Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 19, 2015

D. Singer
Apple, Inc.
A. Begen
Cisco

October 16, 2014

URLs and HTTP Response Forms for Multicast
draft-singer-appsawg-mcast-url-00

Abstract

This document motivates the need for defining a URL for multicast delivery and provides a proposal for this purpose.

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 April 19, 2015.

Copyright Notice

Table of Contents

1. Introduction 2
2. Glossary of Terms 4
3. Use Cases 4
4. Required Information 4
5. Suggested URL Form 5
5.1. Introduction 5
5.2. Information on FLUTE (RFC 6726) 5
5.3. URL Form Requirements 6
5.4. Base URL Form 6
6. FCAST Metainformation field 8
7. HTTP Status Codes and Their Applicability to FCAST 8
7.1. Useful 8
7.2. Unlikely but Possible 9
7.3. Probably Inapplicable 9
8. Operation of the URL Handler 10
9. Security Considerations 10
10. IANA Considerations 11
11. References 11
11.1. Normative References 11
11.2. Informative References 11
Authors' Addresses 11

1. Introduction

Various multicast file-delivery protocols are defined by the IETF and 3GPP (notably File Delivery over Unidirectional Transport (FLUTE) and FCAST). However, they are hard to adopt into other services, partly because they do not follow conventions on how these transports are addressed and what information they deliver. Notably:

1. Much of the Web (Internet) assumes that if a file can be used, it can be referred to by a URL that contains enough information to start to try retrieving it. This is not true for files available over multicast.

2. When a URL form is used, it can be annotated with the information on what it refers to (e.g., a MIME type, a codecs parameter for that MIME type, and so on). If we have no URL, we cannot annotate it.

3. HTTP header responses are widely used to signal the unavailability of an expected resource (404) or that a resource has moved (re-direct), or that there are other choices to retrieve the indicated resource, or to deliver portions of a resource (byte-ranges). Though FCAST uses the metainformation format of HTTP, it misses the status line, so neither unavailability nor re-direct can be signaled. You cannot re-direct from multicast to HTTP, for example.

4. 'Soft' information such as multicast group addresses, transport session identifiers, and so on, are hard-coded into the descriptions (e.g., Session Description Protocol (SDP) files [RFC4566]). In general, the Web/Internet avoids hard-coding such values, preferring to use lookup (e.g., DNS for addresses); lookups can be re-factored as boundaries are crossed.

Traditionally multicasts have been addressed by requiring the client to acquire some pre-knowledge (e.g., an SDP file) by some means out of band. Thus, we require that every protocol that might use multicast be adapted. This is error-prone, limiting, and time-consuming. Instead, if an operating system can have a URL handler for multicast URLs that deliver file objects, with an interface that 'emulates' the interface to HTTP, many (not all) things would 'just work'. Perhaps the most notable is that we might re-direct from HTTP to multicast when the server detects that there is a better way to get the resource (perhaps, at this time and for this client).

The places where URLs occur, and where it would be advantageous to be able to state "this file is available on multicast", are legion. Obvious examples include anything linked into HTML (a Web page or email), especially media (video, audio, images); in HTTP itself where re-direction supplies a URL, and HTTP adaptive streaming systems where many clients could be fetching the same set of content segments. For many of these, operating system support with the same API as HTTP would suffice. Even in the HTTP adaptive streaming case, where it is true that the streaming engine needs to know it is using multicast (as this would make substantial changes to bandwidth estimation, etc.), simplifying the markup and the protocol identification to a URL is a plus. FCAST is closer to HTTP operation than FLUTE; files 'just arrive' and there is no concept of the 'set' as represented by the file delivery table in FLUTE. We therefore focus on FCAST in this document.

2. Glossary of Terms

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. Use Cases

Here are two example use cases.

1. The classic stadium. A sports franchise wants everyone in the stadium to be able to watch a few selected camera streams. They multicast the streams over a tuned WiFi system.

2. A network operator (either Internet service or 3G/4G) wants to enable people to see a video mosaic of the top channels, and click through to get to a channel fast.

The simple solutions are:

1. Provide a QR code that embeds a multicast URL linking to the manifest file for the video content at the entrance, in printed material, on posters, etc. When people tune to that multicast URL with their phones, they get the manifest, and it refers to streams that are also multicast. The act of tuning into the session starts the client caching everything that arrives.

2. The mosaic is a multicast URL, and the segments of each program are also multicast but with short cache-times, and using the same URL label as the unicast address (i.e. ,an HTTP URL). When the user clicks on a program, they fetch the manifest (or perhaps the manifests are also multicast and pre-filled into the cache) and they already have the current segment cached, so startup is effectively instant. As they proceed, there is a good chance the multicast has delivered every segment they need, just in time.

4. Required Information

Currently, tune-in to a multicast involves getting hold of a ’head’ file that gives a variety of information. The possible information can be roughly separated into different classes:

1. Information about alternatives that could be supplied as part of the higher-level protocol (e.g., different representations in HTTP adaptive streaming and HTML5 source elements)

2. Information (IP addresses and the like) that is needed to 'bootstrap' the multicast reception

3. Information about where/how the reception is possible (e.g., protocol parameters, time-ranges, and so on)

4. Information that could be acquired later, in-band, such as feedback addresses, the availability of alternatives and unicast repair servers, and so on (or indeed, a fuller description of the multicast itself)

For the sake of simplicity, we propose that we only include (2) and (3) in the URL form.

