

Content Delivery Networks
Interconnection Working Group
Internet-Draft
Intended status: Informational
Expires: March 31, 2013

J. Famaey
S. Latre
UGent - IBBT
September 27, 2012

Experiments on HTTP Adaptive Streaming over interconnected Content
Delivery Networks
draft-famaey-cdni-has-experiments-00

Abstract

This document reports experimental results on the delivery of HTTP Adaptive Streaming (HAS) content over interconnected Content Delivery Networks (CDNs). Specifically, the implications of CDN request routing between CDNs and HTTP redirection on the quality of delivered HAS content are investigated.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on March 31, 2013.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of

the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

- 1. Introduction 3
- 2. Experimental Setup 3
- 3. Results 6
- 4. Conclusion 8
- 5. Security Considerations 9
- 6. References 9
 - 6.1. Normative References 9
 - 6.2. Informative References 9
- Authors' Addresses 9

1. Introduction

HTTP Adaptive Streaming (HAS) refers to a set of novel streaming approaches, which deliver streaming media content over the HTTP protocol. The content is split into chunks, offered in several quality layers. This allows the client to dynamically adapt the requested quality, based on available network resources and device capabilities. Delivering HAS content across multiple interconnected CDNs introduces some new opportunities and challenges. Specifically, it becomes possible to distribute the chunks of a single HAS content stream across multiple CDNs, based on chunk popularity, Quality of Service requirements, resource availability, economic or other factors.

Every HAS content stream is accompanied by a Manifest File, which lists the chunks of each quality layer and specifies their location in the form of a URL. As stated in [I-D.brandenburg-cdni-has], several alternative methods exist for specifying chunk locations:

- o Relative URLs: The URLs specified in the Manifest File are relative to the Manifest File's location and thus all located on the same surrogate.
- o Absolute URLs with Redirection: The Manifest File specifies the fully qualified URL of each chunk. These URLs, however, direct the client towards the CDN's request routing node, which in turn uses HTTP redirection to send the client to the surrogate hosting the actual chunk.
- o Absolute URLs without Redirection: The URL points directly to the surrogate hosting the chunk, effectively allowing the client to circumvent the CDN request routing process.

This document aims to evaluate and compare different request routing policies for HAS content, derived from these addressing mechanisms, that can be used in CDN-interconnection scenarios.

2. Experimental Setup

The scenario used as a basis for the experiments consists of two interconnected CDNs. The downstream CDN is located close to the end-user (e.g., a telco CDN), while the upstream CDN is positioned further (e.g., in the core Internet). The upstream CDN is assumed to be the main storage facility of the original content. As such, it hosts the Manifest file but can offload content chunks to one or more downstream CDNs. Figure 1 graphically depicts the scenario and lists the parameters that were varied in the course of the experiments.

The upstream CDN request router, upstream CDN content server, downstream CDN request router and downstream CDN content server are depicted as uRR, uCS, dRR and dCS, respectively. During the experiments, three parameters were varied: the one-way Internet delay D , the per-client bandwidth B and the HAS client buffer size P . The bandwidth on all other network links was set to 100 Mbps, while the one-way network delay was set to 5 ms. The round trip time (RTT) between two nodes can be calculated as the sum of the one-way delays of the links on the path between them, multiplied by two. In the performed experiments, the RTT between the client and the dRR/dCS is 40 ms, as the path connecting them consists of 4 links. The RTT between the client and the uRR/uCS equals $(60+2D)$ ms, as the path between them contains 6 links and the Internet path. Note that the figure presents a high level, simplified view of the network topology and does not show all individual network links and routers. The processing delay on the CDN surrogates is not taken into account, as it is assumed to be negligible compared to the network delay.

