

Enabling Conflict-free Collaborations with Cloud Storage Services

Minghao Zhao, Jian Chen, and Zhenhua Li
Tsinghua University

IEEE International Conference on Parallel and Distributed Systems
13-15 December 2021, Beijing

How do you collaborate with your colleagues & teammates?



Emails & IM Apps

traditional, easy-to-grasp

low work efficiency

cannot work concurrently



Dedicated Online Editors

web-based, easy-to-use

limited functions

“walled-garden” concerns



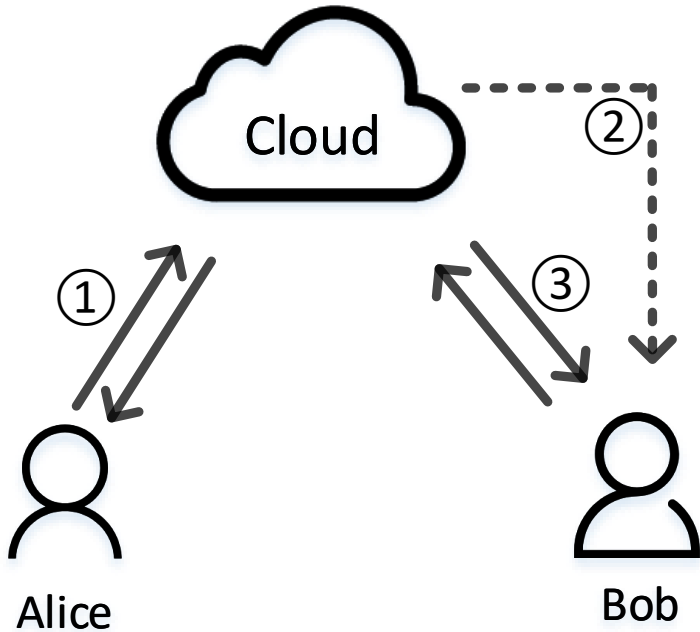
Version Control Systems

fine-grained conflict resolution

complex instructions

not user-friendly


Collaboration with cloud storage services




- ① Automatic file update
- ② Check and Notify
- ③ Automatic propagation

 main.tex

 main (Alice's conflicted copy 2019-12-21).tex

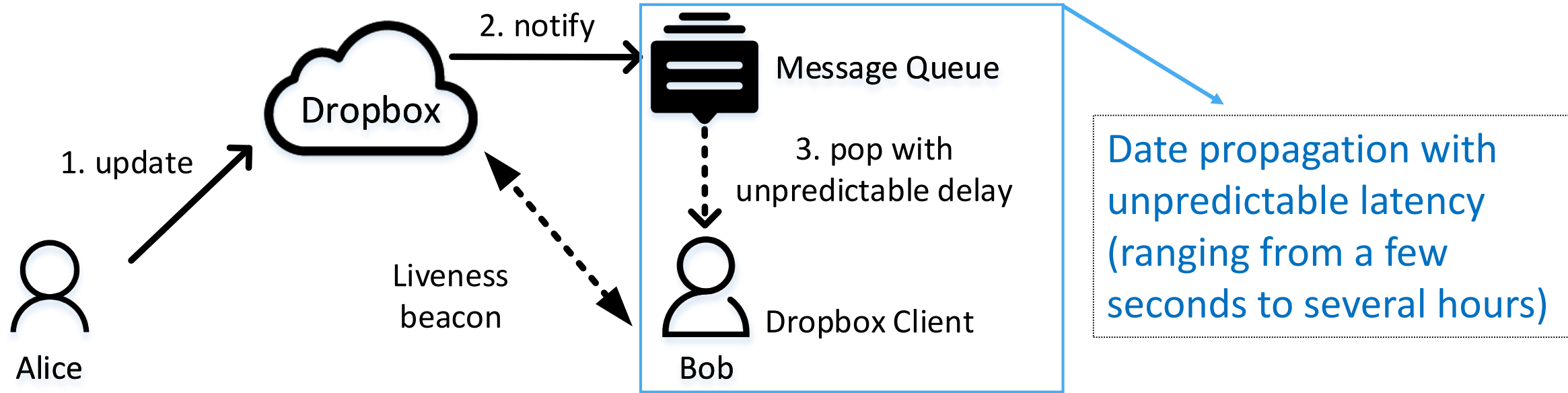
 main (Alice's conflicted copy 2019-12-21)(2).tex

