Models for adaptive-streaming-aware CDNI - Introduction

draft-brandenburg-cdni-has-02, section 1 & 2

CDNI Extended Design Team Meeting

Virtual Meeting

June 28, 2012

Ray van Brandenburg (ray.vanbrandenburg@tno.nl)

Why this draft?

- Although CDNI should be content-agnostic, HAS content poses some unique challenges
  - Very large number of (possibly distributed) files
  - Session-less nature makes logging difficult
  - Manifest file poses problems for Request Routing
  - Etc…

- This draft…
  - Is meant to spur discussion on HAS and CDNI
  - Introduces terminology
  - Discusses some of the problem areas when combing HAS and CDNI
  - Introduces different options for level of HAS awareness in CDNI
  - Allows WG to make well-informed decision on which models to support

Differences since -02

- Incorporates many comments and clarifications received during previous conference calls
  - Unique aspects of Live HAS content and dynamic content (e.g. ad insertion)
  - HTTP vs. DNS and effect on HAS optimizations
  - New URL signing section
  - New logging options
  - Request Routing options clarified
  - Etc…
- Provides recommendations
  - Last version only provided options
  - This version includes authors' recommendation on which options to go for

Models for adaptive-streaming-aware CDNI

-

File Management and Content Collections

draft-brandenburg-cdni-has-02, section 3.1

CDNI Extended Design Team Meeting

Virtual Meeting

June 28, 2012

Ray van Brandenburg (ray.vanbrandenburg@tno.nl)

Three candidate approaches for dealing with File Management of HAS content

- Option 1.1: No HAS awareness
  - ‘Do Nothing’-approach
  - dCDN is unaware of relationship between chunks, forced to store chunks as individual files.
- Option 1.2: Allow single file storage of fragmented content
  - Full ‘HAS-awareness’
  - CDNI Metadata Interface signals type of HAS, name of manifest, etc.
  - Allows dCDN to store fragmented content as single file
- Option 1.3: Access correlation hint
  - Add ‘Access Correlation Hint’ to CDNI Metadata of all chunks belonging to same content collection
  - Can be used by dCDN to know which files are likely to be requested after each other in small time window
- **Recommendation:**
  - In initial version of CDNI Interfaces go for Option 1.1
  - Option 1.2 can be considered for re-chartering after initial solution is completed

Models for adaptive-streaming-aware CDNI

-

Content Acquisition and Content Collections

draft-brandenburg-cdni-has-02, section 3.2

CDNI Extended Design Team Meeting

Virtual Meeting

June 28, 2012

Ray van Brandenburg (ray.vanbrandenburg@tno.nl)

Two candidate approaches for dealing with Content Acquisition of HAS content

- Option 2.1: No HAS awareness
  - ‘Do Nothing’-approach
  - dCDN is unaware of relationship between chunks, forced to acquire chunks as individual files
  - Increased overhead

- Option 2.2: Allow single file acquisition of fragmented content
  - Full ‘HAS-awareness’
  - CDNI Metadata Interface signals type of HAS, name of manifest, etc.
  - Allows dCDN to acquire fragmented content as single file

- **Recommendation:**
  - In initial version of CDNI Interfaces go for Option 2.1
  - Option 2.1 can be considered for re-chartering after initial solution is completed

Models for adaptive-streaming-aware CDNI

-

Request Routing and Manifest Files

draft-brandenburg-cdni-has-02, section 3.3

CDNI Extended Design Team Meeting

Virtual Meeting

June 28, 2012

Ray van Brandenburg (ray.vanbrandenburg@tno.nl)

Request Routing and Manifest Files – Recap

- In a sense, Manifest Files can be considered a form of request routing
- Recap, three methods for addressing chunks in a manifest
  - Relative URL (e.g. ‘segments/seg1.ts’)
  - Absolute URL with Redirection (e.g. ‘http://req_routing.cdn……’)
  - Absolute URL without Redirection (e.g. ‘http://surrogate2.cdn….’)
- Some CDNs might prefer one method above the other
  - Some CDNs/CPs might NEED one method (e.g. for security, anti-deeplinking, etc.)
  - [Note: Should this be part of capability exchange?]
- In some cases Content Provider might decide on type of URL used (e.g. in the case where the Content Provider delivers the manifest)
- In some cases delivery of manifest file might be done by Content Provider (invisible to CDN)
- Special attention needs to be had for ‘Live’ manifest files and manifest files containing additional content which might be delivered by other CDN (ad-insertion)

Three candidate approaches for dealing with manifest files and Request Routing

- Option 3.1: No HAS awareness
  - 'Do Nothing'-approach
  - Absolute URLs with Redirection can cause very significant overhead (one full CDNI redirection process for every chunk)
  - Relative URLs support is brittle since dCDN surrogate might not be able to infer that delivery is on behalf of uCDN
  - Absolute URLs without redirection not supported

- Option 3.2: Manifest File rewriting by uCDN
  - Allow uCDN to rewrite manifest file (e.g. change URLs to point to dCDN Request Router)
  - Does not require changes to CDNI Interfaces. Uses existing CDNI RR Interface for obtaining location of dCDN RR (or surrogate)
  - Transparent to dCDN (no HAS awareness required)
  - Can be optional feature (not mandatory for uCDNs)

- Option 3.3: Two-step Manifest File rewriting
  - Also allow dCDN to rewrite manifest file
  - Requires full 'HAS-awareness' on behalf of dCDN
  - Requires changes to CDNI interfaces

Three candidate approaches for dealing with manifest files and Request Routing - 2

- Option 3.1: No HAS awareness
  - ‘Do Nothing’-approach
  - (…)
- Option 3.2: Manifest File rewriting by uCDN
  - Allow uCDN to rewrite manifest file (e.g. change URLs to point to dCDN Request Router)
  - (…)
- Option 3.3: Two-step Manifest File rewriting
  - Also allow dCDN to rewrite manifest file
  - (…)
- **Recommendation:**
  - Mandatory support for Option 3.1
  - Allow Option 3.2 for uCDN that support this
  - Do not support Option 3.3, but mark as candidate for possible re-chartering in the future