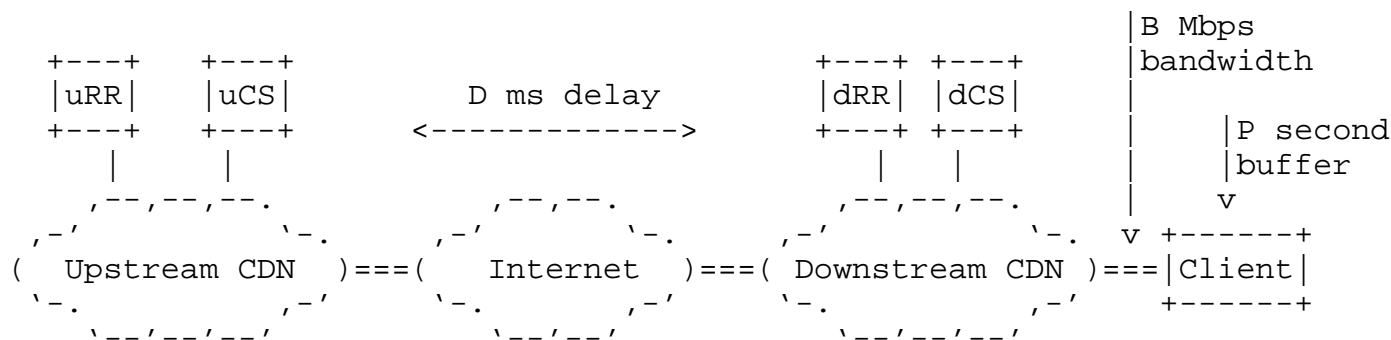


Figure 1: Evaluation scenario and parameters

Three alternative request routing policies are evaluated and compared:

- o **UpstreamRR**: The Manifest File points to the uRR for every chunk. If the chunk is located within the upstream CDN's network, the uRR sends the client a HTTP redirect request to point it to the correct uCS. Otherwise, the uRR redirects the client to dRR, which in turn redirects it to the correct dCS.
- o **DirectRR**: The Manifest File immediately points to the correct request router, which redirects the client to the correct content server. This policy thus allows the client to circumvent going via the upstream CDN's network if the chunk is located downstream.
- o **DirectCS**: The Manifest File immediately points to the correct content server, which allows the client to download segments without being redirected. Compared to the DirectRR policy, the

indirection of contacting the dRR is avoided.

The UpstreamRR policy can be seen as the traditional CDN-I approach, where clients always contact the original CDN and HTTP redirection is used to point them to interconnected CDNs when necessary. It does not require any Manifest File rewriting. Additionally, the upstream CDN does not need any detailed information about chunk locations, as it only needs to redirect clients to the downstream request router. The DirectRR and DirectCS policies are more complex, as they require the upstream CDN to rewrite the original Manifest File. Additionally, when using the DirectCS policy, the downstream CDN either needs to share detailed chunk location information with the upstream CDN or the interconnected CDNs need to collaborate in creating the Manifest File.

The presented policies differ mostly in addressing chunks located in the downstream CDN. As such, the experiments evaluate a scenario where a single client downloads 100 HAS video chunks (of 2 seconds each) from a content server located within the downstream CDN. The constant bitrate video is available in 3 qualities, with bitrates 500 kbps, 1 Mbps and 2 Mbps.

As the end-user Quality of Experience (QoE) depends on several factors, multiple evaluation metrics are used in the comparison:

- o Average played quality: The played quality layer, averaged over all chunks and specified in terms of bitrate. This is expressed in megabits per second (Mbps), representing the bandwidth required for downloading the played quality layers.
- o Total buffer starvation time: The accumulated time during which the client needs to rebuffer the chunks (excluding the original start-up). A rebuffering occurs when a chunk is not available at the client, while it is already required for decoding. This leads to frame freezes, as the client needs to wait for the next chunk to arrive, which significantly reduces QoE.
- o Start-up delay: The time between the initial HTTP request for the first chunk, performed by the client, and the time when the chunk actually starts playing.

All reported results were obtained using the NS-3 simulation environment [ns3] in combination with the Network Simulation Cradle (NSC) [nsc]. NS-3 is a discrete-event network simulator for Internet systems. NSC allows NS-3 to interface directly with the kernel's TCP implementation, generating more accurate and realistic results. The used HAS client rate adaptation algorithm is based on the first version of the client algorithm incorporated in Microsoft's

SmoothStreaming client. The source code of this algorithm can be retrieved from CodePlex [msscode].

3. Results

This section lists and discusses experimental results on the average played video quality and the start-up delay. Results on buffer starvation are omitted, as they did not occur in the depicted scenarios.

Figure 2 depicts the average played quality (in Mbps) as a function of the client bandwidth B , client buffer size P and one-way Internet delay D . The depicted results show that, as expected, the delivered quality for the DirectRR and DirectCS policies is independent of the network delay D between the Interconnected CDNs. On the other hand, when using the standard UpstreamRR policy, quality of the delivered HAS content degenerates significantly as the delay increases. The exact breakpoint at which quality starts degrading does depend on other factors, of which the available bandwidth is the most important. Specifically, at a bandwidth of 5 Mbps, significant quality reductions are visible for upstream CDN network delays of more than 100 ms, while this breakpoint increases to over 200 ms for a bandwidth of 100 Mbps.

