Accelerating SMB2

Mark Rabinovich
Visuality Systems Ltd.

Plan

- Introduction
- WAN issues
- Overview of acceleration methods
- Acceleration challenges
- LAN acceleration
- Q & A

Introduction

Who I am

- Speaker
  - Mark Rabinovich – R&D Manager
- Company
  - Visuality Systems Ltd.
  - NQ - Embedded/Mobile CIFS solution
  - CAX – CIFS Accelerator

What is this session about

This session continues "CIFS Acceleration" theme on SDC as first discussed in 2009.

Currently CAX accelerates SMB traffic.

SMB2 acceleration is under development.

I am going to share our SMB2 acceleration experience.

WAN Issues

CIFS over WAN

When CIFS runs over WAN?

- From a branch office:
  - connected through a satellite link.
  - connected through an overseas link.
- VPN connection:
  - an employee connected from home.

WAN Numbers

Typical WAN latencies:

- Satellite networks – up to one second RTT.
- Overseas networks – 250 milliseconds.

Typical WAN bandwidth:

- Satellite – 1Mbit/sec.
- Oversea networks – 1 Mbit/sec and higher.
- Bandwidth is limited by Internet providers on service overload.

Sensitive Issues

WAN-sensitive issues are:

- Multiplexing. Higher multiplexing is less affected by high latencies.
- Redundancy. Adds more round-trips.
- Limited bandwidth. Effectively limits multiplexing. Latencies again.

The following slides analyze how is this applicable to SMB2.

Multiplexing: Reads & Writes

Known SMB2 implementations are optimized for sequential reads/writes:

```
SMB2    GetInfo Request FS_INFO/SMB2_FS_INFO_03 File: ElizabethTaylor.bmp
SMB2    GetInfo Response
SMB2    Read Request Len:65536 Off:0 File: ElizabethTaylor.bmp
SMB2    Read Request Len:65536 Off:65536 File: ElizabethTaylor.bmp
SMB2    Read Request Len:65536 Off:131072 File: ElizabethTaylor.bmp
SMB2    Read Request Len:65536 Off:196608 File: ElizabethTaylor.bmp
SMB2    Read Request Len:65536 Off:262144 File: ElizabethTaylor.bmp
SMB2    Read Request Len:65536 Off:327680 File: ElizabethTaylor.bmp
SMB2    Read Request Len:65536 Off:393216 File: ElizabethTaylor.bmp
SMB2    Read Request Len:65536 Off:458752 File: ElizabethTaylor.bmp
SMB2    Read Request Len:65536 Off:524288 File: ElizabethTaylor.bmp
SMB2    Read Request Len:65536 Off:589824 File: ElizabethTaylor.bmp
SMB2    Read Request Len:65536 Off:655360 File: ElizabethTaylor.bmp
SMB2    Read Request Len:65536 Off:720896 File: ElizabethTaylor.bmp
SMB2    Read Request Len:65536 Off:786432 File: ElizabethTaylor.bmp
SMB2    Read Request Len:65536 Off:851968 File: ElizabethTaylor.bmp
SMB2    Read Request Len:65536 Off:917504 File: ElizabethTaylor.bmp
SMB2    Read Request Len:63990 Off:983040 File: ElizabethTaylor.bmp
SMB2    Read Response
SMB2    Read Response
SMB2    Read Response
SMB2    Read Response
SMB2    Read Response
SMB2    Read Response
SMB2    Read Response
SMB2    Read Response
SMB2    Read Response
SMB2    Read Response
SMB2    Read Response
SMB2    Read Response
SMB2    Read Response
SMB2    Read Response
SMB2    Read Response
SMB2    Read Response
```

16 outstanding requests

This is a capture fragment of a BMP image showing up over the wire.

*All 16 reads cost about **one RTT**.*

Multiplexing: Reads & Writes

An application does not transparently benefit from this. It should at least implement large buffers.

```
SMB2    Read Request Len:65536 Off:196608 File: 4.14_BE\4.14_BE.pcap
SMB2    Read Response
SMB2    Read Request Len:65536 Off:262144 File: 4.14_BE\4.14_BE.pcap
SMB2    Read Response
SMB2    Read Request Len:65536 Off:327680 File: 4.14_BE\4.14
SMB2    Read Response
SMB2    Read Request Len:65536 Off:393216 File: 4.14_BE\4.14_BE.pcap
SMB2    Read Response
SMB2    Read Request Len:65536 Off:458752 File: 4.14_BE\4.14_BE.pcap
SMB2    Read Response
SMB2    Read Request Len:65536 Off:524288 File: 4.14_BE\4.14_BE.pcap
SMB2    Read Response
SMB2    Read Request Len:65536 Off:589824 File: 4.14_BE\4.14_BE.pcap
SMB2    Read Response
SMB2    Read Request Len:65536 Off:655360 File: 4.14_BE\4.14_BE.pcap
SMB2    Read Response
SMB2    Read Request Len:65536 Off:720896 File: 4.14_BE\4.14_BE.pcap
SMB2    Read Response
SMB2    Read Request Len:65536 Off:786432 File: 4.14_BE\4.14_BE.pcap
SMB2    Read Response
SMB2    Read Request Len:65536 Off:851968 File: 4.14_BE\4.14_BE.pcap
```