 main (Bob's conflicted copy 2019-12-21).tex

 main (Alice's conflicted copy 2019-12-21) (Bob's conflicted copy 2019-12-21).tex

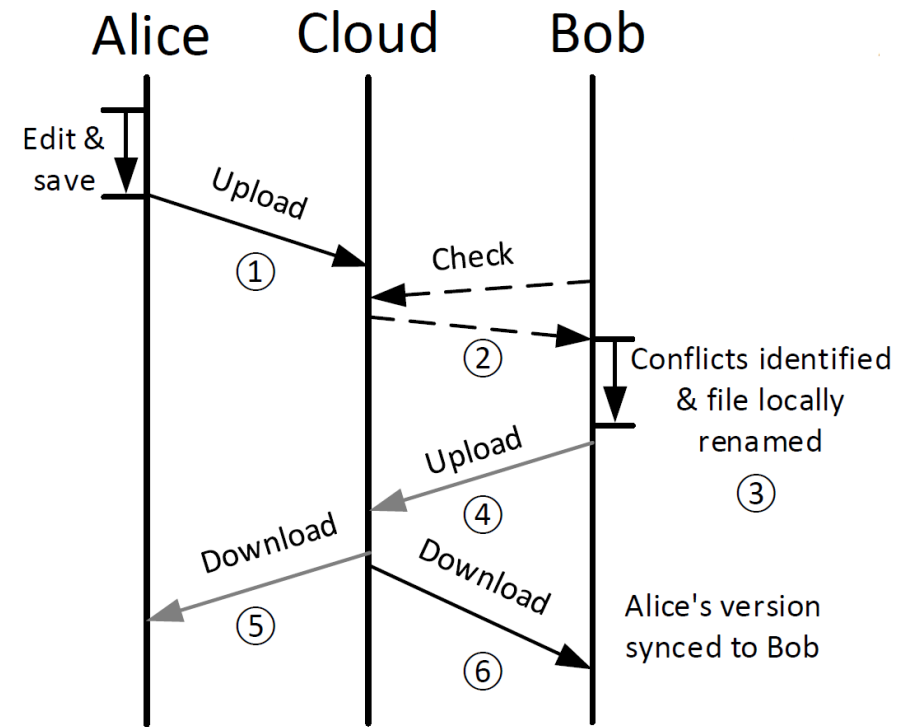
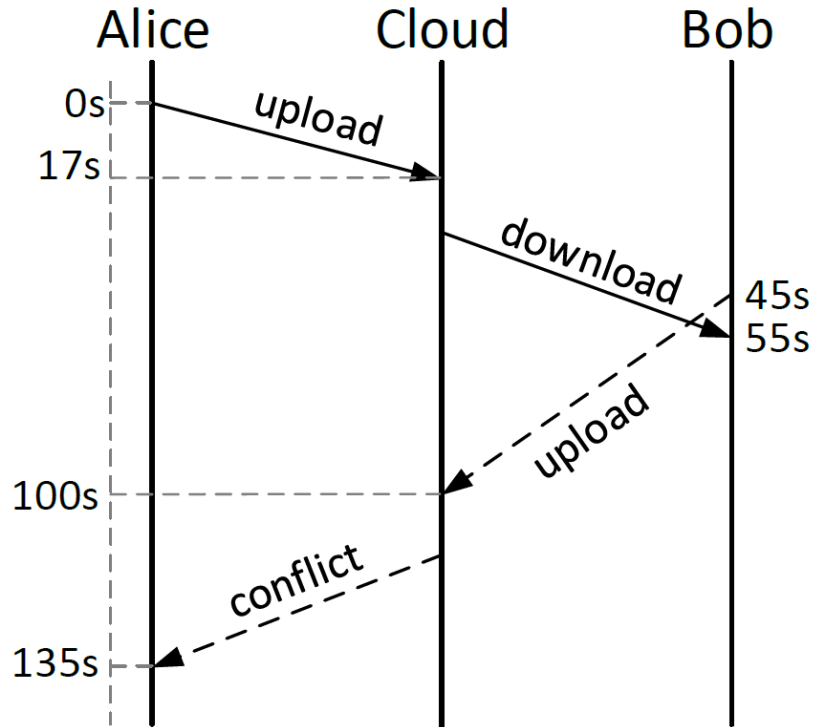
The conflict issue

