DECADE Requirements

draft-gu-decade-reqs-04

Yingjie Gu
Haibin Song
Y. Richard Yang
Richard Alimi
Overview

- Focus on overview of requirements instead of specific language
  - For example, avoid MUST vs. SHOULD vs. MAY (for now)

- Avoid discussion of implementation approaches
  (e.g., designing new protocols, or using/extending existing protocols)

**Purpose**

- *Capture requirements for use by target applications*
- *Explain rationale behind them*
General Principles

- Core data storage operations: read/write/delete
  - Explicit control over in-network storage (contrast to P2P caching)

- Low-latency access
  - P2P applications may have constraints on delivery time

- Efficient transfer among multiple storage servers
  - Data transfer between storage servers avoids last-mile upload

- Client control over resource allocation
  - Bandwidth (e.g., rate/proportion/priority), storage quota, connections

- Allow for small object size
  - Some P2P apps designed to deliver data in small chunks (e.g., 16KB)
Data Access

- User can read/write from own storage
  - May also allow negotiation of data transport protocol
- Define and enforce access control policies for remote peers
  - Note that remote peers may be in different admin/security domains
Data Management

- Allow user to explicitly delete stored data or specify TTL

- User can get current resource usage and limits (including list of stored objects)
  - Make local resource allocation decisions; application restarts
Data Management

- Major considerations
  - Semantics under multiple writers and read/write conflicts
    - Save detailed discussion for the WG...
  - *Significant influence on complexity*

- Initially-considered requirements
  - Allow multiple, concurrent readers
    - P2P client uploads to multiple peers concurrently
  - Allow readers to access data before fully-written
    - Avoid store-and-forward delays to reduce latency

- Possible non-requirements (to reduce complexity)
  - Avoid update operation for already-written data (immutability)

- Possible optimization through relaxing consistency requirements
Authorization

- Per-peer, per-data read access
  - Authorize particular peers to retrieve particular content

- Per-peer write access
  - Authorize particular set of peers to store content

- For discussion (in WG)
  - Mechanism to define access control for remote peers' requests
    - Decision has impacts on latency and load on server
Resource Control

- Allow user to define resource control policies between concurrently-running applications
  - Apps may be on different machines, or not directly communicate
- Allow per-peer, per-data resource control
  - e.g., per-peer BW control or certain blocks with higher priority
- For discussion (in WG)
  - Mechanism to define access control for remote peers' requests
    - Decision has impacts on latency and load on server
Data Availability

- Allow (authorized) offline-access to user's storage
  - Handle intermittent connectivity, or when no app actively running
Error Conditions

- Indicate error if insufficient resources
  - Requested resources (e.g., storage) not available

- Indicate error if content unavailable or deleted
  - Provider may need reject, delete or quarantine data
  - DECADE does NOT indicate how such data identified
  - … but should not cause applications to break

- Allow server to reject requests/connections if overloaded
  - Server should not be forced to undertake new work if overloaded
Other Requirements

- Other requirements for discussion in WG
  - Data naming
  - Reliability/persistence
Comments and questions?