Background

- **Personal cloud services are gaining popularity**
  - Many providers enter the market. (e.g. Dropbox, Google, Microsoft, Box.com, Apple and etc.)
  - Cheaper and larger storage space
  - Different services are combined with the storage (photo browsing, email attachment, social info publication)


Dropbox


GoogleDrive


OneDrive

box


iCloud

… …

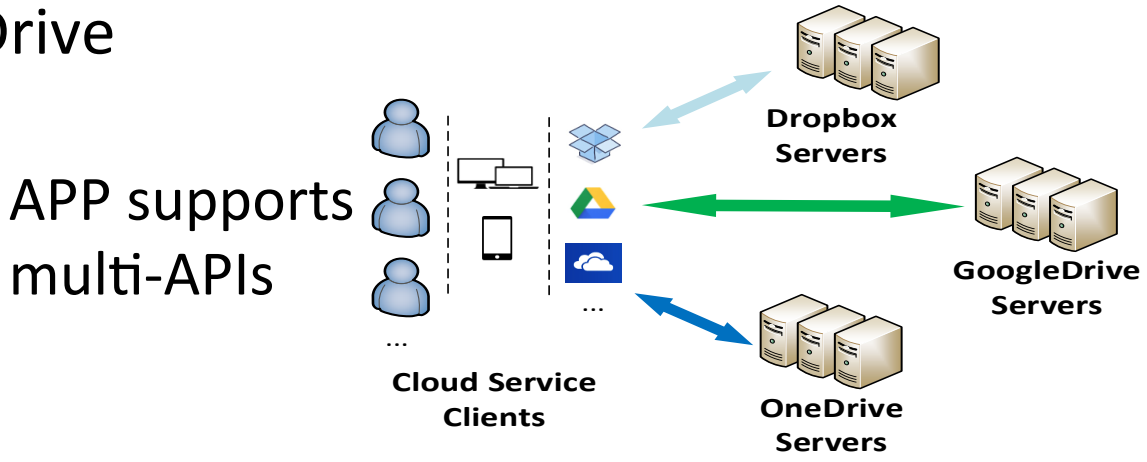
Background

- **Significant traffic produced**
  - Dropbox accounts for approximately 4% of the total traffic or around one third of the YouTube traffic at the same network (2012) [1].
- **Huge number of users**
  - e.g. Dropbox has more than 400 million registered users[2].

- [1]Drago I, Mellia M, M Munafo M, et al. Inside dropbox: understanding personal cloud storage services[C] IMC. ACM, 2012: 481-494.
- [2]http://techcrunch.com/2015/06/24/dropbox-hits-400-million-registered-users/

Background: Usage & Arch.

- Sync **local** files with servers in the **cloud**
- **HTTPS** or **HTTP** as the carrier protocol
- **Multi-device** and **multi-platform**
  - PC, laptop, smartphone
- **Their own proprietary sync protocols**
- **Provide APIs to support third party app**
  - IFTTT: uses APIs of Dropbox, OneDrive, Box, Google Drive



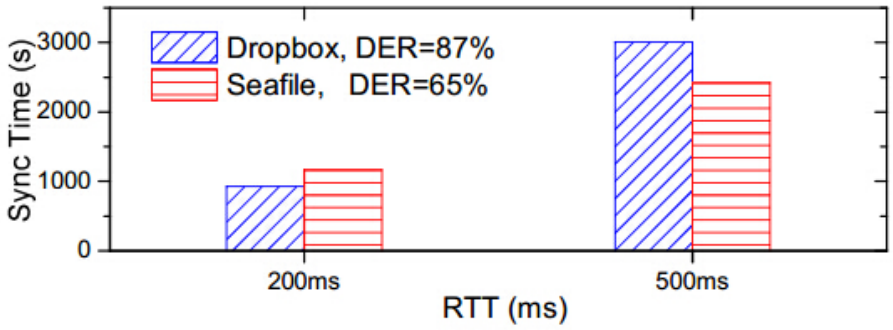
Current Problems

- One user needs multiple similar clients
  - User tends to use **multiple** cloud services
  - Better functionality: e.g. Dropbox may be better at file processing, GoogleDrive may be better at mail attachment
  - Increase the storage space, improve the reliability, …
- Third-party apps need to use multiple APIs
- Measurement: protocols need improvement [3][4]
  - Measurement study on Dropbox, GoogleDrive, OneDrive, Box
  - Different protocols have pros and cons at different aspects
  - But no one work well based on our extensive measurement