Reveal the root causes of conflicts



- Dropbox never locks an edited file to avoid conflicts.
- Unpredicted latency caused by network turbulence and Message Queue
 1. edits updates to cloud (asap);
 2. the notification is pushed to a message queue;
 3. the notification is popped to Bob by the message queue;

Conflicts despite coordination (sequential edits)

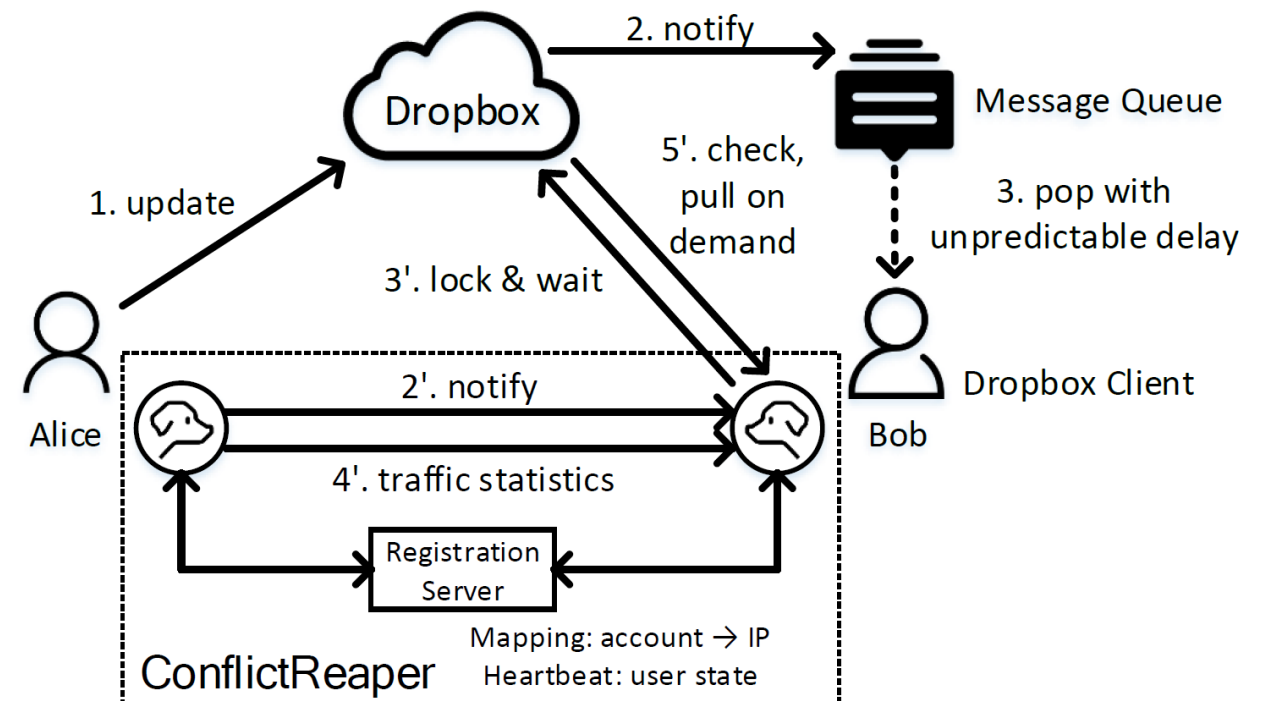


An overview of sequential edits.
The unpredictability & long tail latency
break the time order among file updates