B (Mbps)	P (s)	D (ms)	Average played quality (Mbps)				
			UpstreamRR	DirectRR	DirectCS		
5	6	50	1.94	1.96	1.96		
		100	1.94	1.96	1.96		
		200	0.98	1.96	1.96		
		400	0.50	1.96	1.96		
	24	50	1.93	1.98	1.98		
		100	1.86	1.98	1.98		
		200	0.94	1.98	1.98		
		400	0.50	1.98	1.98		
		100	6	50	1.96	1.98	1.98
				100	1.94	1.98	1.98
200	1.94			1.98	1.98		
400	0.50			1.98	1.98		
24	50		1.96	1.98	1.98		
	100		1.96	1.98	1.98		
	200		1.88	1.98	1.98		
	400		0.50	1.98	1.98		

Figure 2: The average played quality as a function of client bandwidth B, client buffer size P and one-way delay D

Results on start-up delay are depicted in Figure 3. As the results are minimally influenced by the available bandwidth B and client buffer P, start-up delays are only shown for B equal to 5Mbps and P equal to 6s.

D (ms)	Start-up delay (s)		
	UpstreamRR	DirectRR	DirectCS
50	0.90	0.58	0.47
100	1.10	0.58	0.47
200	1.50	0.58	0.47
400	2.30	0.58	0.47

Figure 3: The start-up delay as a function of one-way delay D , for $B = 5\text{Mbps}$ and $P = 6\text{s}$

In line with the played quality, start-up delay is not negatively influenced by an increasing network delay between the interconnected CDNs when using the DirectRR and DirectCS policies. However, when using the UpstreamRR policy, the start-up delay increases linearly with the network delay D . Note that this start-up delay occurs whenever a new stream is requested. This occurs, for example, when switching channels in Internet TV scenarios as well as when users skips parts of a movie in a Video on Demand scenario.

4. Conclusion

In this document, we proposed, evaluated and compared several policies for routing requests and retrieving HAS content chunks distributed across multiple interconnected CDNs. Concretely, the traditional policy, herein called UpstreamRR, in which the original CDN's request router dynamically redirects the end-users towards the CDN currently hosting the requested content, is compared to two novel policies, called DirectRR and DirectCS. These novel policies employ HAS Manifest File rewriting to directly point end-users to the correct CDN (DirectRR) or even the correct content server (DirectCS).

A thorough evaluation, based on NS-3 simulation results, was conducted. It shows that the end-user QoE suffers greatly as a consequence of the HTTP redirects that occur when employing the standard UpstreamRR policy. Depending on the available bandwidth, QoE degradation can start occurring when the one-way network delay towards the upstream CDN is greater than 100 milliseconds. In contrast, the reported results also show that the novel DirectRR and DirectCS policies perform well under increasing network delays.

In summary, these results prove the need for advanced request routing mechanisms, as well as extensive cooperation between interconnected CDNs, to be able to satisfy end-user quality requirements of state-of-the-art HAS-based services.

5. Security Considerations

Not applicable.

6. References

6.1. Normative References

[I-D.brandenburg-cdni-has]

Brandenburg, R., Deventer, O., Faucheur, F., and K. Leung, "Models for adaptive-streaming-aware CDN Interconnection", draft-brandenburg-cdni-has-03 (work in progress), July 2012.

6.2. Informative References

[msscode] "Microsoft's SmoothStreaming rate adaptation algorithm v1 source code", <<https://slextensions.svn.codeplex.com/svn/trunk/SLExtensions/AdaptiveStreaming/>>.

[ns3] "NS-3 discrete event network simulator", <<http://www.nsnam.org/>>.

[nsc] "Network Simulation Cradle", <<http://research.wand.net.nz/software/nsc.php>>.

Authors' Addresses

Jeroen Famaey
Ghent University - IBBT
Gaston Crommenlaan 8/201
Ghent 9050
Belgium

Phone: +32 9 331 49 38
Email: jeroen.famaey@intec.ugent.be

Steven Latre
Ghent University - IBBT
Gaston Crommenlaan 8/201
Ghent 9050
Belgium

Phone: +32 9 331 49 88
Email: steven.latre@intec.ugent.be