SNIA DEVELOPER CONFERENCE

SDC 24

BY Developers FOR Developers

September 16-18, 2024
Santa Clara, CA

What's New in macOS SMB Client

2024 Version – Sonoma 14.x, Sequoia 15.0

Presented by Brad Suinn

Who am I?

Brad Suinn – Network File Systems Engineer

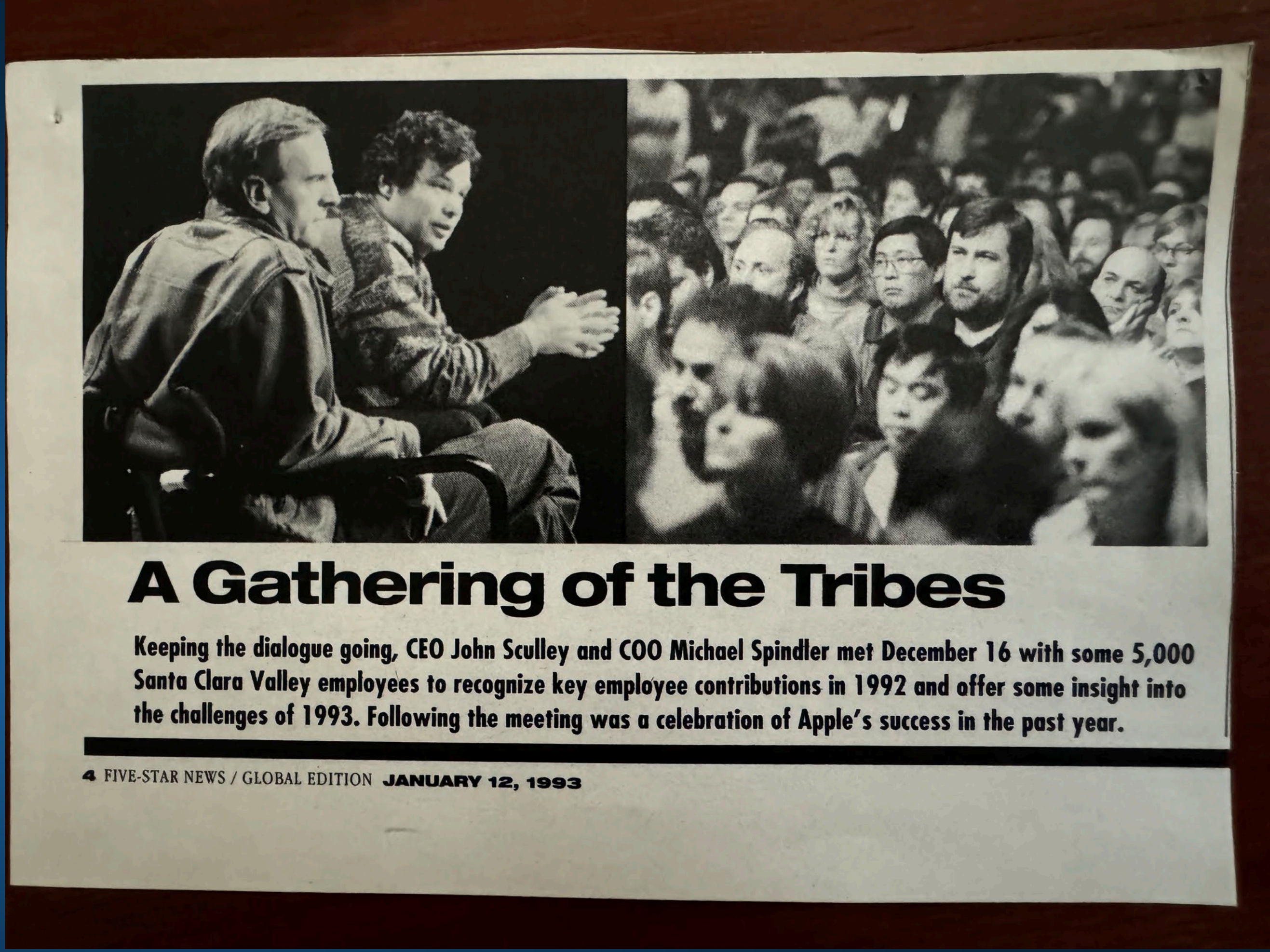
- Joined Apple in late 1989
- 1989 – 1993 QA Engineer
- 1993+ Development Engineer
- First Project I worked on at Apple
  - Macintosh IIfx
    - Motorola 68030 @ 40 MHz
    - 4 MB RAM expandable to 128 MB
    - 80 or 160 MB Hard Disk Drive
    - System 6.x

Topics

- Sonoma Changes
  - Signing algorithm updates
  - nsmb.conf updates
  - connect_to_sharedisk
  - Nanosecond time support
- Sequoia Changes
  - Large directory enumeration improvement
  - Multichannel client side Receive Side Scaling (RSS)
  - SMB Compression
- Questions and maybe answers