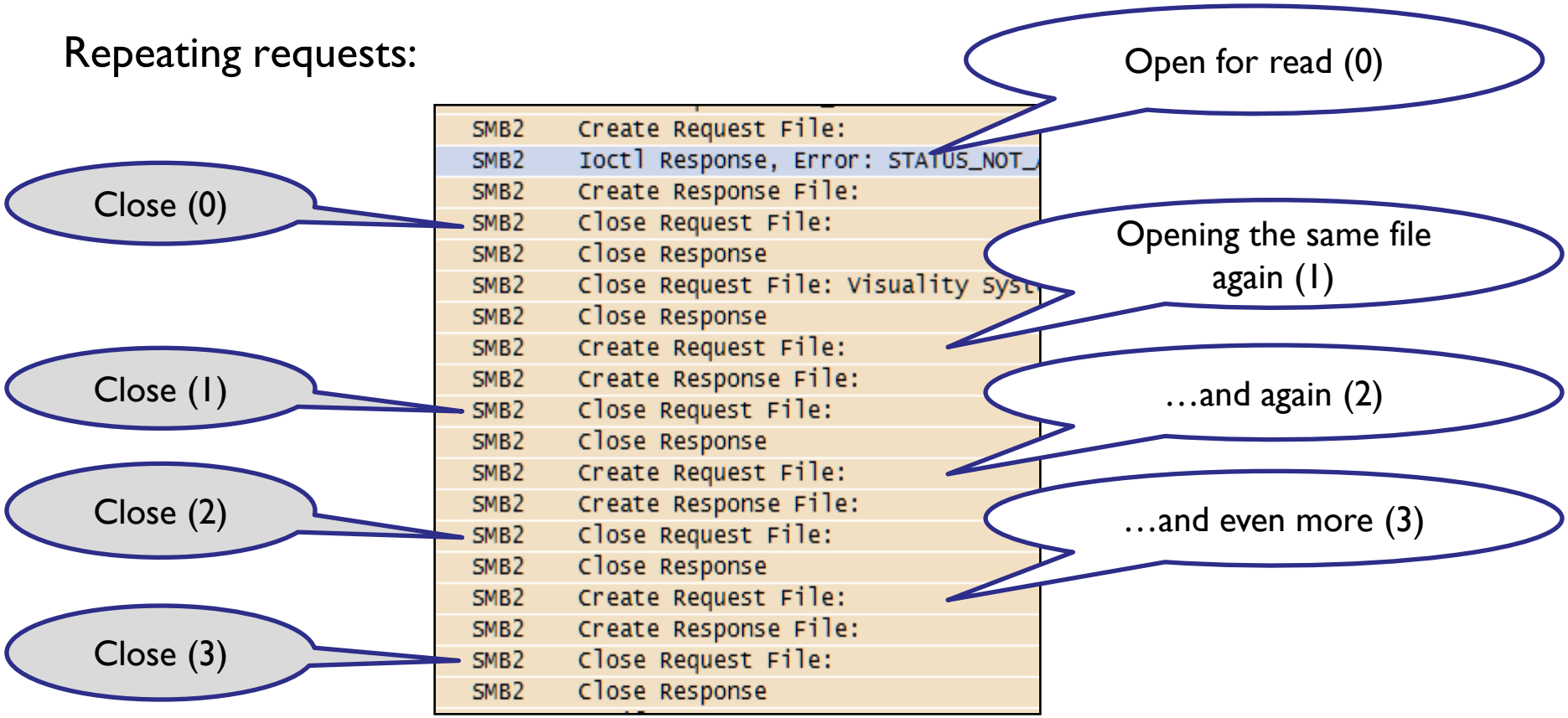
Synchronous reads

This is a capture fragment of an opening of a PCAP file over the wire.

*Each Read costs **one RTT**.*

Redundancy

Repeating requests:



*This is a capture fragment of an opening of MS Word Document over the wire. Four Create(s) are identical. Each Create + Close sequence apparently checks file existence. Each Create + Close pair costs **two RTTs**.*

