SVA CDNI extensions

draft-finkelman-cdni-rr-sva-extensions-01
draft-finkelman-cdni-triggers-sva-extensions-00

Sanjay Mishra, Ori Finkelman

IETF-102, Montreal

July 2018

Refresher

- The Streaming Video Alliance's (SVA) Open Caching (OC) is focused on video-content delivery
  - Open Caching is a specific use case of CDNI where the uCDN is a commercial CDN and the dCDN is an ISP caching layer
- CDNI extensions for Capabilities, Metadata and Triggers are implemented by the SVA Open Caching architecture
  - Motivations and use cases were presented in IETF100 CDNI meeting
  - See relevant slides here: slides-100-cdni-draft-finkelman-cdni-sva-extensions-01
- This presentation gives overview of the two I-D's. One I-D relates to Request Router for footprint and discovery functions and second I-D relates to extensions to the Triggers Interface for content management functions

Agenda

- Request Routing extensions
  - Redirect target address
  - uCDN fallback address
- Control Interface / Triggers extensions
  - Versioning
  - Content selection
  - Trigger generic extensions
  - Extensions specifications: LocationPolicy, TimePolicy
  - Footprint and Capability objects for trigger functions

REQUEST ROUTING

draft-finkelman-cdni-rr-sva-extensions-01

dCDN redirect target capability object

- Requirements:
  - Advertise the redirect target to the uCDN
  - Redirect targets can vary by footprint
  - Redirect target can change over time
  - It should be possible for the dCDN request router to resolve the uCDN redirecting address from the dCDN redirect target address
- **RedirectTarget** Capability object properties
  - **redirecting-hosts** – a list of uCDN hostnames
  - **dns-target** – target address to be used by uCDN for CNAME delegation
  - **http-target** – target URL for HTTP redirect

```
{
    "capabilities": [
     {
      "capability-type": "FCI.RedirectTarget",
      "capability-value": {
        "redirecting-hosts": [
          "a.service123.ucdn.example.com",
          "b.service123.ucdn.example.com"
        ]
        "dns-target": {
          "host": "service123.ucdn.dcdn.com"
        }
        "http-target": {
          <Properties of an HttpTarget object>
        }
      },
      "footprints": [
       <Footprint objects>
      ]
     }
    ]
}
```

Redirect target objects

- **dns-target** properties
  - **host** – hostname without a port number
- **http-target** properties
  - **host** – hostname or IP address with optional port
  - **path-prefix** – add path prefix before the origin path
  - **include-redirecting-host** – add the name of the redirecting host as first path segment
- **Discussion point**: should we use ABNF for construction of the address instead of the proposed “path-prefix” and “include-redirecting-host”?

```
{
    "host": "service123.ucdn.dcdn.com"
}
```

```
{
    "host": "us-east1.dcdn.com",
    "path-prefix": "/cache/1/",
    "include-redirecting-host": true
}
```

uCDN fallback address

- Requirements:
  - Advertise the fallback address to the dCDN
  - Fallback address is specific to a uCDN host / host + path
- **FallbackTarget** Metadata object properties
  - **host** – endpoint, with the limitation that in case of CNAME redirection it can only be a hostname without port number

```
{
    "generic-metadata-type": "MI.FallbackTarget",
    "generic-metadata-value":
     {
       "host": "fallback-a.service123.ucdn.example"
     }
}
```

TRIGGERS INTERFACE

draft-finkelman-cdni-triggers-sva-extensions-00

Triggers interface extensions for video use cases

- Versioning
  - Define versioning scheme for triggers
  - Define version 2 of the trigger commands and objects
  - Advertise version support via FCI
- Select content for trigger by:
  - Regular expression
  - Video playlist (manifest)
- Add generic extension objects
  - List of generic extension objects
  - Initial set of extension objects: LocationPolicy, TimePolicy
  - Support for FCI advertisement for the extension objects

Versioning

- trigger.v2 new properties
  - **content.regexs**
  - **content.playlists**
  - **extensions**

```
{
   "trigger.v2": {
     "type": "invalidate",
     "content.regexs": [ <list of RegexMatch objects> ],
     "content.playlists": [ <list of Playlist objects> ],
     "extensions": [ <list of GenericTriggerExtension objects ]
    },
   "cdn-path": [ "AS64496:1" ]
}
```

- Version 2 of CI/T interface
  - **ci-trigger-command.v2** – use “trigger.v2” instead of “trigger”
  - **ci-trigger-status.v2** – use “trigger.v2” instead of “trigger”
  - **ci-trigger-collection** – no changes
- Version compatibility advertisement
  - Using FCI to advertise CI/T version compatibility
  - More on that later …
- **Discussion point**: is this versioning scheme acceptable? Are there other alternatives?

