CDNi Request Routing Redirection with Loop Prevention
draft-choi-cdni-req-routing-redir-loop-prevention-01.txt

Taesang Choi (choits@etri.re.kr)
Young-IL Seo (yohan.seo@kt.com)
Dong-Ju Kim (dj.kim@kt.com)
Jongmin Lee (jminlee@sk.com)
Ja-Ryeong Koo (wjbkoo@lguplus.co.kr)
John Dongho Shin (eastsky@solbox.com)
Kyungsoo Park (kspark@kaist.ac.kr)

November 8, 2012
85th IETF, Atlanta
CDNi WG
Update Overview

• Presented at 84th IETF, Vancouver
• First revision based on comments made at 84th IETF
  – Comments on Content-Provider-ID formats
  – Comment on URL length limitation
  – Comment on loop prevention algorithm
• Experimentations & results
Main Changes (1)

• In 00 version, "CDN-Provider-ID" was described as a list of CDN provider Names and MaxNumRedHops
• In 01 version, changed to a list of CDN-Provider-Names followed by MaxNumRedHops.
• Note that a list of CDN-Provider-ID is conveyed in URI string to deal with HTTP URL length limitation
• Example: 
Main Changes (2)

• In 00 version, we specified loop prevention algorithm in pseudo code

• In 01 version, we changed it to specify the following in descriptive form:
  – a mechanism to allow loop detection
  – post processing, that is, who is responsible and in what quality (service availability vs quality) for resolving the situation
Experimentation of Loop Prevention

- Built a PoC test-bed with our consortium members (KT, SKT, LGU+, SolBox)
  - Tested in a simple ring type cascaded topology
- Implemented both Iterative HTTP-/DNS-based request routing redirection. Recursive method is under way
- Objectives
  - Verify the feasibility of the proposed method
  - Measure delays incurred during RRR
  - Impact on the size and transmission performance of redirection messages
Experimentation Findings

- Delays in various hop count settings: 5, 10, …, n hops
  - Iterative vs recursive
  - HTTP vs DNS
- Impact on the size and transmission performance
- Miscellaneous
  - 302 HTTP redirection supports upto 20 redirections
  - DNS CNAME supports upto 38 redirections
  - DNS redirection which retains initiating CDN (uCDN) domain name doesn’t work. Work-around: replace with the immediate parent domain name instead
Delay graph

HTTP Redirection

DNS Redirection
HTTP Redirections Trace

Summary & Next Step

- Minor updates were made based on the comments to 00 version
- Some experiments performed with initial results
- Further tests will be performed and reported in the next IETF
- Propose to merge loop prevention mechanism with request routing redirection draft, draft-he-cdni-routing-request-redirection-03
- Any comments or suggestions for improvements are invited
Chair’s Questions & Answers

• Do you feel that the scheme is well understood (i.e. what has to be signaled, how to encode it, how to process it)?
  – Encoding is simple. Currently part of URI query string or CNAME. It can also be encoded in JSON or other encoding formats
  – Signaling is done currently as a part of HTTP or DNS but can be done by RRRI interface protocol, for example, RRRI request & response message in the he’s draft
  – Processing is a Request Router’s operation behavior which is a part of redirection decision making process. And it is also simple. It can cover both loop prevention and detection with associated post detection processes
  – The same scheme can be equally applicable for both iterative and recursive redirections
  – For scheme’s feasibility, performance impact, we performed experiments with some initial results

• Are you clear that it does not have any impact at all on other interfaces?
  – Loop detection is optional requirement then Capabilities advertisement needs to specify it as a part of capabilities
  – In case of metadata, if operational metadata is specified, loop detection or prevention can be one example.
  – For logging, loop detection & prevention event can be part of logging processes
  – For control interface, not affected by trigger interface but not clear for other control aspects since they are defined yet