Redundancy

Multiple requests for the same information:

```
SMB2    GetInfo Request FILE_INFO/SMB2_FILE_STANDARD_INFO File: srvsvc
SMB2    GetInfo Response
DCERPC  Bind: call_id: 2, 2 context items, 1st SRVSVC V3.0
SMB2    Write Response
SMB2    Read Request Len:1024 Off:0 File: srvsvc
DCERPC  Bind_ack: call_id: 2 Unknown result (3), reason: Local limit exce
SRVSVC  NetShareGetInfo request
SRVSVC  NetShareGetInfo response
SMB2    Close Request File: srvsvc
SMB2    Close Response
SMB2    Create Request File: wkssvc
SMB2    Create Response File: wkssvc
SMB2    GetInfo Request FILE_INFO/SMB2_FILE_STANDARD_INFO File: wkssvc
SMB2    GetInfo Response
DCERPC  Bind: call_id: 2, 2 context items, 1st WKSSVC V1.0
SMB2    Write Response
SMB2    Read Request Len:1024 Off:0 File: wkssvc
DCERPC  Bind_ack: call_id: 2 Unknown result (3), reason: Local limit exce
WKSSVC  NetWkstaGetInfo request Level:100
WKSSVC  NetWkstaGetInfo response
SMB2    Close Request File: wkssvc
SMB2    Close Response
SMB2    Create Request File: srvsvc
SMB2    Create Response File: srvsvc
SMB2    GetInfo Request FILE_INFO/SMB2_FILE_STANDARD_INFO File: srvsvc
SMB2    GetInfo Response
```

Two requests for the same information. This information did not change between the requests

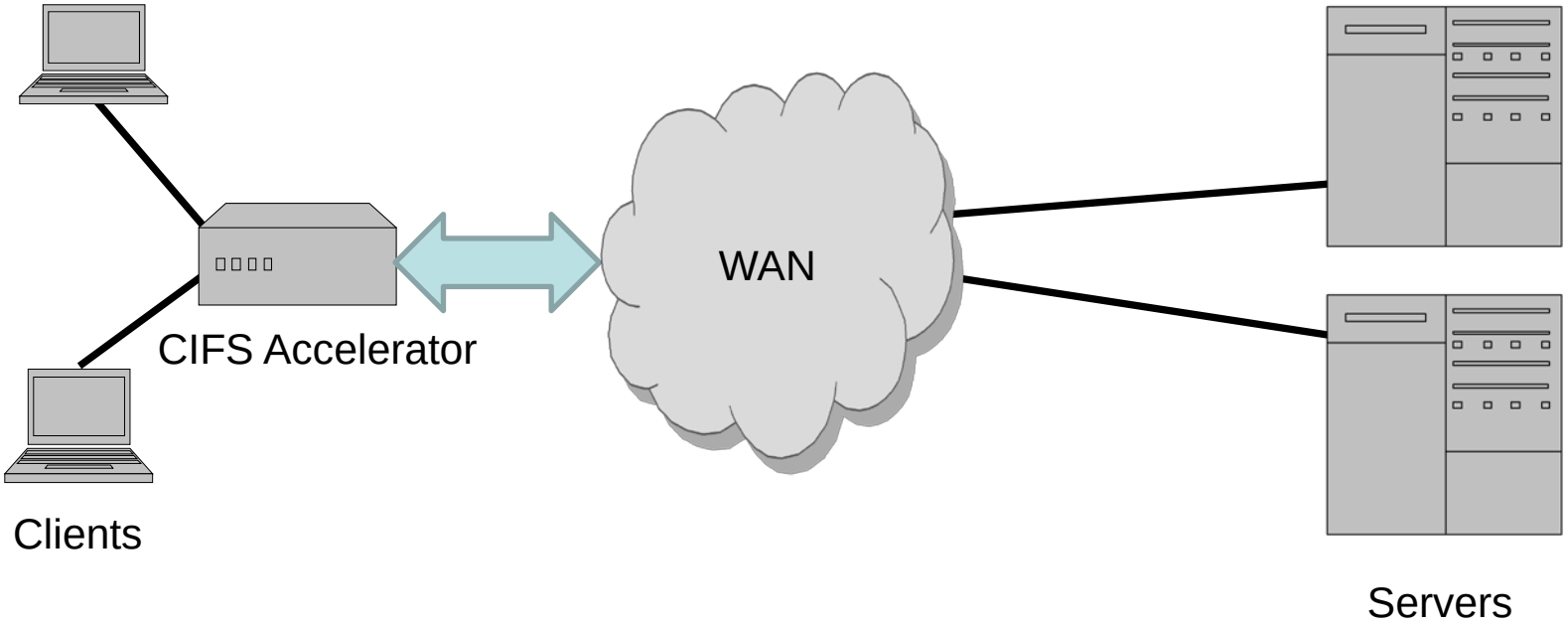
This is a capture fragment of an opening of MS Word Document over the network.

Conclusions

- SMB2 concurrency (multiplexing) is better than SMB(1) yet not always sufficient.
- SMB2 is redundant.

Acceleration Methods

Model



- CIFS Accelerator behaves as CIFS Proxy.
- Client-side (LAN side) asymmetrical solution.
- We do not consider symmetrical solutions.

Techniques

- **Caching**

  We can respond to repeating client requests without accessing the server.