- [3] Yong Cui, Zeqi Lai, et al. Improving Network-level Sync Efficiency for Personal Cloud Storage Services. ACM MobiCom, 2015
- [4] Yong Cui, Zeqi Lai, et al. A First Look at Mobile Cloud Storage Services: Architecture, Experimentation and Challenge. Submitted to IEEE Network

Problem: Sync Inefficiency

- Typical capabilities in cloud storage systems
  - **1. Deduplication**: avoid retransmission of existing content in the cloud (detect redundancy)
  - **2. Chunking**: split file into small chunks, **smaller size** is better for eliminating redundancy
- **Network-aware design is important**
  - Detecting more redundancy is not always efficient
  - Trade-off: computation time and transmission time

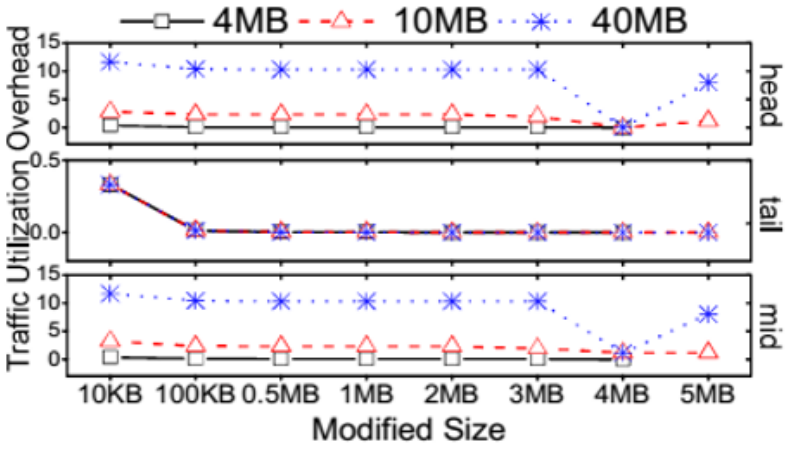


Dropbox: static large trunking,
Seafile: dynamic small trunking

Problem: Sync Inefficiency

- **3. Delta encoding**
  - Only synchronize modified data
- Delta encoding is **not always adopted**
  - Delta encoding is efficient to reduce traffic overhead
  - With improper trunking, file modification may result in sync traffic 10 times that of the modified size

