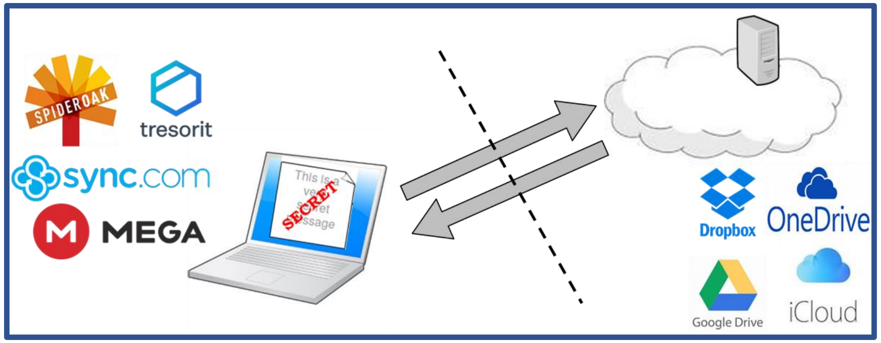
The Overhead of Confidentiality and Client-side Encryption in Cloud Storage Systems

Eric Henziger, *Linköping University*

Niklas Carlsson, *Linköping University*

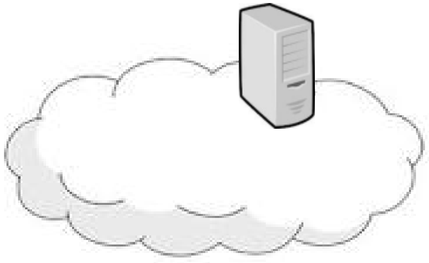
LINKÖPING
UNIVERSITY



Proc. IEEE/ACM UCC, Auckland, New Zealand, Dec. 2019

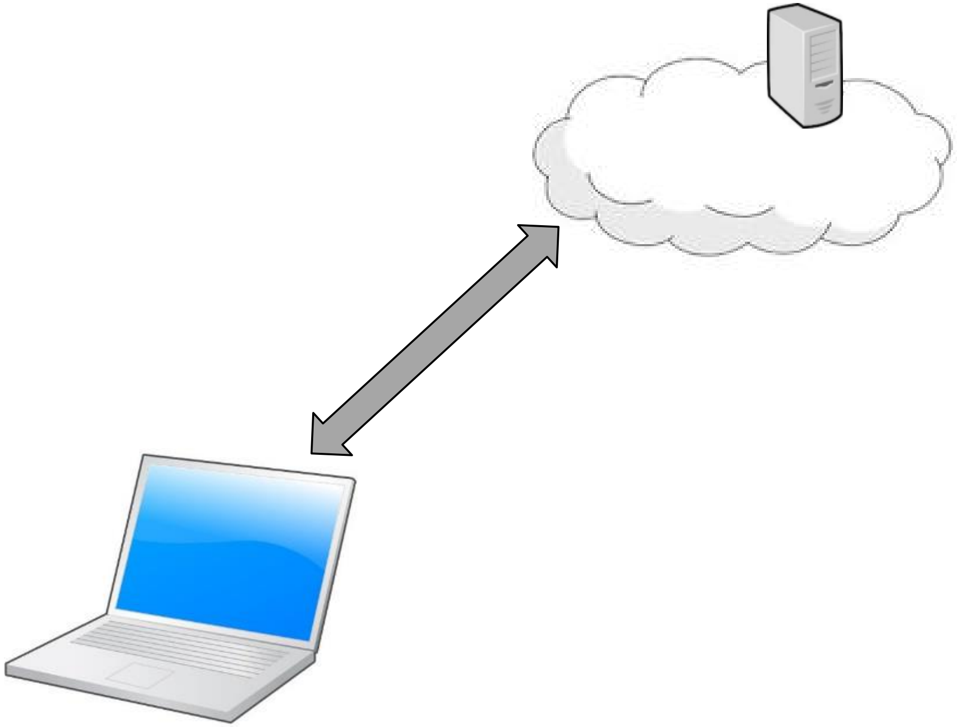
Motivation and problem

- Popular services: Some with 100s of millions of active users each month



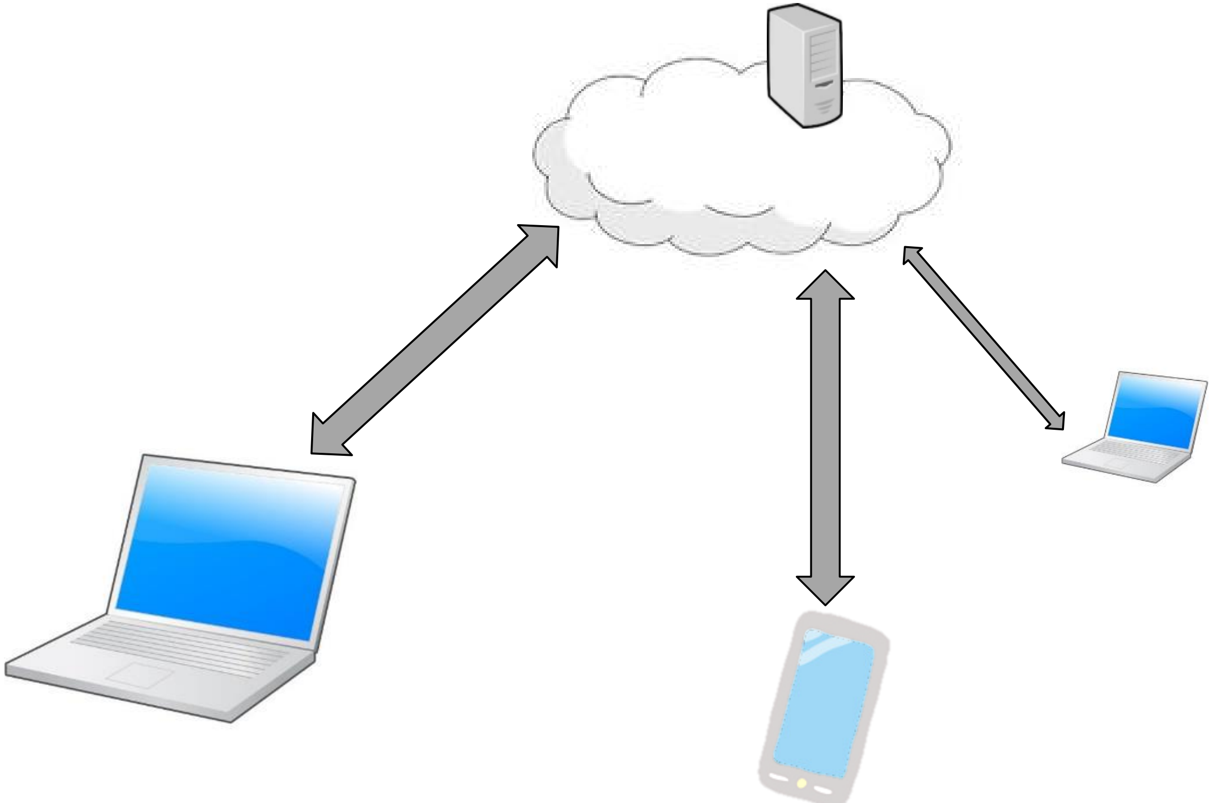
Motivation and problem

- Popular services: Some with 100s of millions of active users each month
- Cloud services have changed how users store and access data



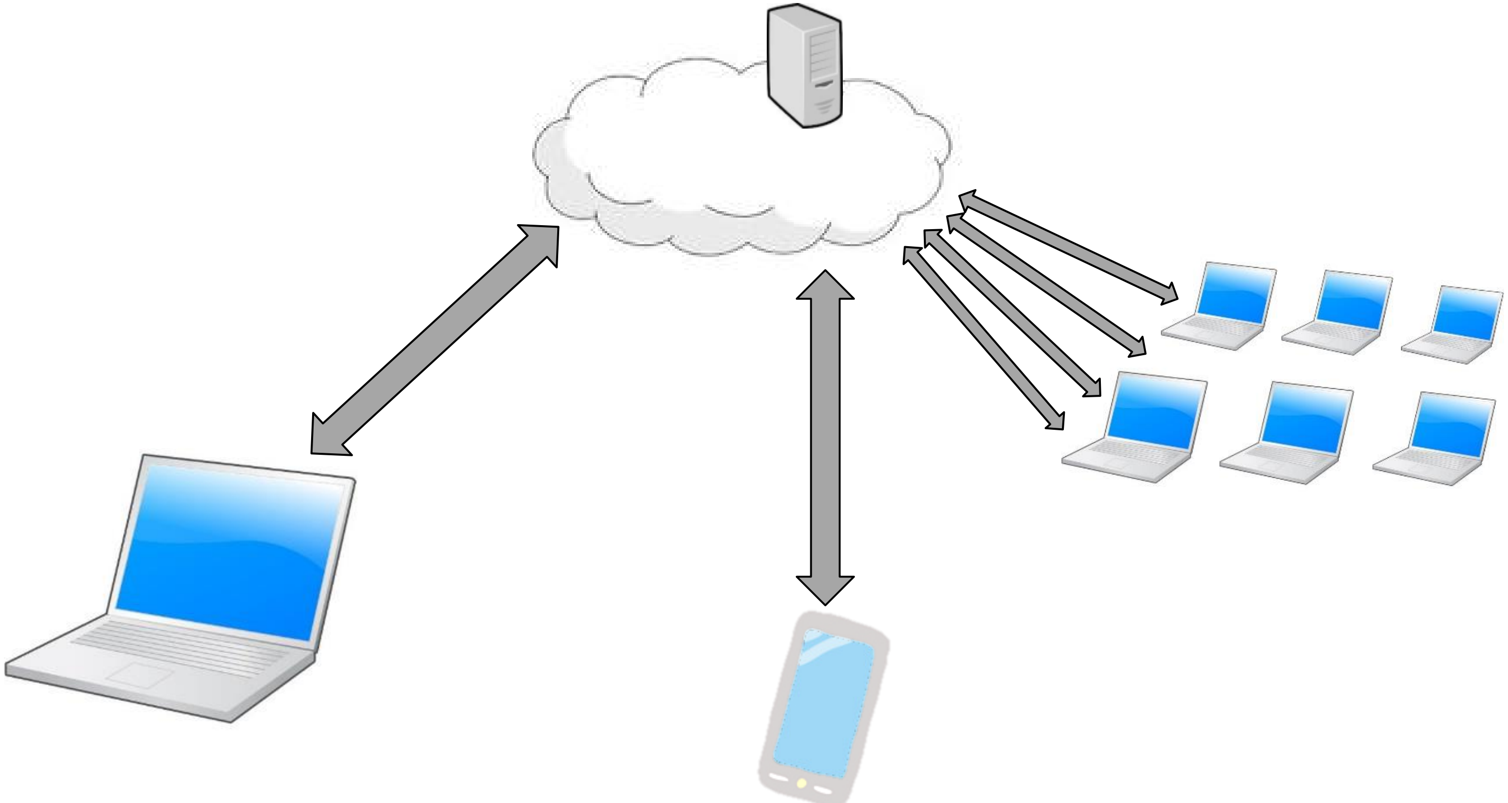
Motivation and problem

- Popular services: Some with 100s of millions of active users each month
- Cloud services have changed how users store and access data
    - E.g., often transparently across multiple devices



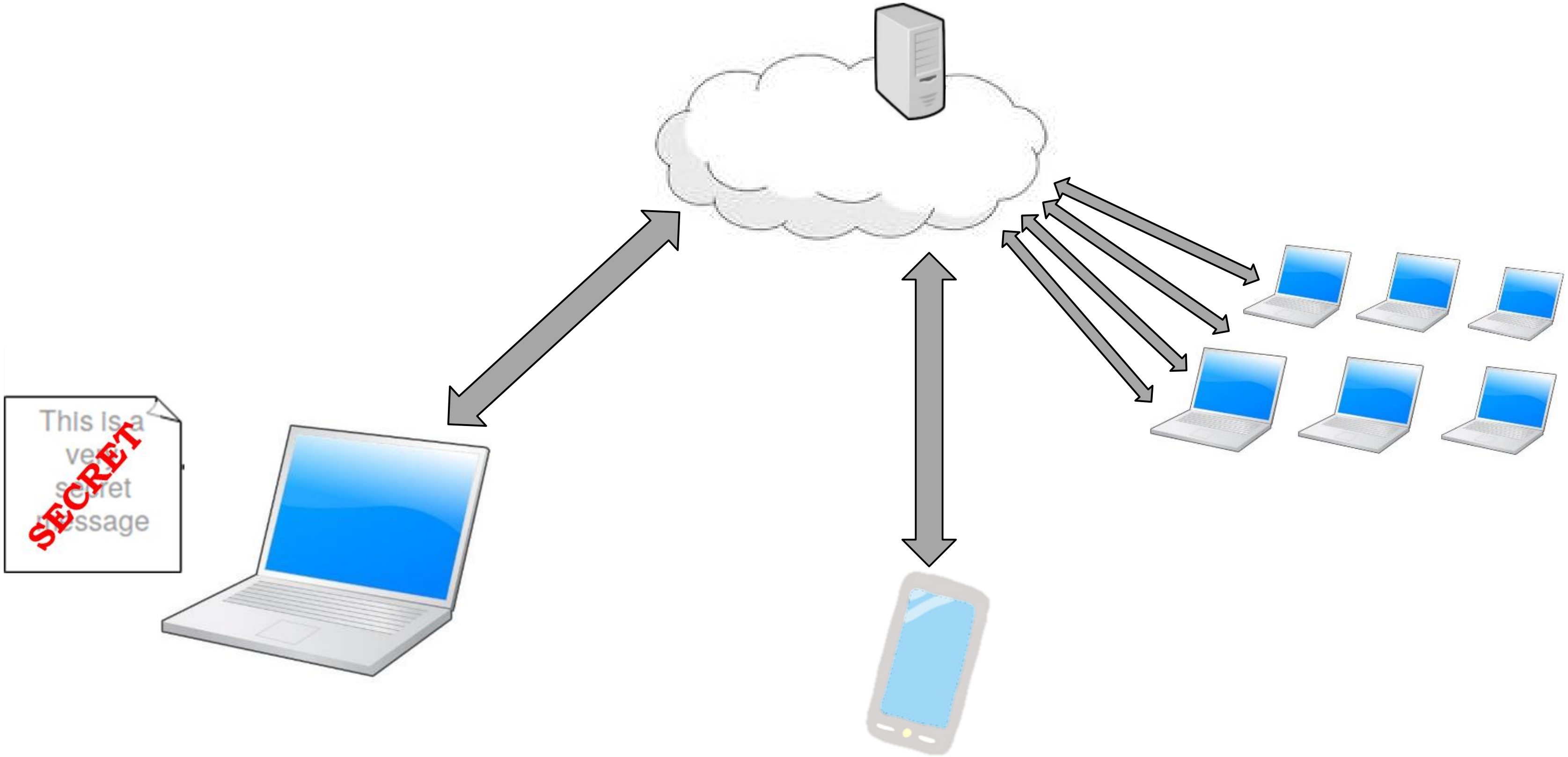
Motivation and problem

- Popular services: Some with 100s of millions of active users each month
- Cloud services have changed how users store and access data
  - E.g., often transparently across multiple devices or users



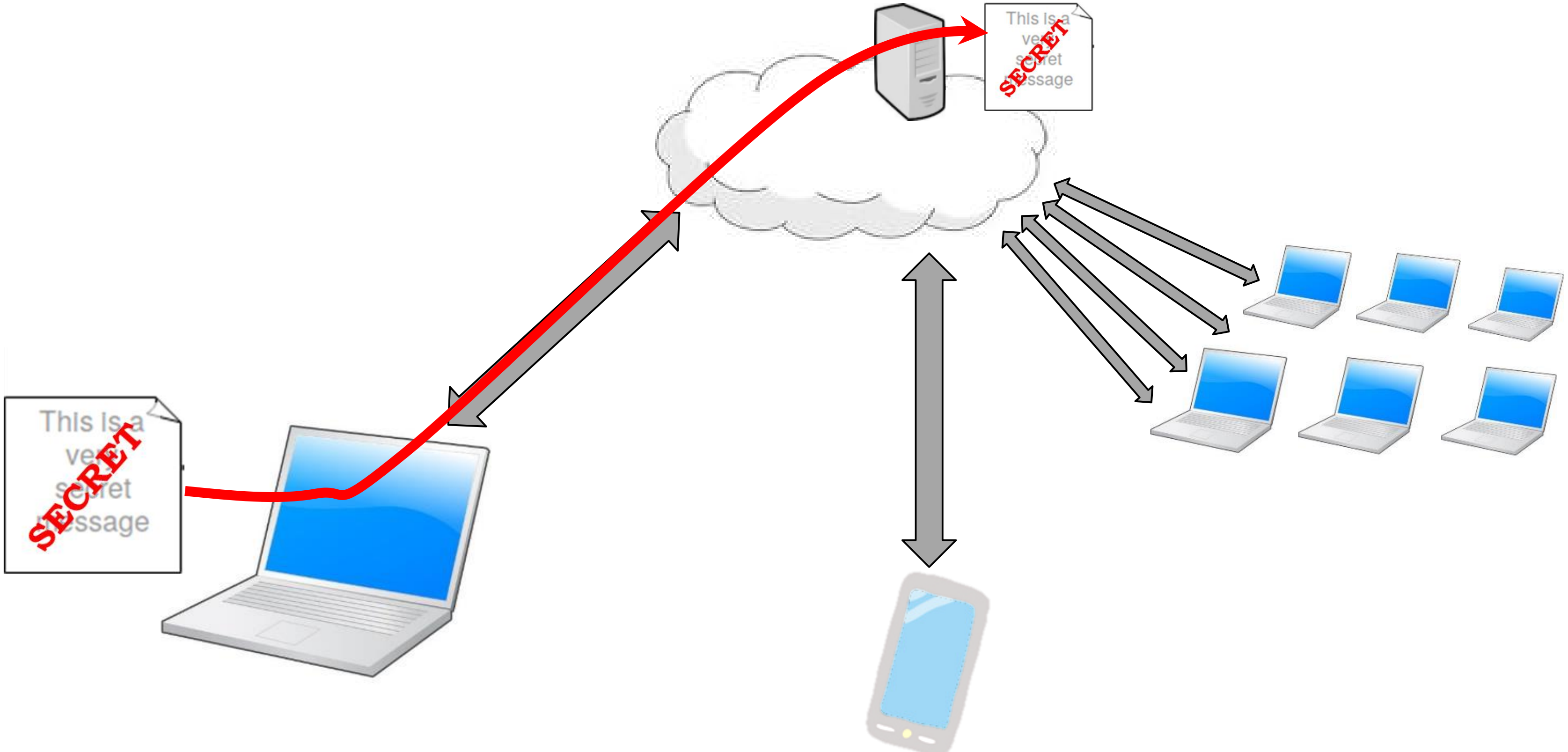
Motivation and problem

- Popular services: Some with 100s of millions of active users each month
- Cloud services have changed how users store and access data
  - E.g., often transparently across multiple devices or users
- Most services require that users fully trust the provider
  - Services gets access to all data and information



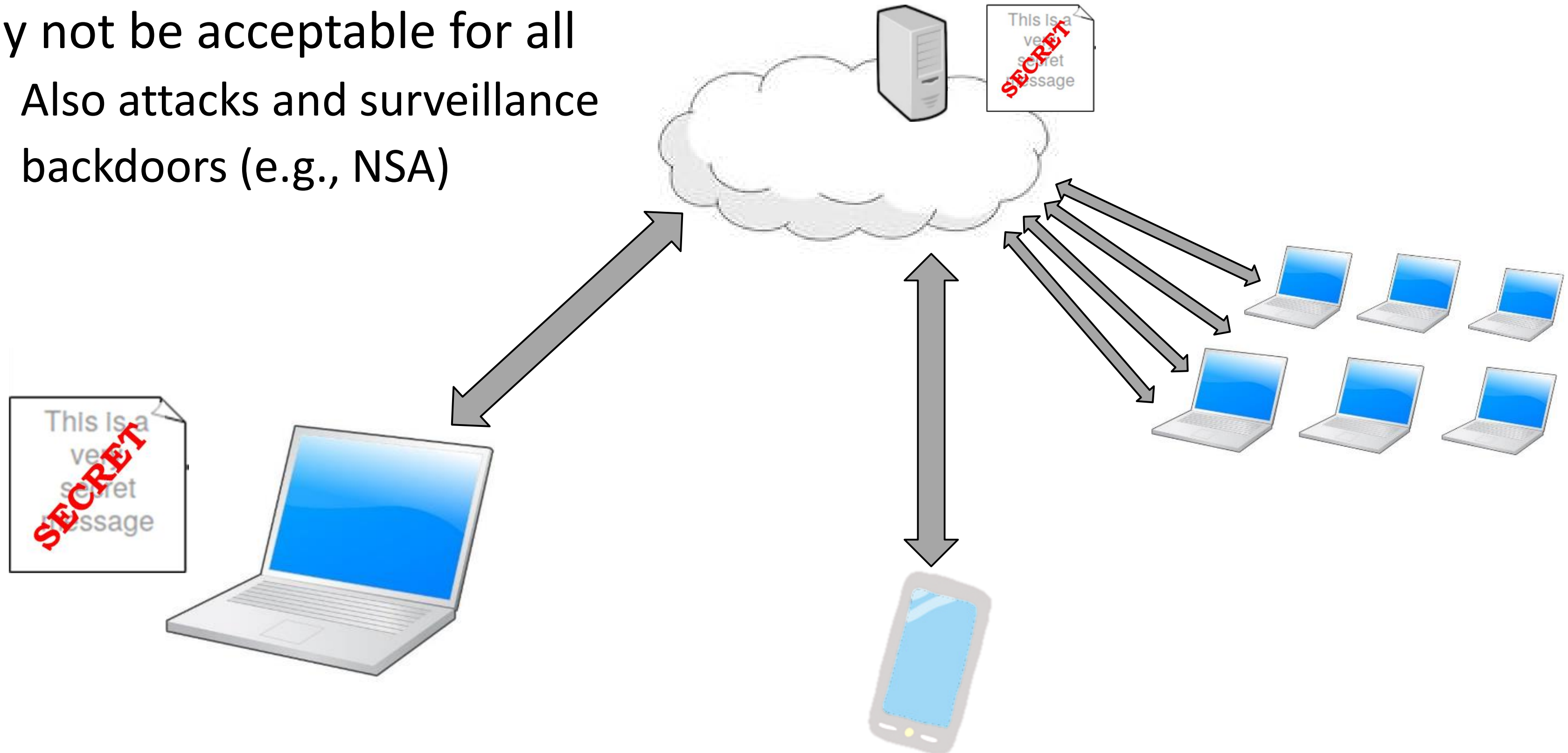
Motivation and problem

- Popular services: Some with 100s of millions of active users each month
- Cloud services have changed how users store and access data
  - E.g., often transparently across multiple devices or users
- Most services require that users fully trust the provider
  - Services gets access to all data and information



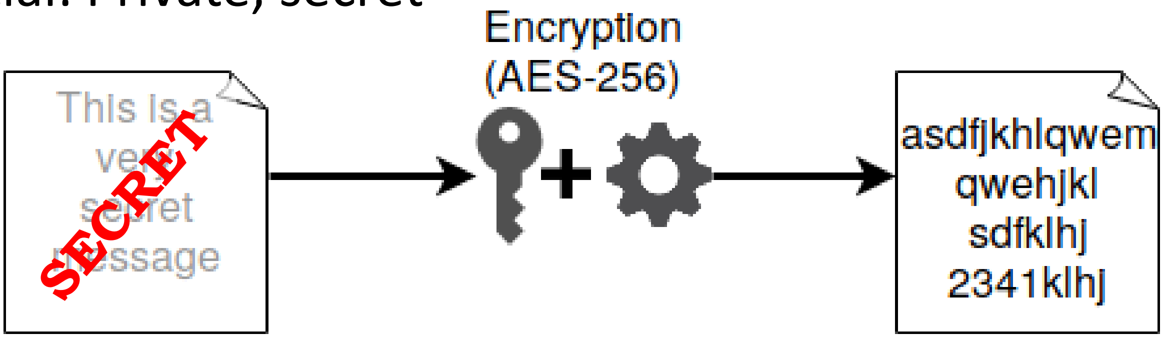
Motivation and problem

- Popular services: Some with 100s of millions of active users each month
- Cloud services have changed how users store and access data
  - E.g., often transparently across multiple devices or users
- Most services require that users fully trust the provider
  - Services gets access to all data and information
- May not be acceptable for all
  - Also attacks and surveillance backdoors (e.g., NSA)



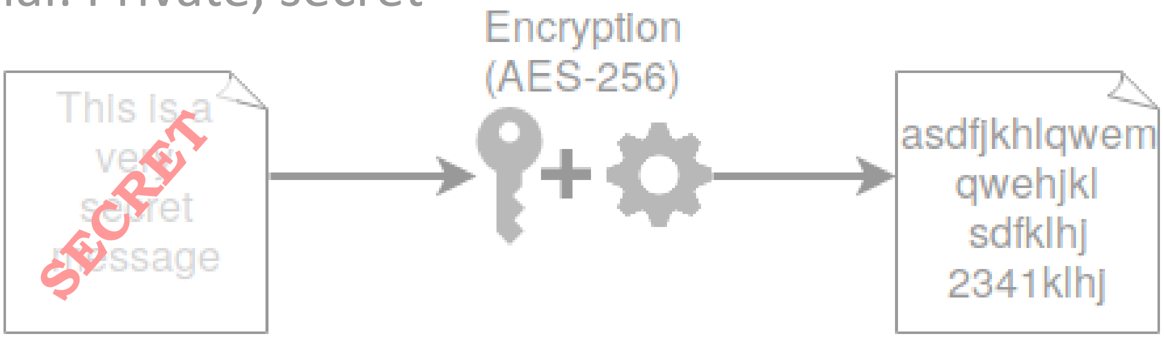
Client-side encryption (CSE)

- Confidential: Private, secret

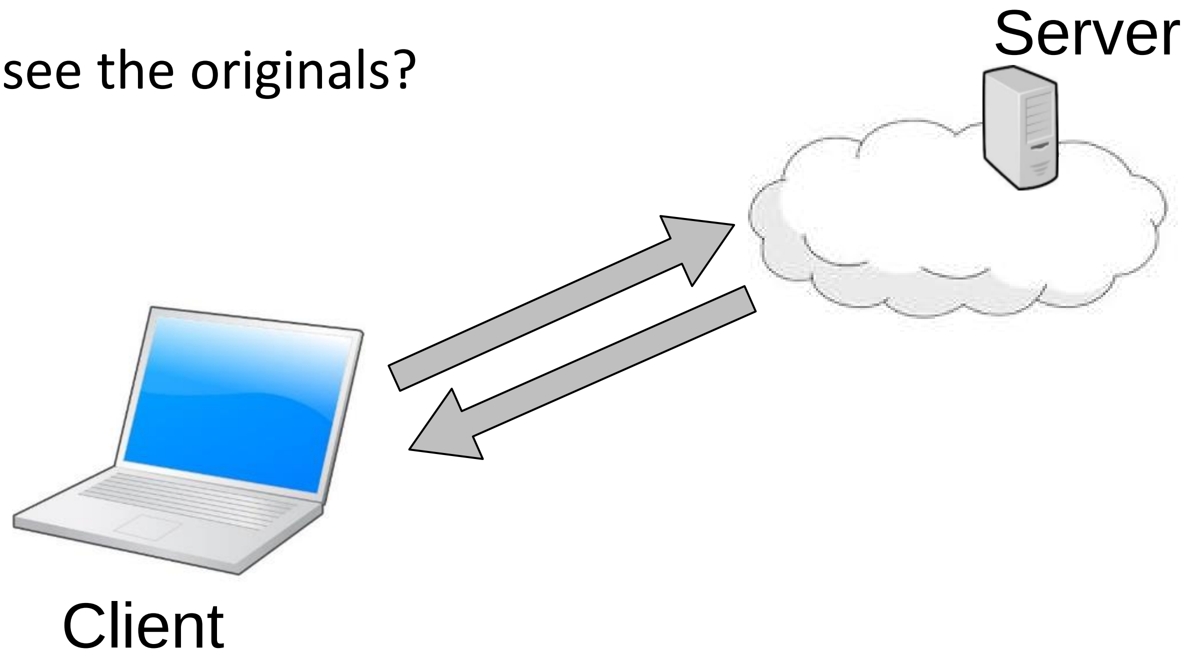

Encryption
(AES-256)
This is a very secret message
SECRET
+
asdfjkhlqwem qwehjkl sdfklhj 2341klhj

Client-side encryption (CSE)

- Confidential: Private, secret

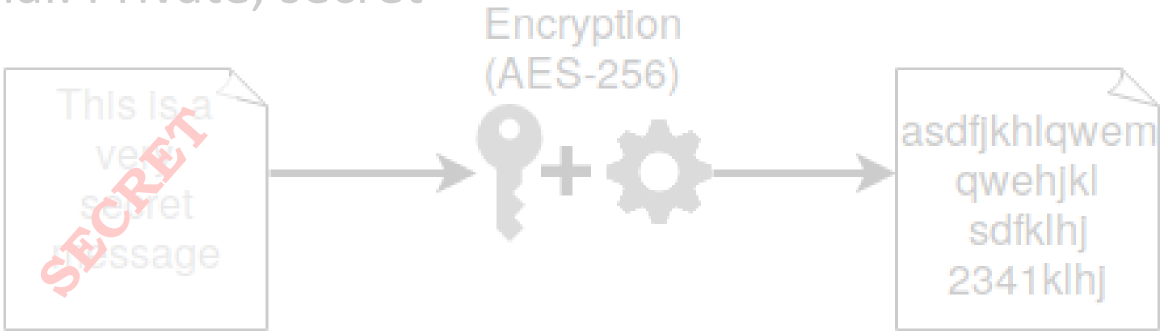

This is a very secret message
SECRET
Encryption (AES-256)
asdfjkhlqwem qwehjkl sdfklhj 2341klhj

- Who can see the originals?