Signing Algorithm Updates

Sonoma and later



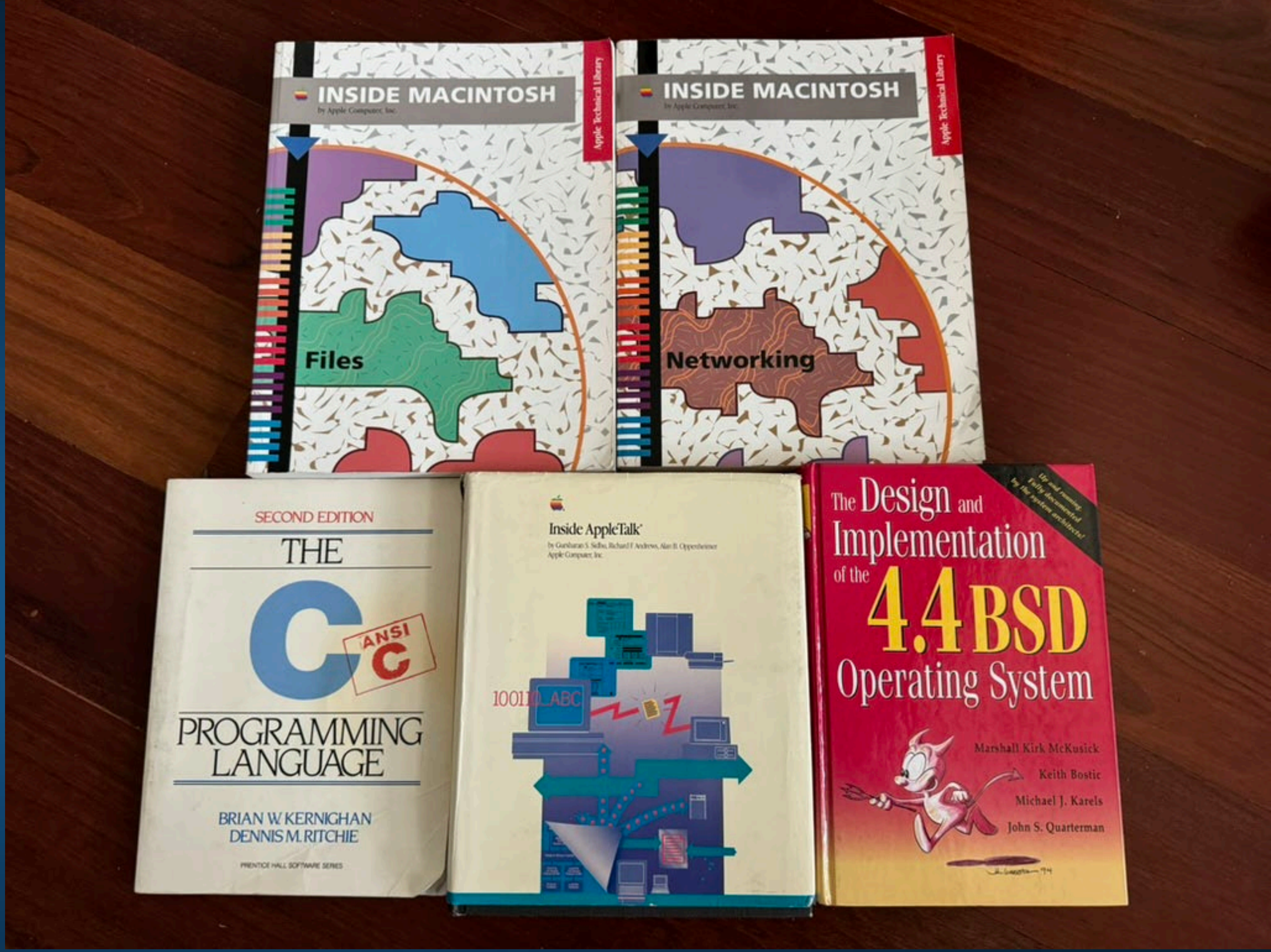
A Gathering of the Tribes

Keeping the dialogue going, CEO John Sculley and COO Michael Spindler met December 16 with some 5,000 Santa Clara Valley employees to recognize key employee contributions in 1992 and offer some insight into the challenges of 1993. Following the meeting was a celebration of Apple's success in the past year.

4 FIVE-STAR NEWS / GLOBAL EDITION JANUARY 12, 1993

Signing Algorithm Updates

- Negotiate Context SMB2_SIGNING_CAPABILITIES
  - AES-GMAC
  - AES-CMAC
- nsmb.conf in global or server sections
  - signing_alg_map=<bitmap>
  - Bit 0 – enable AES-CMAC
  - Bit 1 – enable AES-GMAC
  - Default has AES-GMAC and AES-CMAC enabled
    - AES-GMAC is listed first in Negotiate Context



nsmb.conf Updates

Sonoma and later

nsmb.conf Keywords Moved To New Sections

- Has three sections: default, server, server:share
  - default – applies to all servers and shares
  - server – applies to just that server and any of its shares
  - server:share – applies to just that server and that specific share
- Keywords moved from “default” to “default or server” sections
  - protocol_vers_map
  - signing_required and signing_req_vers
  - validate_neg_off
  - encrypt_cipher_map
  - force_sess_encrypt
  - force_share_encrypt
- Keywords moved from “server:share” to “server” section
  - mc_on and mc_prefer_wired

Share Disk Mode

- For ASi computers, “Share Disk Mode” replaces ”Target Disk Mode”
- This is a special option when booting in Recovery Mode
- The ASi computer will be visible to another client computer connected via USB, USB-C or Thunderbolt cable
- Client computers use Guest to log in to a Share Disk Mode server using SMB
- nsmb.conf “minauth” is not used in this case since Guest is being used

connect_to_sharedisk

- Add new security option “connect_to_sharedisk” to global section
  - Allow/disallow this client to connect to an ASi computer that is booted in Share Disk Mode
  - Default setting is “yes”
- Example:
  - Corporate computer with private data and locked down so it can not copy data to any external storage/network volumes
  - Personal ASi computer brought in, booted into Share Disk Mode and connected via USB-C cable to the corporate computer
  - Corporate computer then mounts the Share Disk Mode server
  - Data can now be copied between the corporate computer and the personal computer

©2024 SNIA. © Apple. All Rights Reserved.

Nanoseconds Support

Sonoma and later

Nanoseconds Support

- In macOS Ventura and earlier, when converting from SMB time to file system time, only the seconds field was converted
- Starting with Sonoma, the file system seconds and nanosecond fields are converted
- Note: SMB time is in 100 nanosecond intervals

Large Directory Enumeration Improvement

Sequoia and later

Pre Sequoia Enumeration Behavior

- Open directory, send query directory(s), parse out a reply entry, add it to enumeration cache, add it to user buffer and keep parsing/adding until user buffer is full.
- When enumeration cache is full and continuing to enumerate
  - Close directory, open directory, send query directory(s), parse reply entries until get to resume entry. Add resume entry to user buffer.
  - Keep parsing entries and adding to user buffer until user buffer is full or entire directory is enumerated.
- Restarting the query directory from the beginning and scanning to the resume entry can have some significant delays
- For very large directories, this restart/resume can happen a lot and scales poorly

©2024 SNIA. © Apple. All Rights Reserved.

Sequoia Enumeration Behavior

- Open directory, send query directory(s), parse out a reply entry, add it to enumeration cache, add it to user buffer and keep parsing/adding until user buffer is full.
  - Save any query directory replies that have not been parsed
- When enumeration cache is full and continuing to enumerate
  - Continue parsing from saved query directory replies and adding them to user buffer
  - If no more saved query directory replies, send query directory(s) and parse the replies until user buffer is full or entire directory is enumerated
- Pattern should be reduced to OpenDir/QueryDir, QueryDir(s), then CloseDir
- For non AAPL Create Context/ReadDirAttr servers, still have the extra request/replies to collect the extra meta data

Multichannel Client Side RSS

Sequoia and later

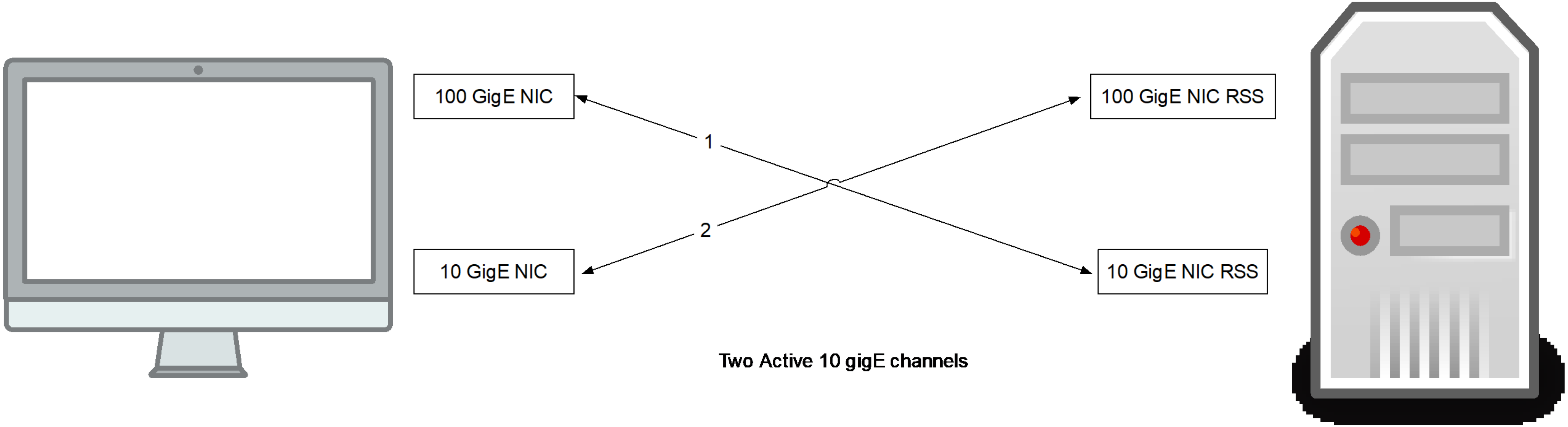
Client Side RSS Support

- In the previous behavior, only one SMB channel was allowed per client Network Interface Controller (NIC)
- Starting in Sequoia, if the client NIC supports RSS, then up to 4 SMB channels are allowed by default
- “smbutil multichannel –a” will show if client or server NICs support RSS
- nsmb.conf
  - mc_max_channels – max number of channels between client and server
  - mc_srvr_rss_channels – max RSS channels per server NIC
  - mc_clnt_rss_channels – max RSS channels per client NIC