- **Predicting**

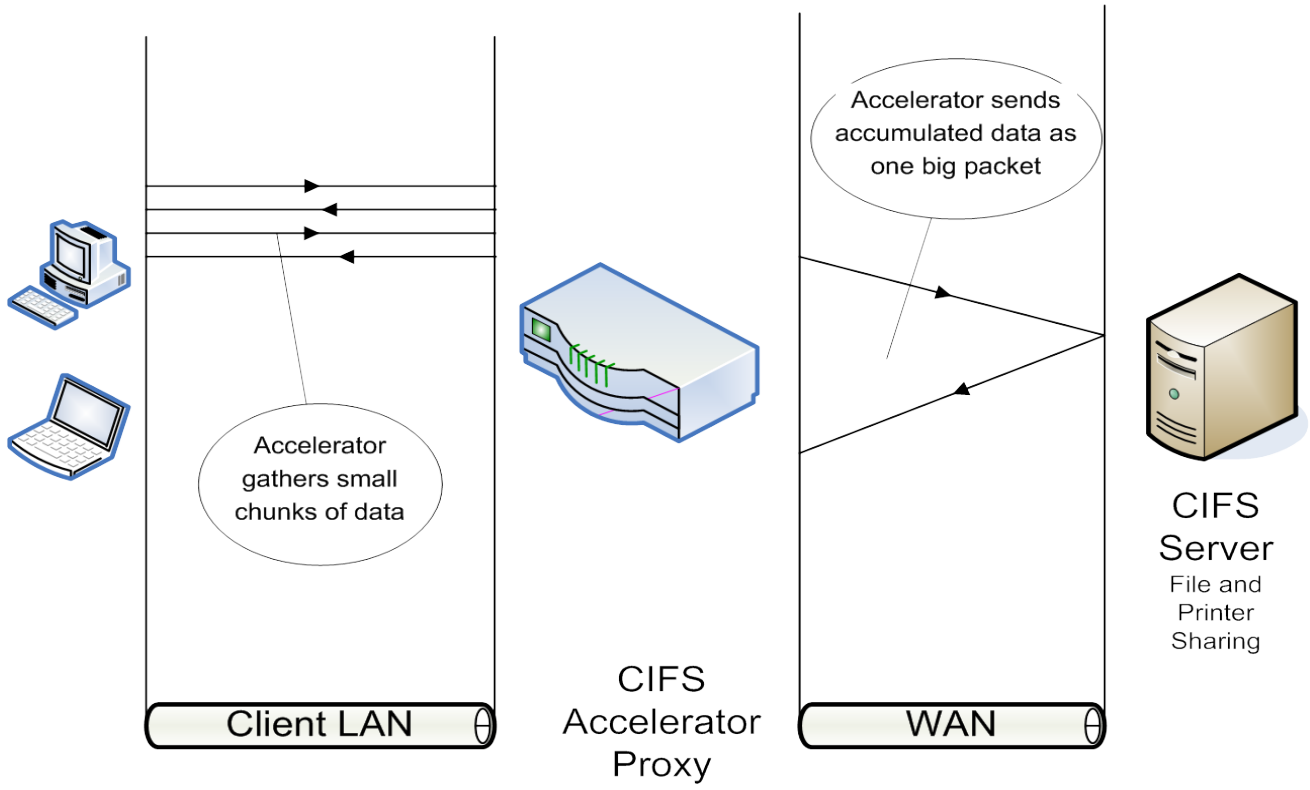
  We can recognize well-known sequences and perform actions in advance.

- **Aggregating**

  We can bring the same information with less requests.

Which of these methods are applicable for SMB2?

Aggregating data



Time without acceleration – two roundtrips,
with acceleration – less then one roundtrip.

Aggregating When?