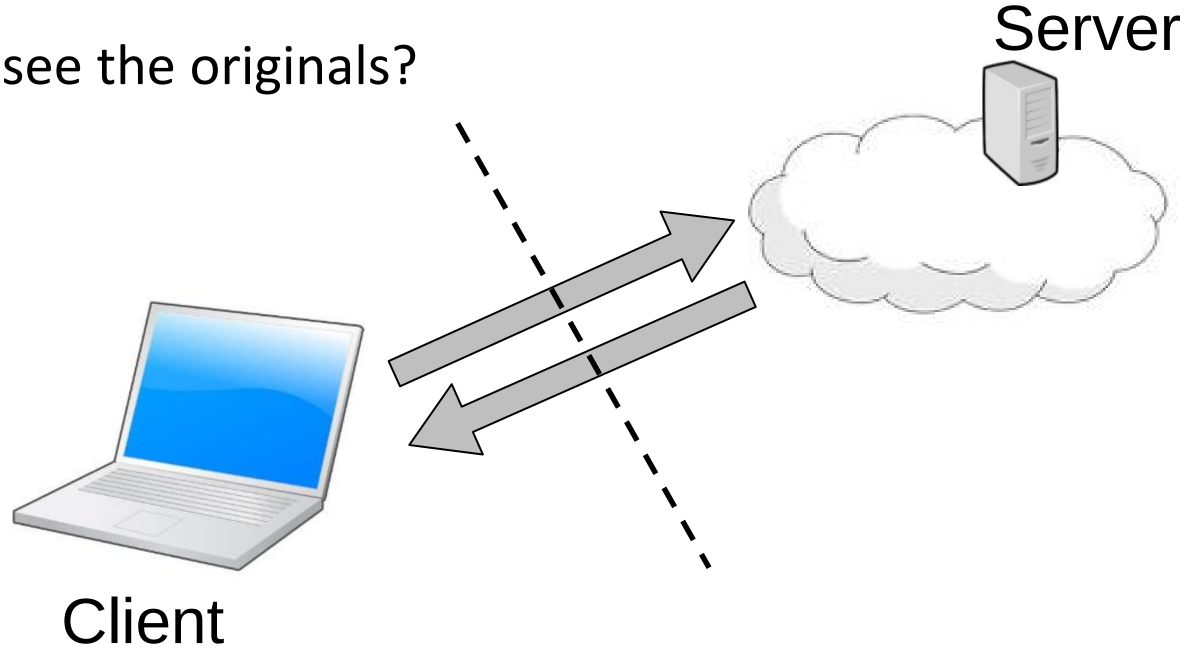


Client-side encryption (CSE)

- Confidential: Private, secret

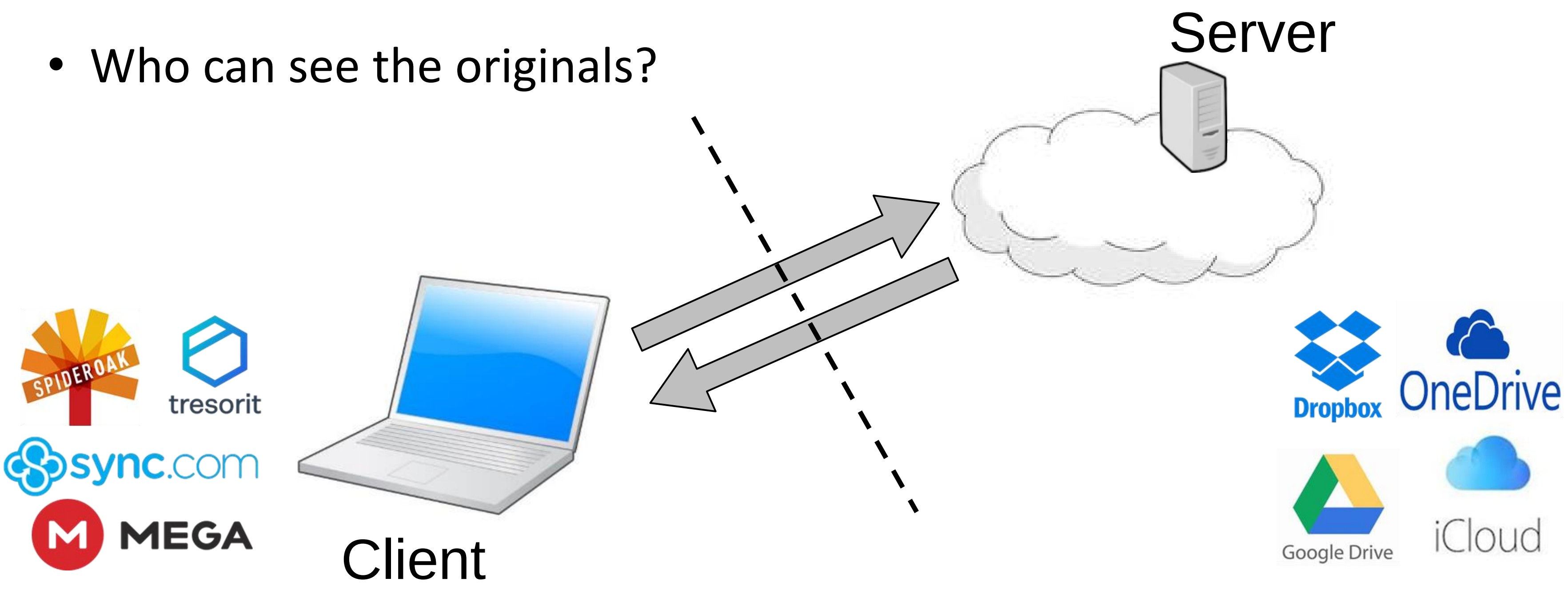

This is a very secret message
SECRET
Encryption (AES-256)
asdfjkhlqwem qwehjkl sdfklhj 2341klhj

- Who can see the originals?



Client-side encryption (CSE)

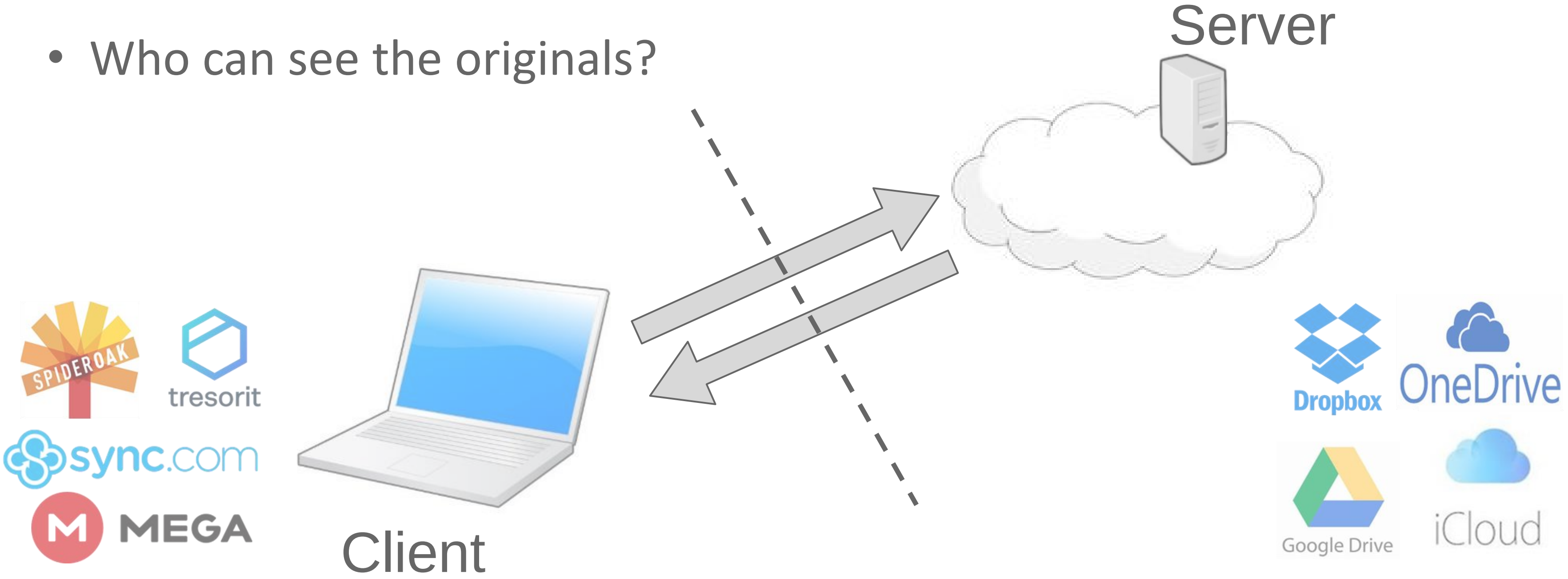
- Who can see the originals?



Client-side encryption (CSE)

However, CSE complicates some bandwidth saving features such as deduplication and delta encoding …

- Who can see the originals?



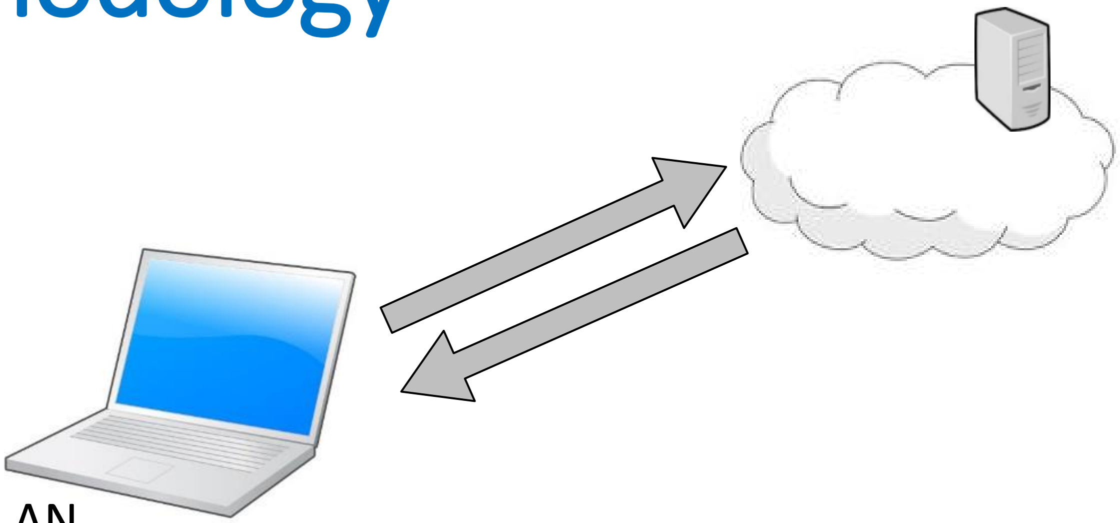
Contributions

Empirically investigate the potential overhead penalty associated with CSE through comparisons of four CSEs and four non-CSEs

1. Controlled experiments to compare and contrast the security and bandwidth saving features implemented
2. Performance tests to compare non-traffic related client-side overheads (e.g., CPU, disk, memory)
3. Targeted example experiments to demonstrate some weaknesses in existing delta encoding solutions

To the best of our knowledge, this is the first research paper that focuses on the difference between CSE and non-CSE supporting services

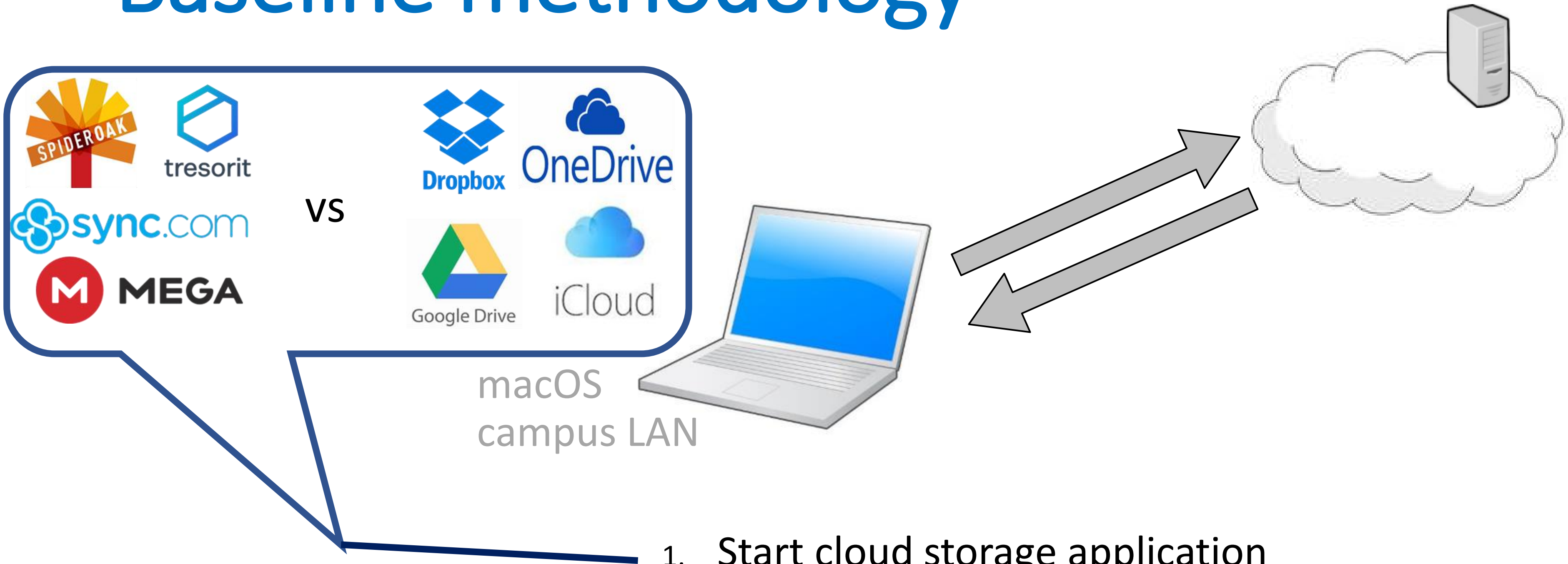
Baseline methodology



macOS
campus LAN

E. Bocchi, I. Drago, and M. Mellia, “Personal Cloud Storage Benchmarks and Comparison,” IEEE Transactions on Cloud Computing, vol. 5, no. 4, pp. 751–764, 2017.

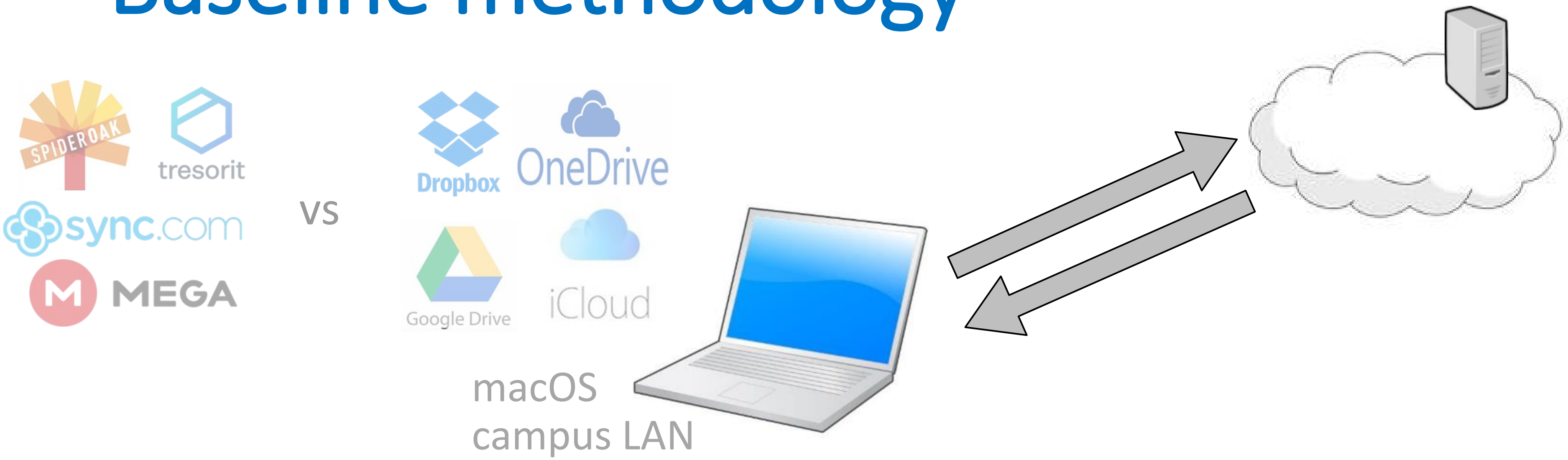
Baseline methodology



1. Start cloud storage application

E. Bocchi, I. Drago, and M. Mellia, "Personal Cloud Storage Benchmarks and Comparison," IEEE Transactions on Cloud Computing, vol. 5, no. 4, pp. 751–764, 2017.

Baseline methodology



1. Start cloud storage application
2. Capture network traffic
3. Measure CPU, memory, disk utilization

python™

- netifaces
- pcapy
- psutil
- numpy
- scipy

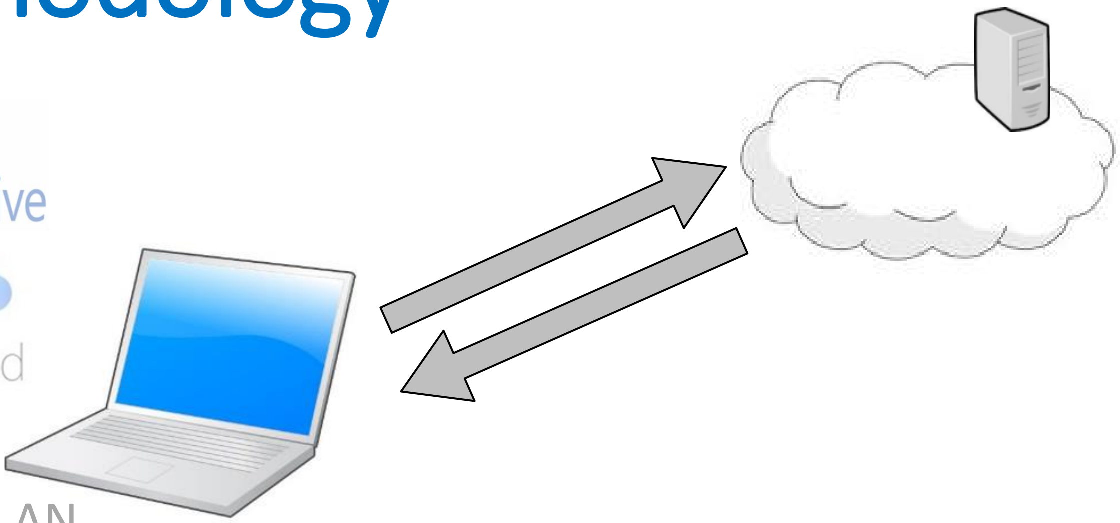
E. Bocchi, I. Drago, and M. Mellia, "Personal Cloud Storage Benchmarks and Comparison," IEEE Transactions on Cloud Computing, vol. 5, no. 4, pp. 751–764, 2017.

Baseline methodology

SPIDEROAK tresorit sync.com MEGA

vs

Dropbox OneDrive Google Drive iCloud



macOS
campus LAN

python

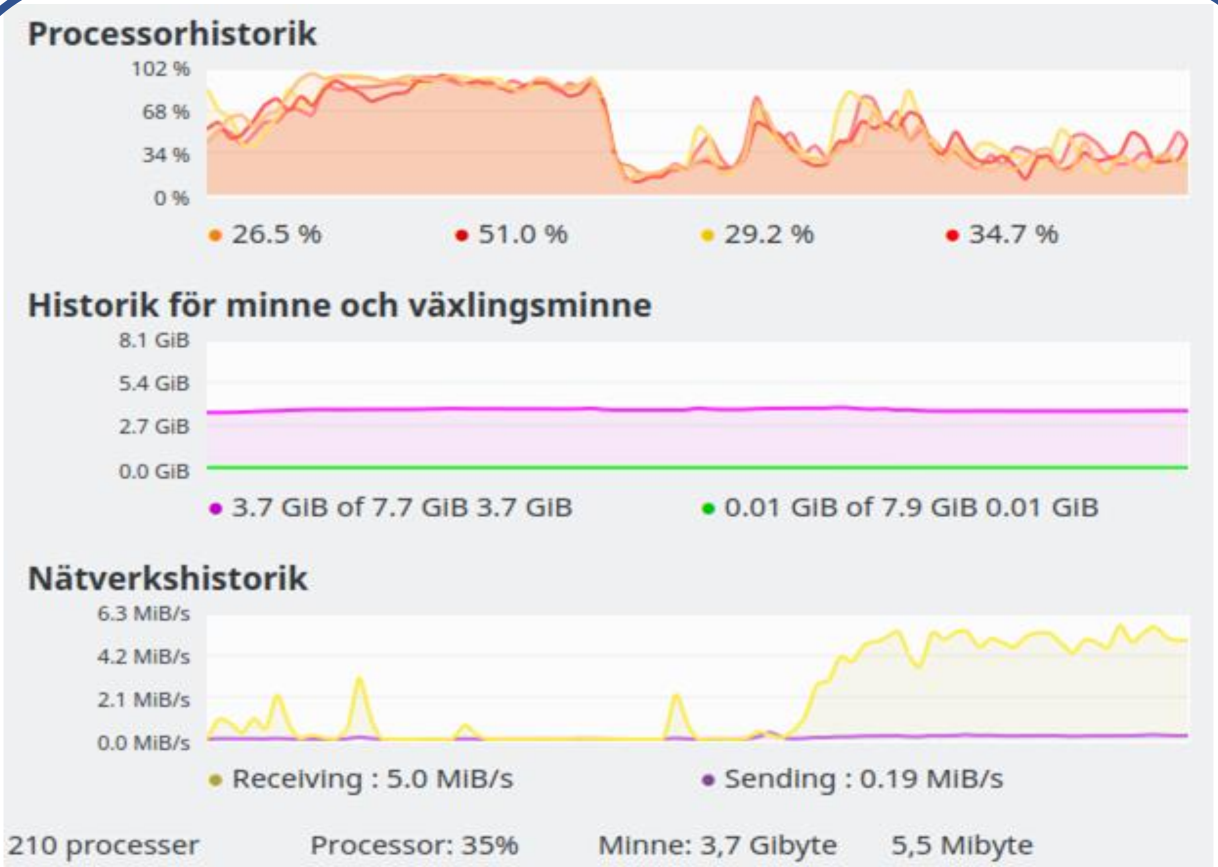
- netifaces
- pcapy
- psutil
- numpy
- scipy

1. Start cloud storage application
2. Capture network traffic
3. Measure CPU, memory, disk utilization
4. Place file in sync folder
5. Wait for synchronization to finish
6. Process capture files and measurements

E. Bocchi, I. Drago, and M. Mellia, "Personal Cloud Storage Benchmarks and Comparison," IEEE Transactions on Cloud Computing, vol. 5, no. 4, pp. 751–764, 2017.

Performance costs; client overheads

- CPU, memory, disk, network traffic



Processorhistorik

● 26.5 % ● 51.0 % ● 29.2 % ● 34.7 %

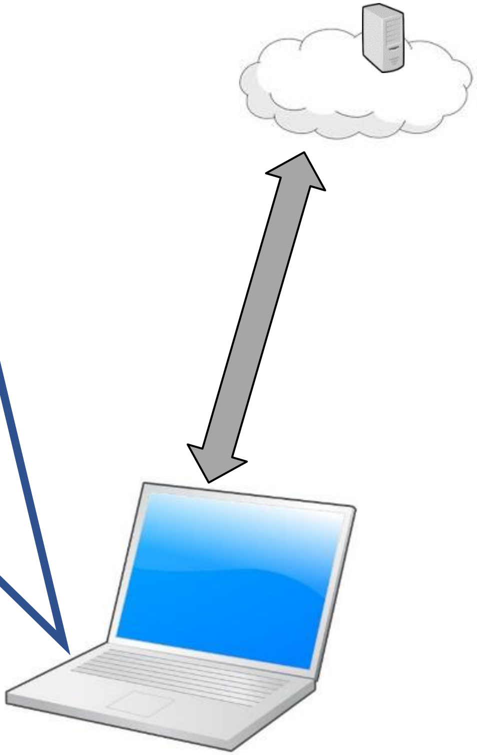
Historik för minne och växlingsminne

● 3.7 GiB of 7.7 GiB 3.7 GiB ● 0.01 GiB of 7.9 GiB 0.01 GiB

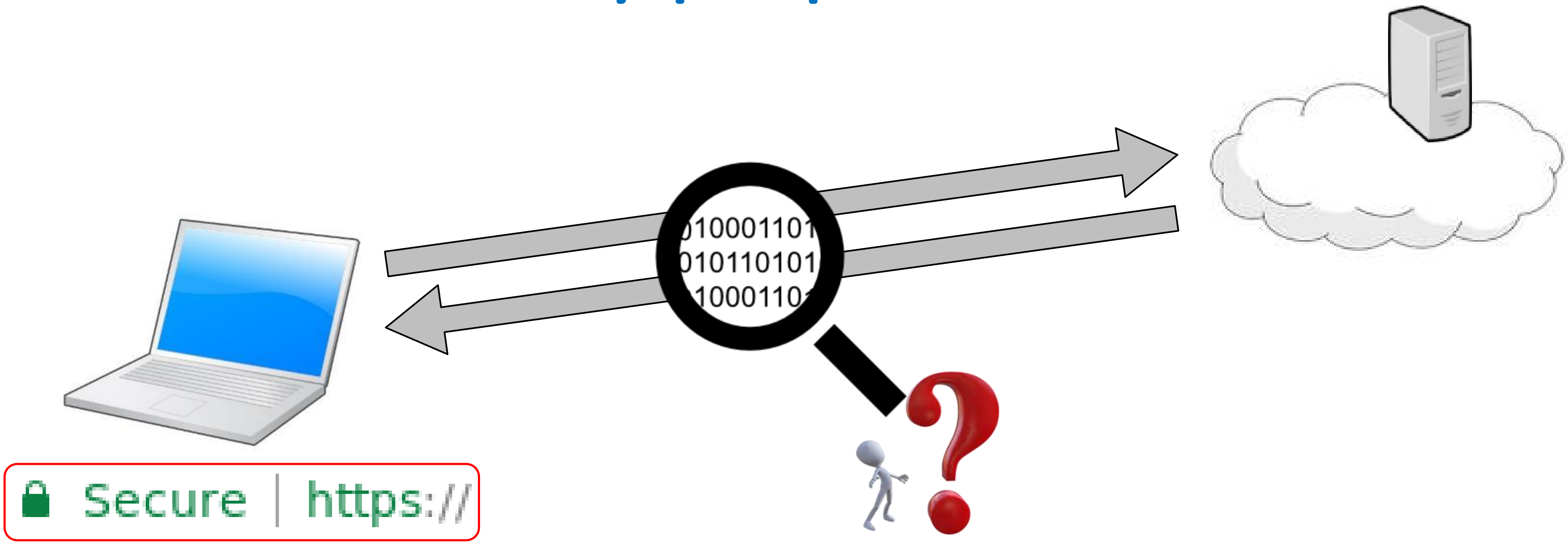
Nätverkshistorik

● Receiving : 5.0 MiB/s ● Sending : 0.19 MiB/s

210 processer Processor: 35% Minne: 3,7 Gibyte 5,5 Mibyte

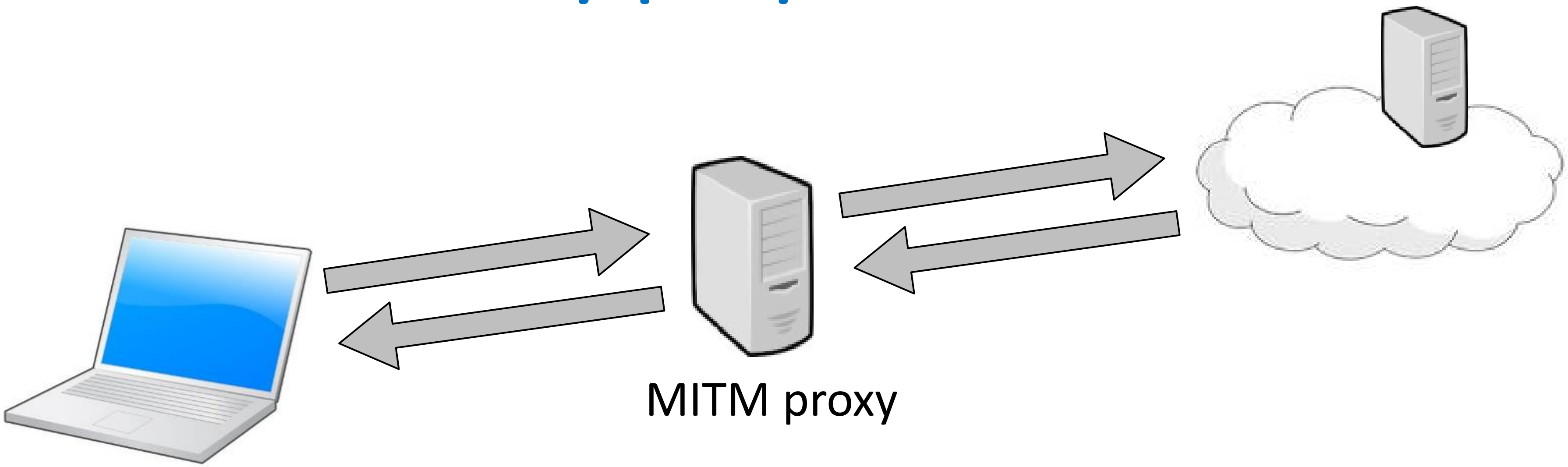


Basic security properties



- All services except Mega use HTTPS (** Mega defaults to HTTP, but has HTTPS)
- Mega and SpiderOak use TLS 1.0; rest TLS 1.2
- All use reasonable signatures (e.g., SHA256+RSA or SHA256+ECC) and encryption for transfer RSA 2048 + AES 128/256 (or corresponding EC)
- In Nov. 2017, three non-CSEs (Dropbox, iCloud, and Google Drive) supported SCT for certificate transparency (CT), but none of the CSEs