Client Side RSS Benefits

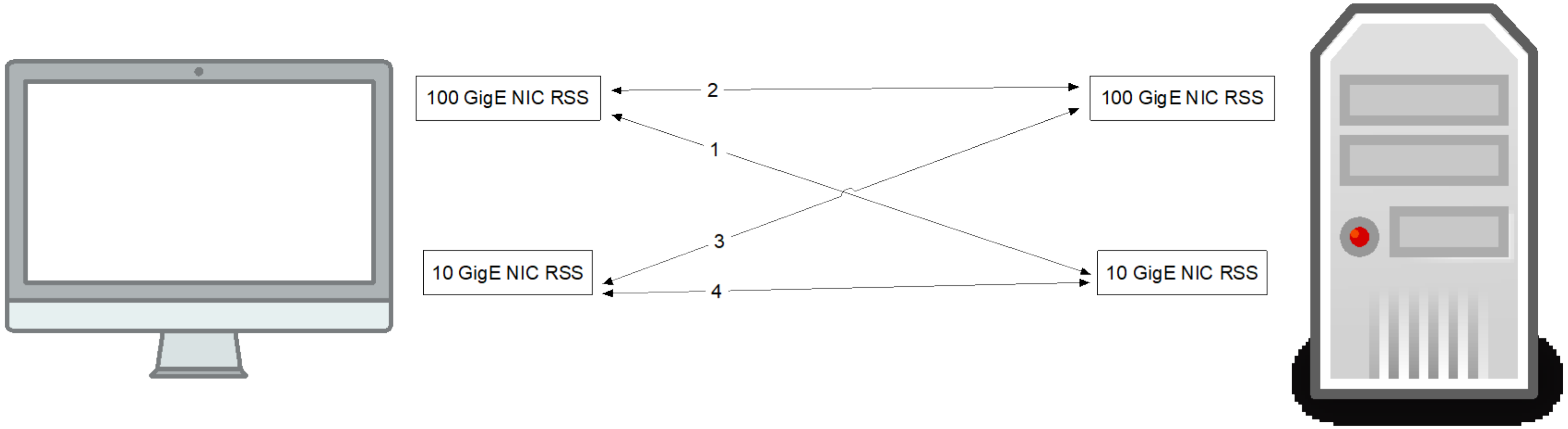
- Allows better selection of channels between client and server when different speed NICs are available
- Allows more SMB channels between client RSS NIC and server RSS NIC which improves performance
- Example Setup:
  - Client has 10 gigE RSS NIC and 100 gigE RSS NIC
  - Server has 10 gigE RSS NIC and 100 gigE RSS NIC

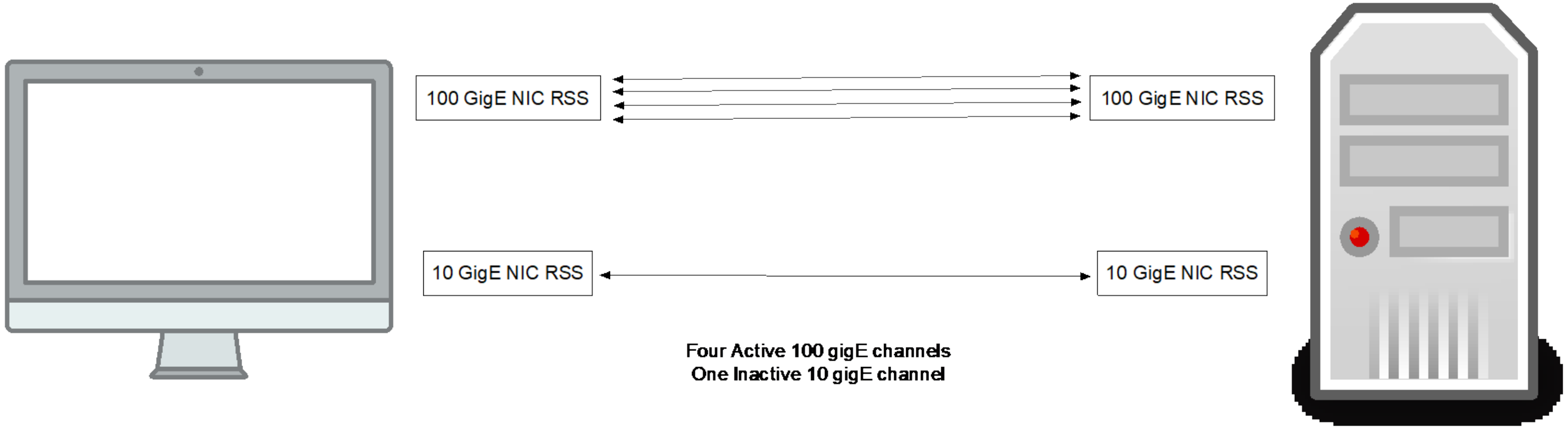
Sonoma Channel Selection Example



Two Active 10 gigE channels

Sequoia Channel Selection Example





Four Active 100 gigE channels
One Inactive 10 gigE channel

SMB Compression

Sequoia and later

MENU
iPod
30GB
Serial No.: JQ515GUGSAY
Designed by Apple in California Assembled in China
Model No.: A1099 EMC No.: 2022 Rated 5-30V ⎓ 1A Max.
TM and © 2005 Apple Computer, Inc. All rights reserved.

SMB Compression Notes

- Ensure Windows is running the latest version
- Microsoft provided example files and matching compressed files for the different algorithms which allowed verification of our algorithms
- For requests, sign first, then compress, then encrypt
- For replies, decrypt first, then decompress, then check signing
- Write data can “fail” to compress and be sent as non compressed
  - Compressed data is larger than original data
  - After too many failures, compression is disabled for that file
- SMB protocol allows any request/reply to be compressed

Supported SMB Compression

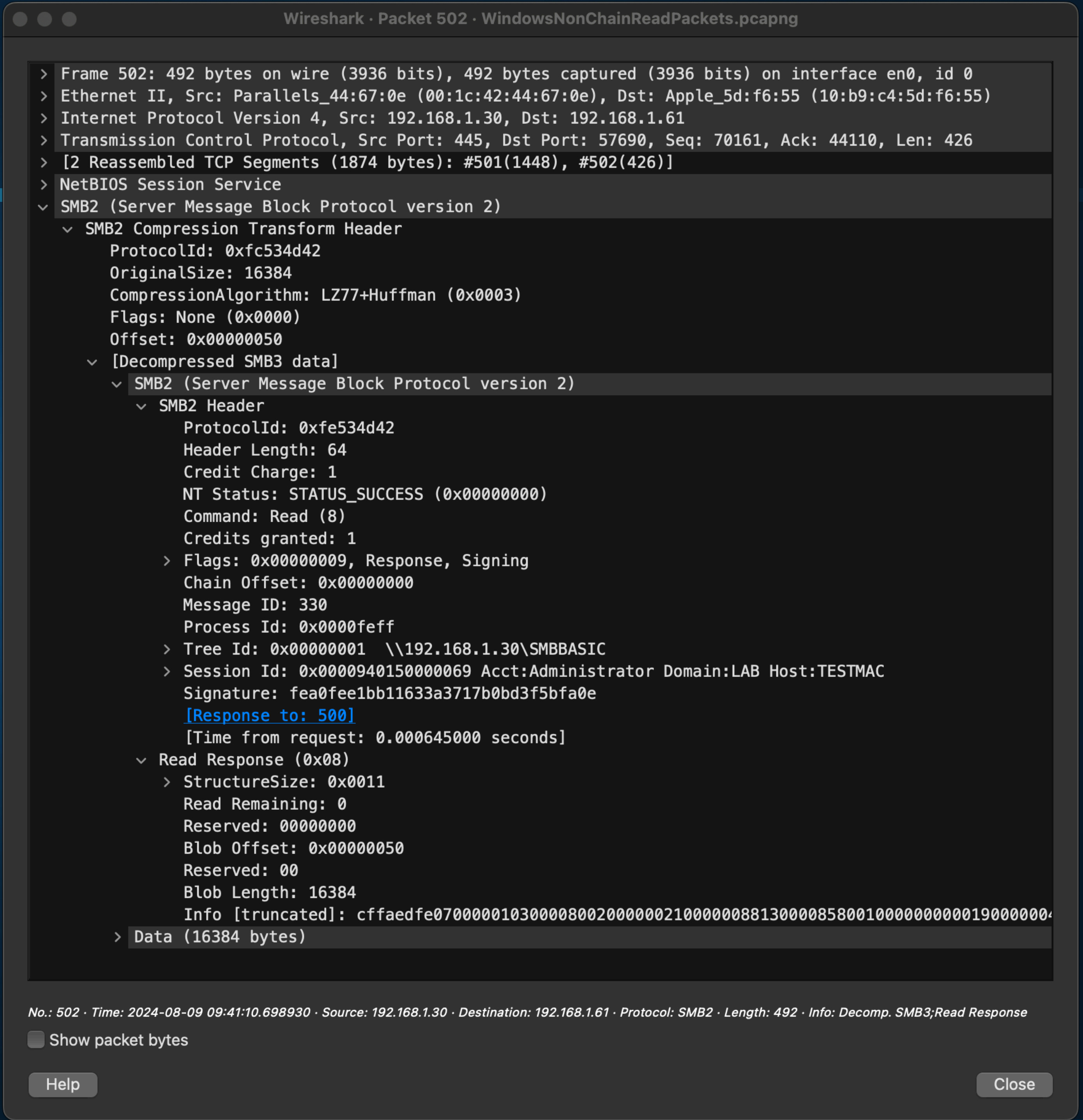
- Negotiate Context SMB2_COMPRESSION_CAPABILITIES
- Supported Algorithms
  - LZ77 + Huffman
    - Compresses data to smallest size but slowest compression speed
  - LZ77
    - Compresses data by a moderate amount but fastest compression speed
  - LZNT1
    - Compresses data the least but with moderate compression speed
  - Pattern_V1
    - Only for chained compression and handles repeating pattern at beginning and end of data
- Chained and non chained compression are supported
- Compression on macOS SMB client is OFF by default
- Only reads and writes of file data are considered for compression