Detailed workflow of Dropbox in
identifying and resolving conflicts

ConflictReaper Design

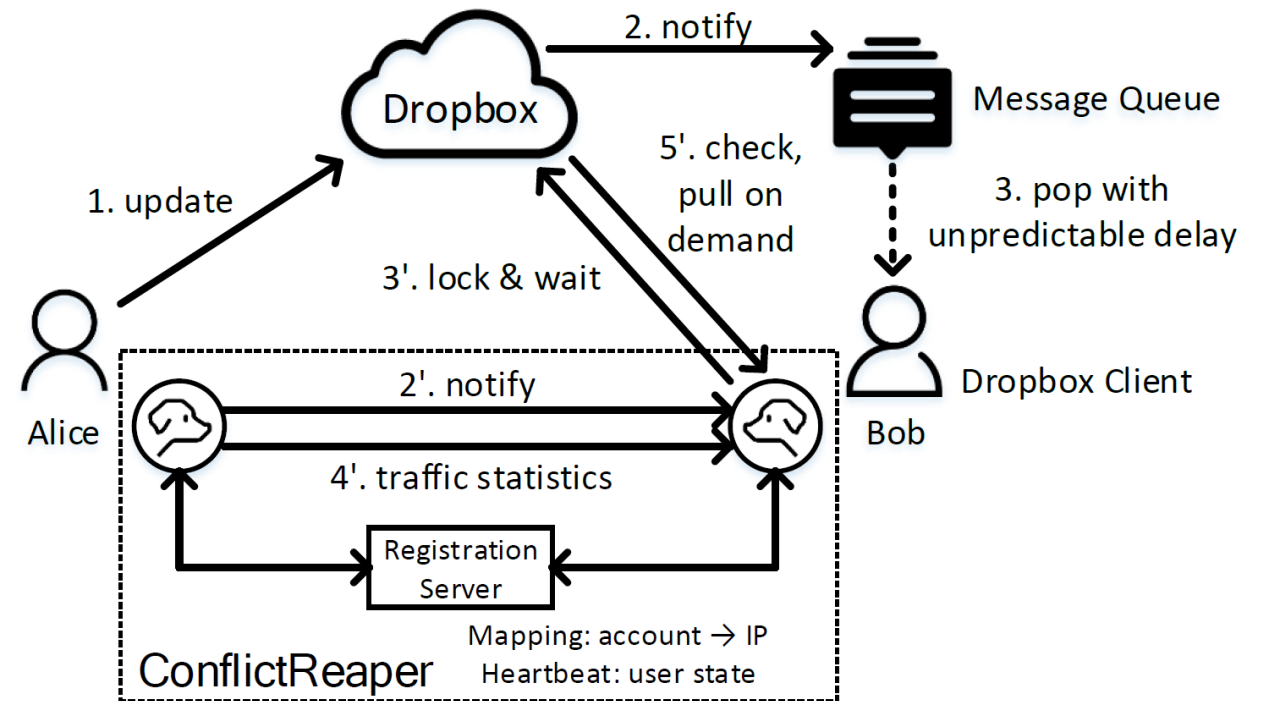
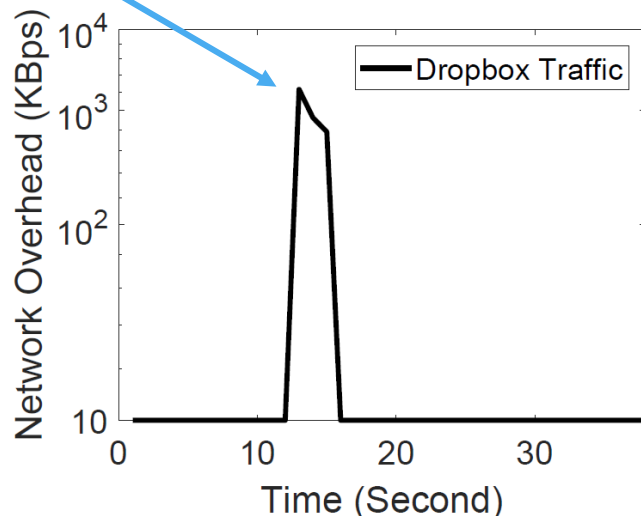
- Using locks to prevent conflicts
- Deduce the sync status from traffic statistics
- Track the latest version with Web APIs



