DECADE Problem Statement

draft-song-decade-problem-statement-01

Richard Alimi
Haibin Song
Y. Richard Yang
Ning Zong
P2P Content Distribution Paradigm

- Highly-scalable
- Robust
- Space for innovation
  - Many novel techniques
  - Many players with novel ideas
P2P Contributes Significant Traffic

- 40-70% of total traffic in many networks

Source: ipoque Internet study 2008/2009
P2P Stress on Infrastructure

- Pure overlay distribution is inefficient
  - Transit
  - Backbone
  - Last mile
In-Network Storage

Effective technique to increase efficiency is to introduce *in-network storage*
Problem 1: Weaknesses of Existing P2P Caches

- Tight coupling with P2P application protocol
  - Cache must implement specific protocol for each application
  - Large number of widely-used, evolving P2P protocols
    - File sharing: BitTorrent, eMule, Pando, …
    - Streaming: PPLive, PPStream, UUSee, Zattoo, Kontiki, TVAnts, Sopcast, Abacast, Solid State Networks, OctoShape, …

- Implication
  - Cache vendor and ISP create and support complex production software
Problem 2: Weak/No Integration with Applications

- Caches only consider policy from ISP perspective
  - *Application is out of the loop*
  - However, some P2P applications rely on resource (e.g., bandwidth) allocation amongst peers

- *Implication*
  - *Application requirements/policies not be reflected by caches*
DECADE Overview

- Reduce production complexity and provide open access
- Integrate with application policies
Example Operation

Native P2P Clients

Client A  P2P Control  P2P Data  Client B

Client A  P2P Data (alternative)  Storage S_a  Storage S_b  Client B

DECADE-enabled P2P Clients
Use Case 1: P2P Users Sharing Content

Client A

Client B

Client C

Source

Storage $S_a$

Storage $S_c$

P2P Control

DECADE

Diagram with numbered arrows indicating the sequence of interactions between clients and storage nodes.
Use Case 2: Content Publisher Distributing Content

Client A

Client B

Client C

Storage $S_a$

Storage $S_c$

Content Publisher

P2P Control

DECADE
Key Benefits

- Reduced complexity compared with existing P2P caching
- Integration with application policies
- Robustness and Incremental deployment
  - P2P applications may still use existing mechanisms
- Open access to applications
- Open innovation by applications
Working Group Goal

*Design architecture for P2P content distribution applications to utilize in-network storage*
Scope

**In-scope**

- Requirements and Architecture for P2P applications to utilize in-network storage
  - *Seek rechartering if protocol development needed*
- Consideration for additional content distribution applications
  - Impact to DECADE complexity MUST be considered
- Integration examples with one or two applications

**Out of scope**

- Details of integration with specific applications
- Implementation of policies regarding copyright-protected/illegal content
- Locating the “best” in-network storage
- Development of a new data transport protocol
Comments and questions?
Backup Slides
Key Components of In-network Storage

- Content Storage Mechanism
  - How are P2P contents detected and stored to in-network storage?

- Content Retrieval Mechanism
  - How are P2P contents discovered and read from in-network storage?

- Communication Protocol
  - What is the protocol to communicate with in-network storage?
Existing Solution 1: Transparent P2P Cache

- Content Storage Mechanism
  - DPI detects content; content written to cache
- Content Retrieval Mechanism
  - Cache masquerades as peer
- Communication Protocol
  - Existing P2P protocols
Existing Solution 2: Non-Transparent P2P Cache

- Content Storage Mechanism
  - Cache acts as a peer; content uploaded to it is cached

- Content Retrieval Mechanism
  - Cache acts as a peer; clients download as they would from any other peer

- Communication Protocol
  - Existing P2P protocols