©2024 SNIA. © Apple. All Rights Reserved.

Non Chained Compression

- Only one algorithm is chosen in Negotiate exchange
  - Pattern_V1 is never used
- Offset indicates how much uncompressed data is in packet
  - Windows leaves the SMB header and read/write structures uncompressed and just compress the data
  - macOS SMB Client follows this same behavior

Nonchained Example



```
> Frame 502: 492 bytes on wire (3936 bits), 492 bytes captured (3936 bits) on interface en0, id 0
> Ethernet II, Src: Parallels_44:67:0e (00:1c:42:44:67:0e), Dst: Apple_5d:f6:55 (10:b9:c4:5d:f6:55)
> Internet Protocol Version 4, Src: 192.168.1.30, Dst: 192.168.1.61
> Transmission Control Protocol, Src Port: 445, Dst Port: 57690, Seq: 70161, Ack: 44110, Len: 426
> [2 Reassembled TCP Segments (1874 bytes): #501(1448), #502(426)]
> NetBIOS Session Service
v SMB2 (Server Message Block Protocol version 2)
    v SMB2 Compression Transform Header
        ProtocolId: 0xfc534d42
        OriginalSize: 16384
        CompressionAlgorithm: LZ77+Huffman (0x0003)
        Flags: None (0x0000)
        Offset: 0x00000050
      v [Decompressed SMB3 data]
        v SMB2 (Server Message Block Protocol version 2)
            v SMB2 Header
                ProtocolId: 0xfe534d42
                Header Length: 64
                Credit Charge: 1
                NT Status: STATUS_SUCCESS (0x00000000)
                Command: Read (8)
                Credits granted: 1
              > Flags: 0x00000009, Response, Signing
                Chain Offset: 0x00000000
                Message ID: 330
                Process Id: 0x0000feff
              > Tree Id: 0x00000001  \\192.168.1.30\SMBBASIC
              > Session Id: 0x0000940150000069 Acct:Administrator Domain:LAB Host:TESTMAC
                Signature: fea0fee1bb11633a3717b0bd3f5bfa0e
                [Response to: 500]
                [Time from request: 0.000645000 seconds]
            v Read Response (0x08)
              > StructureSize: 0x0011
                Read Remaining: 0
                Reserved: 00000000
                Blob Offset: 0x00000050
                Reserved: 00
                Blob Length: 16384
                Info [truncated]: cffaedfe070000010300008002000000210000008813000085800100000000001900000004
        > Data (16384 bytes)
```

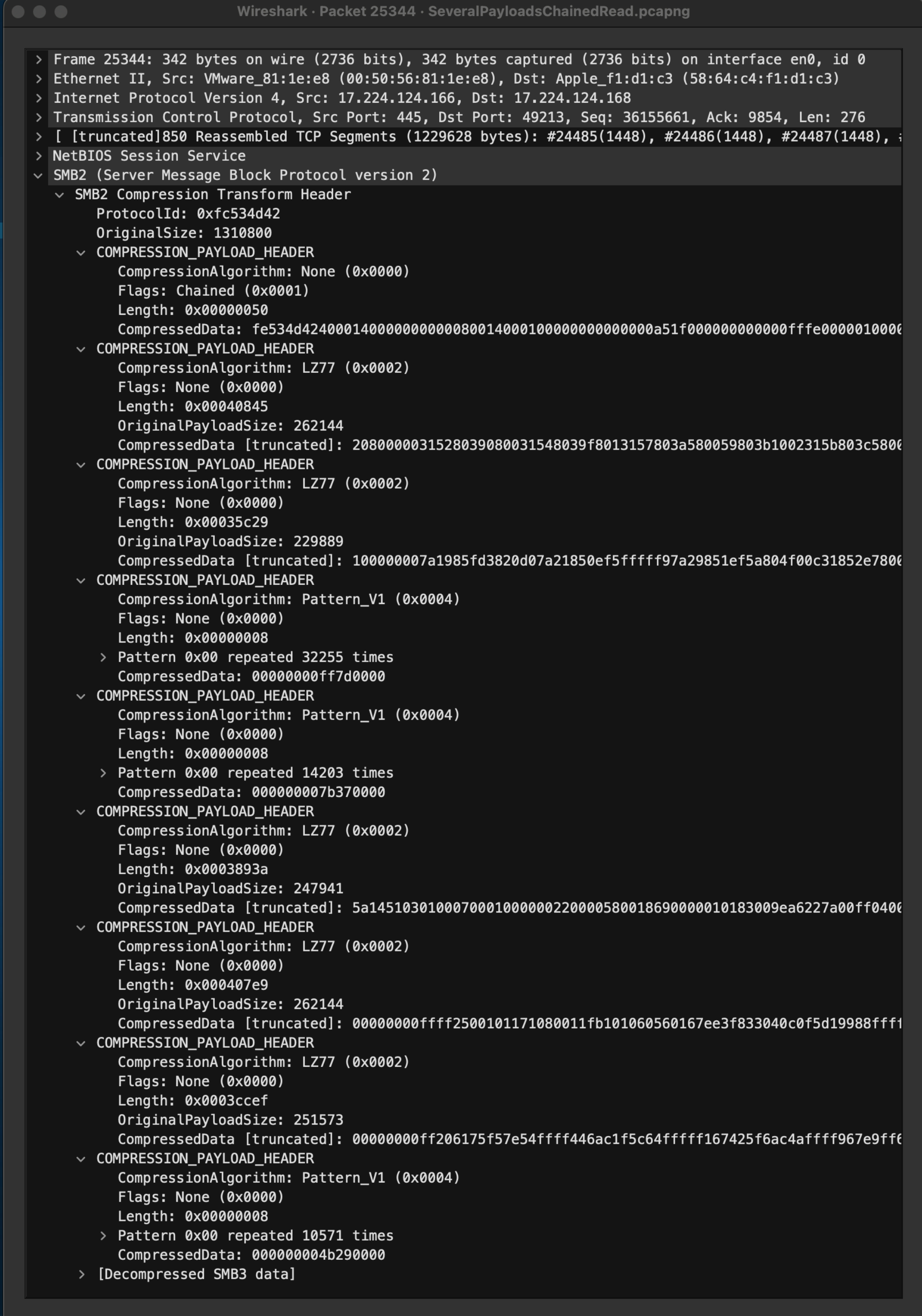
No.: 502 · Time: 2024-08-09 09:41:10.698930 · Source: 192.168.1.30 · Destination: 192.168.1.61 · Protocol: SMB2 · Length: 492 · Info: Decomp. SMB3;Read Response

©2024 SNIA. © Apple. All Rights Reserved.