ConflictReaper Design

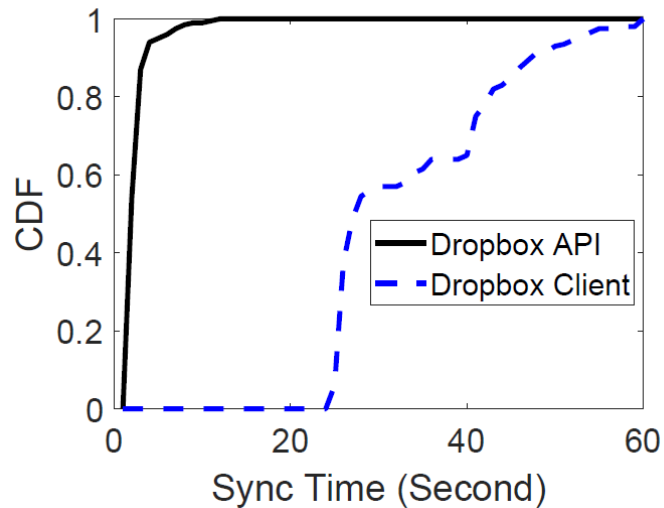
- Using locks to prevent conflicts
- Deduce the sync status from traffic statistics
- Track the latest version with Web APIs

