draft-ietf-decade-reqs-00

Y. Gu, **D. A. Bryan**, Y. R. Yang, R. Alimi

IETF-79, Beijing China, November 11, 2010

Changes since previous version

- Significant changes since previous version (draft-gu-decade-reqs-05)
- These were made mostly in response to comments at IETF-78, so reflect direction from WG, but would like comments on changes
- Refocused on just the clear, true requirements
  - Architectural issues/concepts have instead been moved to draft-alimi-decade-arch-01
  - A few non-requirement issues from previous version specifically discussed by WG remain, but moved to implementation considerations
- The requirements have been split into requirements for the protocol and requirements for the underlying storage system

Format of Talk

- Will list authors' open questions
- Do not plan to go step-by-step through all requirements, but have them in slide form if we need to discuss particular issues or the WG wants to discuss them individually

Open Issues (1)

- Is the break down of issues correct?
  - Current proposed structure
    - DECADE Requirements
      - Protocol requirements
      - Requirements on underlying storage
    - Implementation considerations
  - Is this approach reasonable to the WG?

(Note: implementation considerations may be moved to another document later, but collecting them here until we have a better location)

Open Issues (2)

- The WG has indicated a desire to allow for applications (such as distributed VoD) that closely resemble the data dissemination patterns of P2P, while not building a generic network storage system
  - **Application-Independent API:** The DECADE IAP MUST provide a simple, application-independent API for P2P applications to access in-network storage
- We propose to revise to use a more generic term that is still within the charter constraints
- Need to explicitly state not building a generic network storage system?

Open Issues (3)

- The WG has indicated a desire to allow for applications to do duplicate data removal within a server
  - Have added this as a implementation detail
- Do we want to do duplication removal across multiple servers?
  - This would require protocol considerations

Open Issues (4)

- Discovery
  - We currently do not have requirements for this, but clearly a discovery mechanism is required
  - Looking for input on requirements for discovery
  - (Note: Discovery doesn't include selecting "best" server – explicitly out of scope!)
- Mobility
  - We don't have many requirements around mobility
    - What is the interest?
    - What do we need for ***specific*** requirements?

Open Issues (5)

- Do we need a new requirement to explicitly caution against putting the “smarts” of the system into DECADE?
- Approach in other places has been to provide generic mechanism, offload specific complexity to application
- Should we clarify this, and how do we capture specifically?

Additional Slides

Specific Requirements Follow

Protocol Requirements

- Overall Protocol Requirements
- Connectivity Concerns
- Error and Failure Conditions
- Transfer and Latency Requirements
- Data Access Requirements
- Data Management Requirements
- Resource Control
- Authorization

Overall Protocol Requirements

- **Application-Independent API:** The DECADE IAP MUST provide a simple, application-independent API for P2P applications to access in-network storage.
- **Cross-platform Access:** If DECADE supports the ability to store metadata associated with data objects, the DECADE protocol(s) MUST transmit any metadata using an operating system-independent and architecture-independent format.

Connectivity Concerns

- **NATs and Firewalls:** DECADE SHOULD be usable across firewalls and NATs without requiring additional network support (e.g., Application-level Gateways).
- **Connections to Clients:** DECADE SHOULD NOT require that network connections be made to DECADE clients (e.g., from a server to a DECADE client or from a DECADE client to another DECADE client).

Error and Failure Conditions

- **Overload Condition:** In-network storage, which is operating close to its capacity limit (e.g., too busy servicing other requests), MUST be able to reject requests.
- **Insufficient Resources:** DECADE MUST support an error condition indicating to a DECADE client that resources (e.g., storage space) were not available to service a request (e.g., storage quota exceeded when attempting to store data).
- **Unavailable and Deleted Data:** DECADE MUST support error conditions indicating that (1) data was rejected from being stored, (2) deleted, or (3) marked unavailable by a storage provider.