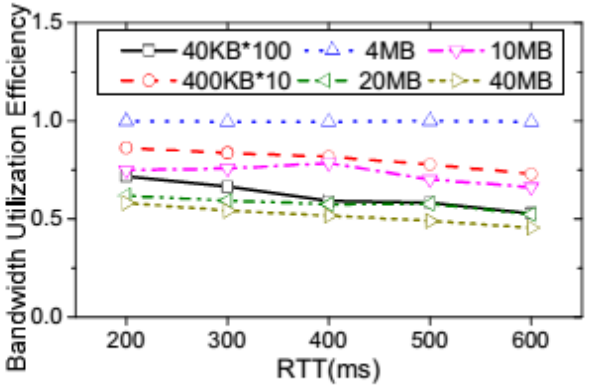
Traffic data / modified data



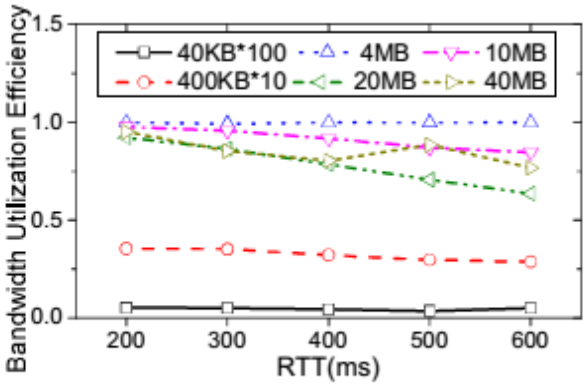
Dropbox with Delta Encoding

Problem: Sync Inefficiency

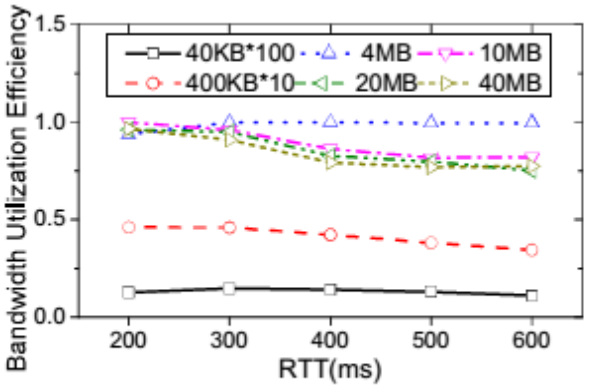
- **4. Bundling**
  - Transmit multiple small chunks as a single big chunk
- **Bundling is not always adopted**
  - Sync throughput slumps when synchronizing many small files
  - GoogleDrive establishes a new connection for one file without bundling (suffering TCP slow start)



(a) Dropbox



(b) GoogleDrive



(c) OneDrive

Root Cause for Sync inefficiency

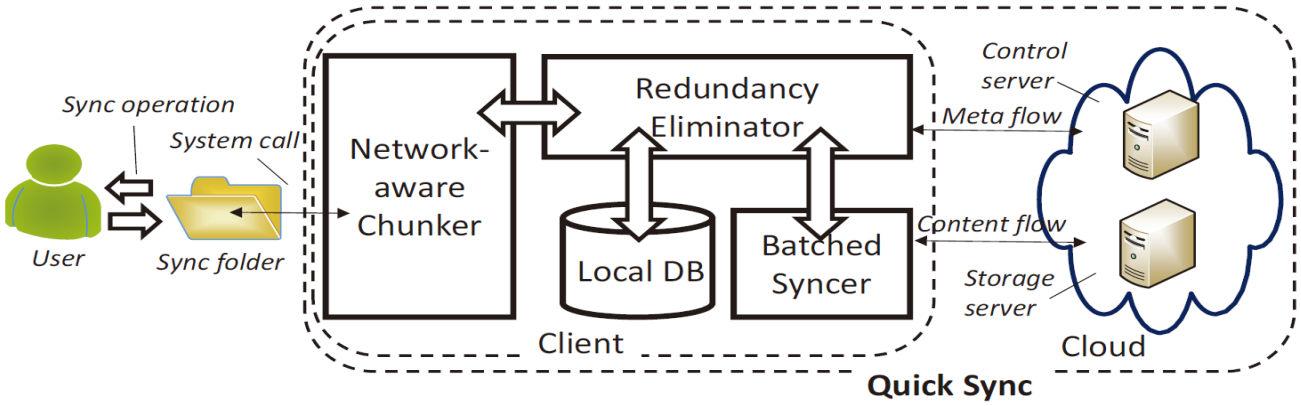
- Client and server: **proprietary sync protocols**
- Different capability configuration & implementation
- Sync protocol is not well designed

Capabilities	Windows			
	Dropbox	Google Drive	OneDrive	Seafile
Chunking	4 MB	8 MB	var.	var.
Bundling	√	×	×	×
Deduplication	√	×	×	√
Delta encoding	√	×	×	×
Data compression	√	√	×	×

*Android versions: very different trunking sizes, only dropbox supports Dedup.