- ❒ Sequential file read

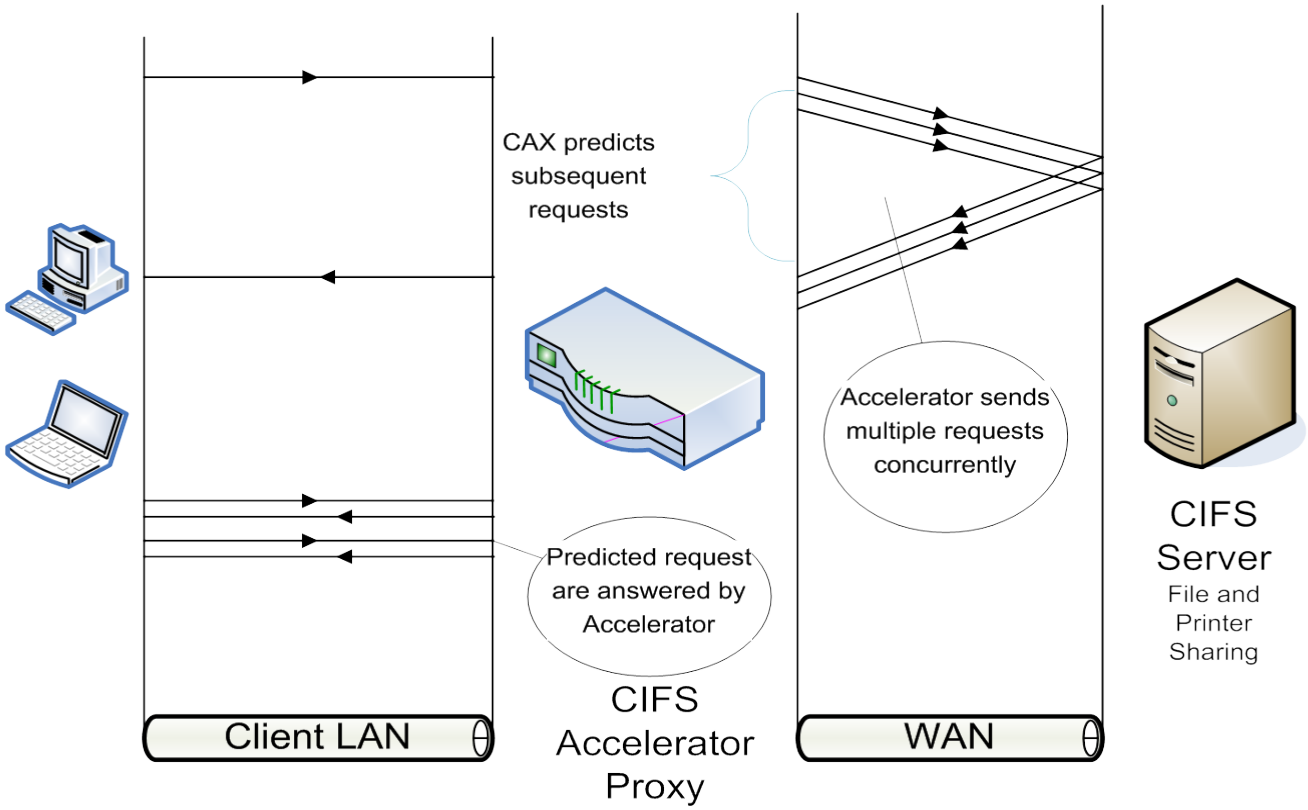
  Depends on application, not applicable when multiplexing.

- ☒ Always issue queries with the most comprehensive info level

  SMB2 clients are mostly using comprehensive levels.

Legend: ☑ - fully applicable, ☒ - not applicable, ☐ - somehow applicable

Predicting



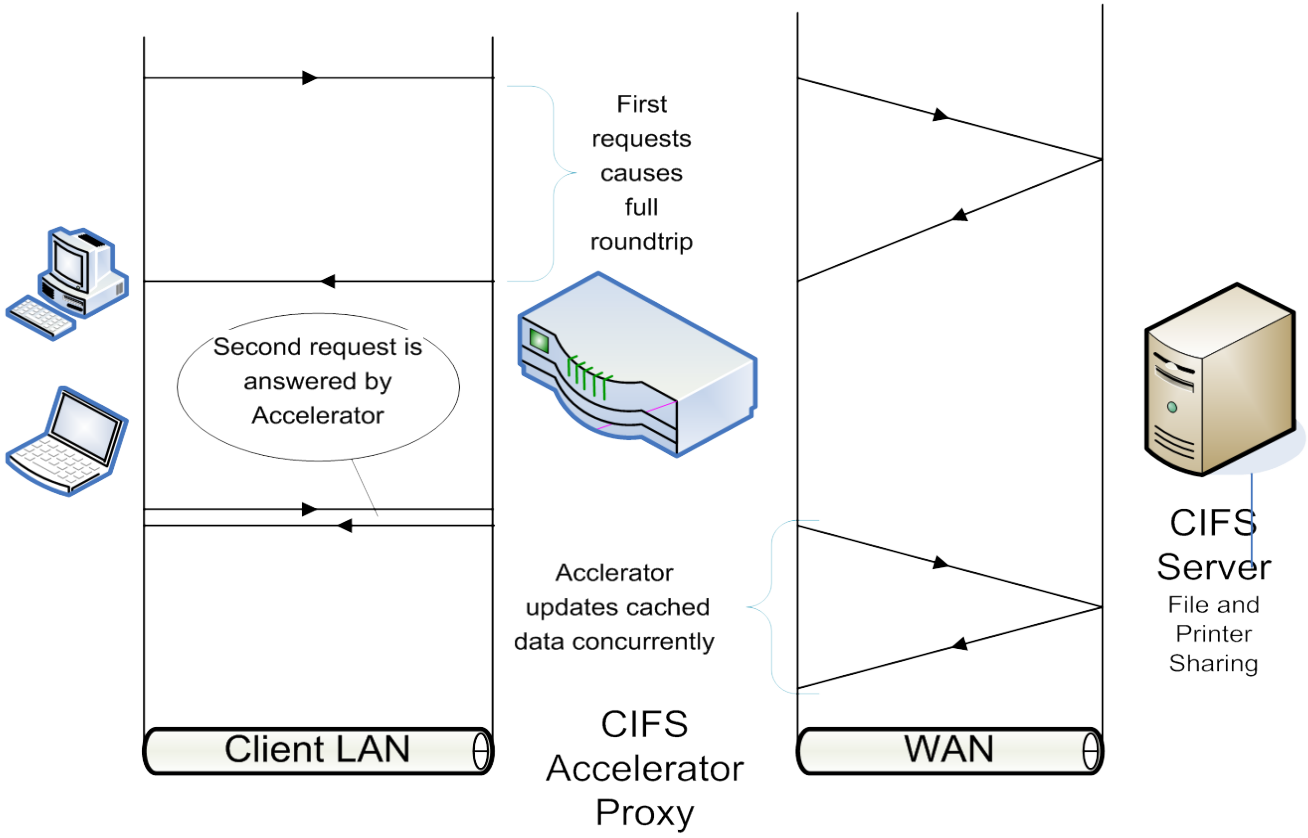
Time without acceleration – three roundtrips, with acceleration – one roundtrip.