Basic security properties



Secure | https://

Set application to trust MITM proxy (add proxy certificate to root store)

- All applications except Mega prevent TLS interception
- Reason: certificate pinning or similar techniques used

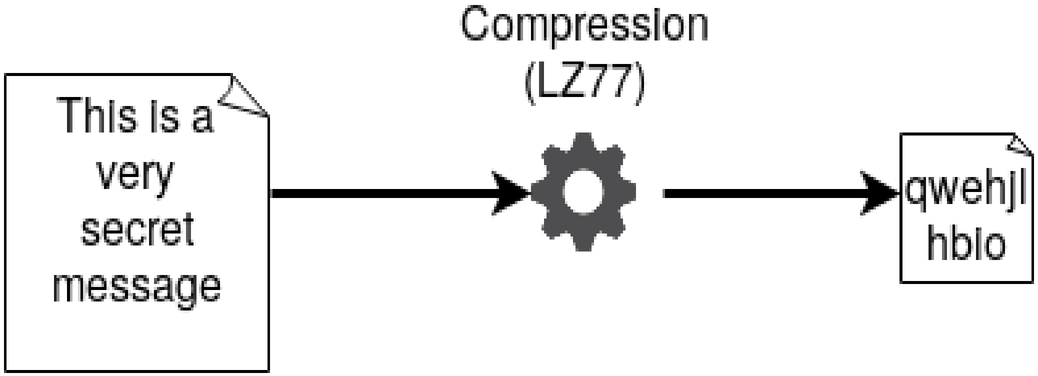
Same, but using their respective interfaces

- Interception successful for all services (except SpiderOak, who does not have a web interface)
- What we see appears to match services CSE claims

Bandwidth saving features

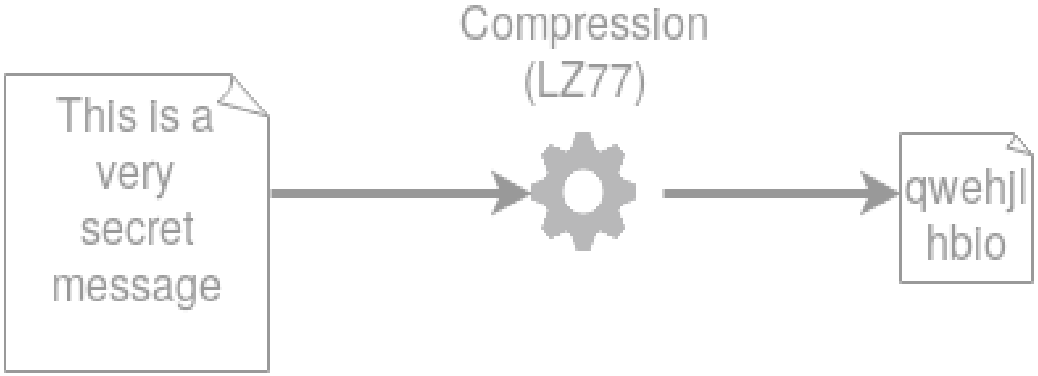
Feature 1: Compression

```
eric@Zipper:/tmp$ ls -l big.txt
-rw-rw-r-- 1 eric eric 6488666 big.txt
eric@Zipper:/tmp$ gzip big.txt
eric@Zipper:/tmp$ ls -l big.txt.gz
-rw-rw-r-- 1 eric eric 2385263 big.txt.gz
```



Feature 1: Compression

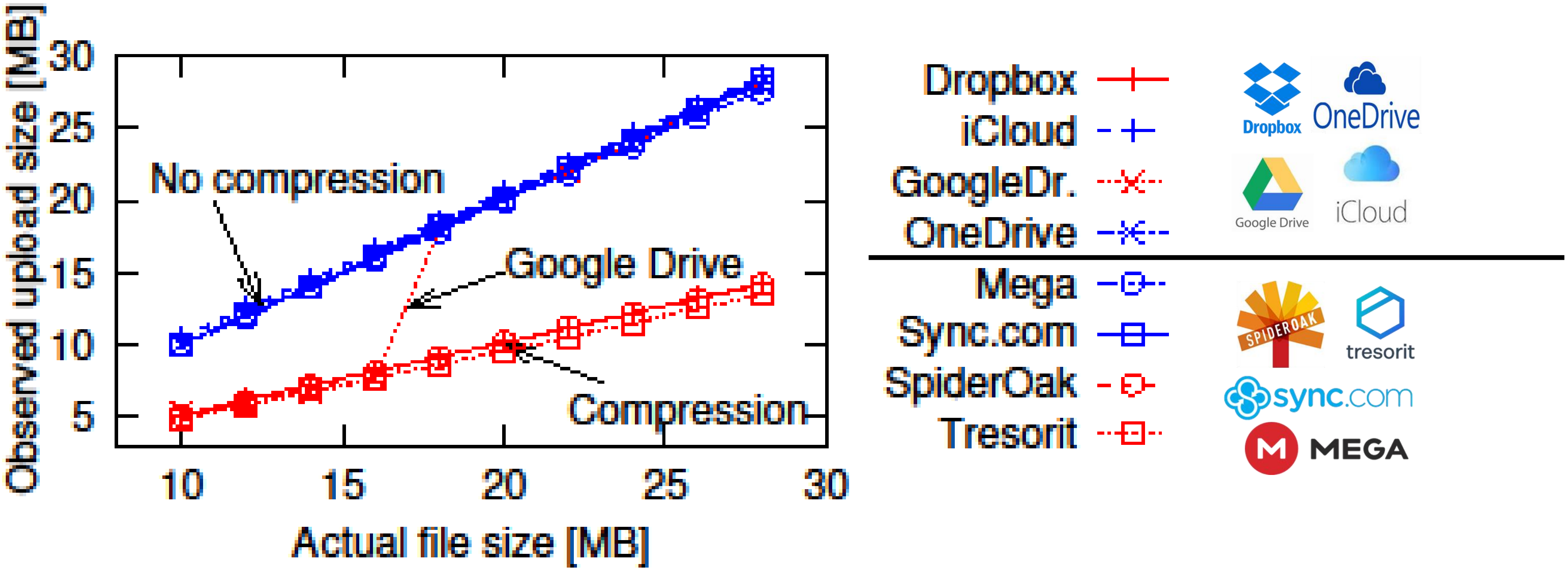
```
eric@Zipper:/tmp$ ls -l big.txt
-rw-rw-r-- 1 eric eric 6488666 big.txt
eric@Zipper:/tmp$ gzip big.txt
eric@Zipper:/tmp$ ls -l big.txt.gz
-rw-rw-r-- 1 eric eric 2385263 big.txt.gz
```



Test procedure

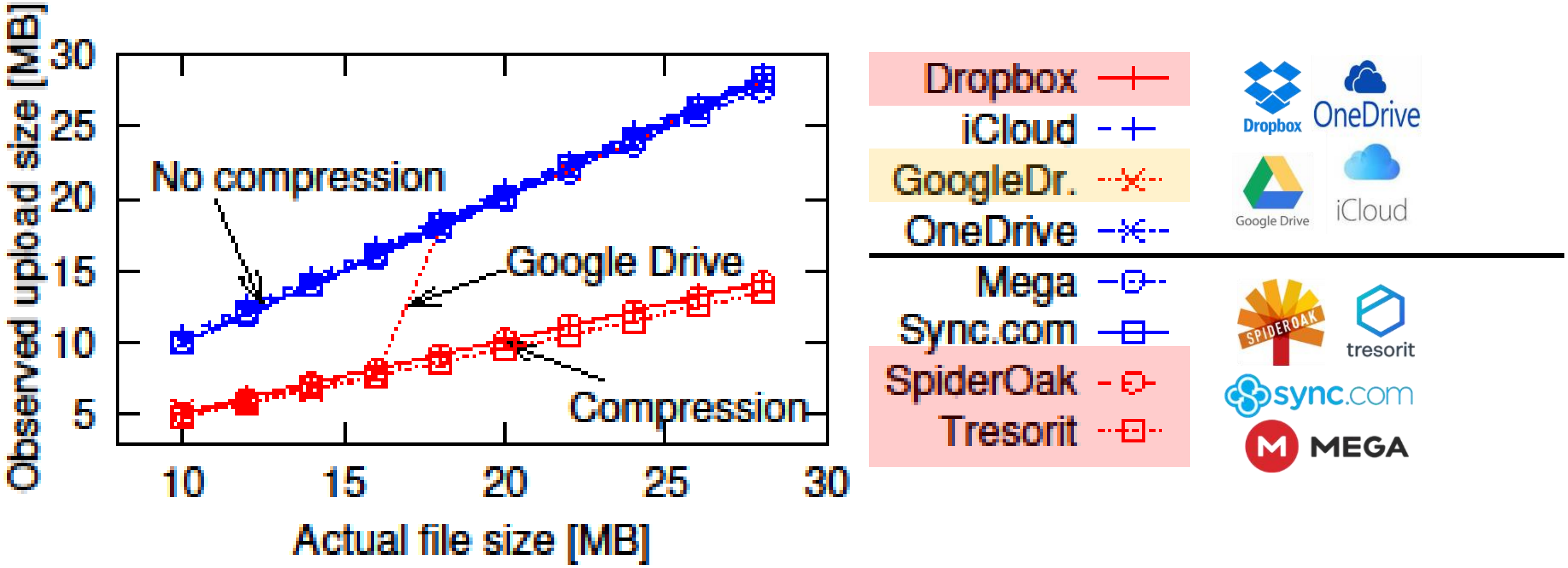
- Create files of sizes 10-28 MB containing random English words
- Determine amount of uploaded bytes
- If uploaded bytes < file size, then compression

Feature 1: Compression



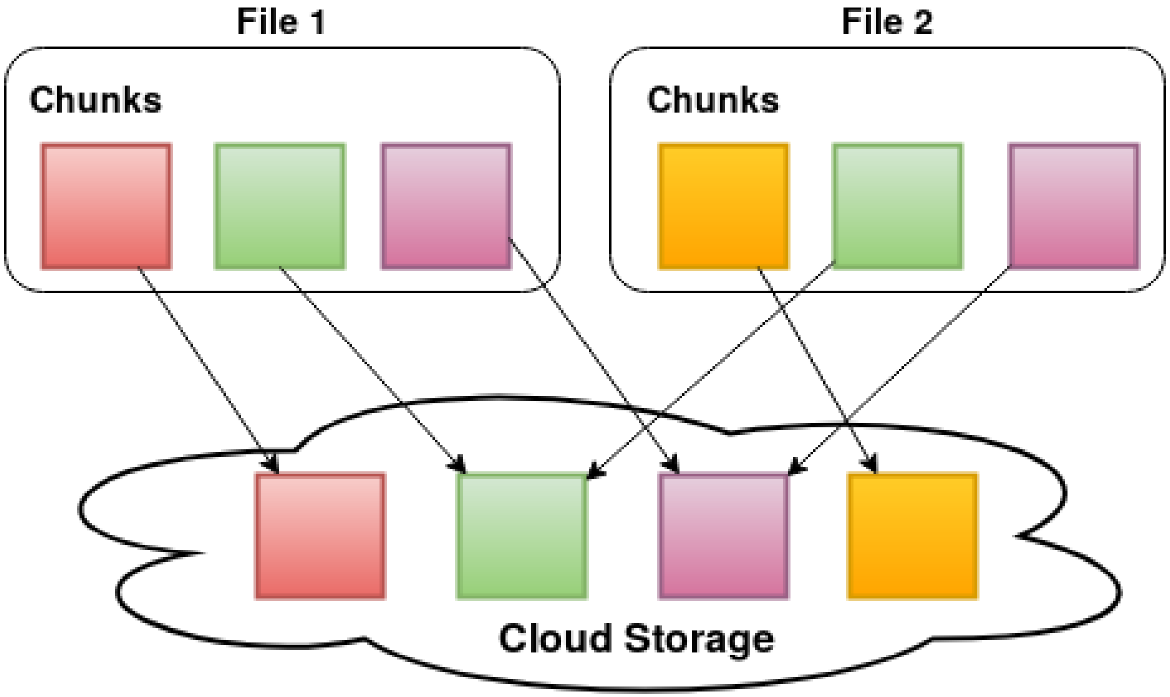
- Dropbox, SpiderOak and Tresorit do compression
- Google Drive does compression if file size is <2^{24} bytes (limit found with binary search)

Feature 1: Compression

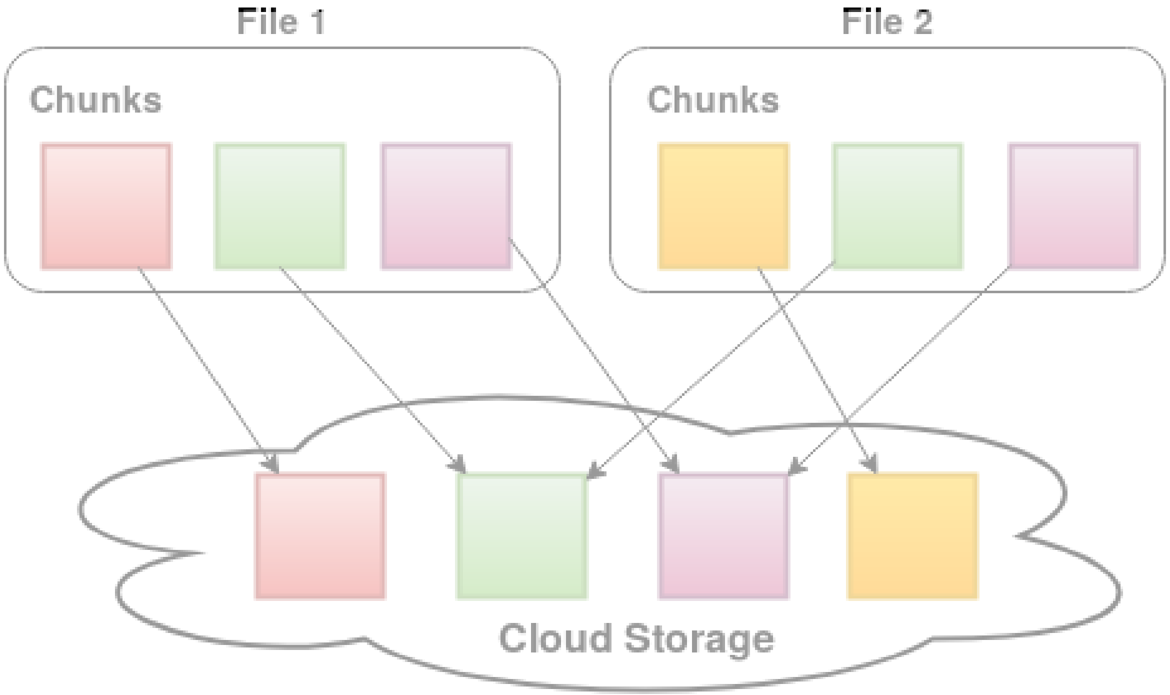


- Dropbox, SpiderOak and Tresorit do compression
- Google Drive does compression if file size is $<2^{24}$ bytes (limit found with binary search)

Feature 2: Deduplication



Feature 2: Deduplication

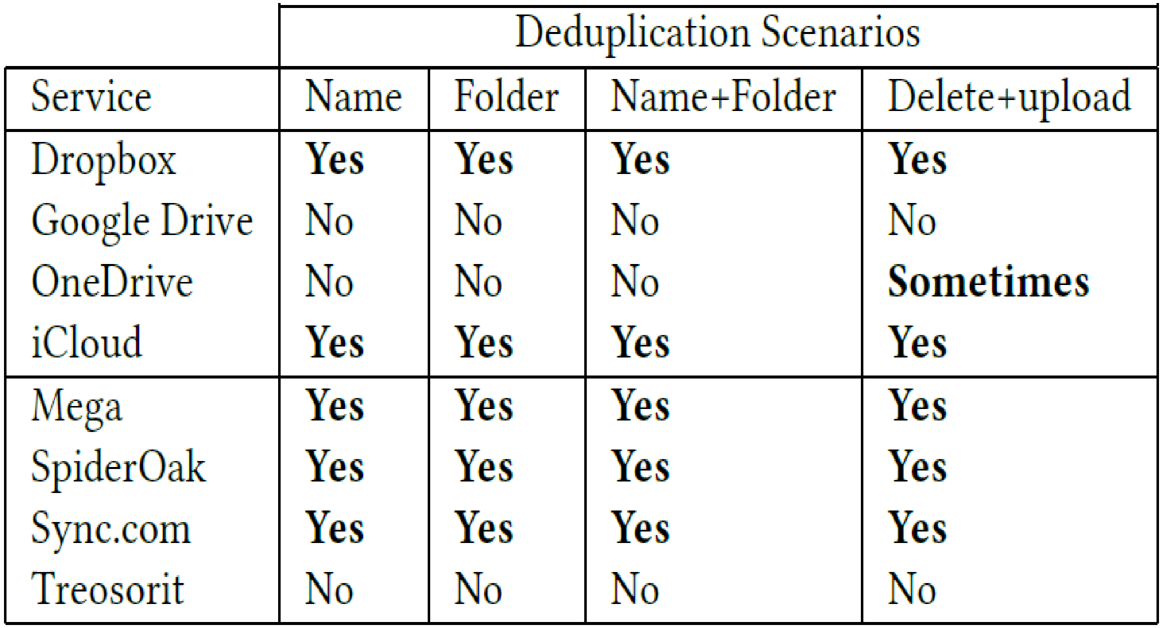


Test procedure

- Store two files with identical content
- If second file is synced without significant upload, then deduplication

Feature 2: Deduplication

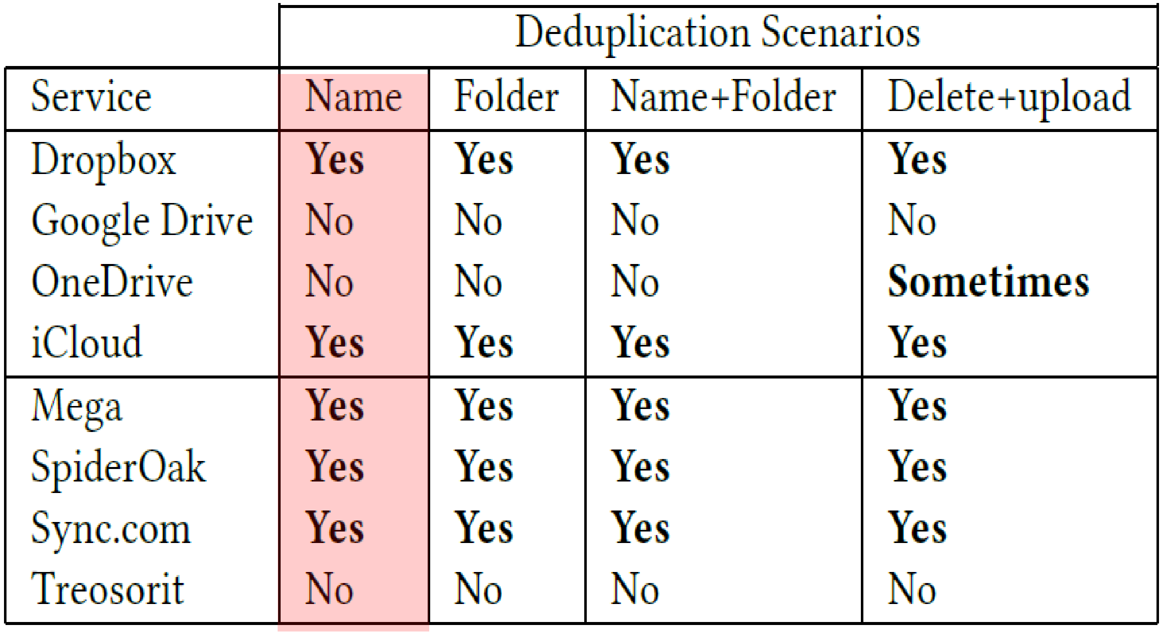
- Store two files with identical content
  - Different file names
  - Different folders
  - Different file name and folder
  - By deleting the file and then re-uploading it



	Deduplication Scenarios			
Service	Name	Folder	Name+Folder	Delete+upload
Dropbox	**Yes**	**Yes**	**Yes**	**Yes**
Google Drive	No	No	No	No
OneDrive	No	No	No	**Sometimes**
iCloud	**Yes**	**Yes**	**Yes**	**Yes**
Mega	**Yes**	**Yes**	**Yes**	**Yes**
SpiderOak	**Yes**	**Yes**	**Yes**	**Yes**
Sync.com	**Yes**	**Yes**	**Yes**	**Yes**
Treosorit	No	No	No	No

Feature 2: Deduplication

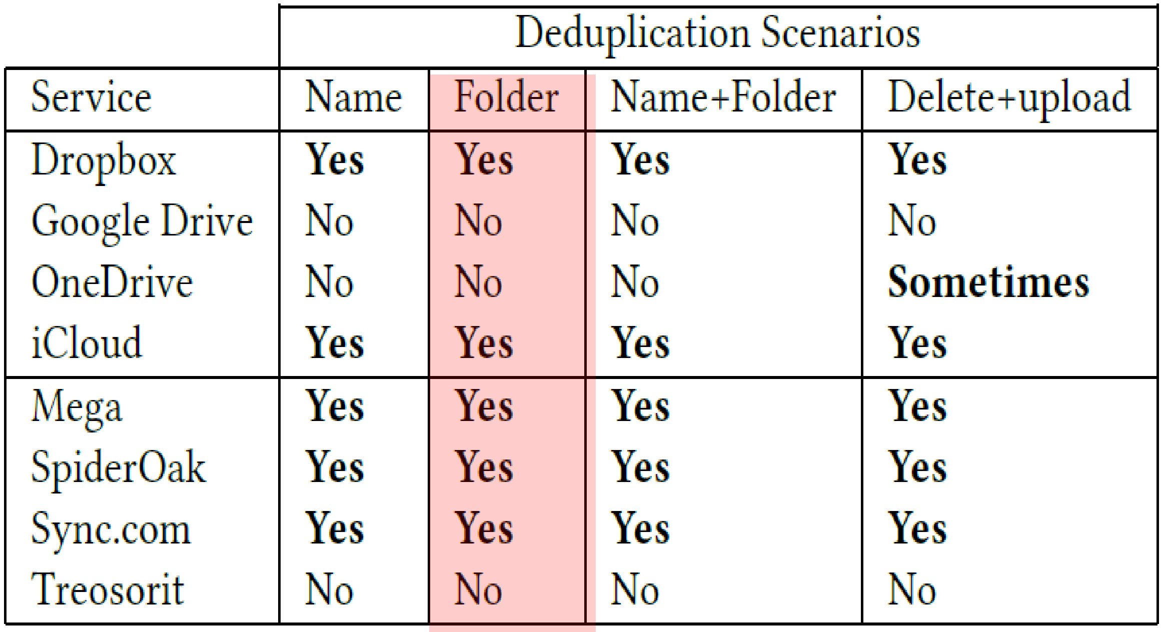
- Store two files with identical content
  - Different file names
  - Different folders
  - Different file name and folder
  - By deleting the file and then re-uploading it



	Deduplication Scenarios			
Service	Name	Folder	Name+Folder	Delete+upload
Dropbox	**Yes**	**Yes**	**Yes**	**Yes**
Google Drive	No	No	No	No
OneDrive	No	No	No	**Sometimes**
iCloud	**Yes**	**Yes**	**Yes**	**Yes**
Mega	**Yes**	**Yes**	**Yes**	**Yes**
SpiderOak	**Yes**	**Yes**	**Yes**	**Yes**
Sync.com	**Yes**	**Yes**	**Yes**	**Yes**
Treosorit	No	No	No	No

Feature 2: Deduplication

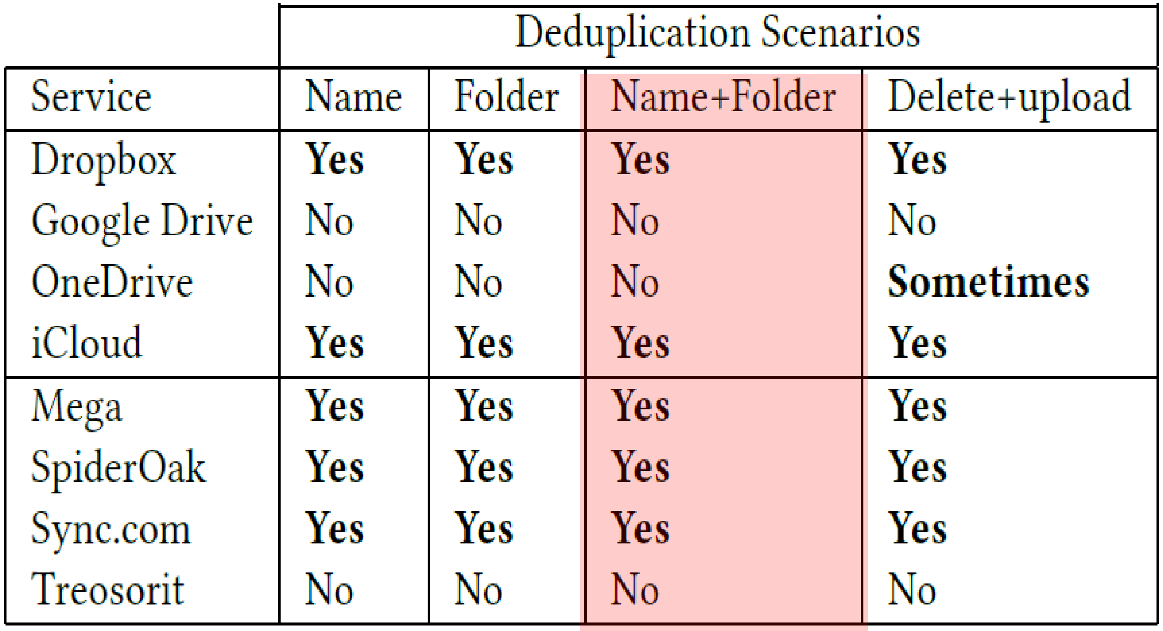
- Store two files with identical content
  - Different file names
  - Different folders
  - Different file name and folder
  - By deleting the file and then re-uploading it



	Deduplication Scenarios			
Service	Name	Folder	Name+Folder	Delete+upload
Dropbox	**Yes**	**Yes**	**Yes**	**Yes**
Google Drive	No	No	No	No
OneDrive	No	No	No	**Sometimes**
iCloud	**Yes**	**Yes**	**Yes**	**Yes**
Mega	**Yes**	**Yes**	**Yes**	**Yes**
SpiderOak	**Yes**	**Yes**	**Yes**	**Yes**
Sync.com	**Yes**	**Yes**	**Yes**	**Yes**
Treosorit	No	No	No	No