Chained Compression

- One algorithm is chosen along with PATTERN_V1 in Negotiate exchange
- Windows adds the SMB header and read/write structures as the first payload with CompressedNone
  - macOS SMB Client follows this same behavior
- Remaining data is processed in “chunks”
  - Check for ForwardDataPattern at beginning and if found, add PATTERN_V1 payload
  - Check for BackwardDataPattern at end and if found, save for later
  - Compress remaining data with algorithm and add algorithmic payload
  - If BackwardDataPattern was found, add PATTERN_V1 payload
  - Repeat process if more uncompressed data is left
- One chain may be built from several processed chunks

Chained Example



Wireshark · Packet 25344 · SeveralPayloadsChainedRead.pcapng

```
> Frame 25344: 342 bytes on wire (2736 bits), 342 bytes captured (2736 bits) on interface en0, id 0
> Ethernet II, Src: VMware_81:1e:e8 (00:50:56:81:1e:e8), Dst: Apple_f1:d1:c3 (58:64:c4:f1:d1:c3)
> Internet Protocol Version 4, Src: 17.224.124.166, Dst: 17.224.124.168
> Transmission Control Protocol, Src Port: 445, Dst Port: 49213, Seq: 36155661, Ack: 9854, Len: 276
> [ [truncated]850 Reassembled TCP Segments (1229628 bytes): #24485(1448), #24486(1448), #24487(1448),
> NetBIOS Session Service
v SMB2 (Server Message Block Protocol version 2)
  v SMB2 Compression Transform Header
      ProtocolId: 0xfc534d42
      OriginalSize: 1310800
    v COMPRESSION_PAYLOAD_HEADER
        CompressionAlgorithm: None (0x0000)
        Flags: Chained (0x0001)
        Length: 0x00000050
        CompressedData: fe534d424000140000000000000800140001000000000000000a51f000000000000fffe000001000
    v COMPRESSION_PAYLOAD_HEADER
        CompressionAlgorithm: LZ77 (0x0002)
        Flags: None (0x0000)
        Length: 0x00040845
        OriginalPayloadSize: 262144
        CompressedData [truncated]: 2080000031528039080031548039f8013157803a580059803b1002315b803c580
    v COMPRESSION_PAYLOAD_HEADER
        CompressionAlgorithm: LZ77 (0x0002)
        Flags: None (0x0000)
        Length: 0x00035c29
        OriginalPayloadSize: 229889
        CompressedData [truncated]: 100000007a1985fd3820d07a21850ef5fffff97a29851ef5a804f00c31852e780
    v COMPRESSION_PAYLOAD_HEADER
        CompressionAlgorithm: Pattern_V1 (0x0004)
        Flags: None (0x0000)
        Length: 0x00000008
      > Pattern 0x00 repeated 32255 times
        CompressedData: 00000000ff7d0000
    v COMPRESSION_PAYLOAD_HEADER
        CompressionAlgorithm: Pattern_V1 (0x0004)
        Flags: None (0x0000)
        Length: 0x00000008
      > Pattern 0x00 repeated 14203 times
        CompressedData: 000000007b370000
    v COMPRESSION_PAYLOAD_HEADER
        CompressionAlgorithm: LZ77 (0x0002)
        Flags: None (0x0000)
        Length: 0x0003893a
        OriginalPayloadSize: 247941
        CompressedData [truncated]: 5a14510301000700010000002200005800186900000010183009ea6227a00ff040
    v COMPRESSION_PAYLOAD_HEADER
        CompressionAlgorithm: LZ77 (0x0002)
        Flags: None (0x0000)
        Length: 0x000407e9
        OriginalPayloadSize: 262144
        CompressedData [truncated]: 00000000ffff2500101171080011fb101060560167ee3f833040c0f5d19988fff
    v COMPRESSION_PAYLOAD_HEADER
        CompressionAlgorithm: LZ77 (0x0002)
        Flags: None (0x0000)
        Length: 0x0003ccef
        OriginalPayloadSize: 251573
        CompressedData [truncated]: 00000000ff206175f57e54ffff446ac1f5c64fffff167425f6ac4affff967e9ff
    v COMPRESSION_PAYLOAD_HEADER
        CompressionAlgorithm: Pattern_V1 (0x0004)
        Flags: None (0x0000)
        Length: 0x00000008
      > Pattern 0x00 repeated 10571 times
        CompressedData: 000000004b290000
    > [Decompressed SMB3 data]
```

nsmb.conf Compression Keywords

- comp_algorithm_map – bitmap of algorithms to enable (0 = all disabled)
- comp_chaining_disable – disable chaining compression (no)
- comp_io_threshold – minimum IO size to attempt compression on (4096 KB)
- comp_chunk_len – chained write chunk size for processing (256 KB)
  - Windows uses 256 KB chunks and macOS SMB Client follows this behavior
- comp_max_fail_cnt – max times write compression can “fail” before disabling compression for that file (5)
- comp_exclude_list – comma separated list of file extensions to add to default exclusion list
- comp_include_list – comma separated list of file extensions to override default exclusion list