Predicting What?

- ☑ Pre-fetching
- ☑ Force requests for complex scenarios
- ☑ Some SMBs (almost) always succeed

Legend: ☑ - fully applicable, ☒ - not applicable, ☐ - somehow applicable

Caching



Time without acceleration – two roundtrips, with acceleration – one roundtrip

Caching What?

- ☑ File Information
- ☑ Share Information
- ☑ Server Information
- ☑ File data
- ☑ File nonexistence
- ☑ Stream-full/stream-less file

Legend: ☑ - fully applicable, ☒ - not applicable, ☐ - somehow applicable

What happens in SMB2?

- Aggregating:
  - SMB2 implementations can chain commands
  - Most (but not all) of applications over SMB2 utilize maximal buffers
  - Aggregating has little effect on SMB2
- Predicting
  - Applications over SMB2 use more multiplexing than when run over SMB(1).
  - Predicting is less effective with SMB2 but still very effective.
- Caching is as effective with SMB2 as with SMB(1).

Acceleration Challenges

Message Sequence

- SMB2 applies strong policy of message sequence handling
- Accelerated WAN sequences differ from LAN sequences
- Accelerator should keep trek of outstanding requests and translate LAN-side message ID into WAN-side message ID and vice versa

Predicting

- One should predict transactions rather than single commands as typical for SMB(1)
- A transaction starts from *Create* and ends with *Close*.
- *Access Mask* in *Create* may be a hint
- Predicted transactions may go wrong way, so that one needs to keep rollback in mind