Content selection by regular expression

- **content.regexs**
  - List of RegexMatch objects
- **RegexMatch** object properties
  - **regex** – regular expression as defined by PCRE
  - **case-sensitive** – true for case sensitive match, otherwise false
  - **match-query-string** – true if matching should be done on query string, otherwise false

```
{
   "regex": "^(https:\\/\\/video\\.example\\.com)\\/([a-z])\\/movie1\\/([1-7])\\/*(index.m3u8|\\d{3}.ts)$",
   "case-sensitive": true,
   "match-query-string": false
}
```

Content selection by playlists

- **content.playlists**
  - List of Playlist objects
- **Playlist** object properties
  - **playlist** – a URL to the playlist file
  - **abr-protocol** – AbrProtocol
- **AbrProtocol** – one of the strings to be registered as AbrProtocol types

```
{
 "playlist": "https://www.example.com/hls/title/index.m3u8",
 "abr-protocol": "hls"
}
```

AbrProtocol Type	Description	Type Specification	Protocol Specification
hls	HTTP Live Streaming	RFCthis	RFC 8216
mss	Microsoft Smooth Streaming	RFCthis	MSS
dash	Dynamic Adaptive Streaming over HTTP (MPEG-DASH)	RFCthis	MPEG-DASH

- **Discussion point:** should we register protocols with version numbers, for example “hls.v7” instead of “hls”?

Generic extensions

- **extensions**
  - List of GenericTriggerExtension objects – can be used for any purpose to manipulate the trigger
- **GenericTriggerExtension** object properties
  - **generic-trigger-extension-type** – string specifying the type of the object contained in the generic-trigger-extension-value property
  - **generic-trigger-extension-value** – specific trigger extension object
  - **mandatory-to-enforce** – flag indicating if the enforcement of this extension is mandatory
  - **safe-to-redistribute** – flag indicating if this trigger extension is safe to redistribute
  - **incomprehensible** – flag indicating if any CDN along the delegation path failed to understand or transform this extension
- **Discussion point:** do we need safe-to-redistribute? Does it make sense to redistribute when an extension is not comprehensible?

```
{
  "generic-trigger-extension-type":
   <Type of this trigger extension object>,
  "generic-trigger-extension-value":
   {
     <properties of this trigger extension object>
   },
  "mandatory-to-enforce": true,
  "safe-to-redistribute": true,
  "incomprehensible": false
}
```

Initial set of extensions objects

- **Location policy**
  - Control the cache locations where a trigger should be execute
  - Examples:
    - Preposition everywhere but NYC
    - Purge from all caches only in SF
- **Time policy**
  - Control the time slot in which a trigger is executed
  - Examples - preposition a new episode at off-peak time a day before release
- **Discussion point:** how to designate time in different time zones, for example off-peak hours?
  - Current proposal is to use TimePolicy in conjunction with LocationPolicy and to execute multiple triggers
  - Alternative option is to use local-time

Location Policy

- **CIT.LocationPolicy** object properties
  - **locations** – a list of LocationRule objects (see Section 4.2.2.1 of [RFC8006])

```
{
      "generic-trigger-extension-type": "CIT.LocationPolicy",
      "generic-trigger-extension-value":
      {
        "locations": [
         {  "action": "allow", "footprints": [ {  "footprint-type": "countrycode",  "footprint-value": ["us"] } ] },
         {  "action": "deny",  "footprints": [ {  "footprint-type": "countrycode",  "footprint-value": ["ca"] } ] }
        ]
      },
      "mandatory-to-enforce": true,
      "safe-to-redistribute": true,
      "incomprehensible": false
}
```

Time Policy

- **CIT.TimePolicy** object properties
  - **window** – TimeWindow object (see Section 4.2.3.2 of [RFC8006])

```
{
      "generic-trigger-extension-type": "CIT.TimePolicy",
      "generic-trigger-extension-value":
      {
            "window": {
                  "start": 946717200,
                   "end": 946746000
             }
       },
      "mandatory-to-enforce": true,
      "safe-to-redistribute": true,
      "incomprehensible": false
}
```

Capability objects for trigger functions

- **FCI.TriggerVersion** properties
  - **versions** – list of supported trigger interface versions represented as strings

```
{
  "capability-type": "FCI.TriggerVersion",
  "capability-value": { "versions": [ "2", "2.1" ] },
  "footprints": [ <Footprint objects> ]
}
```

- **FCI.TriggerPlaylistProtocol** properties
  - **abr-protocols** – list of AbrProtocol types

```
{
  "capability-type": "FCI.TriggerPlaylistProtocol",
  "capability-value": { "abr-protocols": ["hls", "dash"] },
  "footprints": [ <Footprint objects> ]
}
```

- **FCI.TriggerGenericExtension** properties
  - **trigger-extension** – list supported GenericTriggerExtension types

```
{
  "capability-type": "FCI.TriggerGenericExtensionl",
  "capability-value": {
      "trigger-extension": ["CIT.LocationPolicy", "CIT.TimePolicy"]
   },
  "footprints": [ <Footprint objects> ]
}
```

- Q & A
- Next steps ?
- Thank You !