Traffic peak

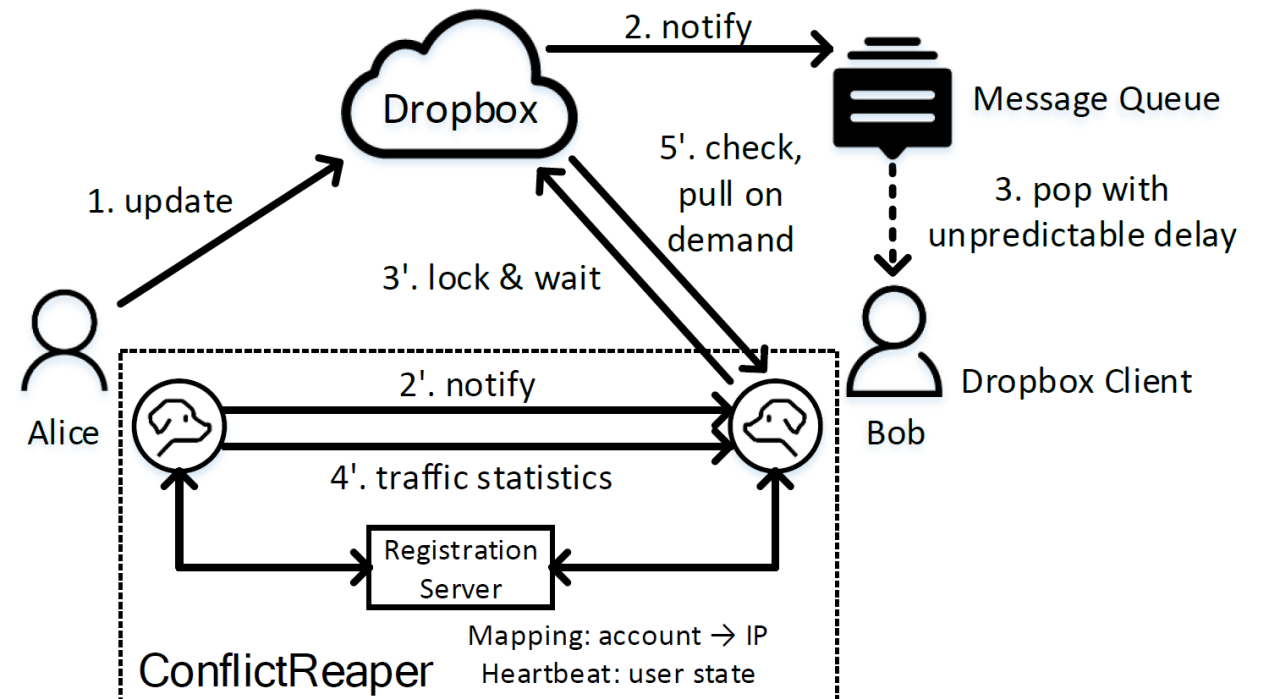


ConflictReaper Design

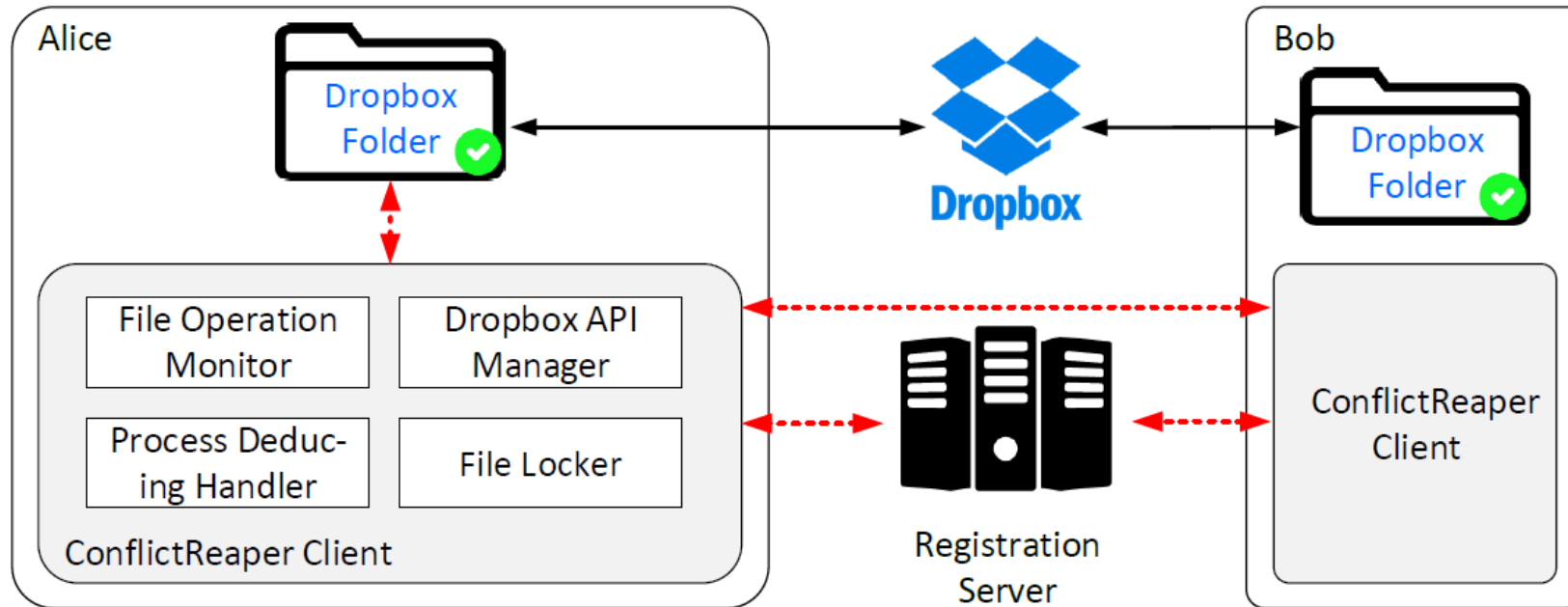
- Using locks to prevent conflicts
- Deduce the sync status from traffic statistics
- Track the latest version with Web APIs



$$T_{wait} = \frac{file_size - sync_traffic}{bandwidth} + T_{P99}$$



ConflictReaper Implementation



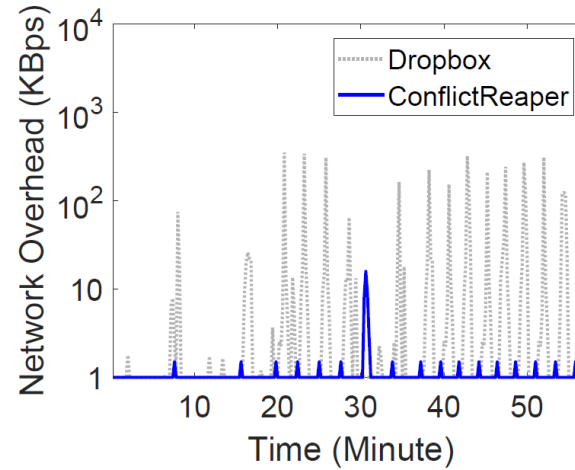
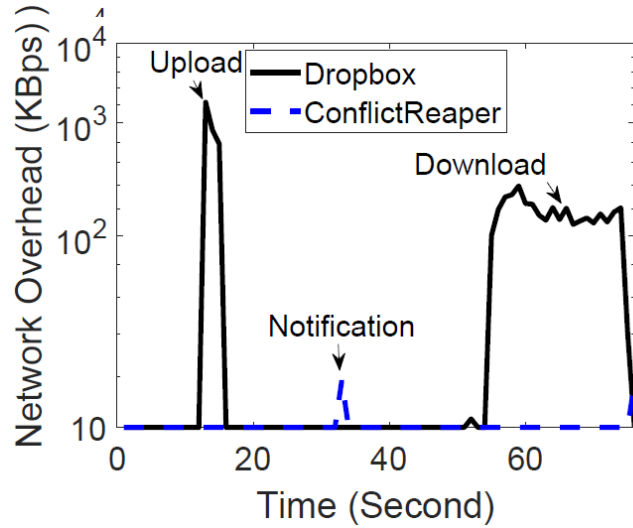
~2500 lines of C# codes for clients