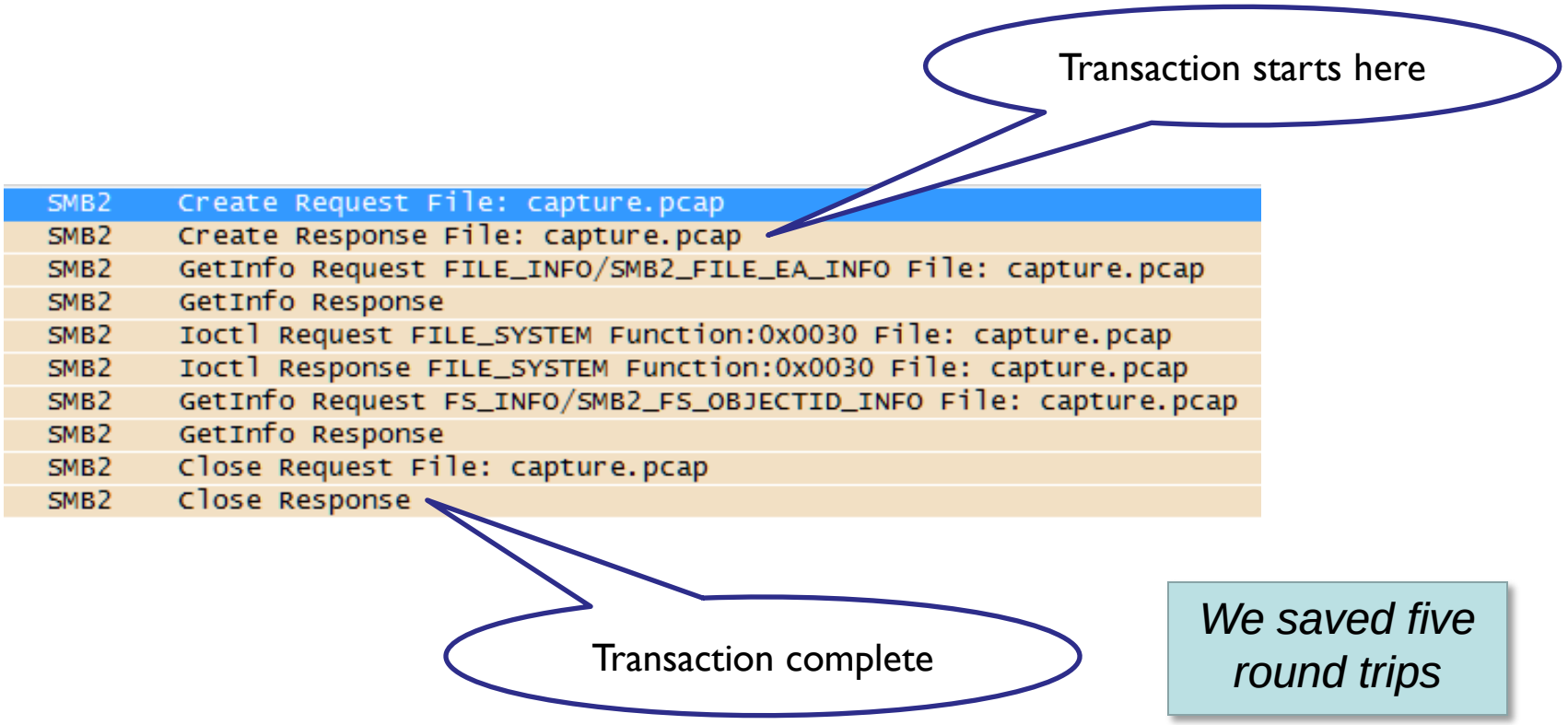
Accelerating Create

- Accelerating SMB2 will never be effective without accelerating Create(s)
- Known Access Mask values may indicate a start of typical transactions:
  - 0x100080 usually starts a QueryInfo/IOCTL transaction
  - 0x100081 frequently starts a QueryDirectory transaction
- When a typical transaction is assumed we can start a sequence of instant responses

Internal File IDs

- Accelerating Create(s) requires generation of FIDs in the proxy
- Client-side FID versus Server-side FID
  - LAN-side file IDs are generated by proxy
  - WAN-side file IDs are assigned by server

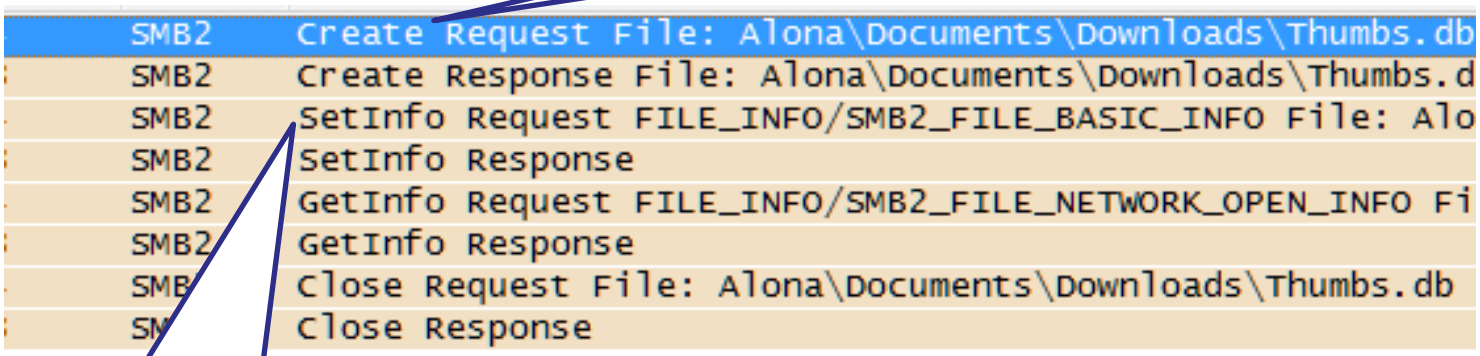
Internal File IDs (cont.)



Accelerator can follow this transaction and it can answer instantly since the file has been cached.
Only LAN FID is used.

Internal File IDs (cont.)

Transaction starts here



we have to transfer this request to server

Here Accelerator breaks the sequence of instant responses and obtains WAN FID to continue with the transaction sequence.

We **saved two round trips** by instantly answering on GetInfo and Close

Internal File IDs (cont.)

On a broken sequence we:

- Send Create to server and get FID in response.
- Assign this server-side FID to client-side FID that we has already generated.
- Send commands already answered internally.
- Continue with the transaction.
- While delegating requests to server we translate LAN FID into WAN FID.
- While delegating responses to server we translate WAN FID into LAN FID.

Caching

Caching for SMB2 uses mostly the same principles as for SMB(1), except for the following:

- Pre-fetching involves Create and Close commands.
- In SMB(1) pre-fetching queries are sent as separate (yet concurrent) packets. In SMB2 they may be chained in the framework of the same Create and Close.