Feature 2: Deduplication

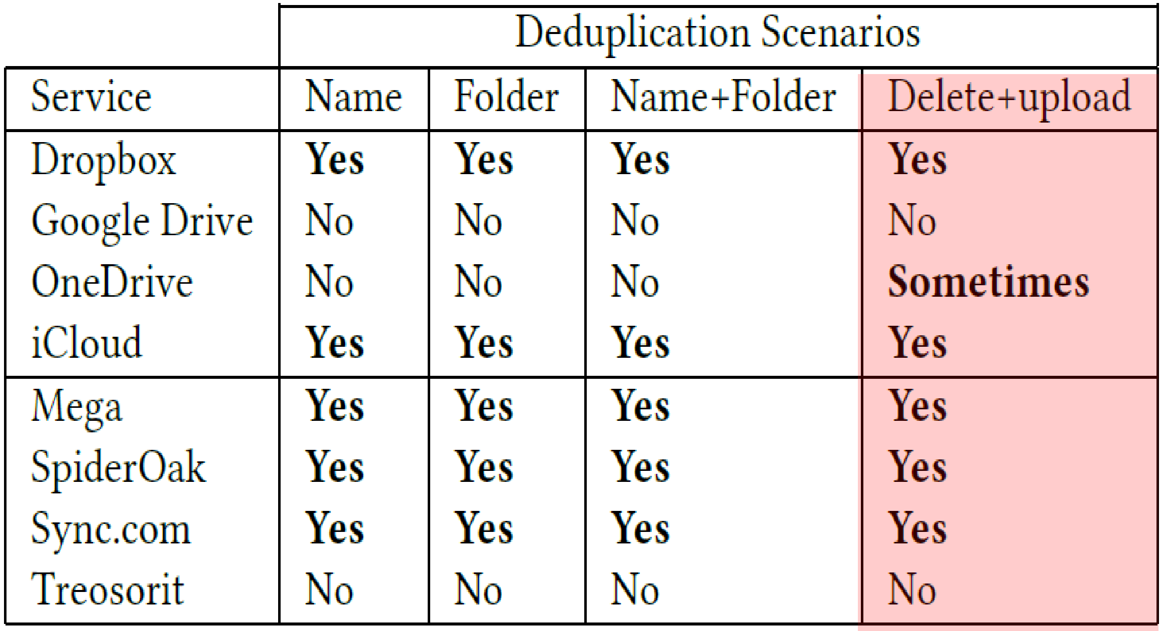
- Store two files with identical content
  - Different file names
  - Different folders
  - Different file name and folder
  - By deleting the file and then re-uploading it



	Deduplication Scenarios			
Service	Name	Folder	Name+Folder	Delete+upload
Dropbox	**Yes**	**Yes**	**Yes**	**Yes**
Google Drive	No	No	No	No
OneDrive	No	No	No	**Sometimes**
iCloud	**Yes**	**Yes**	**Yes**	**Yes**
Mega	**Yes**	**Yes**	**Yes**	**Yes**
SpiderOak	**Yes**	**Yes**	**Yes**	**Yes**
Sync.com	**Yes**	**Yes**	**Yes**	**Yes**
Treosorit	No	No	No	No

Feature 2: Deduplication

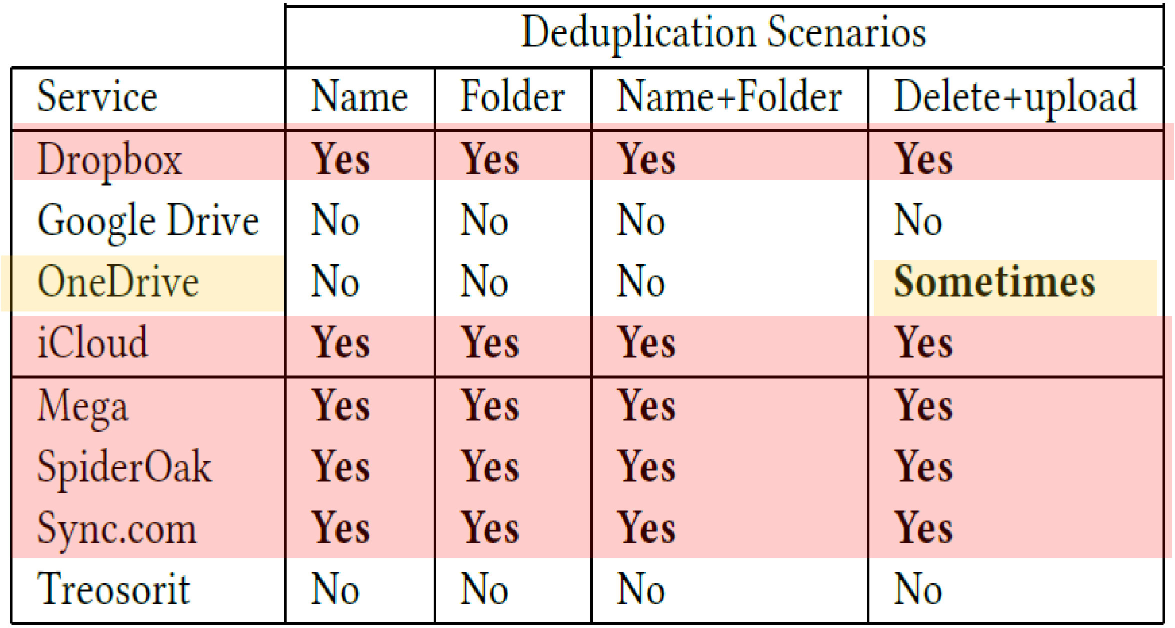
- Store two files with identical content
  - Different file names
  - Different folders
  - Different file name and folder
  - By deleting the file and then re-uploading it



	Deduplication Scenarios			
Service	Name	Folder	Name+Folder	Delete+upload
Dropbox	**Yes**	**Yes**	**Yes**	**Yes**
Google Drive	No	No	No	No
OneDrive	No	No	No	**Sometimes**
iCloud	**Yes**	**Yes**	**Yes**	**Yes**
Mega	**Yes**	**Yes**	**Yes**	**Yes**
SpiderOak	**Yes**	**Yes**	**Yes**	**Yes**
Sync.com	**Yes**	**Yes**	**Yes**	**Yes**
Treosorit	No	No	No	No

Feature 2: Deduplication

- Store two files with identical content
  - Different file names
  - Different folders
  - Different file name and folder
  - By deleting the file and then re-uploading it



	Deduplication Scenarios			
Service	Name	Folder	Name+Folder	Delete+upload
Dropbox	**Yes**	**Yes**	**Yes**	**Yes**
Google Drive	No	No	No	No
OneDrive	No	No	No	**Sometimes**
iCloud	**Yes**	**Yes**	**Yes**	**Yes**
Mega	**Yes**	**Yes**	**Yes**	**Yes**
SpiderOak	**Yes**	**Yes**	**Yes**	**Yes**
Sync.com	**Yes**	**Yes**	**Yes**	**Yes**
Treosorit	No	No	No	No

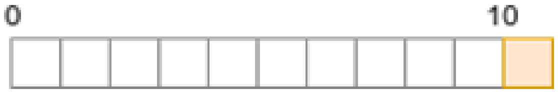
Feature 3: Delta encoding

Test method

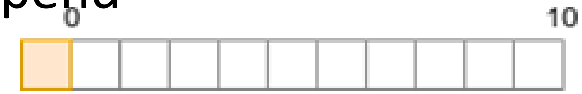
- Make sequence of changes
- Measure size of updates (full vs part)

File modifications considered

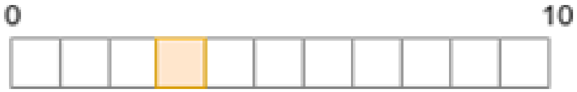
- Append



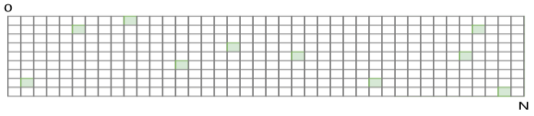
- Prepend



- Insert



- N random byte changes



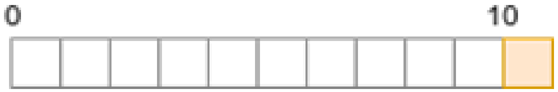
Feature 3: Delta encoding

Test method

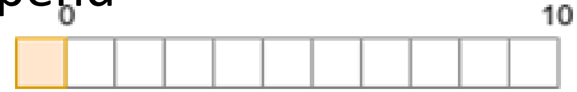
- Make sequence of changes
- Measure size of updates (full vs part)

File modifications considered

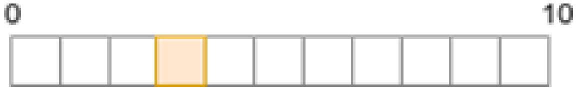
- Append



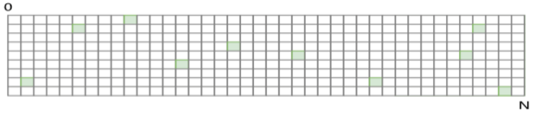
- Prepend



- Insert



- N random byte changes



	Yes	No
Non-CSE	Dropbox iCloud	Google Drive OneDrive
CSE	SPIDEROAK	sync.com MEGA tresorit

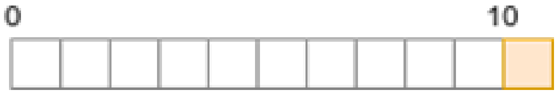
Feature 3: Delta encoding

Test method

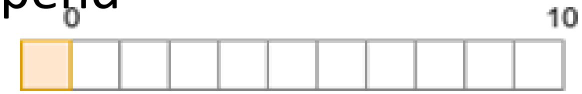
- Make sequence of changes
- Measure size of updates (full vs part)

File modifications considered

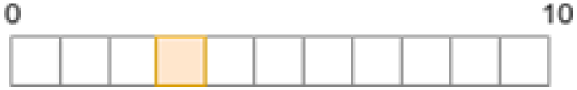
- Append



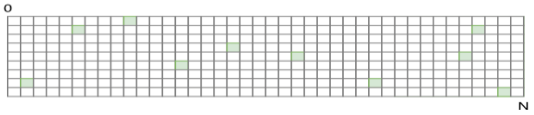
- Prepend



- Insert

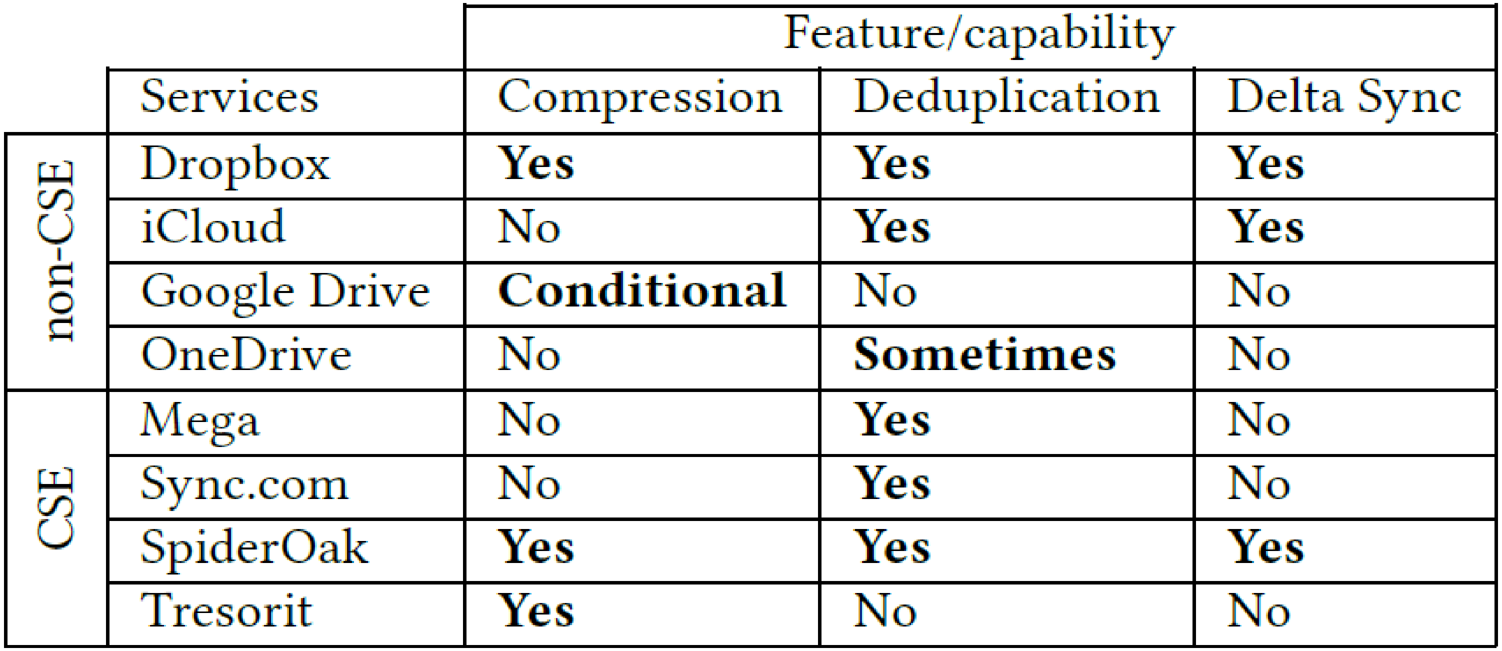


- N random byte changes



	Yes	No
Non-CSE	Dropbox iCloud	Google Drive OneDrive
CSE	SPIDEROAK	sync.com MEGA tresorit

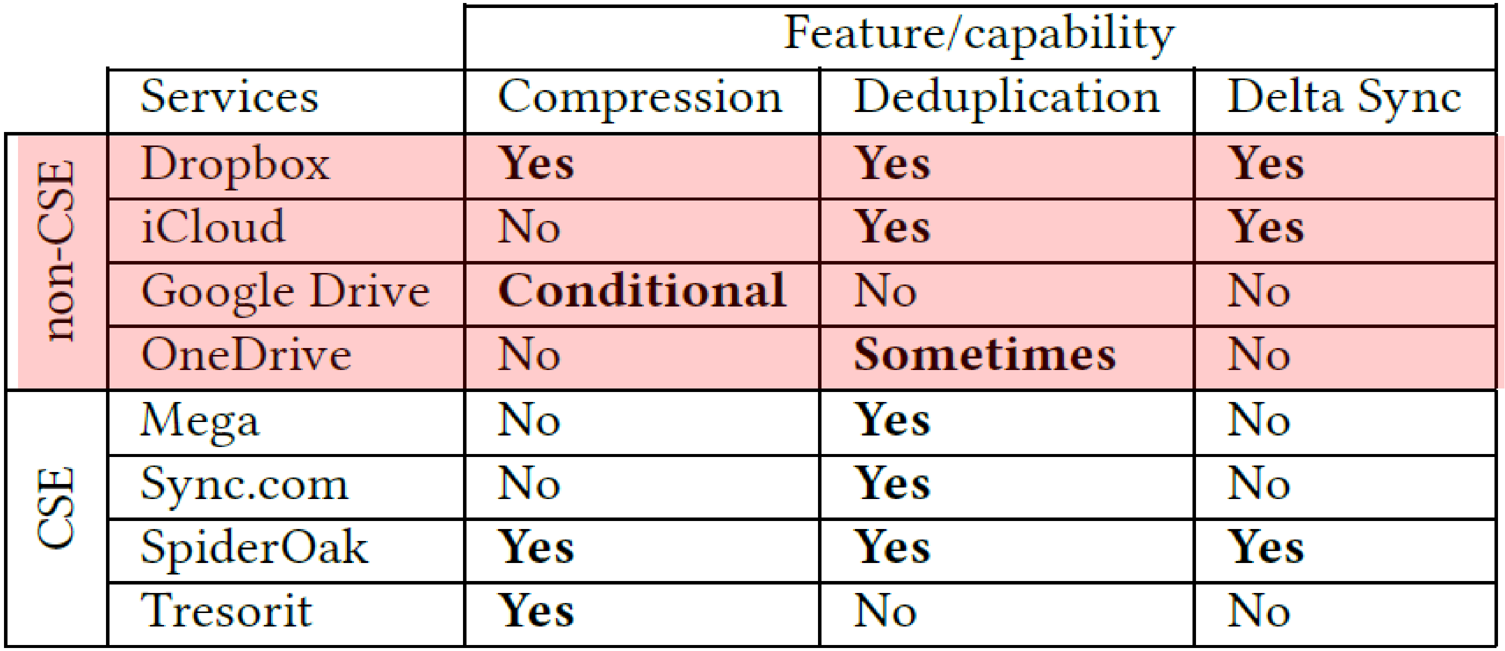
Feature summary



		Feature/capability		
	Services	Compression	Deduplication	Delta Sync
non-CSE	Dropbox	**Yes**	**Yes**	**Yes**
	iCloud	No	**Yes**	**Yes**
	Google Drive	**Conditional**	No	No
	OneDrive	No	**Sometimes**	No
CSE	Mega	No	**Yes**	No
	Sync.com	No	**Yes**	No
	SpiderOak	**Yes**	**Yes**	**Yes**
	Tresorit	**Yes**	No	No

- No clear difference between CSE vs non-CSEs
- Instead, large variations within each group

Feature summary



		Feature/capability		
	Services	Compression	Deduplication	Delta Sync
non-CSE	Dropbox	**Yes**	**Yes**	**Yes**
	iCloud	No	**Yes**	**Yes**
	Google Drive	**Conditional**	No	No
	OneDrive	No	**Sometimes**	No
CSE	Mega	No	**Yes**	No
	Sync.com	No	**Yes**	No
	SpiderOak	**Yes**	**Yes**	**Yes**
	Tresorit	**Yes**	No	No

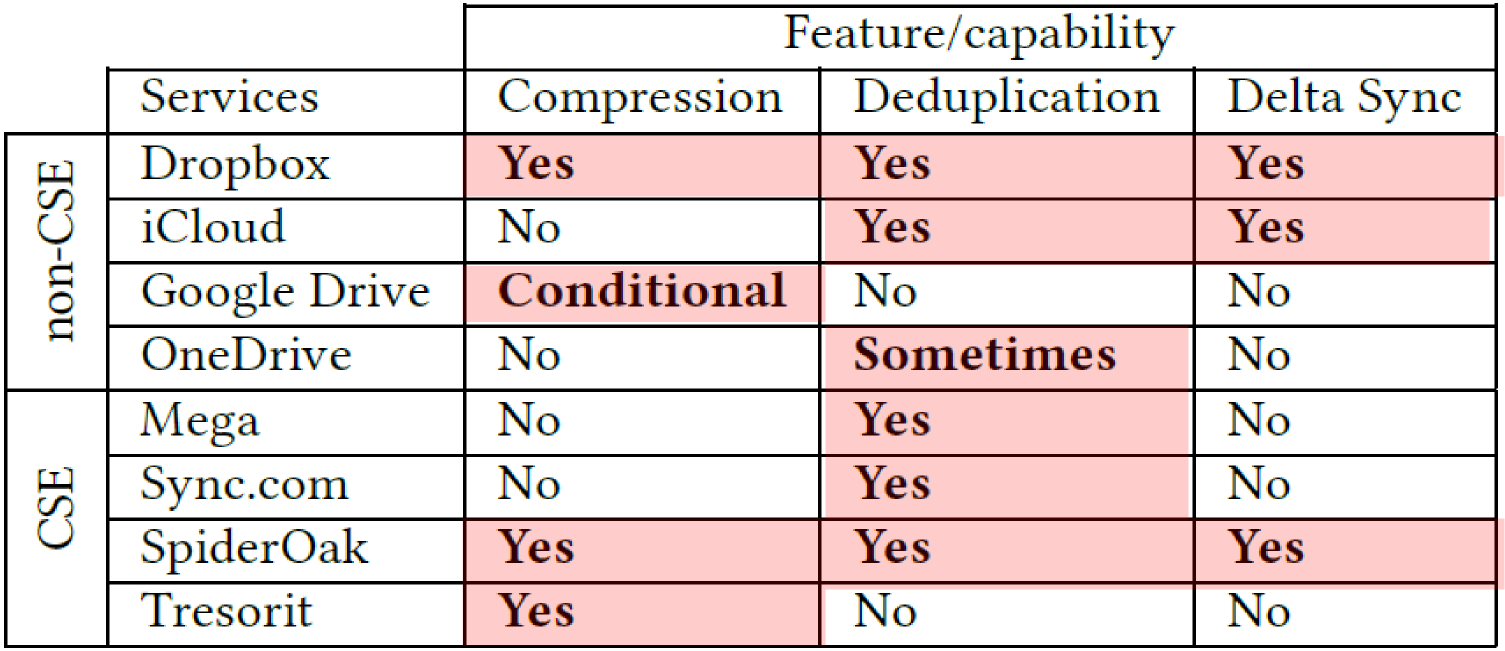
- No clear difference between CSE vs non-CSEs
- Instead, large variations within each group

Feature summary

		Feature/capability		
	Services	Compression	Deduplication	Delta Sync
non-CSE	Dropbox	**Yes**	**Yes**	**Yes**
	iCloud	No	**Yes**	**Yes**
	Google Drive	**Conditional**	No	No
	OneDrive	No	**Sometimes**	No
CSE	Mega	No	**Yes**	No
	Sync.com	No	**Yes**	No
	SpiderOak	**Yes**	**Yes**	**Yes**
	Tresorit	**Yes**	No	No

- No clear difference between CSE vs non-CSEs
- Instead, large variations within each group

Feature summary



		Feature/capability		
	Services	Compression	Deduplication	Delta Sync
non-CSE	Dropbox	**Yes**	**Yes**	**Yes**
	iCloud	No	**Yes**	**Yes**
	Google Drive	**Conditional**	No	No
	OneDrive	No	**Sometimes**	No
CSE	Mega	No	**Yes**	No
	Sync.com	No	**Yes**	No
	SpiderOak	**Yes**	**Yes**	**Yes**
	Tresorit	**Yes**	No	No

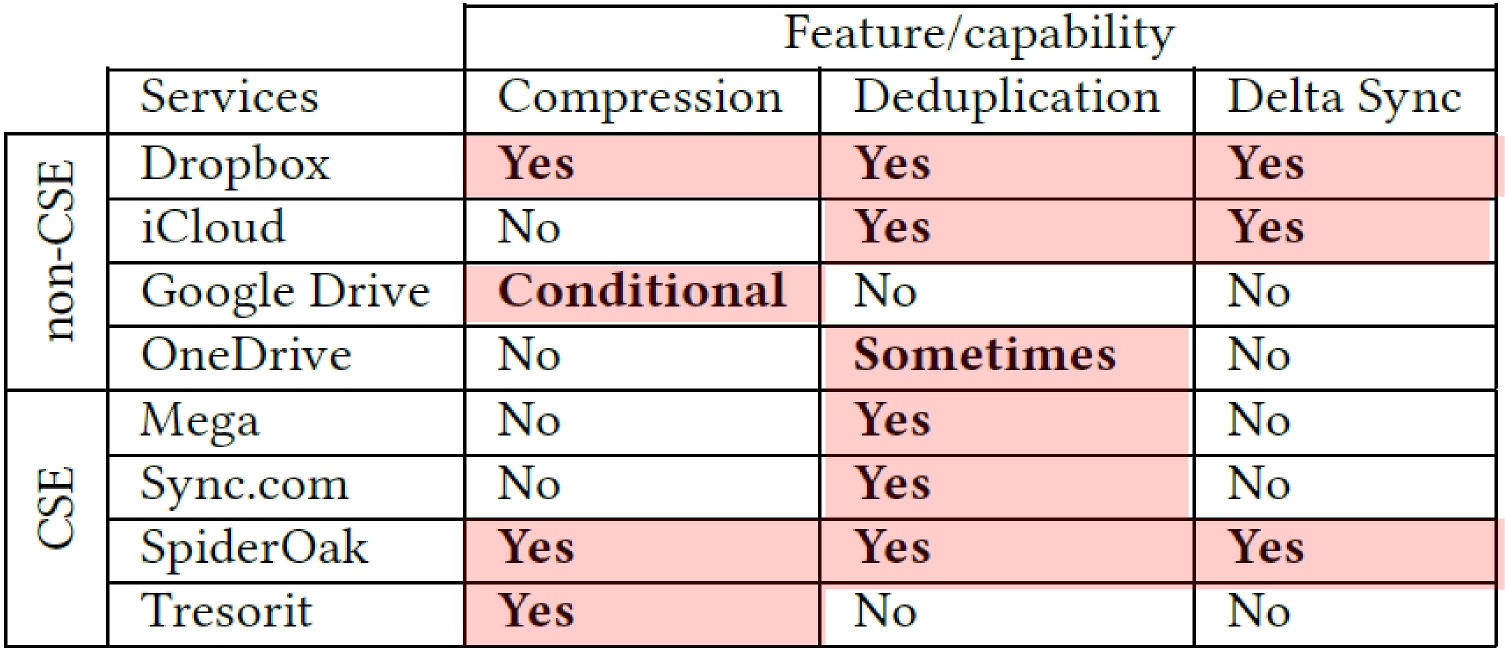
- No clear difference between CSE vs non-CSEs
- Instead, large variations within each group

Feature summary

		Feature/capability		
	Services	Compression	Deduplication	Delta Sync
non-CSE	Dropbox	**Yes**	**Yes**	**Yes**
	iCloud	No	**Yes**	**Yes**
	Google Drive	**Conditional**	No	No
	OneDrive	No	**Sometimes**	No
CSE	Mega	No	**Yes**	No
	Sync.com	No	**Yes**	No
	SpiderOak	**Yes**	**Yes**	**Yes**
	Tresorit	**Yes**	No	No

- No clear difference between CSE vs non-CSEs
- Instead, large variations within each group
- Only Dropbox (non-CSE) and SpiderOak (CSE) has all three features

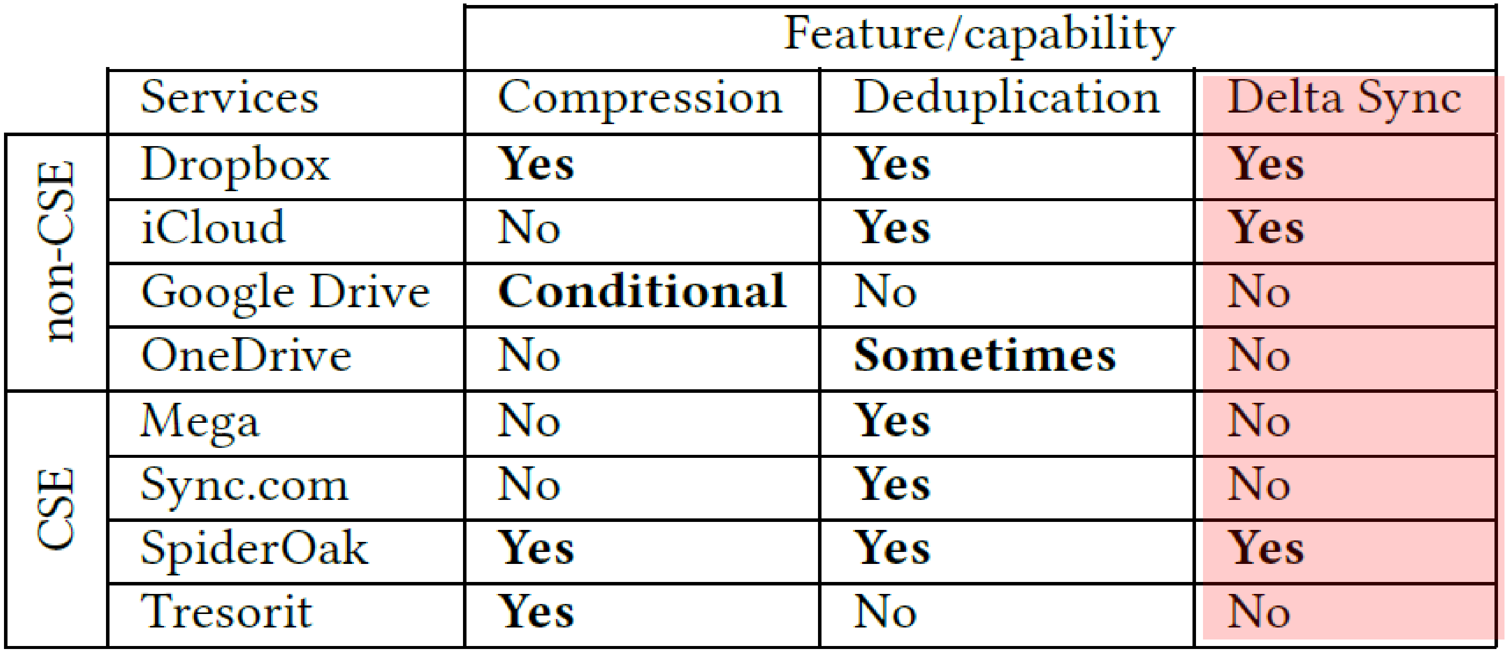
Feature summary



		Feature/capability		
	Services	Compression	Deduplication	Delta Sync
non-CSE	Dropbox	Yes	Yes	Yes
	iCloud	No	Yes	Yes
	Google Drive	Conditional	No	No
	OneDrive	No	Sometimes	No
CSE	Mega	No	Yes	No
	Sync.com	No	Yes	No
	SpiderOak	Yes	Yes	Yes
	Tresorit	Yes	No	No

- No clear difference between CSE vs non-CSEs
- Instead, large variations within each group
- Only Dropbox (non-CSE) and SpiderOak (CSE) has all three features
- All services implement at least some feature (but different)