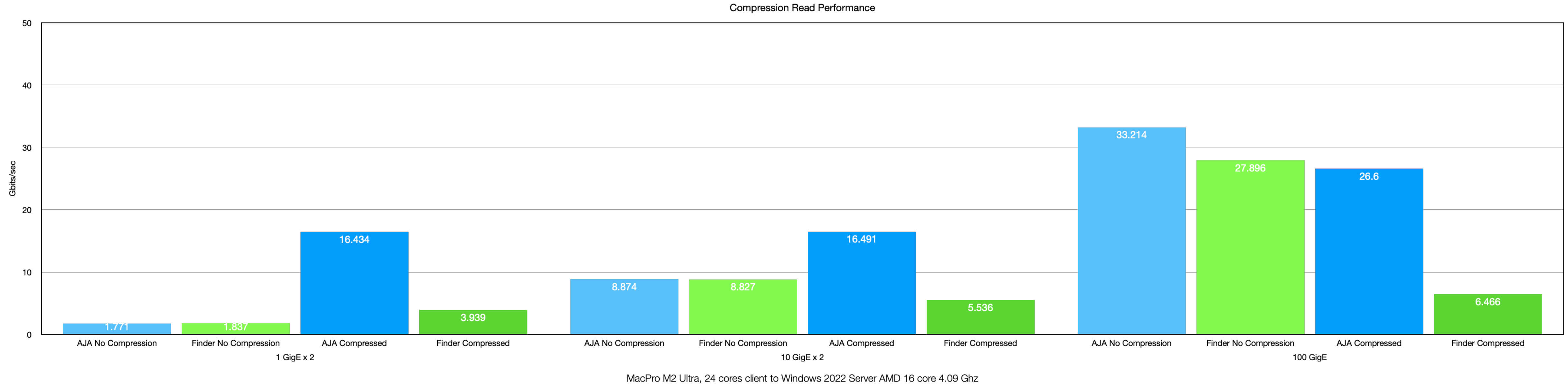
Default Compression Exclusion List

- Compression should not be attempted on files that are already compressed
- File extensions that are expected to not compress well
  - "7z", "aa3", "aac", "aes", "asf", "avchd", "avi", "bik", "bsf", "bz", "bz2", "bzip", "bzip2", "chm", "cpgz", "cr2", "divx", "dng", "docm", "docx", "dotm", "dotx", "emz", "epub", "f4v", "flv", "gif", "gpg", "graffle", "gz", "gzip", "hdmov", "heic", "heif", "hxs", "j2c", "jar", "jpeg", "jpg", "lzma", "m4a", "m4a", "m4v", "mint", "mkv", "mov", "mp2", "mp3", "mp4", "mpa", "mpe", "mpeg", "mpg", "mpq", "mshc", "msi", "mts", "nef", "odp", "ods", "odt", "opus", "otp", "ots", "ott", "pack", "pages", "png", "pptm", "pptx", "pspimage", "qt", "ra", "rar", "rpm", "sea", "sit", "tgz", "tif", "tiff", "vob", "war", "wav", "webarchive", "webm", "webp", "wma", "wmv", "wtv", "wv", "xlsb", "xlsm", "xlsx", "xps", "xz", "zip", "zstd”

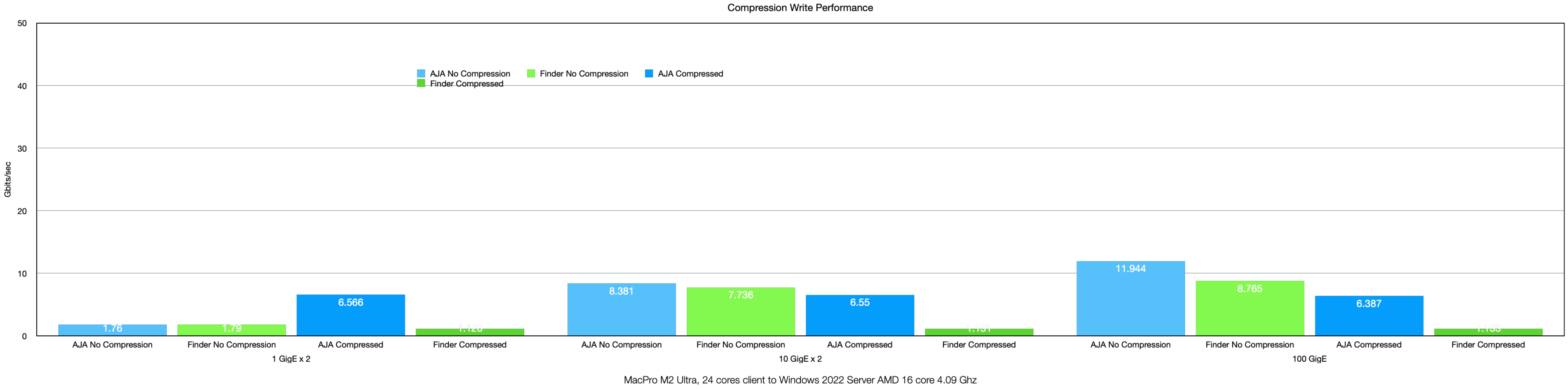
Compression Performance

- Benefits of compression is situational as performance is data dependent
  - Highly compressible data results in much better performance
  - Low compressible data ends up with compression disabled so about same performance as non compressed transfers
- Slower networks benefit more than fast networks
  - If network is fast, then compression just adds more delays
- Chained and non chained have about equivalent performance
  - Chained could be faster if data had more repeatable patterns in it due to PATTERN_V1 support
- Compressed traffic can use much less network bandwidth
- Setup:
  - Chained, LZ77H, two 1 gigE NICs, two 10 gigE NICS, one 100 gigE NIC
  - Sequoia 1.50 using drivers that come with OS (DriverKit)

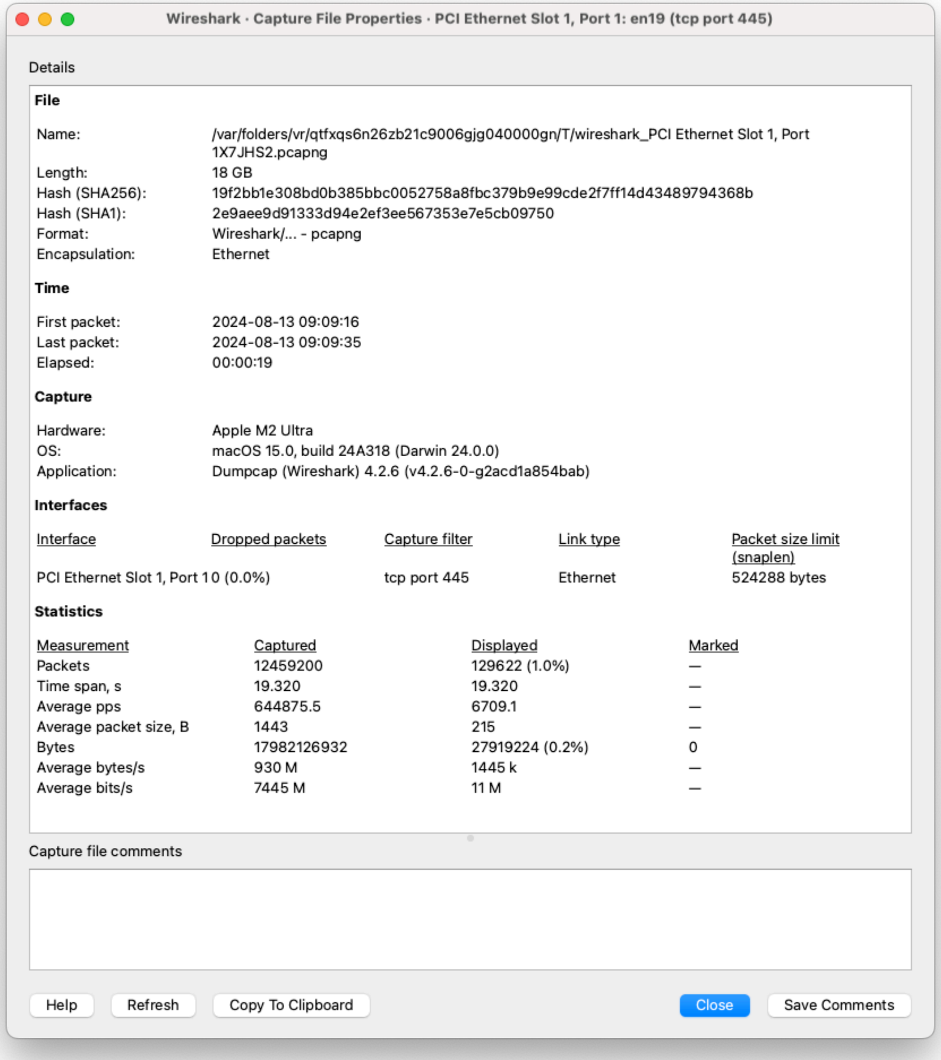
Read Compression – 1 GigE, 10 gigE, 100 gigE



Write Compression – 1 GigE, 10 gigE, 100 gigE



Bandwidth – AJA System Test Lite



Wireshark · Capture File Properties · PCI Ethernet Slot 1, Port 1: en19 (tcp port 445)

Details

File

Name: /var/folders/vr/qtfxqs6n26zb21c9006gjg040000gn/T/wireshark_PCI Ethernet Slot 1, Port 1X7JHS2.pcapng
Length: 18 GB
Hash (SHA256): 19f2bb1e308bd0b385bbc0052758a8fbc379b9e99cde2f7ff14d43489794368b
Hash (SHA1): 2e9aee9d91333d94e2ef3ee567353e7e5cb09750
Format: Wireshark/... - pcapng
Encapsulation: Ethernet