Caching (cont.)


```
SMB2 (Server Message Block Protocol version 2)
  SMB2 Header
  Create Request (0x05)
SMB2 (Server Message Block Protocol version 2)
  SMB2 Header
  GetInfo Request (0x10)
    Length: 40
    .... .... .... ...1 = Dynamic Part: True
    Class: FS_INFO (0x02)
    InfoLevel: SMB2_FS_INFO_05 (0x05)
    Max Response Size: 924
    unknown: 00000000000000000000000000000000
    GUID handle
SMB2 (Server Message Block Protocol version 2)
  SMB2 Header
  GetInfo Request (0x10)
    Length: 40
    .... .... .... ...1 = Dynamic Part: True
    Class: FS_INFO (0x02)
    InfoLevel: SMB2_FS_INFO_01 (0x01)
    Max Response Size: 924
    unknown: 00000000000000000000000000000000
    GUID handle
SMB2 (Server Message Block Protocol version 2)
  SMB2 Header
  GetInfo Request (0x10)
    Length: 40
    .... .... .... ...1 = Dynamic Part: True
    Class: FS_INFO (0x02)
    InfoLevel: SMB2_FS_OBJECTID_INFO (0x08)
    Max Response Size: 924
    unknown: 00000000000000000000000000000000
    GUID handle
SMB2 (Server Message Block Protocol version 2)
  SMB2 Header
  Close Request (0x06)
```

- This is a fragment of server-side traffic.
- Immediately after TreeConect Accelerator pre-fetches and caches all FS information. Three FS queries are sent as one packet of chained commands.
- Later on Accelerator will answer instantly on client's FS queries without querying the server.

LAN Acceleration

About LAN Acceleration

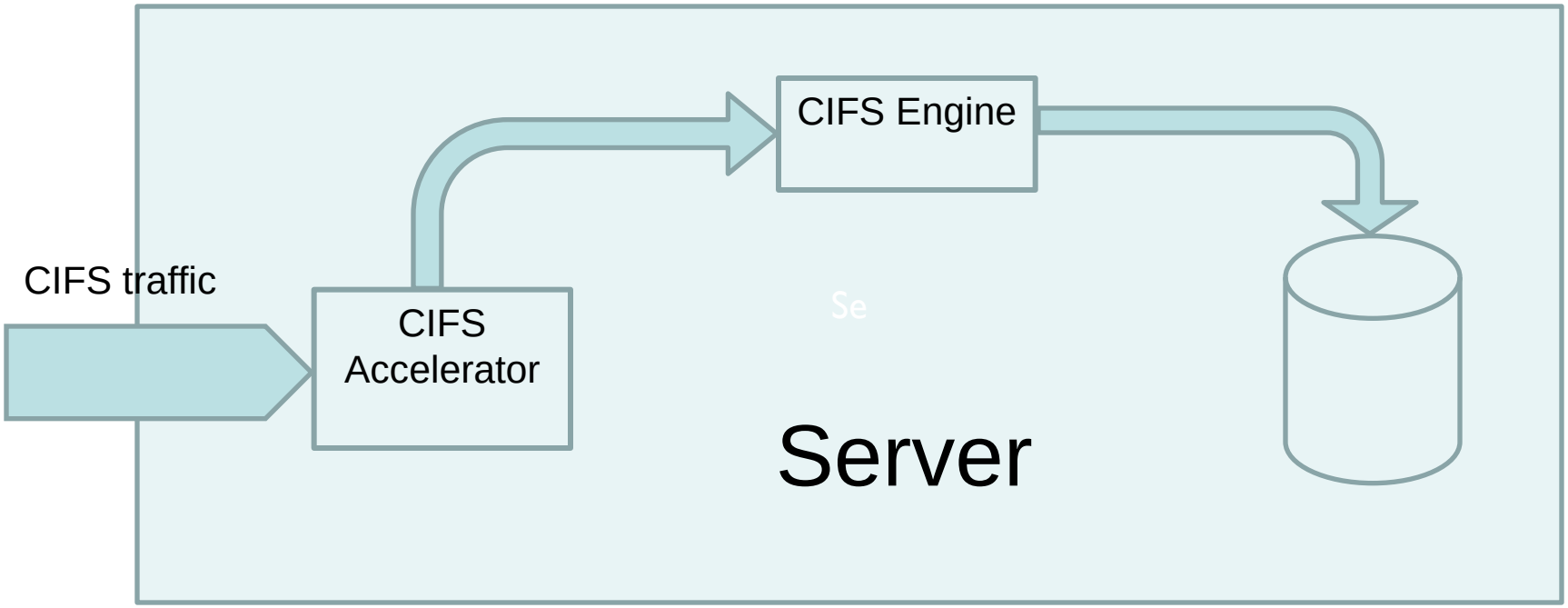
Are WAN acceleration methods applicable to LAN?

Remark: this section's aim is to raise a discussion rather than share any valuable results

Considerations

- On 10/100Gbit networks disk transfer rates become major performance factor rather than bandwidth
- CIFS acceleration on server-side may decrease disk access thus increasing performance

Model



CIFS Accelerator minimizes CIFS traffic thus decreasing access to the storage

LAN vs WAN Acceleration

- LAN acceleration introduces more challenges.
- CIFS acceleration techniques should be reviewed to minimize overheads which are more significant than for WAN acceleration.
- Some techniques should be just avoided.

	WAN	**LAN**
Overhead of CPU consumption	Negligible	Significant
Overhead of broken sequence of instant responses	Negligible	Significant

Q & A