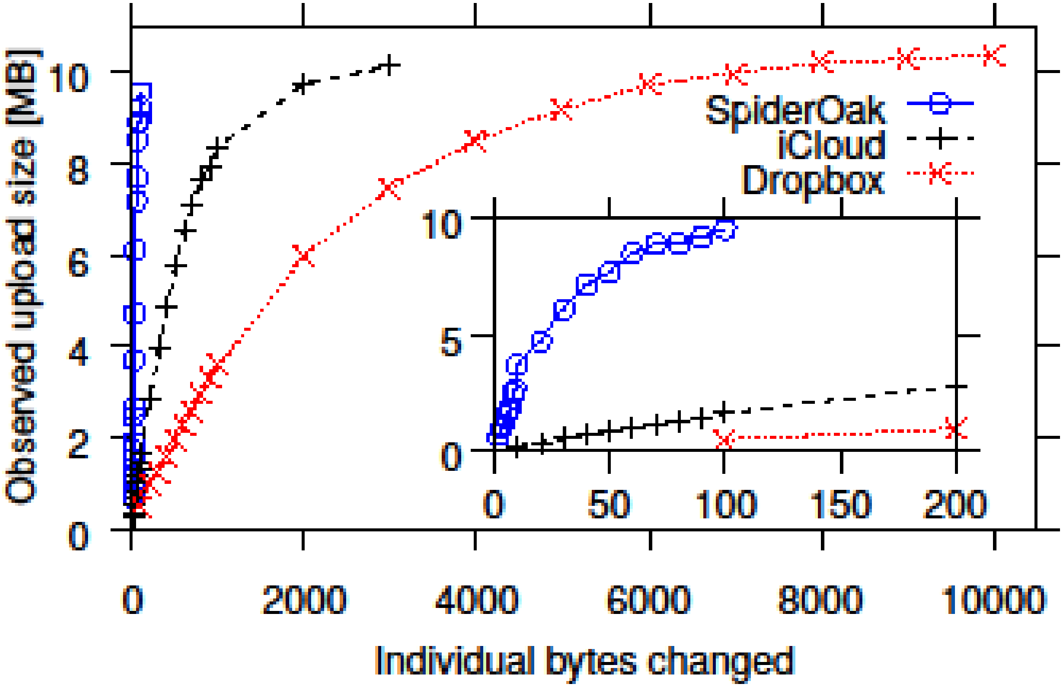
Feature summary

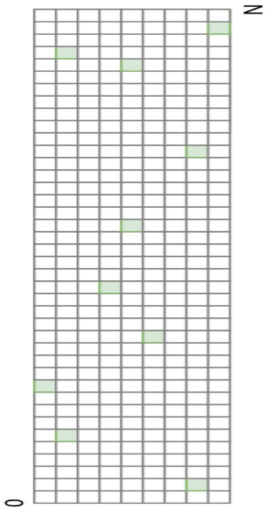


		Feature/capability		
	Services	Compression	Deduplication	Delta Sync
non-CSE	Dropbox	**Yes**	**Yes**	**Yes**
	iCloud	No	**Yes**	**Yes**
	Google Drive	**Conditional**	No	No
	OneDrive	No	**Sometimes**	No
CSE	Mega	No	**Yes**	No
	Sync.com	No	**Yes**	No
	SpiderOak	**Yes**	**Yes**	**Yes**
	Tresorit	**Yes**	No	No

- No clear difference between CSE vs non-CSEs
- Instead, large variations within each group
- Only Dropbox (non-CSE) and SpiderOak (CSE) has all three features
- All services implement at least some feature (but different)
- Furthermore: Delta encoding efficiency differ substantially …

Delta encoding efficiency …





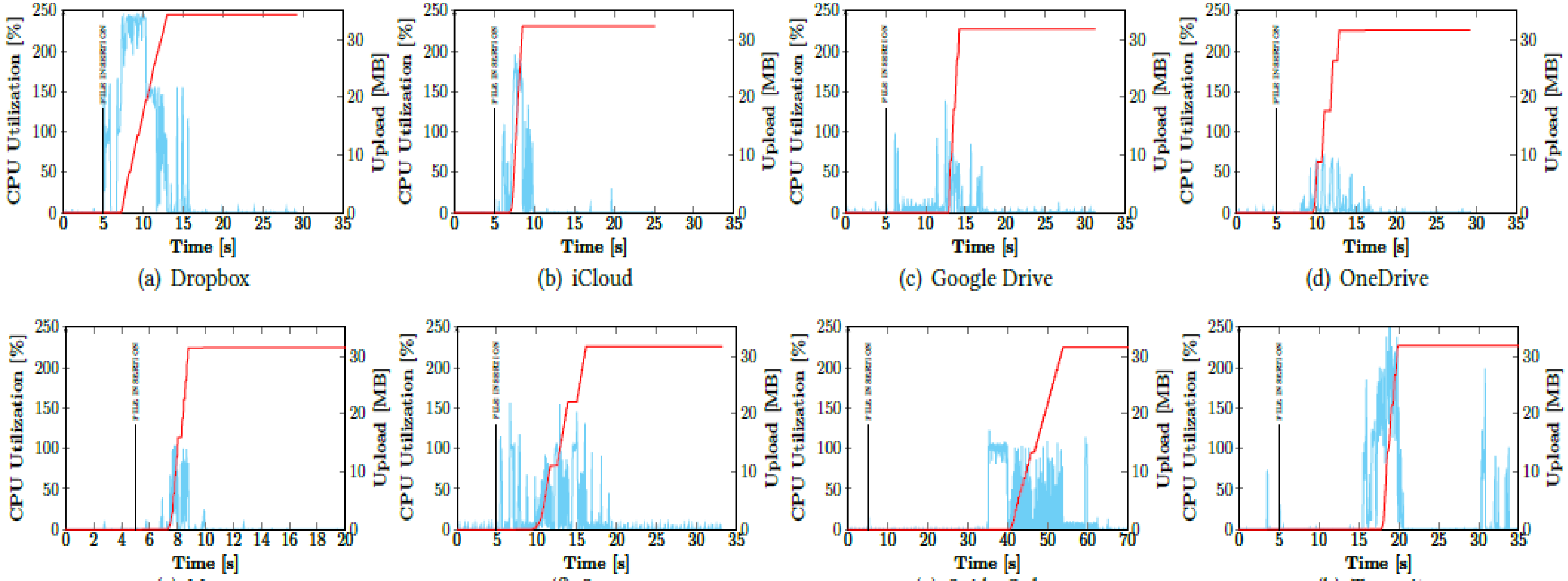
Large differences among service implementing (some) delta encoding

- SpiderOak (CSE) performs much worse than iCloud (non-CSE) and Dropbox (non-CSE)

Note: More detailed delta-encoding analysis and optimized delta encoding policies for CSE in our IEEE CloudCom 2019 paper (next week)

Performance evaluation

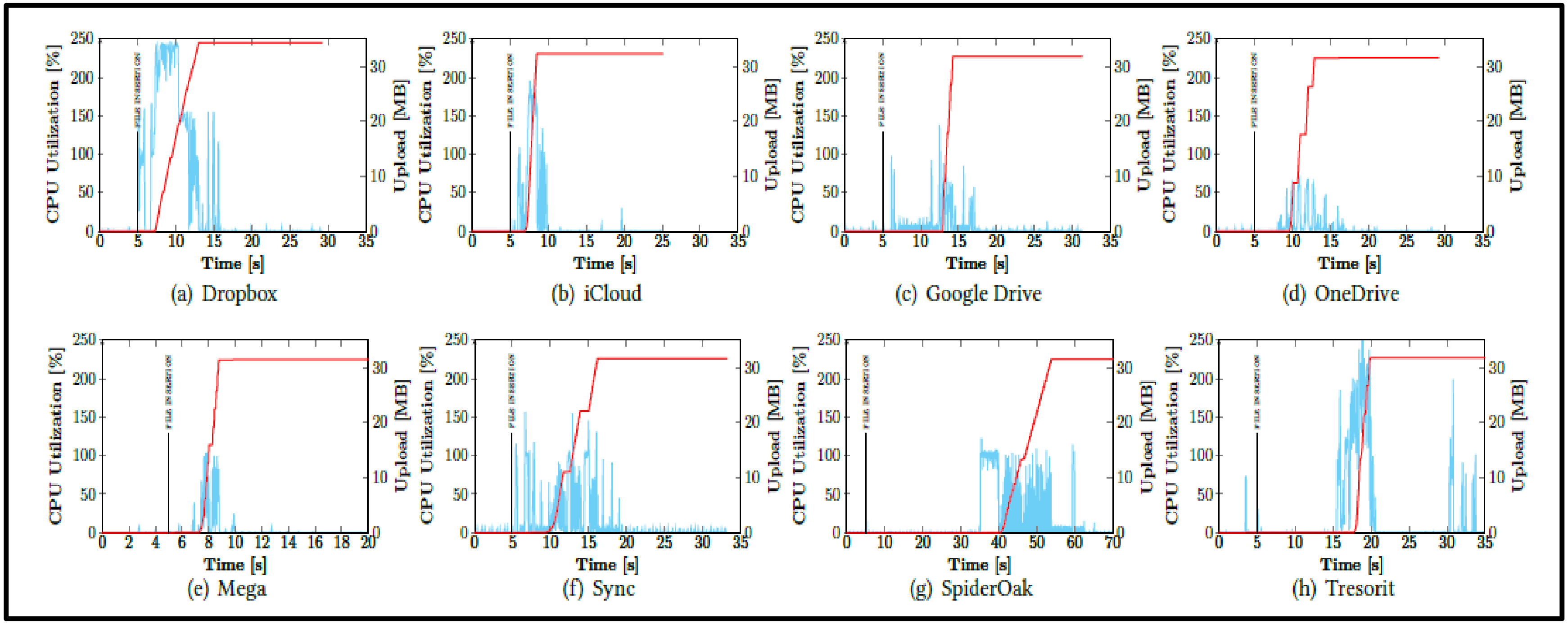
Performance: CPU



(a) Dropbox (b) iCloud (c) Google Drive (d) OneDrive

(e) Mega (f) Sync (g) SpiderOak (h) Tresorit

Performance: CPU



(a) Dropbox

(b) iCloud

(c) Google Drive

(d) OneDrive

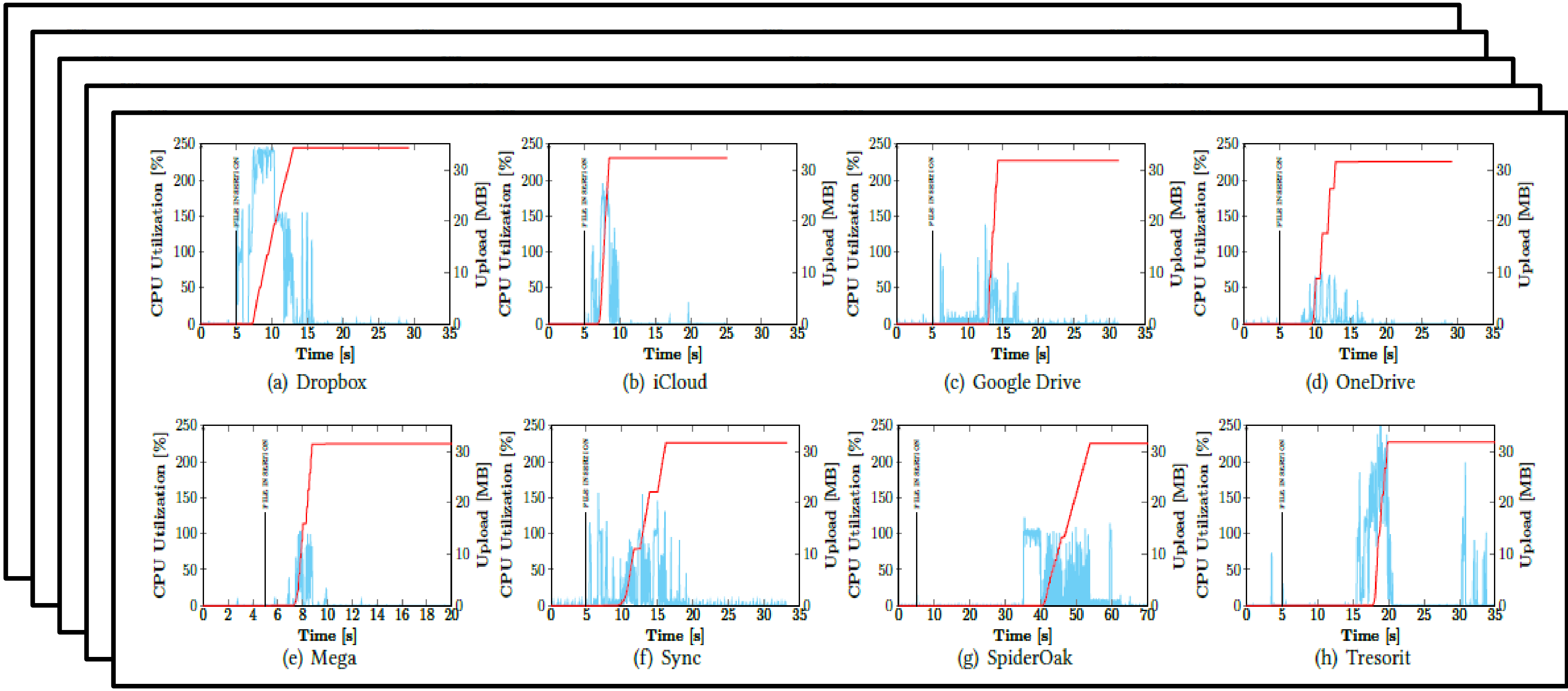
(e) Mega

(f) Sync

(g) SpiderOak

(h) Tresorit

Performance: CPU



(a) Dropbox

(b) iCloud

(c) Google Drive

(d) OneDrive

(e) Mega

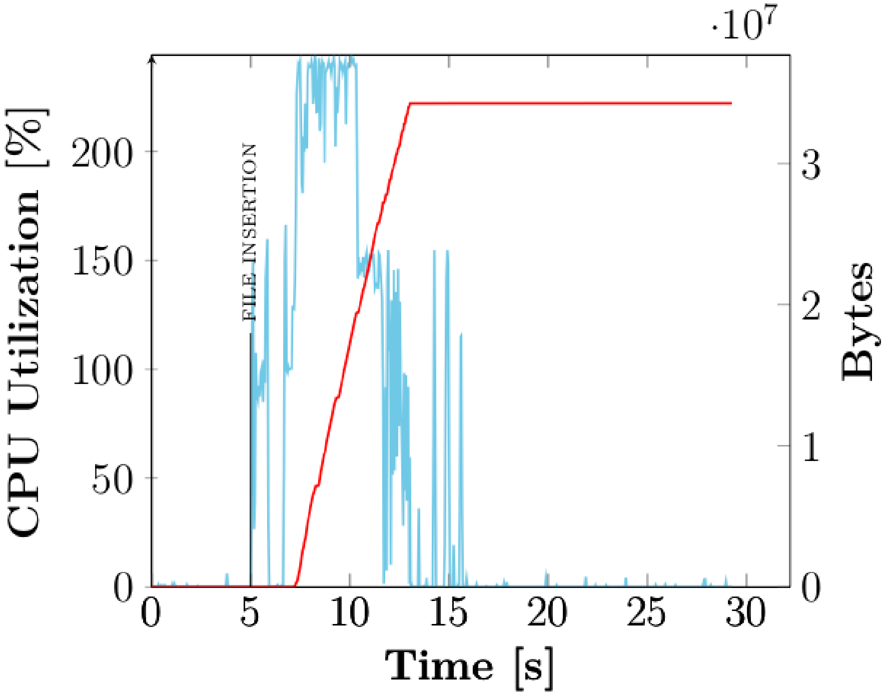
(f) Sync

(g) SpiderOak

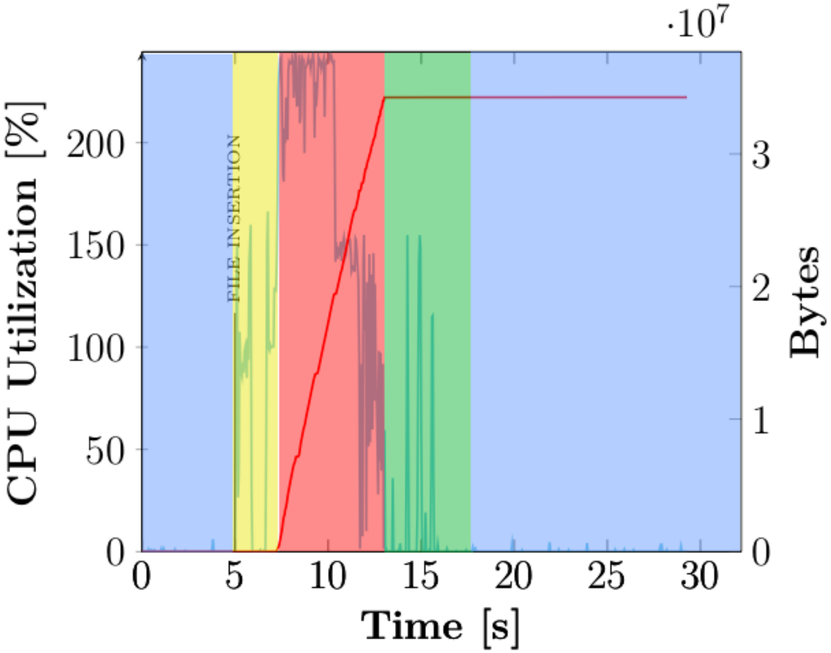
(h) Tresorit

×25

Performance: CPU



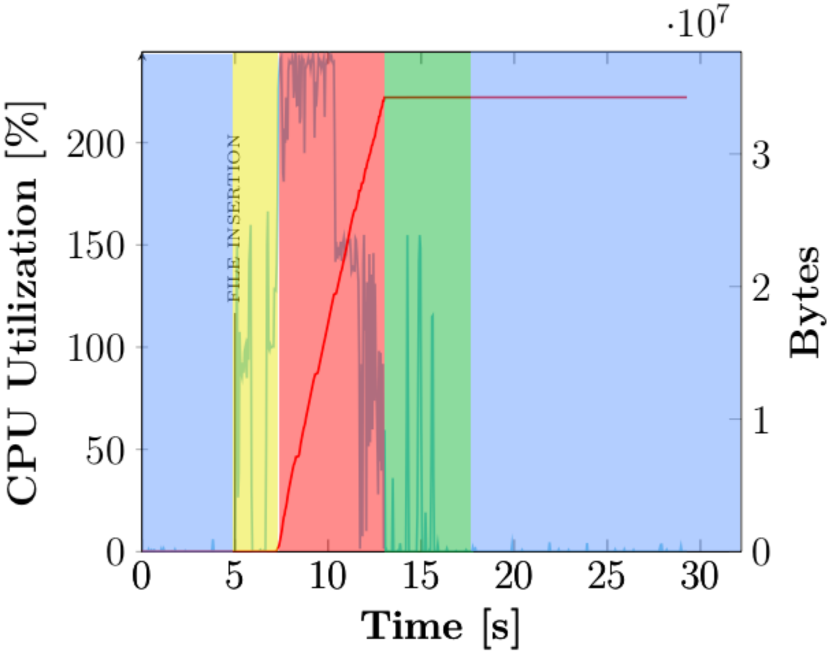
Performance: CPU

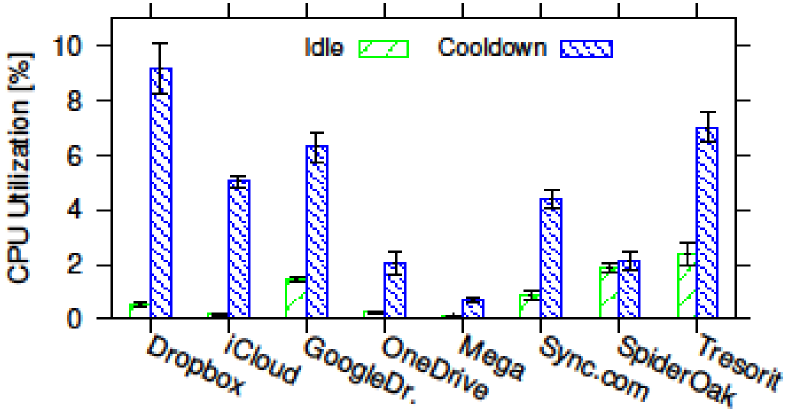


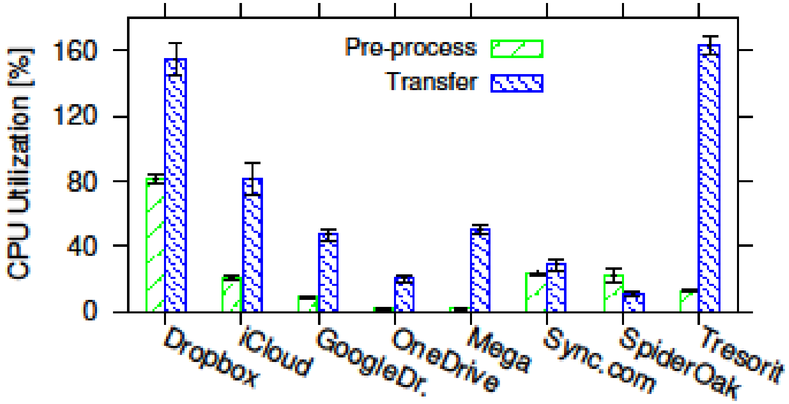
Synchronization phases

- Idle
- Pre-processing
- Transfer
- Cooldown

Performance: CPU



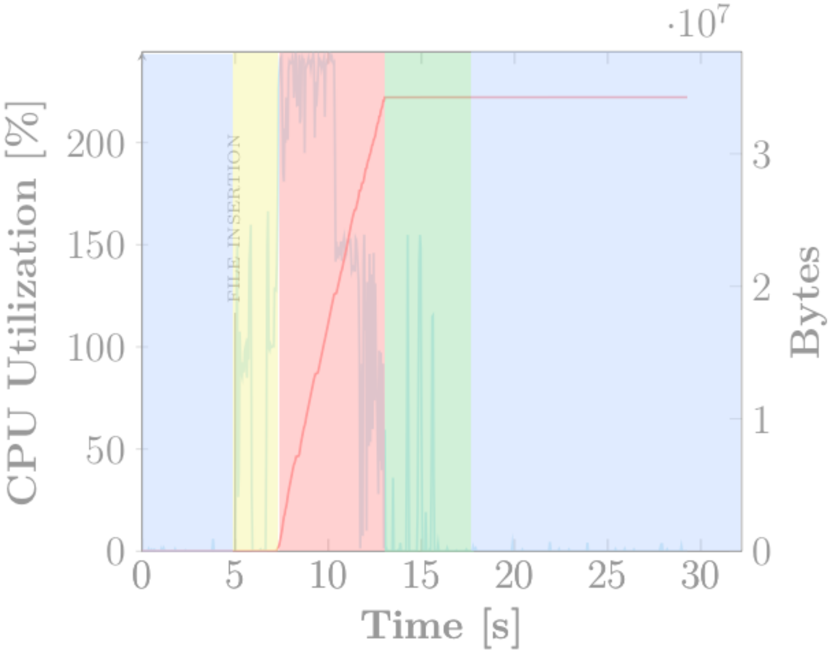


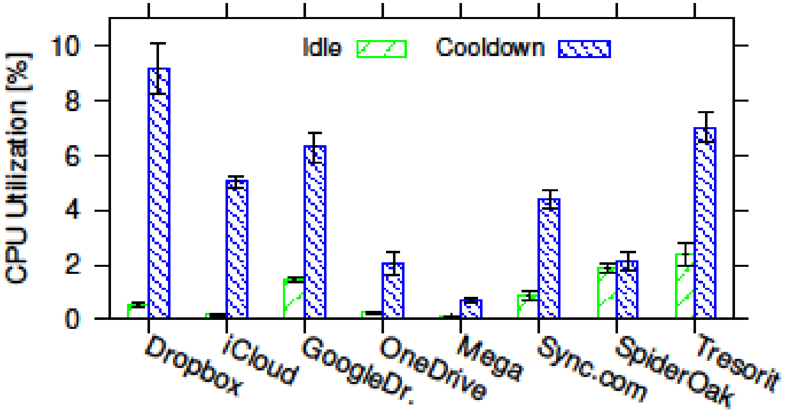


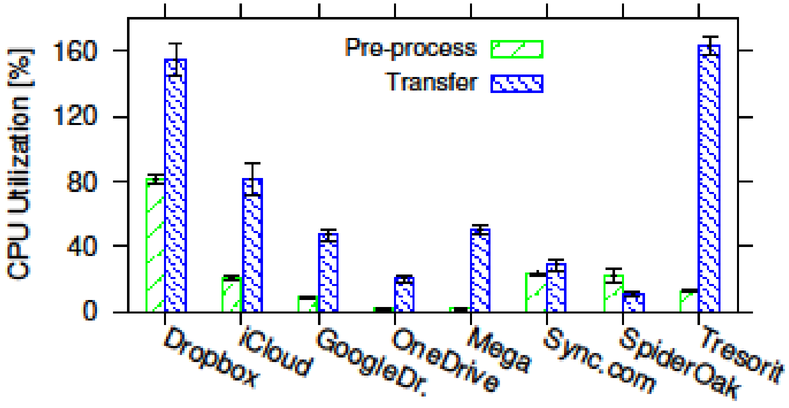
Synchronization phases

- Idle
- Pre-processing
- Transfer
- Cooldown

Performance: CPU



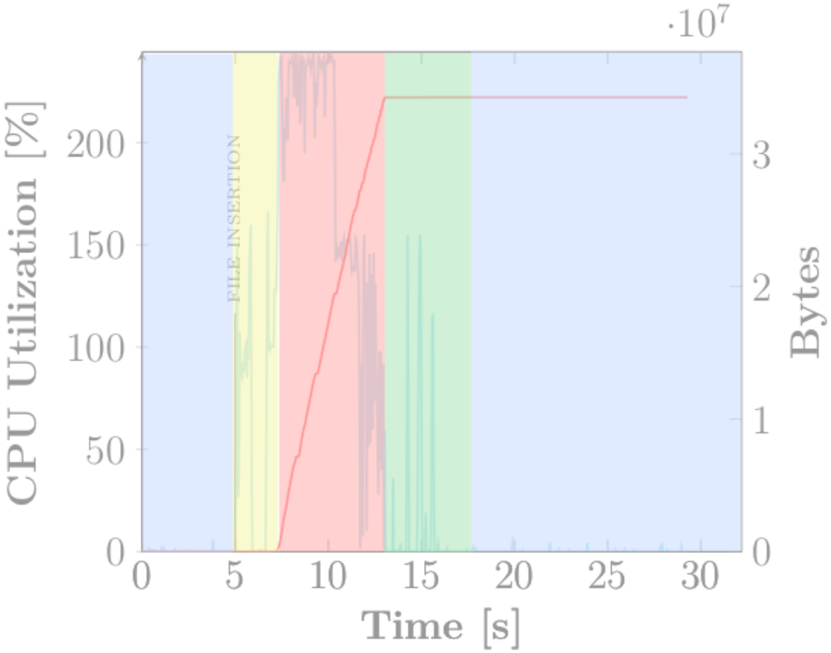




Synchronization phases

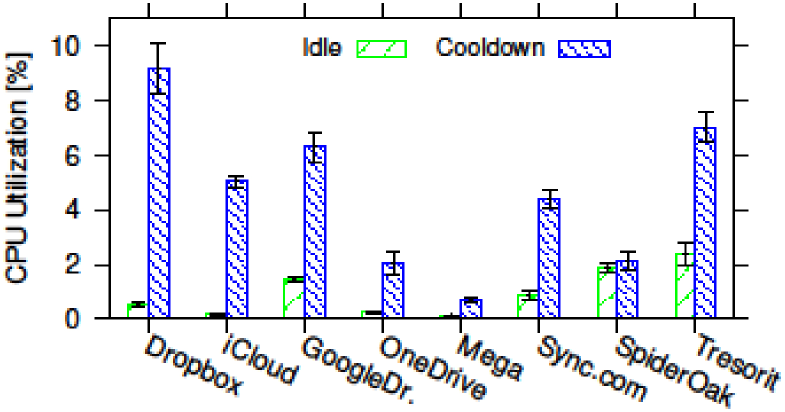
- Idle
- Pre-processing
- Transfer
- Cooldown