Time

First packet: 2024-08-13 09:09:16
Last packet: 2024-08-13 09:09:35
Elapsed: 00:00:19

Capture

Hardware: Apple M2 Ultra
OS: macOS 15.0, build 24A318 (Darwin 24.0.0)
Application: Dumpcap (Wireshark) 4.2.6 (v4.2.6-0-g2acd1a854bab)

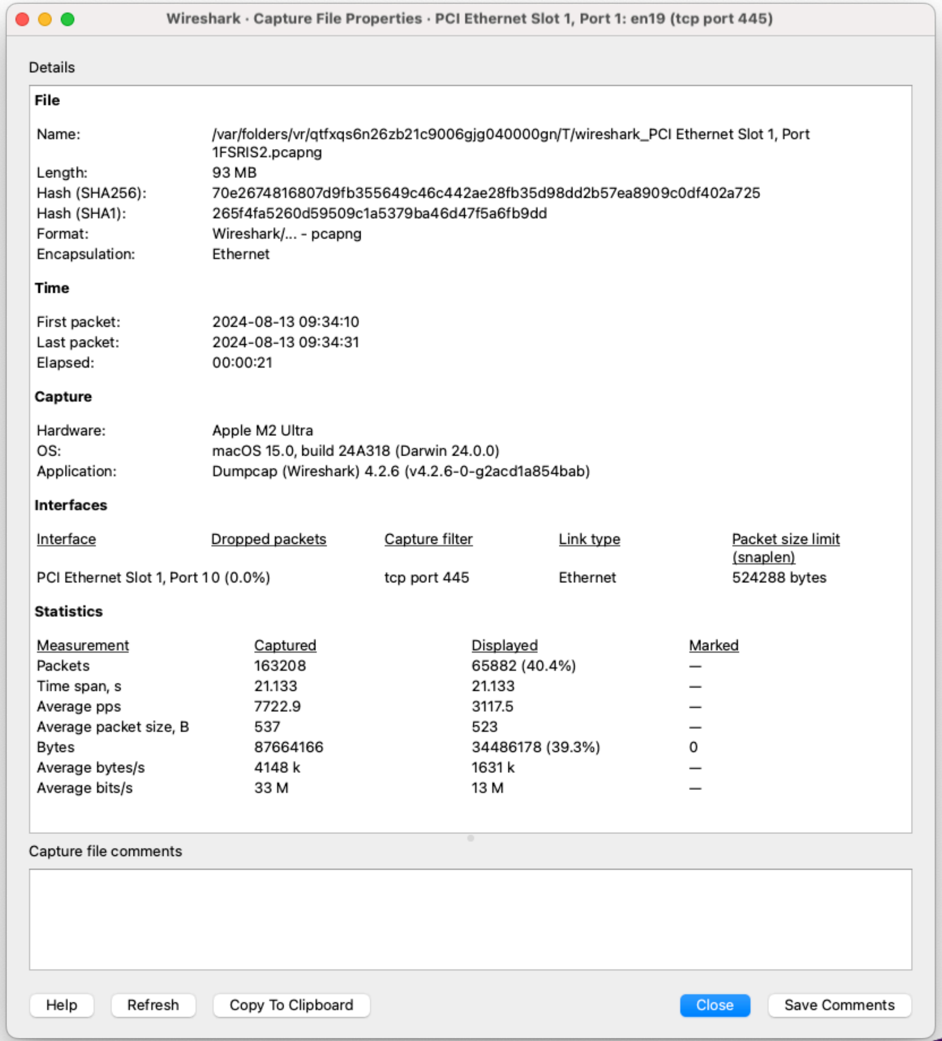
Interfaces

Interface	Dropped packets	Capture filter	Link type	Packet size limit (snaplen)
PCI Ethernet Slot 1, Port 1	0 (0.0%)	tcp port 445	Ethernet	524288 bytes

Statistics

Measurement	Captured	Displayed	Marked
Packets	12459200	129622 (1.0%)	—
Time span, s	19.320	19.320	—
Average pps	644875.5	6709.1	—
Average packet size, B	1443	215	—
Bytes	17982126932	27919224 (0.2%)	0
Average bytes/s	930 M	1445 k	—
Average bits/s	7445 M	11 M	—

Capture file comments



Wireshark · Capture File Properties · PCI Ethernet Slot 1, Port 1: en19 (tcp port 445)

Details

File

Name: /var/folders/vr/qtfxqs6n26zb21c9006gjg040000gn/T/wireshark_PCI Ethernet Slot 1, Port 1FSRIS2.pcapng
Length: 93 MB
Hash (SHA256): 70e2674816807d9fb355649c46c442ae28fb35d98dd2b57ea8909c0df402a725
Hash (SHA1): 265f4fa5260d59509c1a5379ba46d47f5a6fb9dd
Format: Wireshark/... - pcapng
Encapsulation: Ethernet

Time

First packet: 2024-08-13 09:34:10
Last packet: 2024-08-13 09:34:31
Elapsed: 00:00:21

Capture

Hardware: Apple M2 Ultra
OS: macOS 15.0, build 24A318 (Darwin 24.0.0)
Application: Dumpcap (Wireshark) 4.2.6 (v4.2.6-0-g2acd1a854bab)

Interfaces

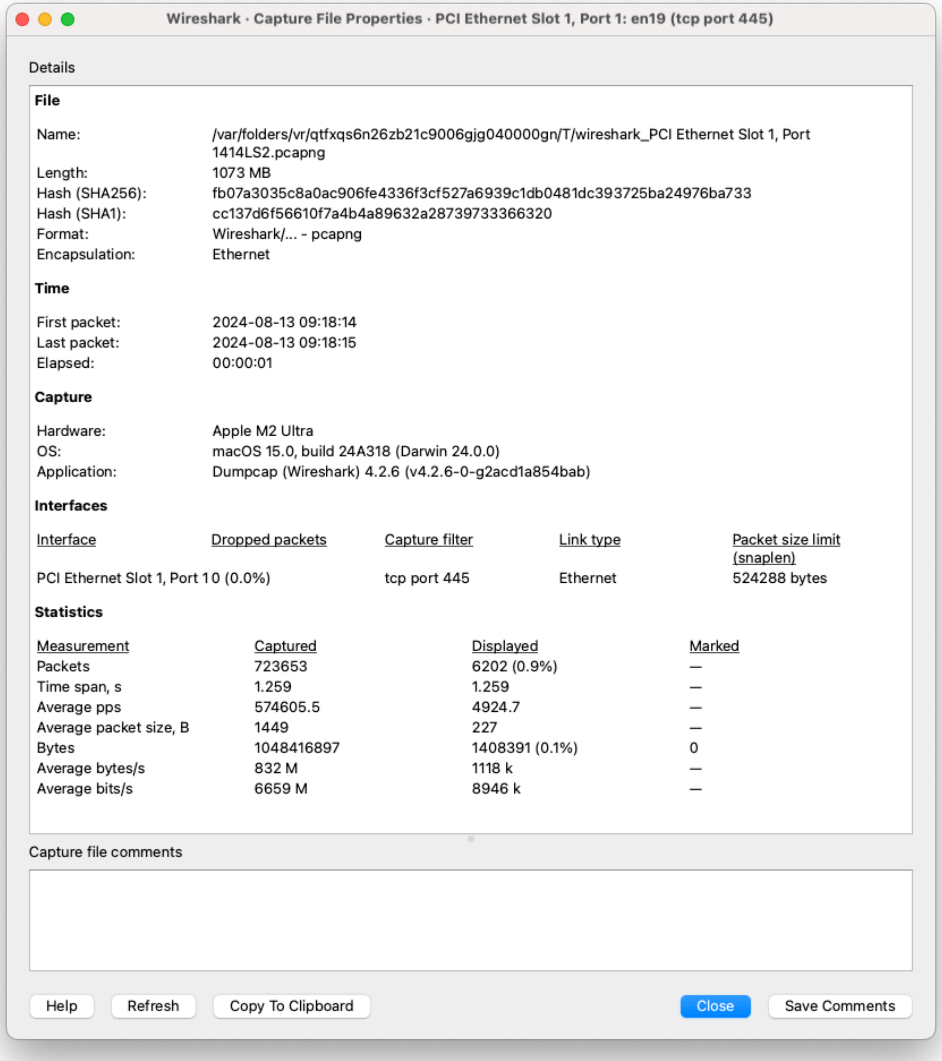
Interface	Dropped packets	Capture filter	Link type	Packet size limit (snaplen)
PCI Ethernet Slot 1, Port 1	0 (0.0%)	tcp port 445	Ethernet	524288 bytes

