Performance Linux->Linux

- client Ubuntu with 3.16-rc4 with Pavel's patches, srv Fedora 20 (3.14.9 kernel Samba server version 4.1.9)
- dd if=/mnt/testfile of=/dev/null bs=50M count=30
- testfile is 1.5GB existing file, unmount/mount in between each large file copy to avoid any caching effect on client (although server will have cached it)

- SMB3 averaged 199MB/sec reads (copy from server)
- CIFS averaged 170MB/sec reads (copy from server)
- NFSv3 averaged 116MB/sec (copy from server)
- NFSv4 and v4.1 averaged 110MB/sec (copy from server)

- Write speeds (doing dd if=/dev/zero of=/mnt/testfile bs=60M count=25) more varied but averaged similar speeds for copy to server for both NFSv3/v4/v4.1 and SMB3 (~175MB/s)
- NB: Additional NFS server and client scalability patches have recently been added to kernel (it is possible that they may help these cases)

Testing … testing … testing

- One of the goals last summer was to improve automated testing of cifs.ko
  - Multiple cifs bugs found, test automation much improved, approximately 5 bugs/features remain to be fixed for full xfstest compatibility
  - See https://wiki.samba.org/index.php/Xfstesting-cifs
- Functional tests:
  - Xfstest is the standard file system test bucket for Linux
    - Runs over local file systems, nfs, and now cifs/smb3
      - Found multiple bugs when ran this first (including Samba bug – with times before Epoch e.g.)
    - Challenge to figure out which tests *should* work (since some tests are skipped when run over nfs and cifs)
  - Other functional tests include cthon, dbench, fsx. Cthon also has recently been updated to better support cifs
- Performance/scalability testing
  - Specsfs works over cifs mounts (performance testing)
  - Big recent improvements in scalability of dbench (which can run over mounts)
  - Various other linux perf fs tests work over cifs (iozone etc.)
  - Need to figure out how to get synergy with iostats/nfsstats/nfsometer

XFSTEST current status

- Multiple server bugs found too
- Client bugs:
  - As with NFS, there are some intractable mtime consistency problems due to server/client last write time differences/delays, but these tests could be skipped
  - Generic tests: 011 (dirstress), 023 and 245 (rename), 075/091/127/263 (fsx failures fallocate related), 239 (need ACLs), 313 (timestamps)

- The Future of SMB3 and Linux is very bright
- Let's continue its improvement!

Thank you for your time