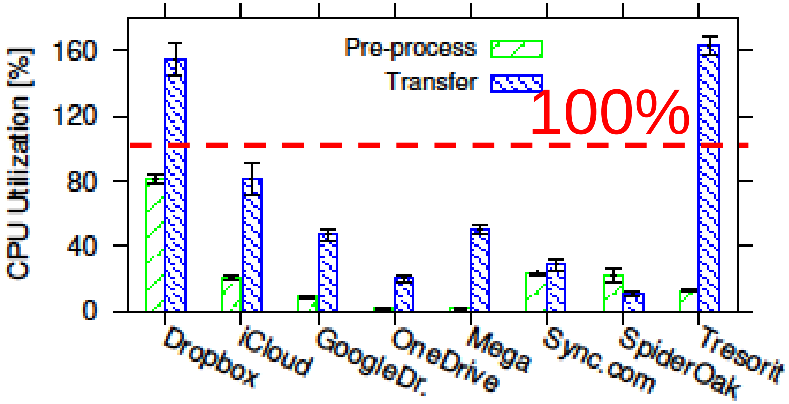
Performance: CPU



Synchronization phases

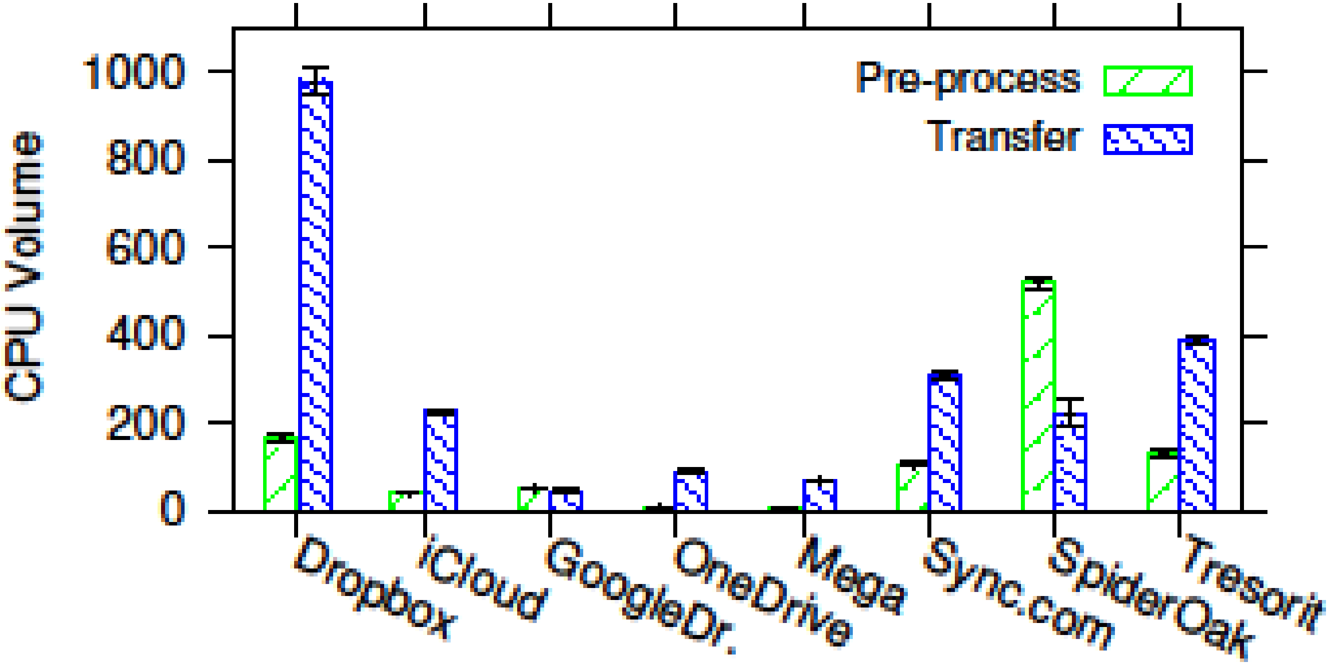
- Idle
- Pre-processing
- Transfer
- Cooldown





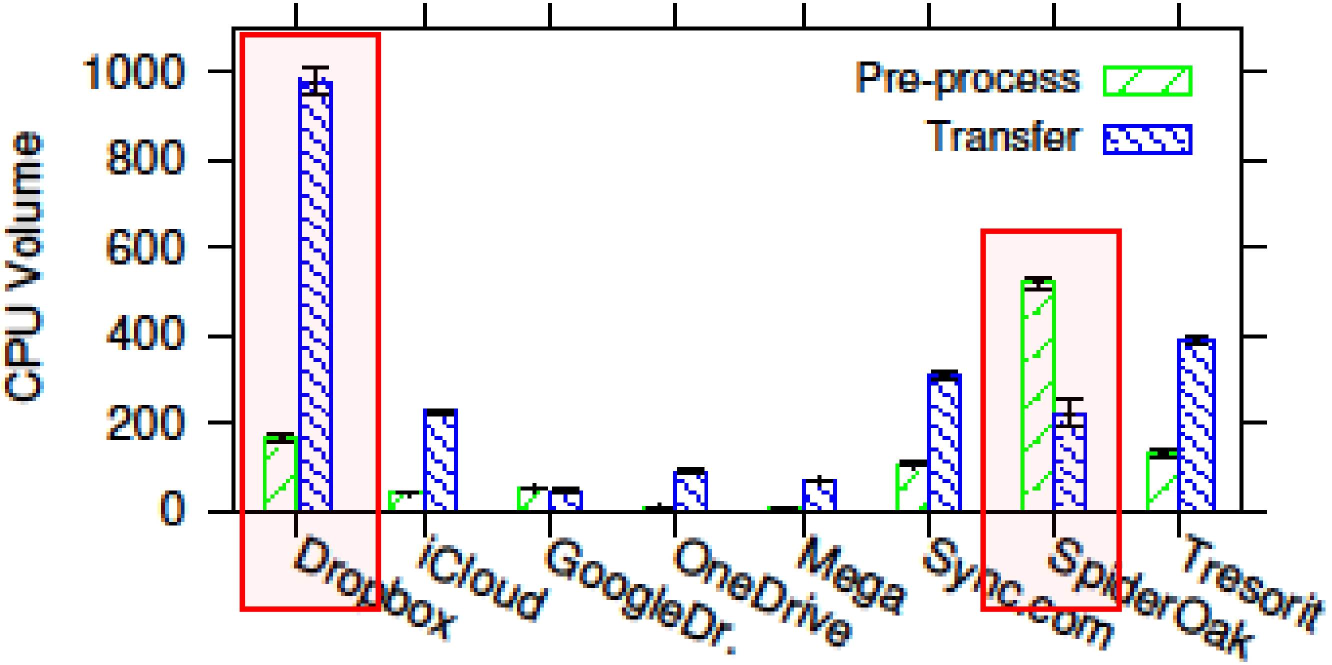
Note: Some values above 100%, due to multithreaded service using at least 2 cores

Performance: CPU



CPU Volume = (Mean “extra” CPU * Phase duration),
where “extra” is relative the “idle” baseline

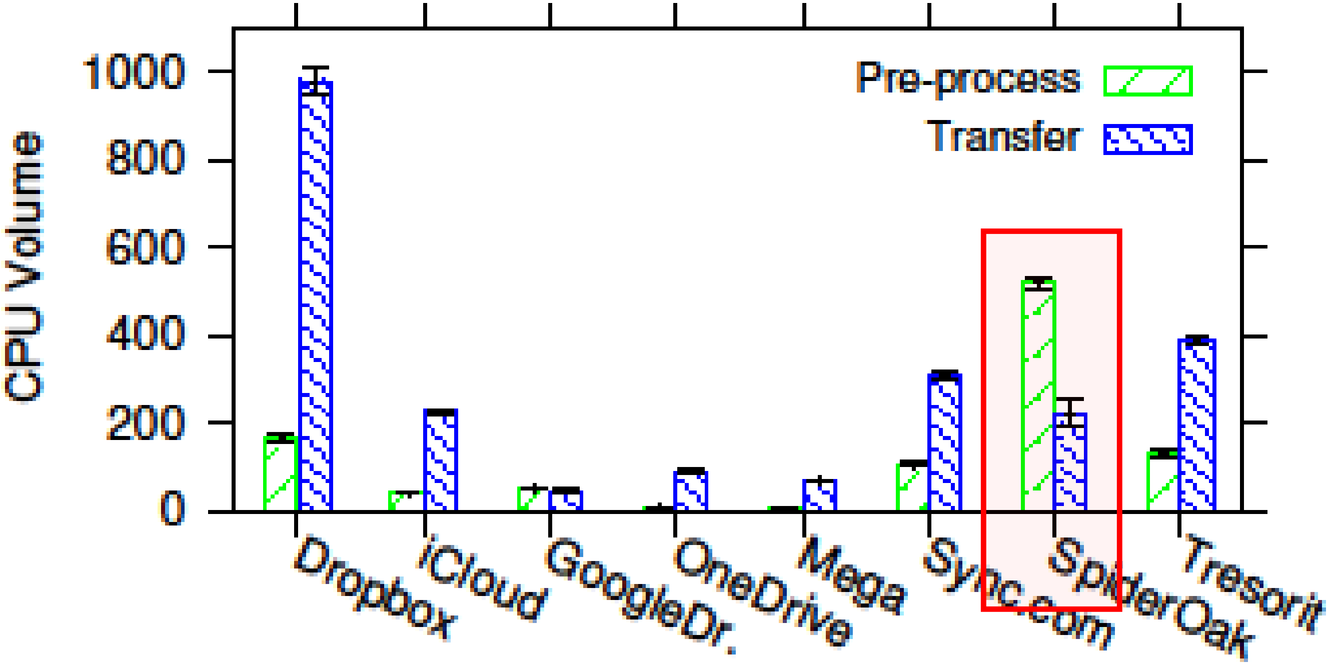
Performance: CPU



CPU Volume = (Mean “extra” CPU * Phase duration),
where “extra” is relative the “idle” baseline

- Highest among feature rich services (e.g., Dropbox and SpiderOak)

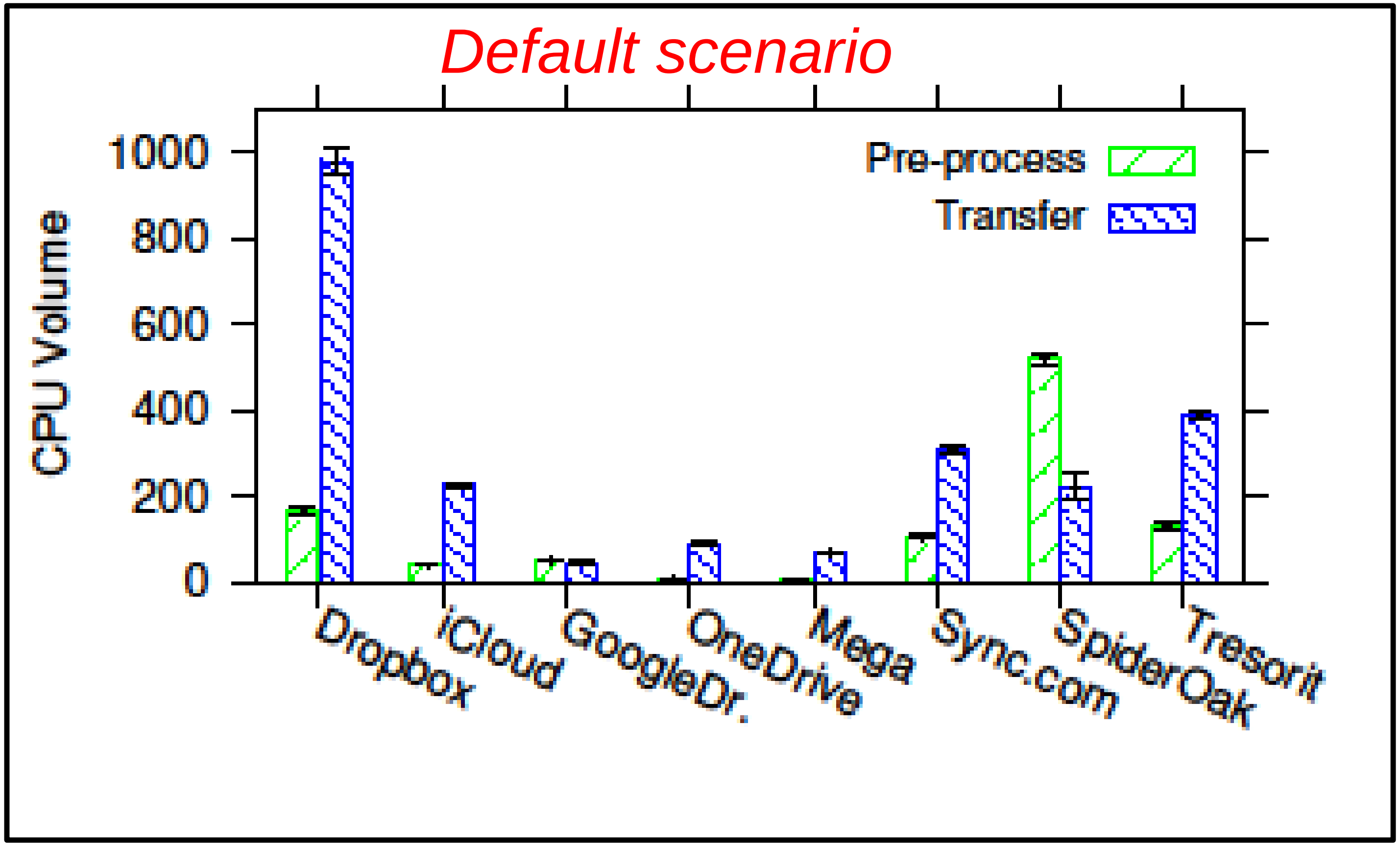
Performance: CPU



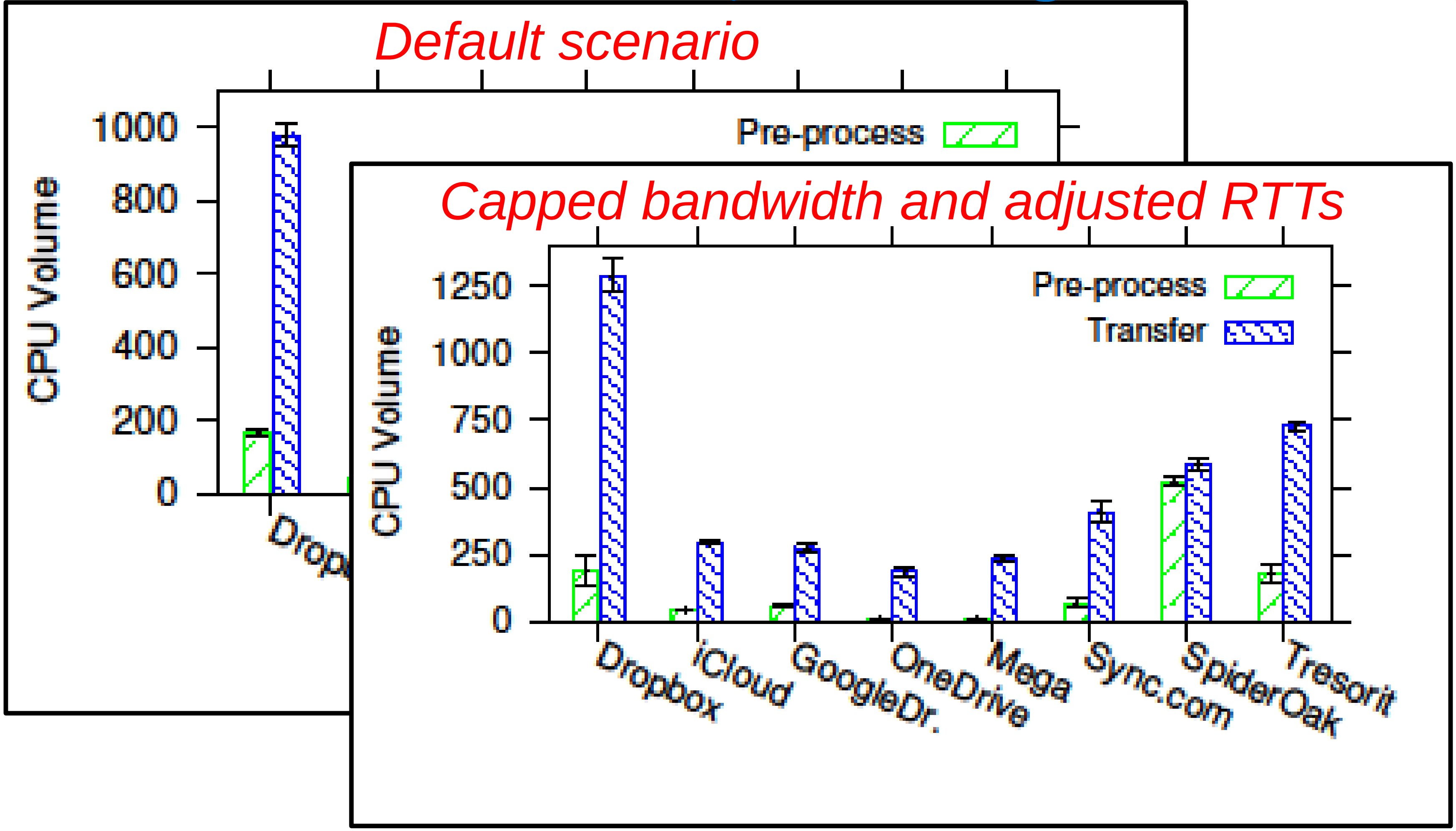
CPU Volume = (Mean "extra" CPU * Phase duration), where "extra" is relative the "idle" baseline

- Highest among feature rich services (e.g., Dropbox and SpiderOak)
- SpiderOak does most pre-processing (incl. storing copy to disk)
- Other services' CPU usage dominated by transfer

Performance: CPU



Performance: CPU (matching conditions)



Default scenario

Capped bandwidth and adjusted RTTs

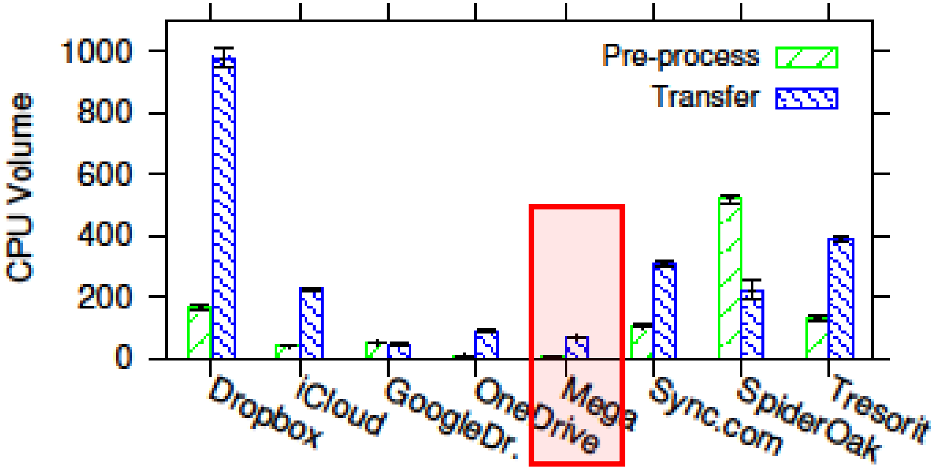
- Increase CPU volumes somewhat, but relative overheads remain …

Performance: HTTP vs HTTPS

MEGA

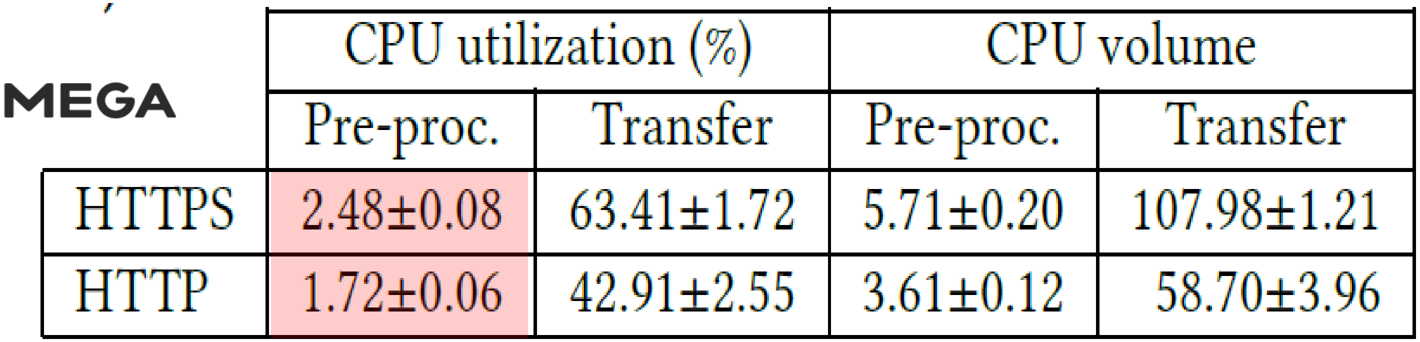
	CPU utilization (%)		CPU volume	
	Pre-proc.	Transfer	Pre-proc.	Transfer
HTTPS	2.48±0.08	63.41±1.72	5.71±0.20	107.98±1.21
HTTP	1.72±0.06	42.91±2.55	3.61±0.12	58.70±3.96

- Small extra HTTPS overhead compared to most other service



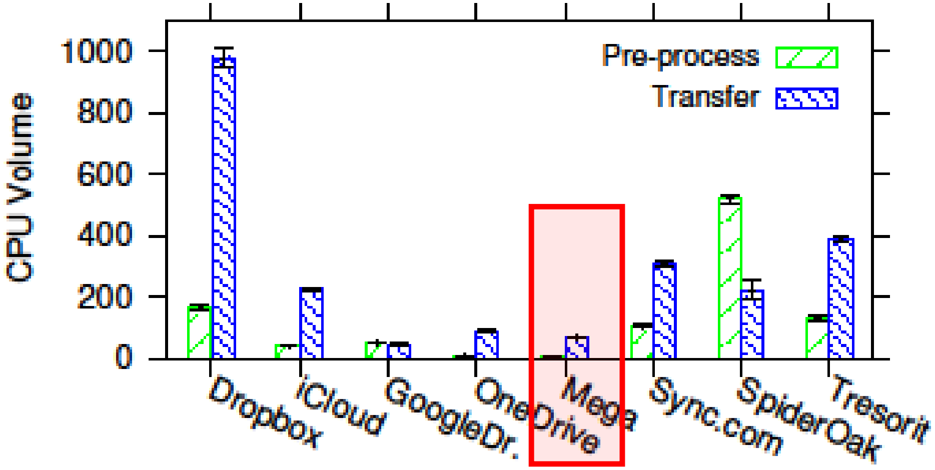
MEGA

Performance: HTTP vs HTTPS

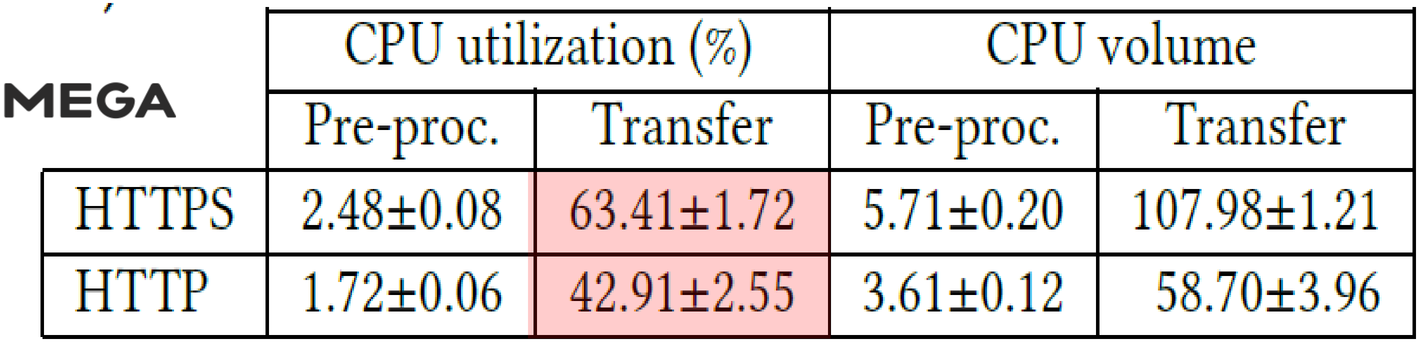


	CPU utilization (%)		CPU volume	
	Pre-proc.	Transfer	Pre-proc.	Transfer
HTTPS	2.48±0.08	63.41±1.72	5.71±0.20	107.98±1.21
HTTP	1.72±0.06	42.91±2.55	3.61±0.12	58.70±3.96

- Small extra HTTPS overhead compared to most other service

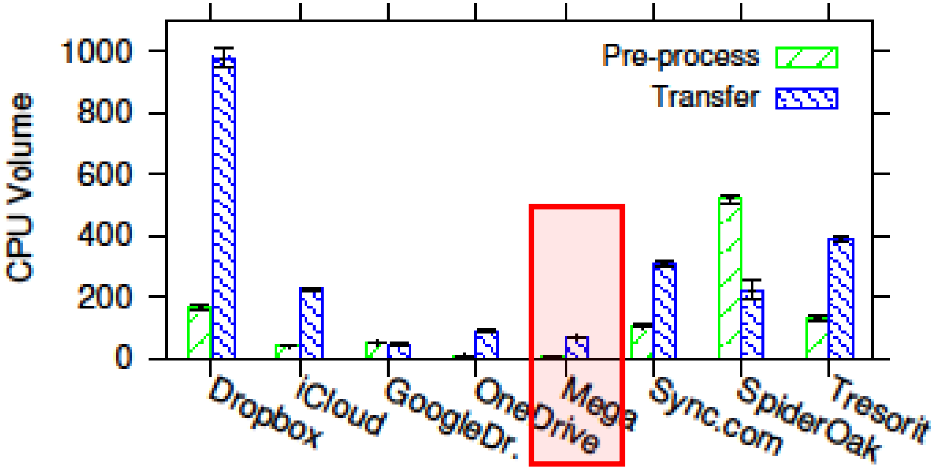


Performance: HTTP vs HTTPS

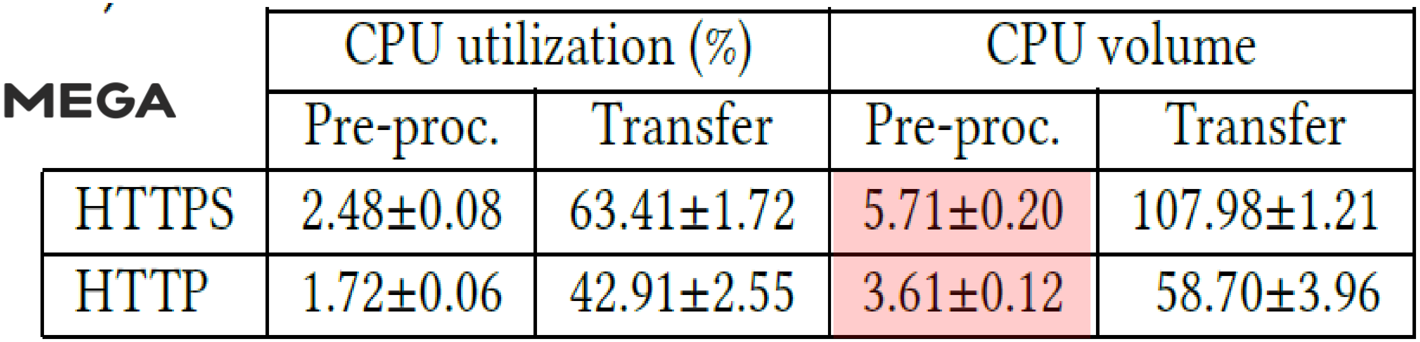


	CPU utilization (%)		CPU volume	
	Pre-proc.	Transfer	Pre-proc.	Transfer
HTTPS	2.48±0.08	63.41±1.72	5.71±0.20	107.98±1.21
HTTP	1.72±0.06	42.91±2.55	3.61±0.12	58.70±3.96

- Small extra HTTPS overhead compared to most other service

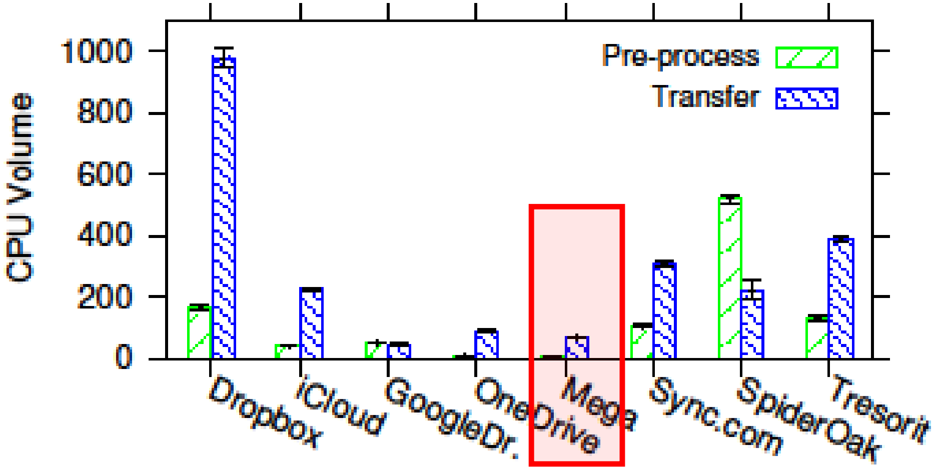


Performance: HTTP vs HTTPS

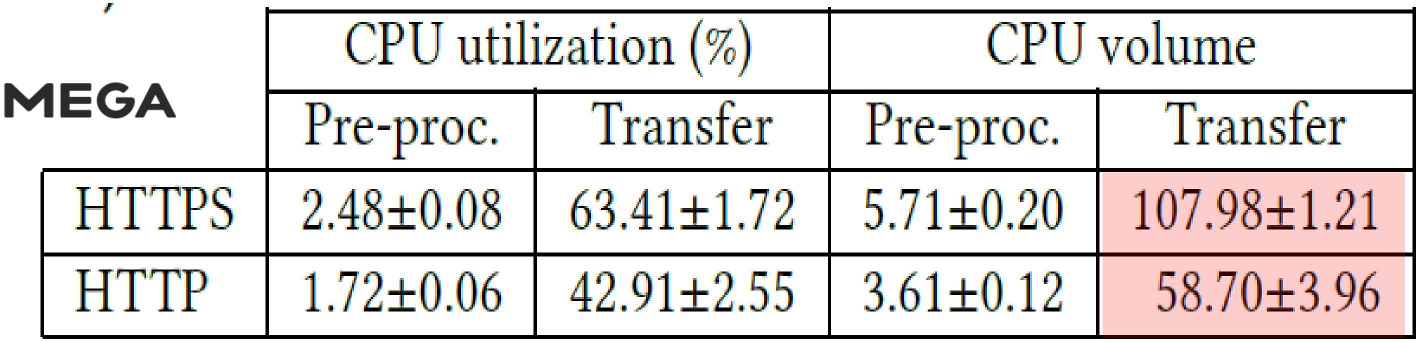


	CPU utilization (%)		CPU volume	
	Pre-proc.	Transfer	Pre-proc.	Transfer
HTTPS	2.48±0.08	63.41±1.72	5.71±0.20	107.98±1.21
HTTP	1.72±0.06	42.91±2.55	3.61±0.12	58.70±3.96