Statistics

Measurement	Captured	Displayed	Marked
Packets	163208	65882 (40.4%)	—
Time span, s	21.133	21.133	—
Average pps	7722.9	3117.5	—
Average packet size, B	537	523	—
Bytes	87664166	34486178 (39.3%)	0
Average bytes/s	4148 k	1631 k	—
Average bits/s	33 M	13 M	—

Capture file comments

Bandwidth – DSCopy enwik9



Wireshark · Capture File Properties · PCI Ethernet Slot 1, Port 1: en19 (tcp port 445)

Details

File

Name:	/var/folders/vr/qtfxqs6n26zb21c9006gjg040000gn/T/wireshark_PCI Ethernet Slot 1, Port 1414LS2.pcapng
Length:	1073 MB
Hash (SHA256):	fb07a3035c8a0ac906fe4336f3cf527a6939c1db0481dc393725ba24976ba733
Hash (SHA1):	cc137d6f56610f7a4b4a89632a28739733366320
Format:	Wireshark/... - pcapng
Encapsulation:	Ethernet

Time

First packet:	2024-08-13 09:18:14
Last packet:	2024-08-13 09:18:15
Elapsed:	00:00:01

Capture

Hardware:	Apple M2 Ultra
OS:	macOS 15.0, build 24A318 (Darwin 24.0.0)
Application:	Dumpcap (Wireshark) 4.2.6 (v4.2.6-0-g2acd1a854bab)

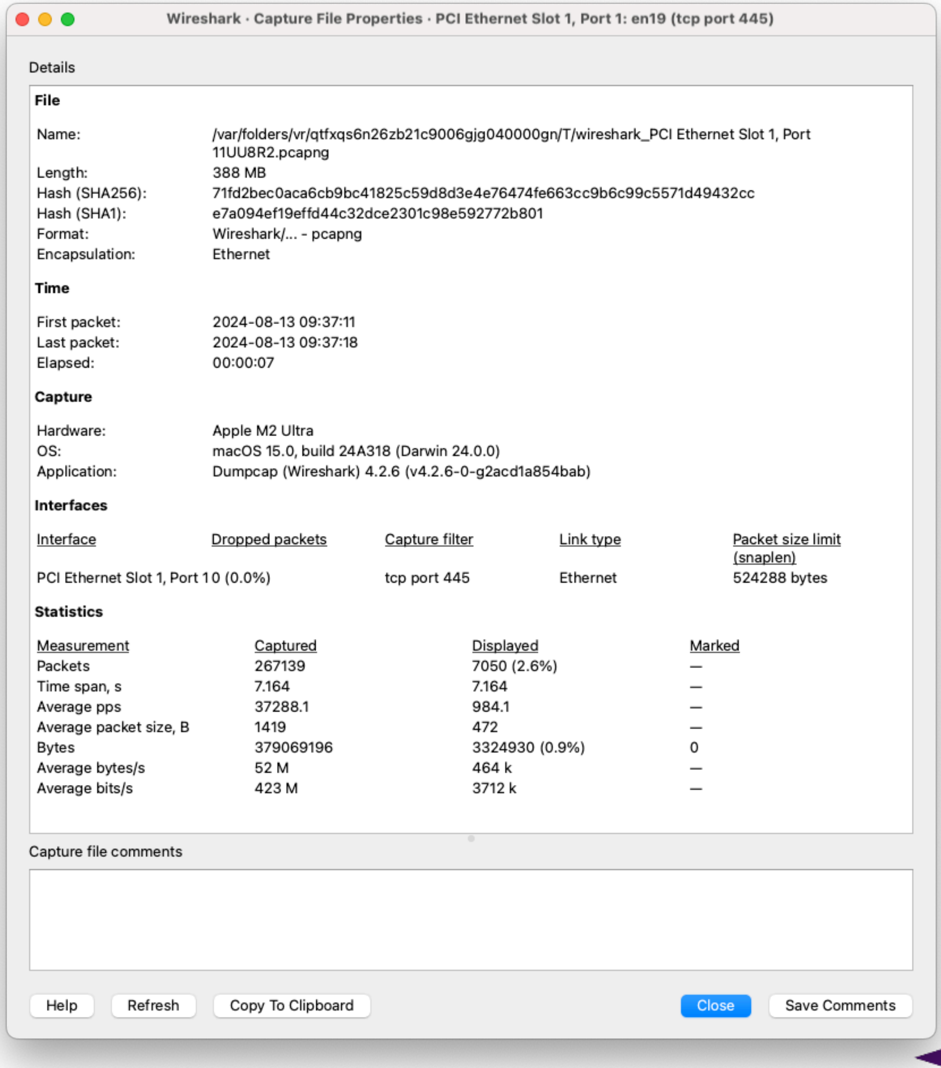
Interfaces

Interface	Dropped packets	Capture filter	Link type	Packet size limit (snaplen)
PCI Ethernet Slot 1, Port 1	0 (0.0%)	tcp port 445	Ethernet	524288 bytes

Statistics

Measurement	Captured	Displayed	Marked
Packets	723653	6202 (0.9%)	—
Time span, s	1.259	1.259	—
Average pps	574605.5	4924.7	—
Average packet size, B	1449	227	—
Bytes	1048416897	1408391 (0.1%)	0
Average bytes/s	832 M	1118 k	—
Average bits/s	6659 M	8946 k	—

Capture file comments



Wireshark · Capture File Properties · PCI Ethernet Slot 1, Port 1: en19 (tcp port 445)

Details

File

Name:	/var/folders/vr/qtfxqs6n26zb21c9006gjg040000gn/T/wireshark_PCI Ethernet Slot 1, Port 11UU8R2.pcapng
Length:	388 MB
Hash (SHA256):	71fd2bec0aca6cb9bc41825c59d8d3e4e76474fe663cc9b6c99c5571d49432cc
Hash (SHA1):	e7a094ef19effd44c32dce2301c98e592772b801
Format:	Wireshark/... - pcapng
Encapsulation:	Ethernet

Time

First packet:	2024-08-13 09:37:11
Last packet:	2024-08-13 09:37:18
Elapsed:	00:00:07

Capture

Hardware:	Apple M2 Ultra
OS:	macOS 15.0, build 24A318 (Darwin 24.0.0)
Application:	Dumpcap (Wireshark) 4.2.6 (v4.2.6-0-g2acd1a854bab)

Interfaces

Interface	Dropped packets	Capture filter	Link type	Packet size limit (snaplen)
PCI Ethernet Slot 1, Port 1	0 (0.0%)	tcp port 445	Ethernet	524288 bytes

Statistics

Measurement	Captured	Displayed	Marked
Packets	267139	7050 (2.6%)	—
Time span, s	7.164	7.164	—
Average pps	37288.1	984.1	—
Average packet size, B	1419	472	—
Bytes	379069196	3324930 (0.9%)	0
Average bytes/s	52 M	464 k	—
Average bits/s	423 M	3712 k	—

Capture file comments

Questions?

©2024 SNIA. © Apple. All Rights Reserved.

Please take a moment to rate this session.

Your feedback is important to us.