Transfer and Latency Requirements

- **Low-Latency Access:** DECADE SHOULD provide "low-latency" access for application clients. DECADE MUST allow clients to specify at least two classes of services for service: lowest possible latency and latency non-critical.
- **Indirect Transfer:** DECADE MUST allow a user's in-network storage to directly fetch from other user's in-network storage.
- **Data Object Size:** DECADE MUST allow for efficient data transfer of small objects (e.g., 16KB) between a DECADE client and in-network storage with minimal additional latency required by the protocol.
- **Communication Among Storage Elements:** DECADE SHOULD support the ability for two in-network storage elements in different administrative domains to store and/or retrieve data directly between each other. If such a capability is supported, this MAY be the same (or a subset or extension of) as the IAP used by clients to access data.

Data Access Requirements

- **Reading/Writing Own Storage:** DECADE MUST support the ability for a DECADE client to read data from and write data to its own in-network storage.
- **Access by Other Users:** DECADE MUST support the ability for a user to apply access control policies to users other than itself for its storage. The users with whom access is being shared can be under a different administrative domain than the user who owns the in-network storage. The authorized users may read from or write to the user's storage.
- **Negotiable Data Protocol:** DECADE MUST support the ability for a DECADE client to negotiate with its In-network storage about which protocol it can use to read data from and write data to its In-network storage.
- **Separation of Data Operations from Application Control:** The DECADE IAP MUST only provide a minimal set of core operations to support diverse policies implemented and desired by Target Applications.

Data Management

- **Agnostic of Reliability:** DECADE SHOULD remain agnostic of reliability/fault-tolerance level offered by storage provider.
- **Time-to-Live for Stored Data:** DECADE MUST support the ability for a DECADE client to indicate a time-to-live value (or expiration time) indicating a length of time until particular data can be deleted by the in-network storage element.
- **Offline Usage:** DECADE MAY support the ability for a user to provide authorized access to its in-network storage even if the user has no DECADE applications actively running or connected to the network.

Resource Control

- **Multiple Applications:** DECADE SHOULD support the ability for users to define resource sharing policies for multiple applications being run/managed by the user.
- **Per-Peer, Per-Data Control:** A DECADE client MUST be able to assign resource quotas to individual peers for reading from and writing particular data to its in-network storage within a particular range of time. The DECADE server MUST enforce these constraints.
- **Server Involvement:** DECADE client MUST be able to indicate, without contacting the server itself, resource control policies for peers' requests.

Authorization

- **Per-Peer, Per-Data Read Access:** A DECADE Client MUST be able to authorize individual peers to read particular data stored on its in-network storage.
- **Per-User Write Access:** A DECADE Client MUST be able to authorize individual peers to store data into its in-network storage.
- **Authorization Checks:** In-network storage MUST check the authorization of a client before it executes a supplied request. The in-network storage MAY use optimizations to avoid such authorization checks as long as the enforced permissions are the the same.

Authorization (continued)

- **Credentials Not IP-Based:** Access MUST be able to be granted on other credentials than the IP address
- **Server Involvement:** A DECADE client MUST be able to indicate, without contacting the server itself, access control policies for peers' requests.

Storage Requirements

- **Explicit Deletion of Stored Data:** DECADE MUST support the ability for a DECADE client to explicitly delete data from its own in-network storage.
- **Multiple Writing:** DECADE MUST NOT allow multiple writers for the same object. Implementations raise an error to one of the writers.
- **Multiple Reading:** DECADE MUST allow for multiple readers for an object.

Storage Requirements (continued)

- **Reading Before Completely Written:** DECADE MAY allow readers to read from objects before they have been completely written.
- **Writing Model:** DECADE MUST provide at least a writing model (while storing an object) that appends data to data already stored.
- **Storage Status:** A DECADE client MUST be able to retrieve current resource usage (including list of stored data) and resource quotas on its in-network storage.

Storage Non-Requirements

- **No Ability to Update:** DECADE SHOULD NOT provide ability to update existing objects. That is, objects are immutable once they are stored.

Implementation Considerations

- **Resource Scheduling:** Need to consider the implications of scheduling on performance. Must be aware of applications' attempts to "game the system".
- **Removal of Duplicate Records:** Systems should be able to remove duplicate records by using shared data, but this is not protocol level decision, and applications should not be forced to do this.