- Small extra HTTPS overhead compared to most other service

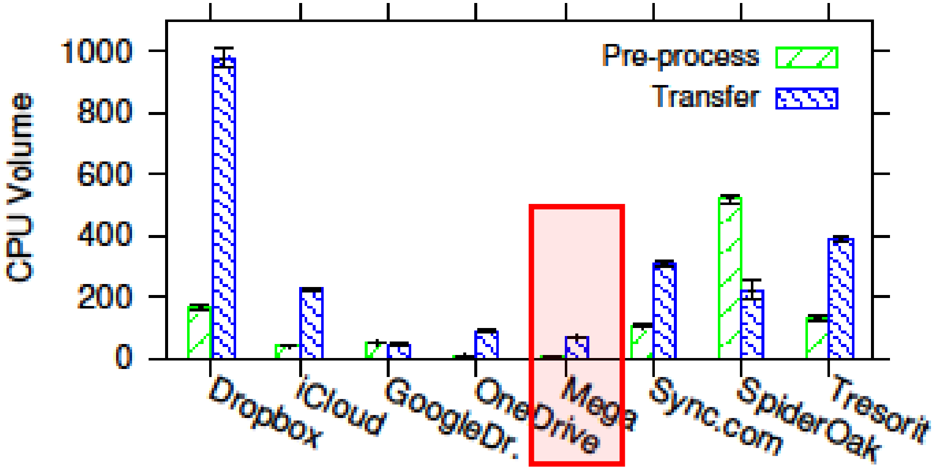


Performance: HTTP vs HTTPS

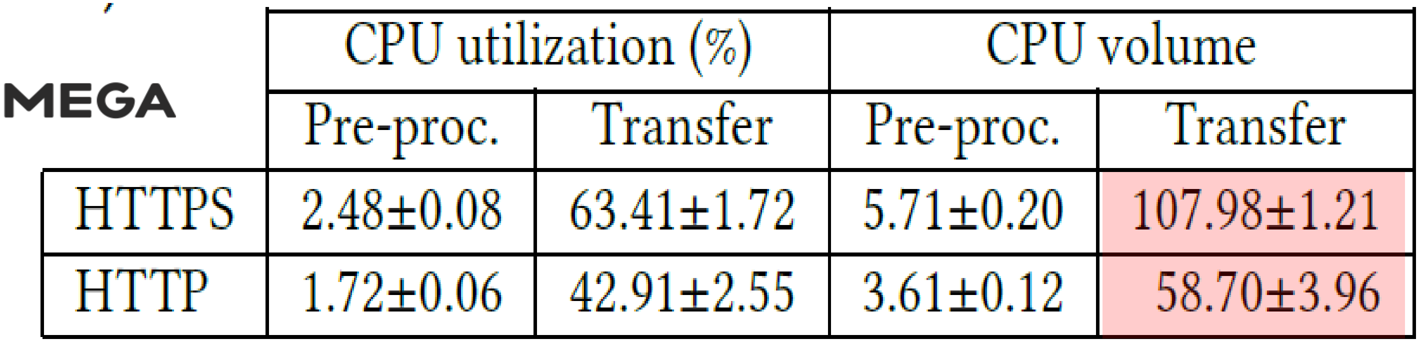


	CPU utilization (%)		CPU volume	
	Pre-proc.	Transfer	Pre-proc.	Transfer
HTTPS	2.48±0.08	63.41±1.72	5.71±0.20	107.98±1.21
HTTP	1.72±0.06	42.91±2.55	3.61±0.12	58.70±3.96

- Small extra HTTPS overhead compared to most other service

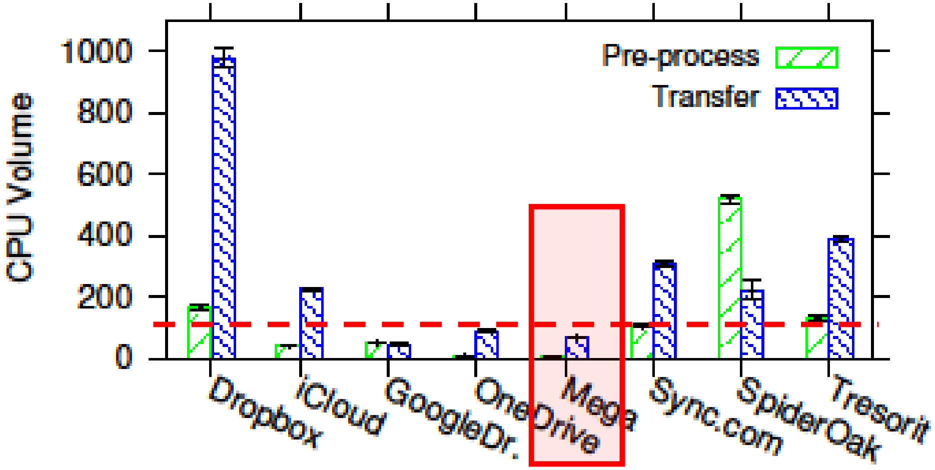


Performance: HTTP vs HTTPS

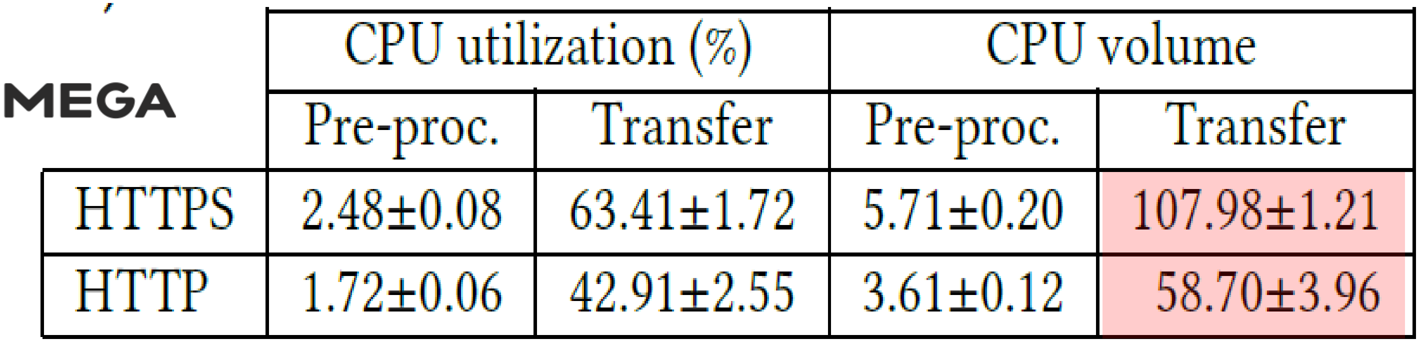


	CPU utilization (%)		CPU volume	
	Pre-proc.	Transfer	Pre-proc.	Transfer
HTTPS	2.48±0.08	63.41±1.72	5.71±0.20	107.98±1.21
HTTP	1.72±0.06	42.91±2.55	3.61±0.12	58.70±3.96

- Small extra HTTPS overhead compared to most other service

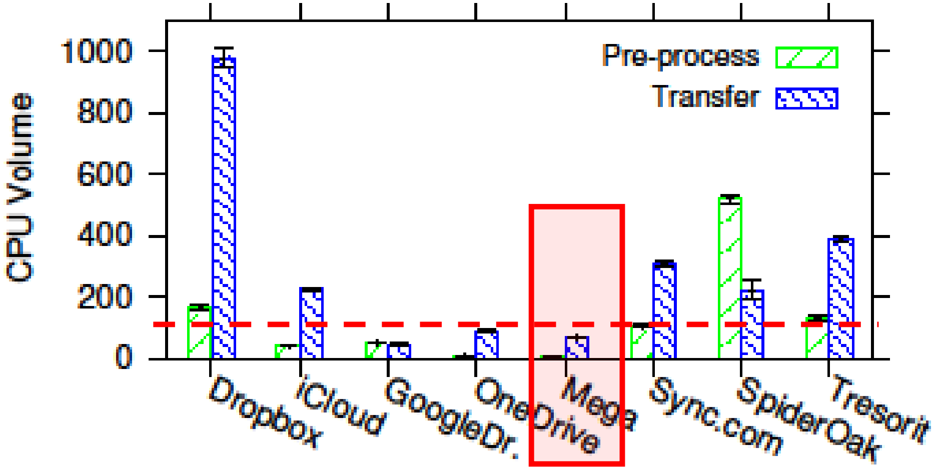


Performance: HTTP vs HTTPS

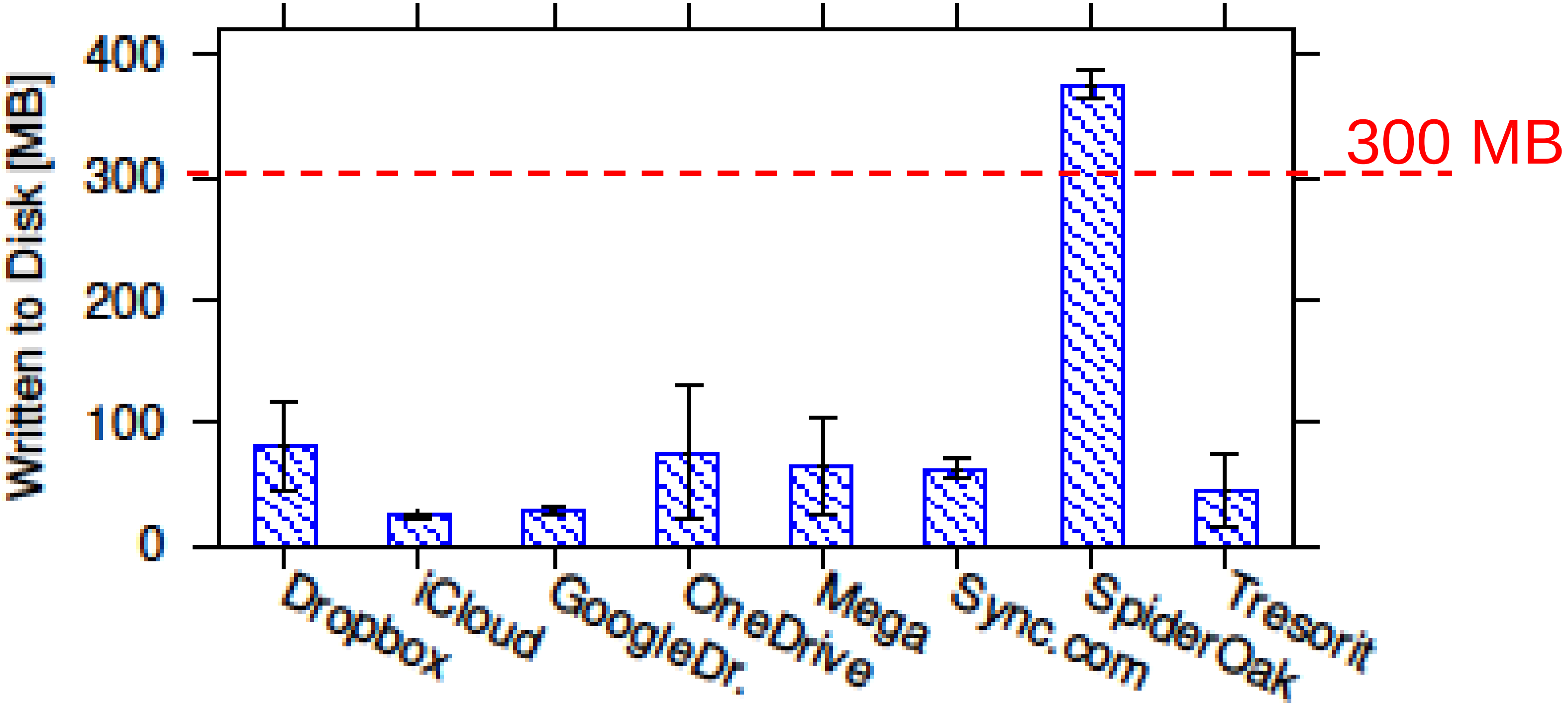


	CPU utilization (%)		CPU volume	
	Pre-proc.	Transfer	Pre-proc.	Transfer
HTTPS	2.48±0.08	63.41±1.72	5.71±0.20	107.98±1.21
HTTP	1.72±0.06	42.91±2.55	3.61±0.12	58.70±3.96

- Small extra HTTPS overhead compared to most other service
- CPU volume seems more dependent on what other features are implemented

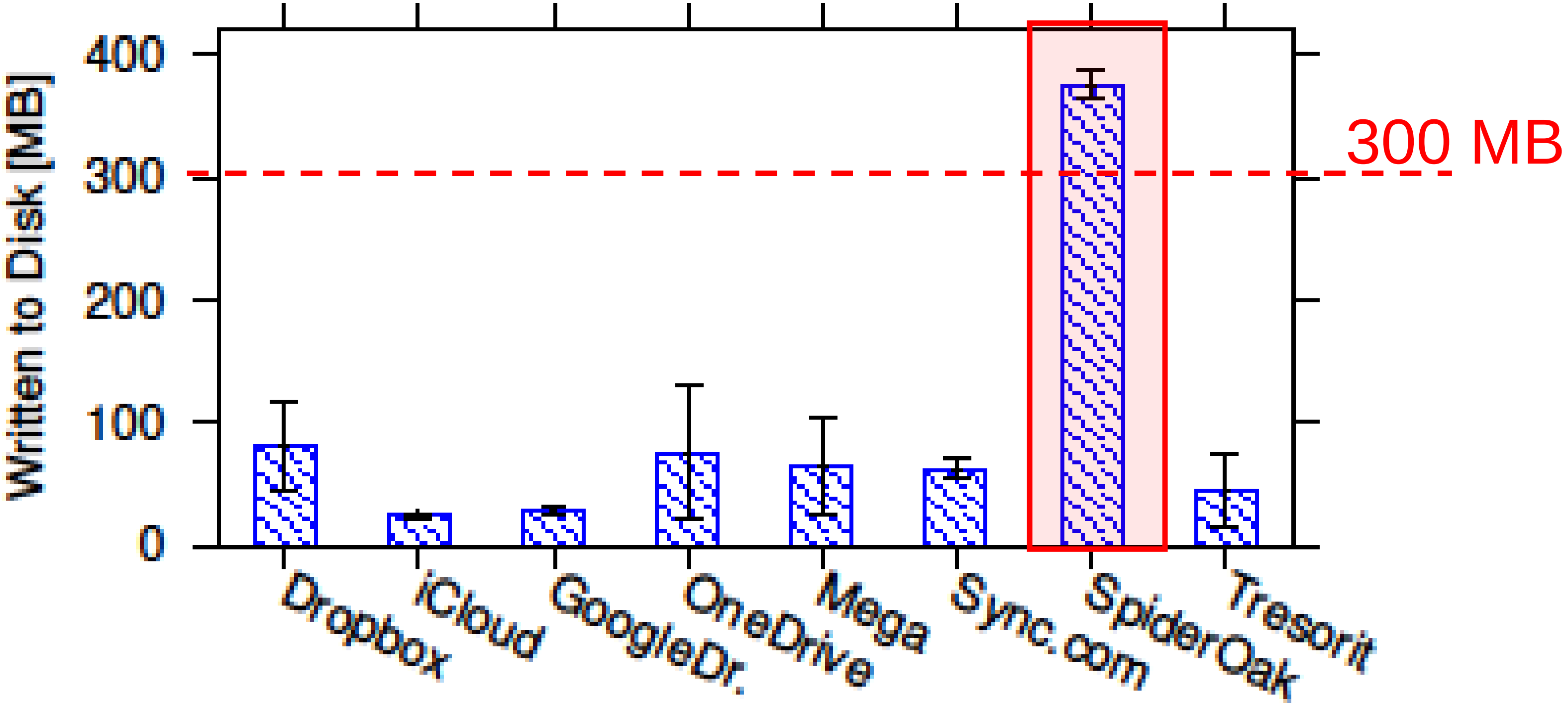


Performance: Disk usage



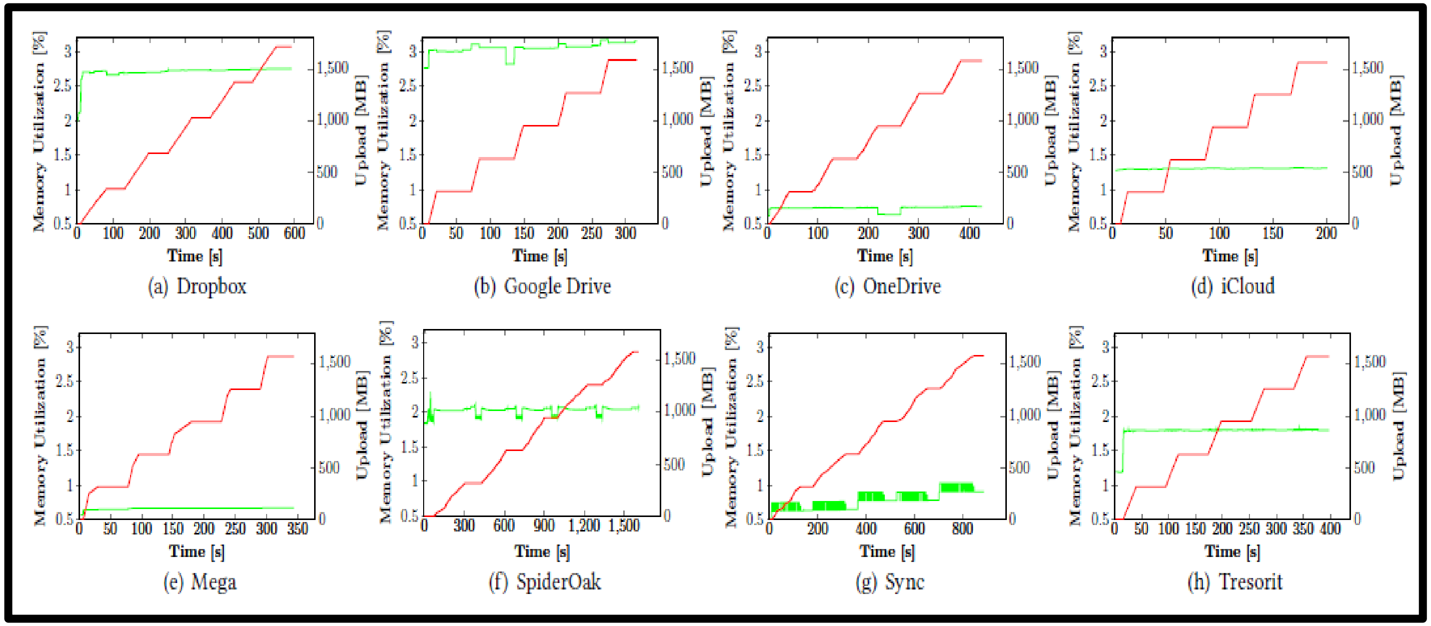
- Example writing 300 MB to cloud
  - Note: Can’t measure per process here (so, noise from other processes …)

Performance: Disk usage



- Example writing 300 MB to cloud
  - Note: Can’t measure per process here (so, noise from other processes …)
- SpiderOak temporarily writes entire file to disk; others do not ...

Performance: Memory usage



(a) Dropbox

(b) Google Drive

(c) OneDrive

(d) iCloud

(e) Mega

(f) SpiderOak

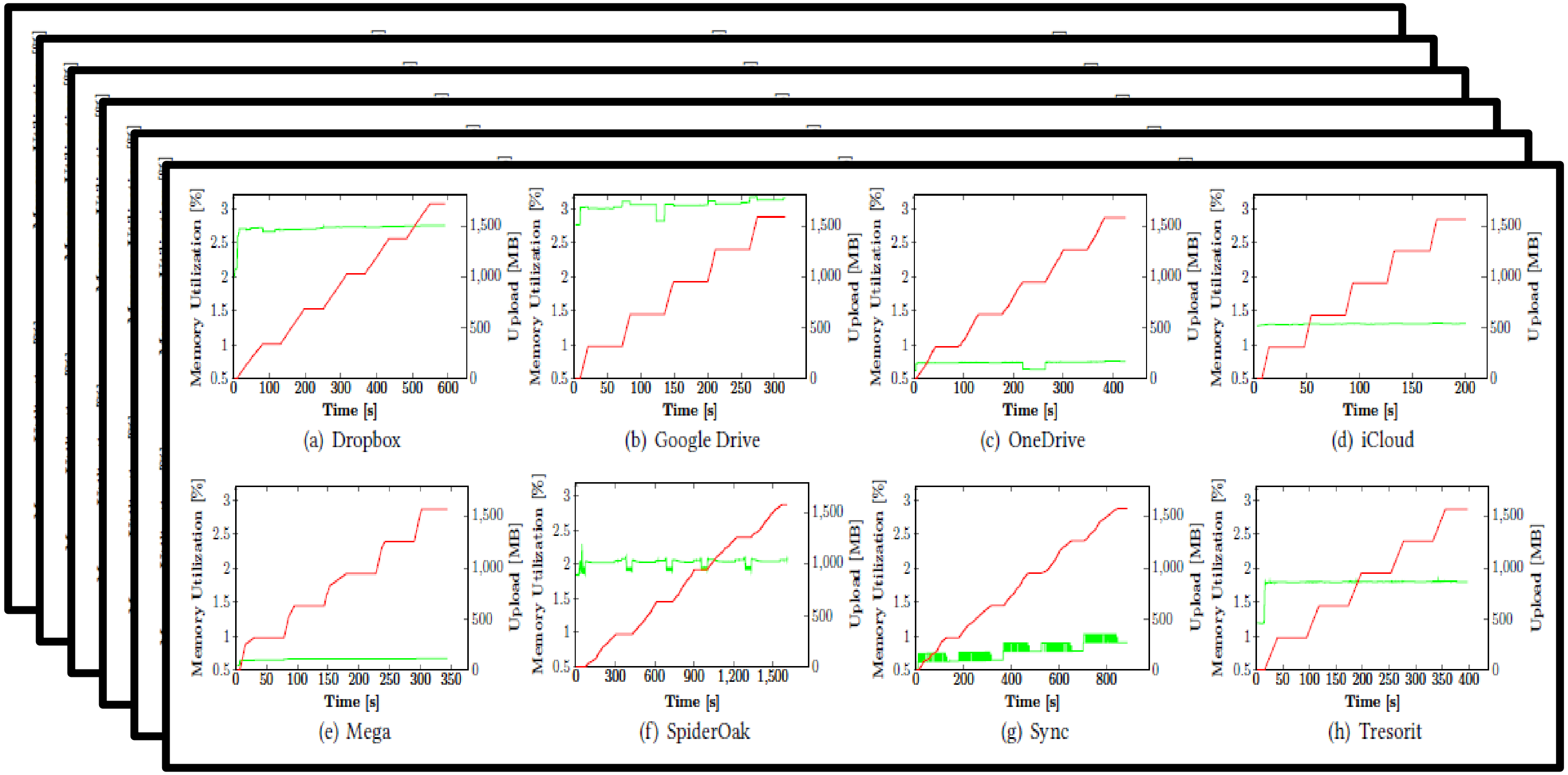
(g) Sync

(h) Tresorit

Test description

- Consecutively upload five different files of 300 MB each

Performance: Memory usage



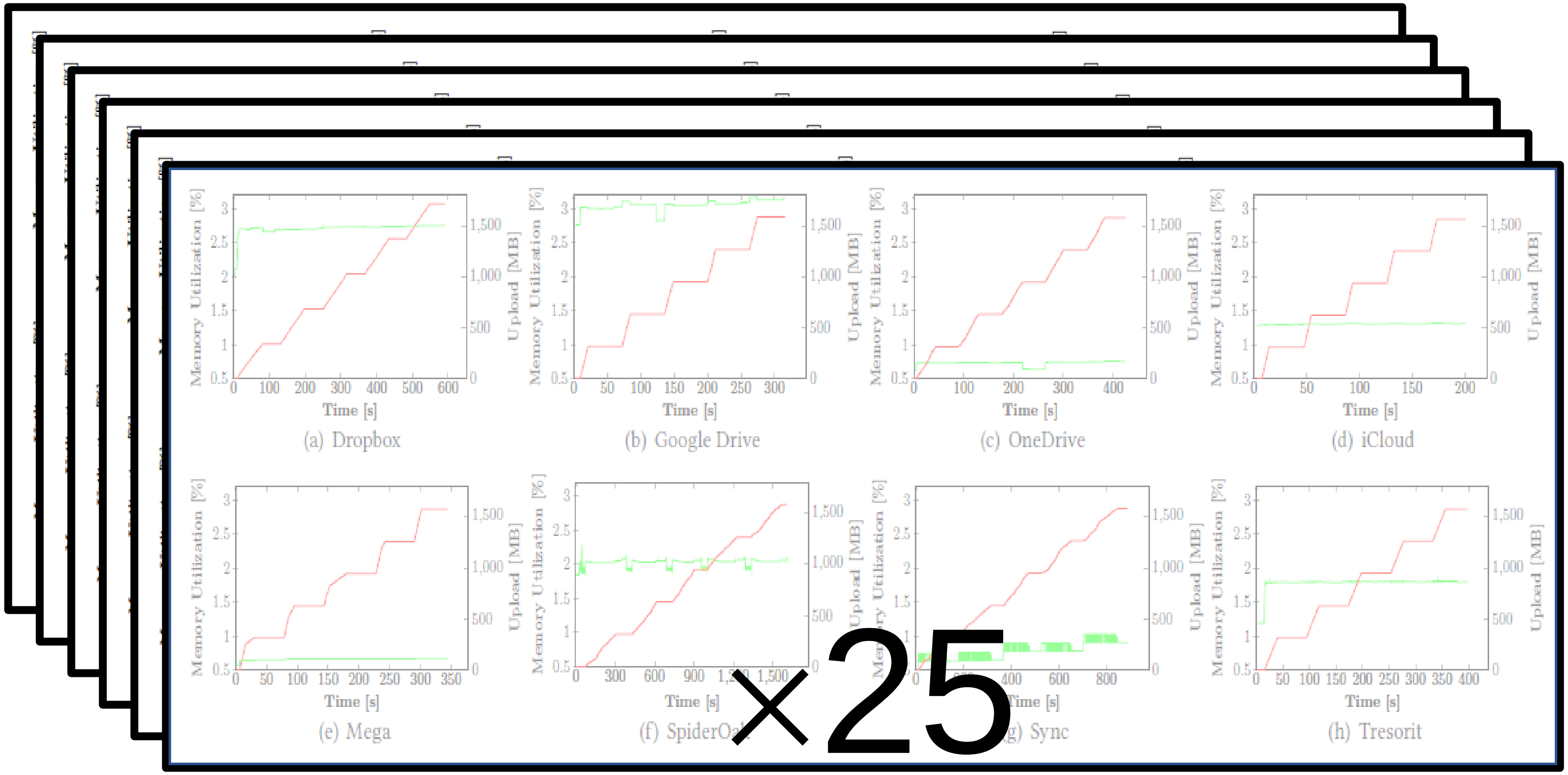
(a) Dropbox (b) Google Drive (c) OneDrive (d) iCloud