~650 lines of Java for registration server,

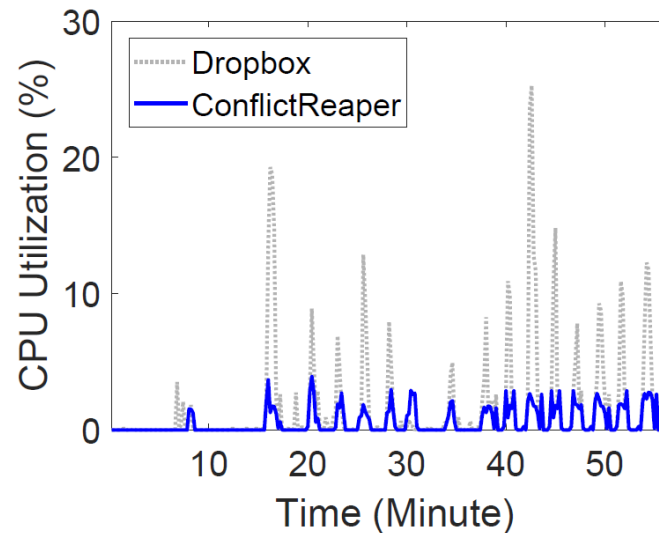
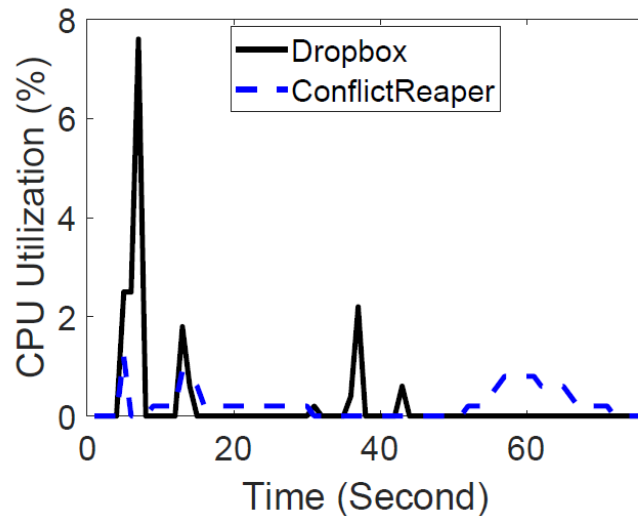
Code available: <https://conflictreaper.github.io/>



Performance Evaluation



Network Overhead



CPU Utilization



(Closely) Related Work

[FAST'20] Jian Chen, Minghao Zhao, et al., Lock-Free Collaboration Support for Cloud Storage Services with Operation Inference and Transformation.

- Studied eight mainstream cloud storage service
- Investigate common root-causes of conflicts in cloud storage-based collaborations (e.g., conflict rates, locks, conflict resolution, latency, and file update method)
- A preliminary design of lock-free, operation-based and fine-grained conflict resolution

[TPDS'22] Minghao Zhao, et al., UFC2: User-Friendly Collaborative Cloud

- Common architecture and general design principles
- Comprehensive design of a system with fine efficiency, universality and user-friendliness



In summary

- Collaboration with cloud storage services
 - Easy-to-use and automatic sync
 - Inevitable conflicts
- Unpredictable delivery latency contribute to conflicts
- ConflictReaper: preventing conflict in an light weight manner

Code available: <https://conflictreaper.github.io/>

Contact us: mh-zhao17@mails.tsinghua.edu.cn;

lizhenhua1983@tsinghua.edu.cn