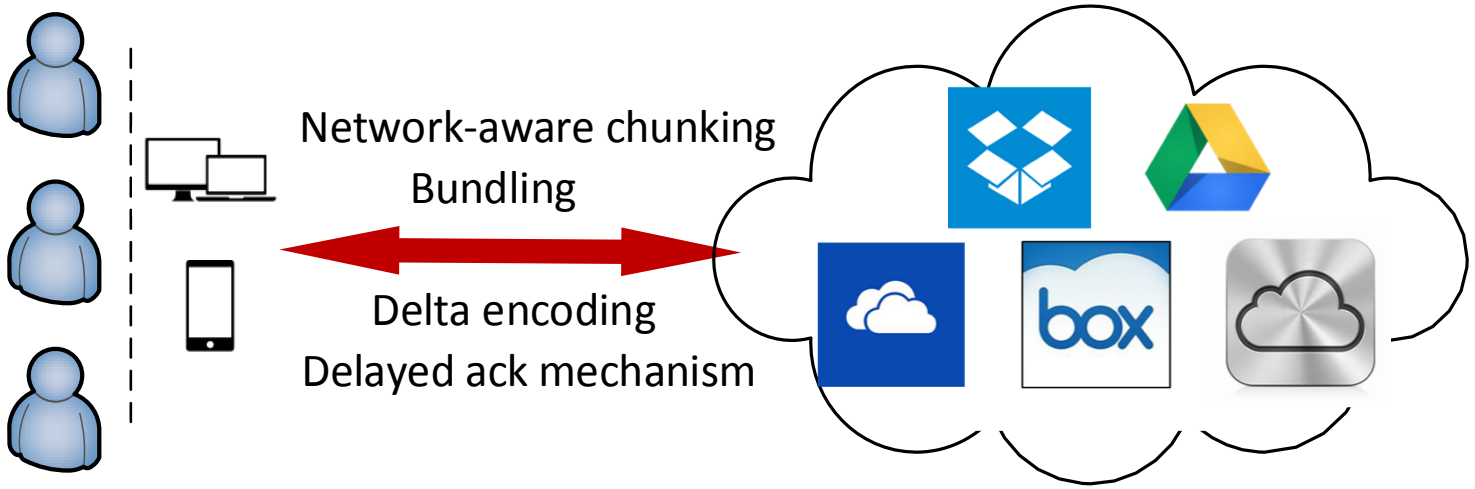
Improving sync efficiency

- QuickSync: a system with three novel techniques to improve the sync efficiency [ACM MobiCom15]
  - Adaptively select the proper chunking strategy
  - Improve delta-encoding to reduce the traffic overhead
  - Improve the network utilization of sync protocol
  - Effectiveness: reducing up to 51.8% sync time in representative sync scenarios



Usage of Standard Sync Protocol

- Advantage
  - One (third-party) client can support multiple services
  - Easy to improve cloud storage services
  - APIs will be unnecessary or simplified



What need to be considered?

- Key elements to improve the protocol
  - Chunking strategy or chunking size
  - Bundling small files together
  - Delayed ack mechanism
    - Sequential ack mechanism: wastes bandwidth
  - Delta encoding: a filed to indicate its validation
  - Deduplication
  - Compression
- Configuration or negotiation in the protocol
- Network-aware will be much better

Open issues

- Is IETF interested in this work?
- Any other issues?
  - Control server protocol for metadata trans.
  - Authentication or security issues?
- A new WG for this topic (BoF)? Scope of the new WG?
- Comments are welcome!

Backup slide: Successful open standard: XMPP

- A set of open **IM protocols**
  - published by IETF in 2004
- An **extensible** and **flexible** protocol
  - gives you the choices and control about how you access your data & services
- Before, there had **already** been
  - **popular** and **mature** proprietary IM apps (protocols)
  - e.g. MSN, ICQ …

Backup slide: Successful open standard: XMPP

- After the XMPP hit the market
  - IM services have gained widespread success
- **Popular** IM apps are/were based on XMPP

- Development of **personal cloud storage service**
  - similar to IM service
  - another XMPP?

Backup slide: Design principles

- Distributed architecture for control and data plane
- Only differences are transmitted
- Network-aware protocol
- Extensible message format
- Easy to understand and implement