(e) Mega (f) SpiderOak (g) Sync (h) Tresorit

Test description

- Consecutively upload five different files of 300 MB each

Performance: Memory usage



(a) Dropbox (b) Google Drive (c) OneDrive (d) iCloud

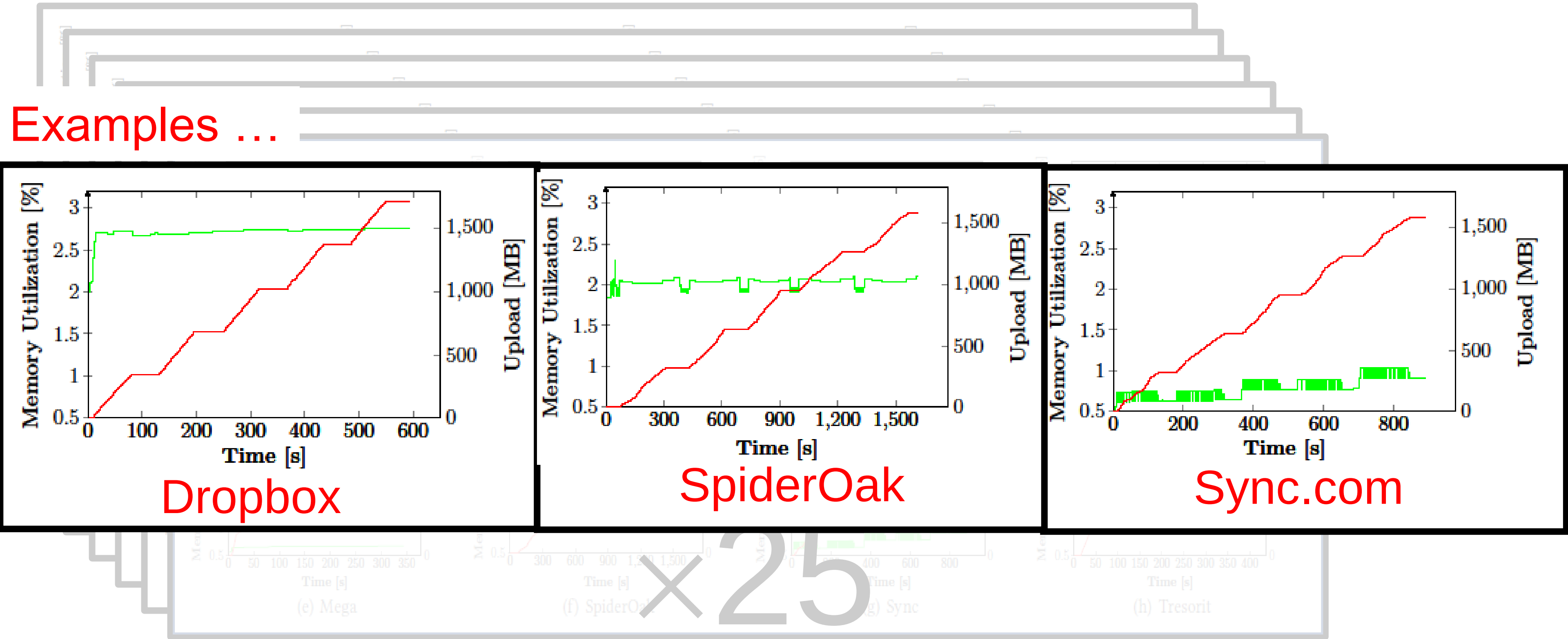
(e) Mega (f) SpiderOak (g) Sync (h) Tresorit

×25

Test description

- Consecutively upload five different files of 300 MB each

Performance: Memory usage



Examples …

Dropbox

SpiderOak

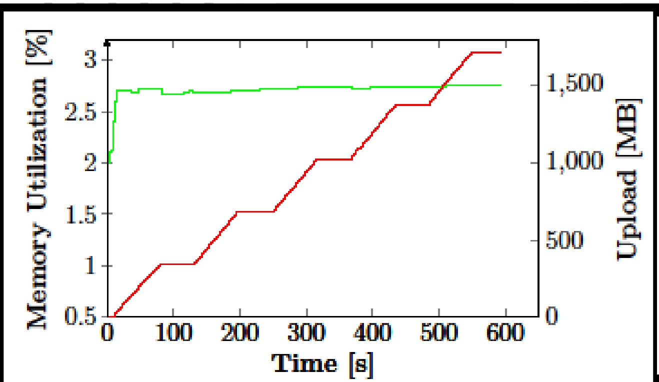
Sync.com

Test description

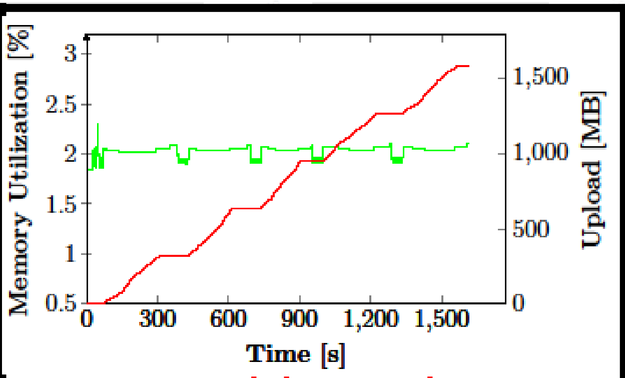
- Consecutively upload five different files of 300 MB each

Performance: Memory usage

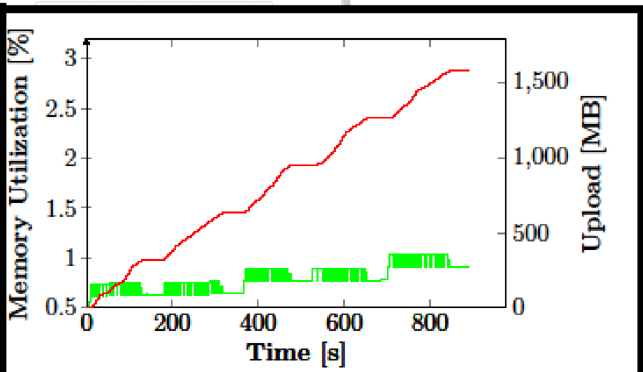
Examples …



Dropbox



SpiderOak



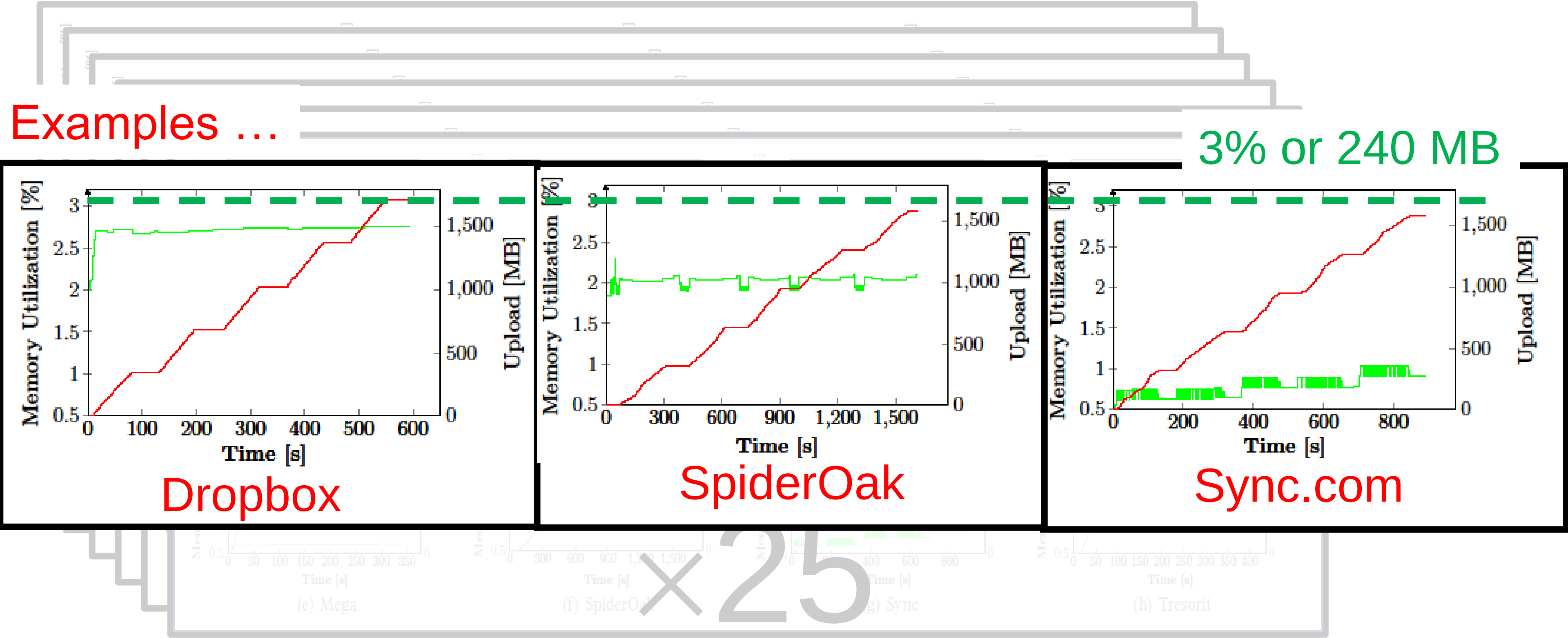
Sync.com

×25

Observations

Performance: Memory usage

Examples …



3% or 240 MB

Dropbox

SpiderOak

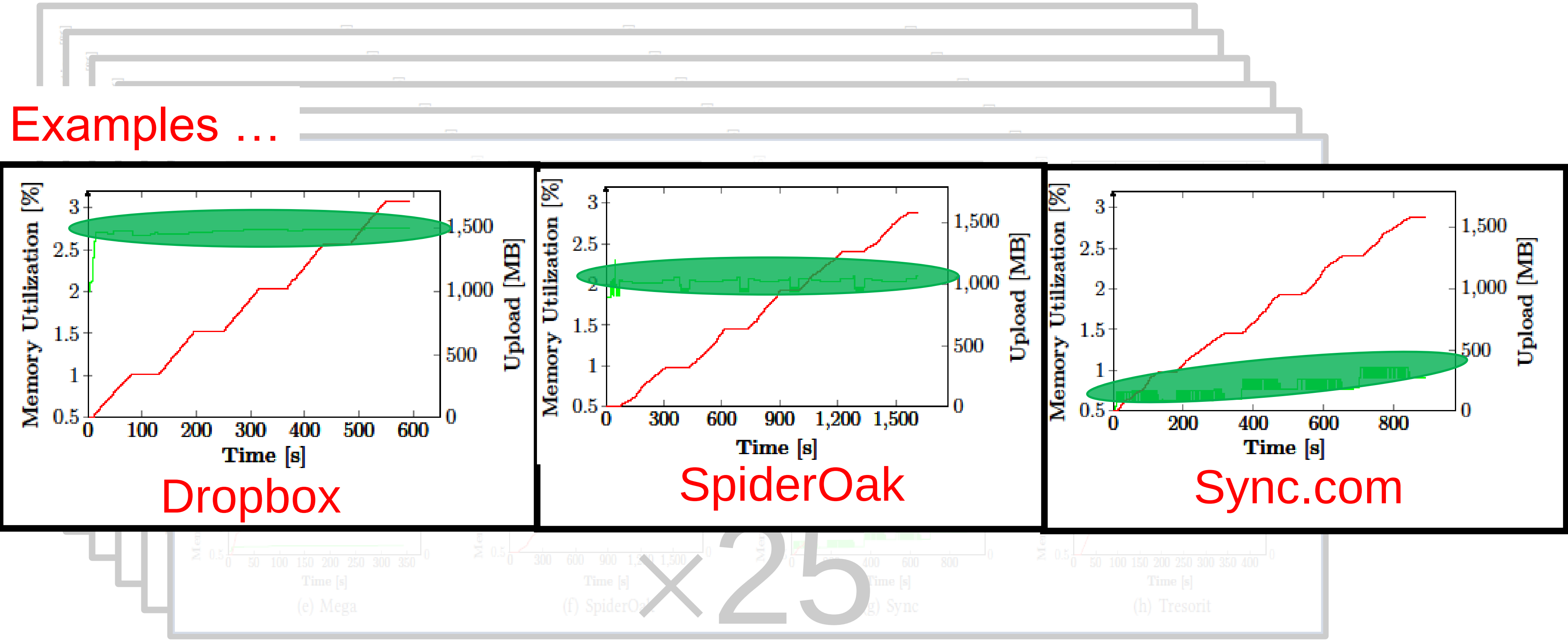
Sync.com

Observations

- None keep full copy in memory (e.g., 3% here is 240 MB)

Performance: Memory usage

Examples …

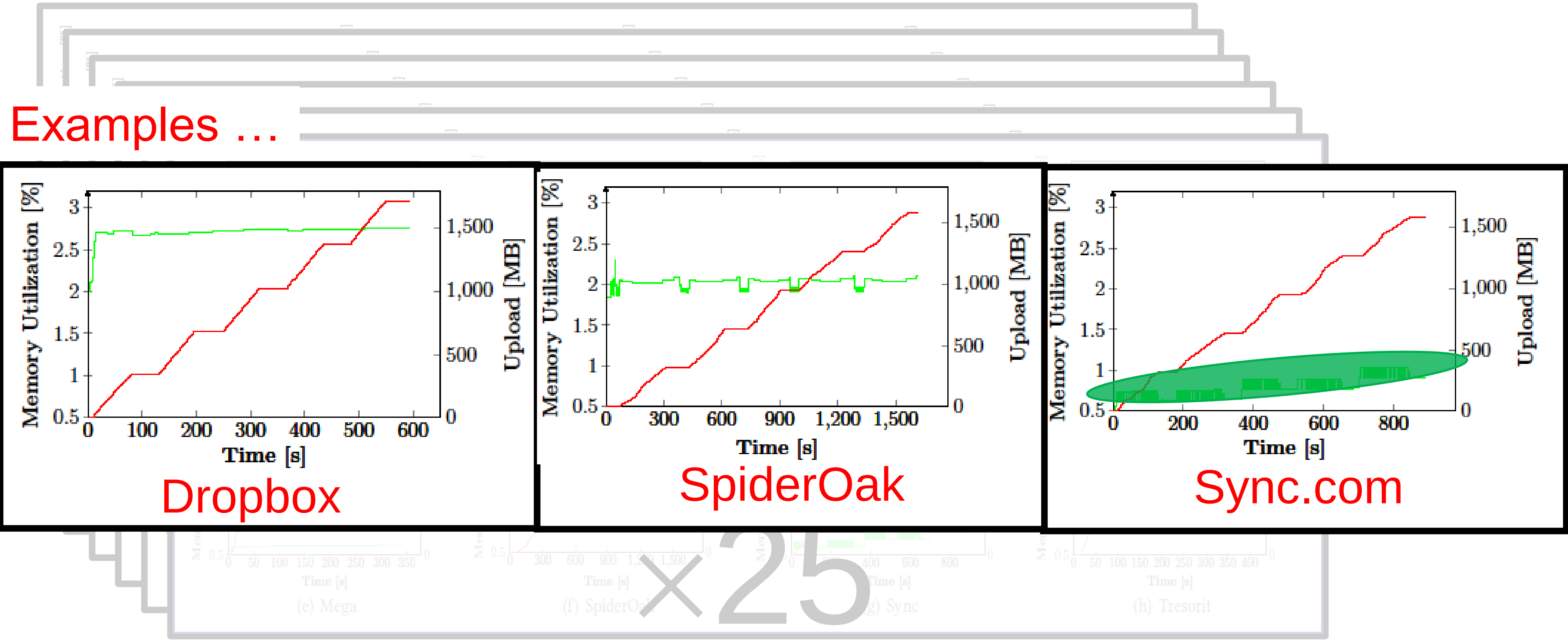


Observations

- None keep full copy in memory (e.g., 3% here is 240 MB)
- Dropbox, SpiderOak again stand out: most memory (with Google Dr.)

Performance: Memory usage

Examples …



Dropbox

SpiderOak

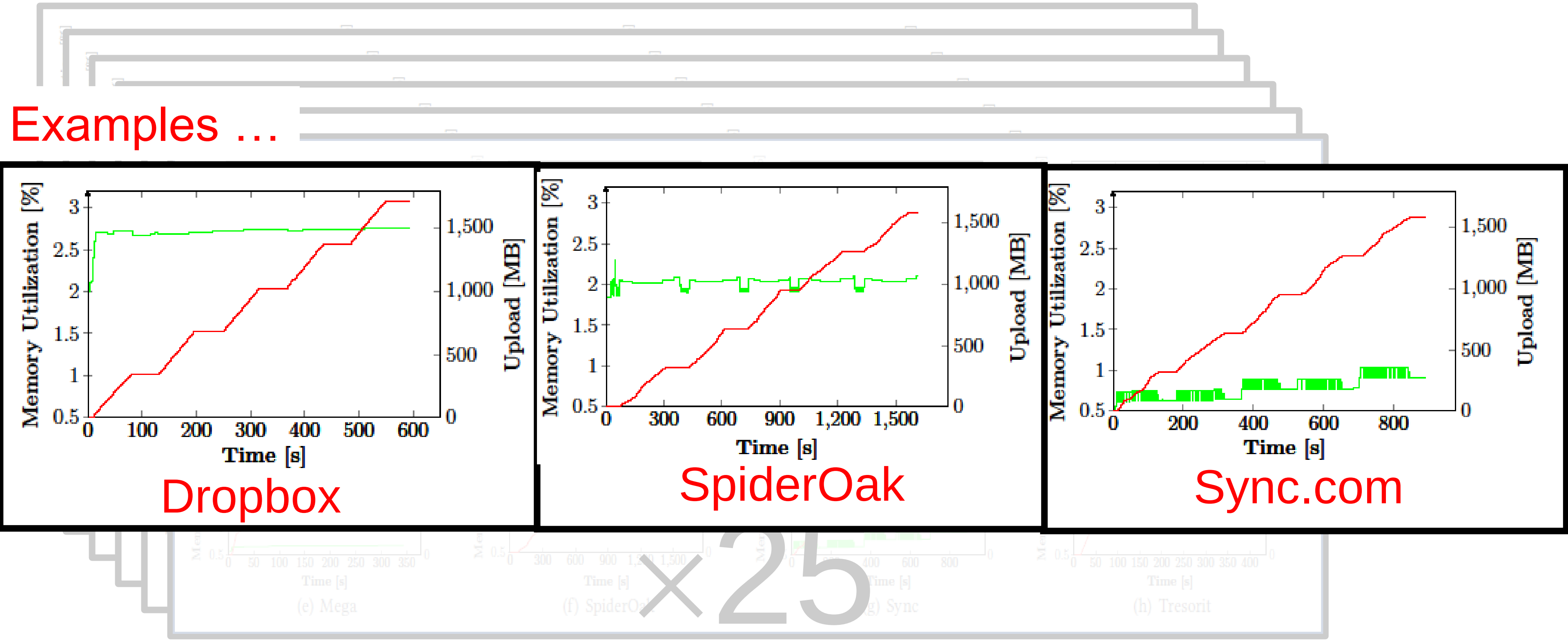
Sync.com

Observations

- None keep full copy in memory (e.g., 3% here is 240 MB)
- Dropbox, SpiderOak again stand out: most memory (with Google Dr.)
- Service unique patterns (e.g., sync.com have some drift in example)

Performance: Memory usage

Examples …

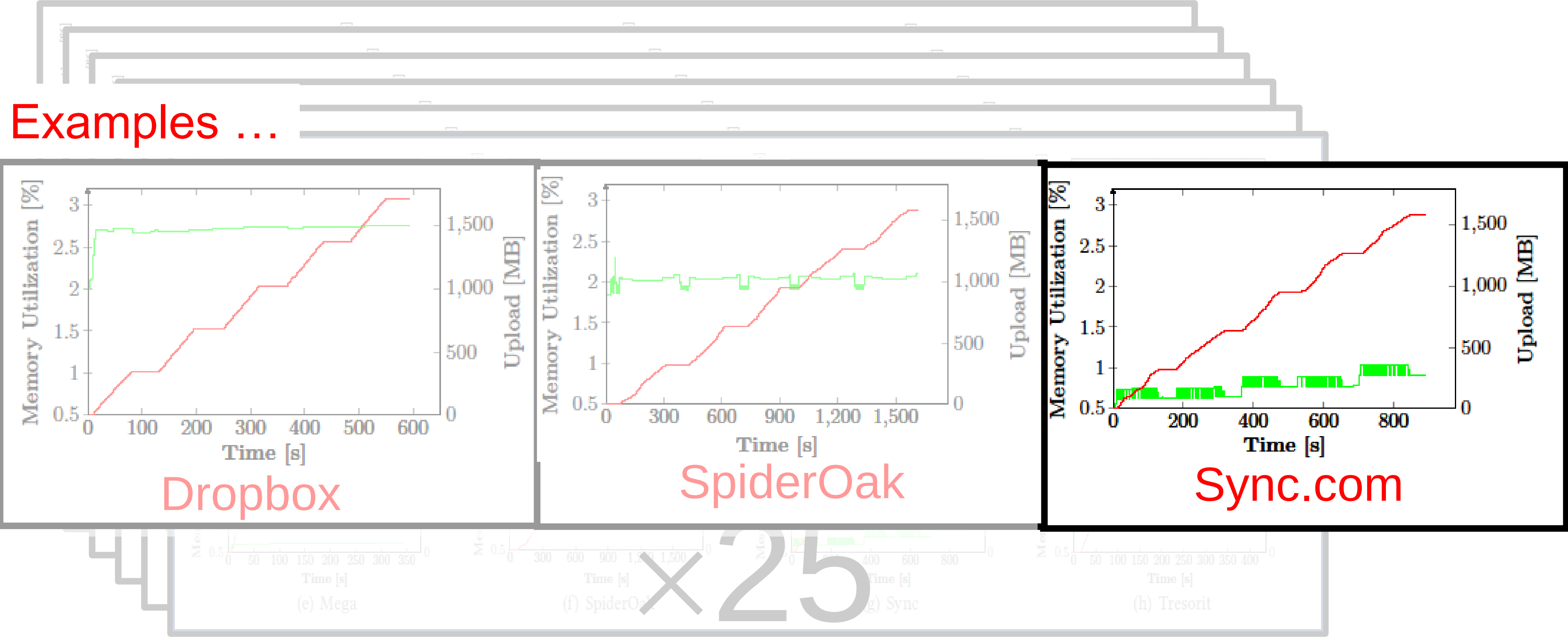


Observations

- None keep full copy in memory (e.g., 3% here is 240 MB)
- Dropbox, SpiderOak again stand out: most memory (with Google Dr.)
- Service unique patterns (e.g., sync.com have some drift in example)

Performance: Memory usage

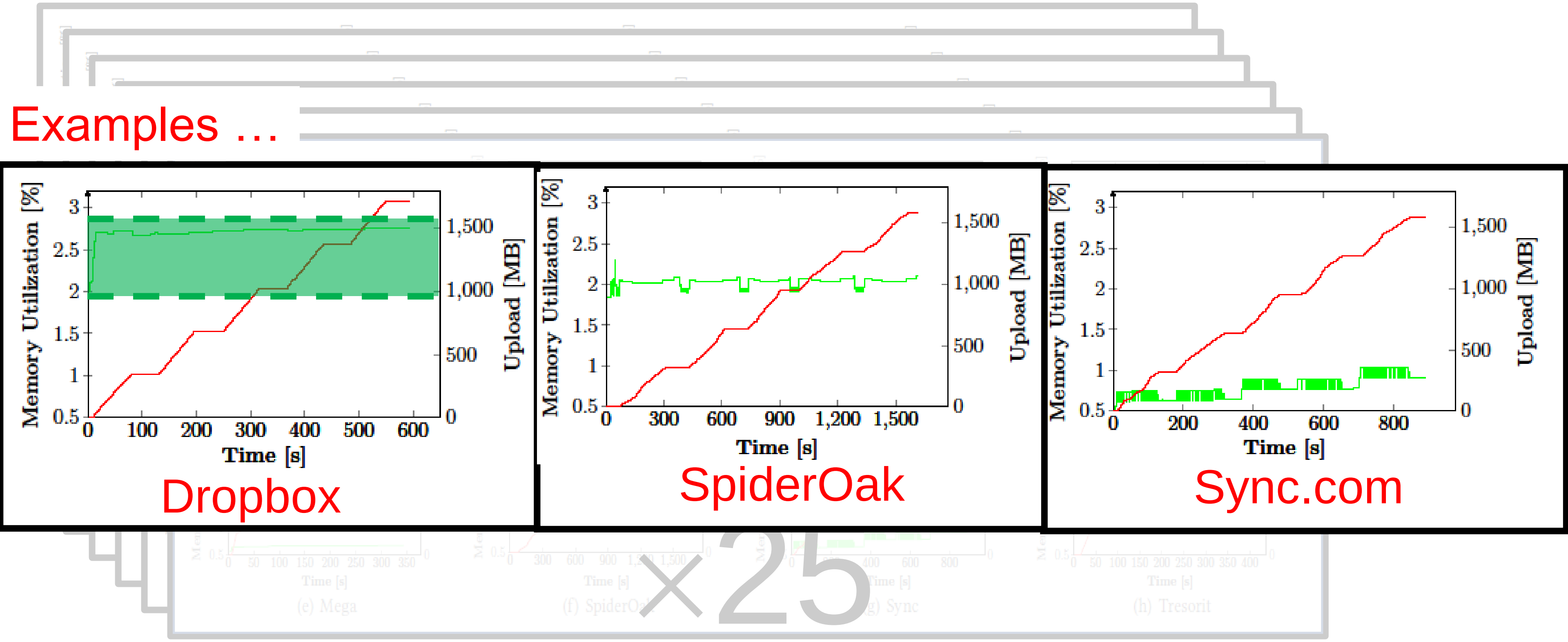
Examples …



Observations

- None keep full copy in memory (e.g., 3% here is 240 MB)
- Dropbox, SpiderOak again stand out: most memory (with Google Dr.)
- Service unique patterns (e.g., sync.com have some drift in example)

Performance: Memory usage



Examples …

Observations

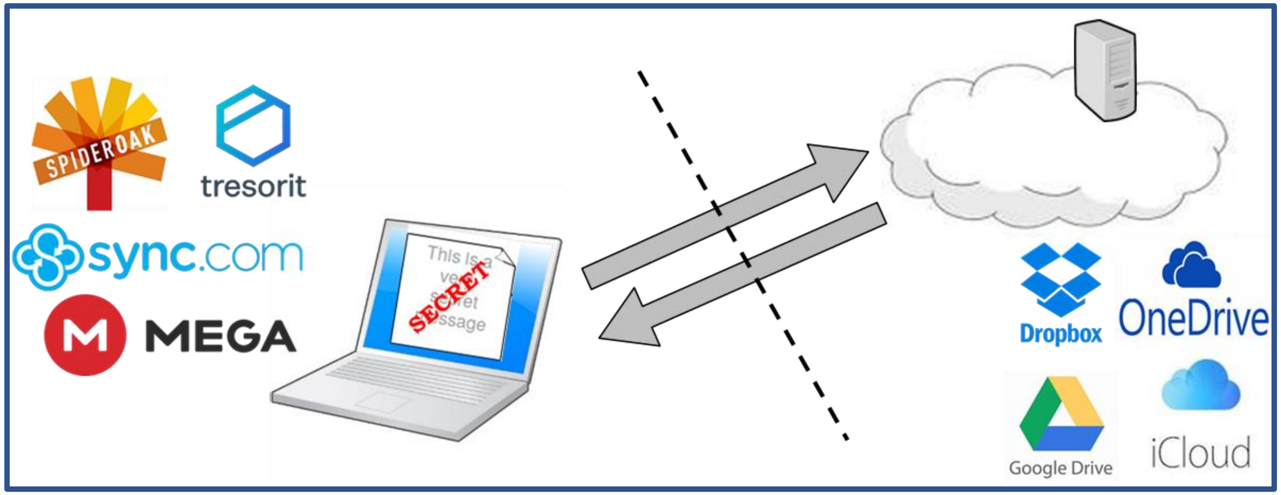
- None keep full copy in memory (e.g., 3% here is 240 MB)
- Dropbox, SpiderOak again stand out: most memory (with Google Dr.)
- Service unique patterns (e.g., sync.com have some drift)
- Mem. increases relative idle small: 4 with > 20MB; max 0.68% (Dropbox)

Conclusions

Conclusions

- First empirical comparison of the overheads of CSEs and non-CSEs
  - Set of security and bandwidth saving features implemented
  - Performance overheads (e.g., CPU volume, disk writes, memory)
- Overheads depend on set of bandwidth saving features implemented
- Bandwidth saving features such as **compression** and **deduplication** come with low additional overhead and achieve similar efficiency
- Main penalty associated with CSE appears to be due to bandwidth, storage, and processing overheads associated with implementing (or not implementing) different forms of **delta encoding together with CSE**
  - Significant differences between the CSE (SpiderOak) and the two non-CSEs (Dropbox, iCloud) implementing delta encoding
  - SpiderOak comes with **higher storage footprint** on the client and servers, has **higher bandwidth overhead** for uploaders and downloaders, and **implements less effective delta encoding** than Dropbox and iCloud
  - Follow-up work: More detailed delta-encoding analysis and optimized delta encoding policies for CSE in our IEEE CloudCom 2019 paper (next week)

Thanks for listening!



The Overhead of Confidentiality and Client-side Encryption in Cloud Storage Systems

Eric Henziger and Niklas Carlsson

LINKÖPING UNIVERSITY

Niklas Carlsson (niklas.carlsson@liu.se)