Ideally, there is something about the multicast itself that allows the client system to assess fairly rapidly whether it is working (the multicast join succeeded, packets are arriving, etc.) and if that fails, the URL handler can give a suitable error indication (maybe an existing one, maybe new).

5. Suggested URL Form

5.1. Introduction

Both FLUTE and FCAST rely on Asynchronous Layered Coding (ALC) [RFC5775] / Layered Coding Transport (LCT) [RFC5651], which in turn has the concept of channels to handle congestion and rate control. We presume the existence of a base channel and indicate how to acquire that.

In an FCAST session, files are identified by URI labels. We suggest that we identify a reserved URN form to indicate 'this is a complete SDP file describing all the sessions'. This allows bootstrapping from the base channel to all of the channels in a session.

FLUTE [RFC6726] is specific about the parameters needed to acquire an ALC/LCT session, and since FCAST [RFC6968] also relies on ALC/LCT, the same analysis applies.

5.2. Information on FLUTE (RFC 6726)

To start receiving a file delivery session, the receiver needs to know transport parameters associated with the session. Interpreting these parameters and starting the reception therefore represents the entry point from which thereafter the receiver operation falls into the scope of this specification. According to [RFC5775], the transport parameters of an ALC/LCT session that the receiver needs to know are:

- The source IP address.
- The number of channels in the session.
- The destination IP address and port number for each channel in the session.
- The transport session identifier (TSI) of the session.
- An indication that the session is a FLUTE session. The need to demultiplex objects upon reception is implicit in any use of FLUTE, and this fulfills the ALC requirement of an indication of whether or not a session carries packets for more than one object (all FLUTE sessions carry packets for more than one object).

5.3. URL Form Requirements

We have at least the following requirements:

- The URLs must be valid according to the RFCs and recent work at the W3C
- The absolute form must exist (obviously)
- Relative URLs must also work
- We should avoid the fragment (#) and query suffices (?) even though, in the latter case, there is no server that the URL is sent to.
- We should permit the URL to self-declare its validity period (and thus enable rapid time-outs when it is requested outside this period)
- Ideally, we also allow it to indicate its ’geographic’ (operator network) availability scope

5.4. Base URL Form

This suggests a URL form in three parts:

- A prefix giving the URL scheme and basic parameters
- A mid-part giving the temporal and geographic scope
- A suffix that is the label of the desired file

Where the prefix is roughly like

fcast://destination:port/source:TSI

with

destination: An explicit multicast address (x.y.z.w) or (better) a name that resolves to one (or more) IP multicast address(es) for the base channel.

port: Port number for the base channel.

source: An explicit IP address (x.y.z.w) or (better) a name that resolves to the source address.

TSI: The transport session identifier for the session.

The mid-part has optional terms that are each formatted as / key:value. The keys and their values are:

start: the absolute start-time of availability

end: likewise, the end time

network: an identification of the network(s) on which the multicast is made available (for the indicated time-span, if any)

The start and end times are each optional and if present are expressed exactly as in SDP, i.e. as the decimal representation of Network Time Protocol (NTP) time values in seconds since 1900. If the URL agent determines it is operating outside this time range, a suitable error SHOULD be returned immediately. If either the start or end time are absent, then the multicast starts (or stops) at an indefinite time.

The network attribute takes a list of domain names, joined by the plus sign; if the URL handler is confident that the machine is not on any of the networks, a suitable error SHOULD be returned immediately, as it knows the multicast reception will not succeed.

The suffix starts with the special key /label: and is followed by the label of the desired file. (We retain at least the forward slashes in the path in the clear so that relative URLs work, but perhaps some characters and maybe some instances of slash should be escaped.)

Example:

fcast://232.0.0.1:5620/broadcast.example.com:527353/start:35776638264 /network:media.example.com/label:http://news.example.com/stuff.mp4

given such a URL, the terminal can (try to) tune into the FCAST session, and retrieve the indicated file.

6. FCAST Metainformation field

In FCAST the metainformation field carries anything that an HTTP metainformation field can carry, but not the status line. This means it is not possible to indicate "this file might be expected here, but it is not here any more" (404) or "this file has moved" (301 or 307) or even that there are multiple choices on where to get this resource (300 'choices'). The most useful, perhaps, is the ability to indicate "you might have expected to get this over this multicast, but it's not here, but over there (re-direct)" perhaps even re-directing back to HTTP, or to another multicast session.

We therefore suggest we define a new form of the FCAST metainformation that also includes a status line formatted exactly as the HTTP status line, but with the HTTP-version replaced by FCAST-version:

Status-Line = FCAST-Version SP Status-Code SP Reason-Phrase CRLF

7. HTTP Status Codes and Their Applicability to FCAST

Here are the status codes available in HTTP 1.1, and a brief statement of whether they could be applicable to FCAST:

7.1. Useful

- 200 OK: Usual status code when the object is supplied, or when just the metainformation is supplied
- 203 Non-Authoritative Information
- 206 Partial Content: Useful to indicate that byte-ranges of the resource are supplied separately
- 300 Multiple Choices: Useful to indicate that there are also other places to get the content
- 301 Moved Permanently: The resource might be expected here, but has moved (re-direct)
- 302 Found
- 303 See Other

- 307 Temporary Redirect: The resource might be expected here, but has moved (re-direct)
- 404 Not Found: The resource might be expected here, but it is no longer available
- 410 Gone

7.2. Unlikely but Possible

- 100 Continue: Unlikely to be of use
- 101 Switching Protocols: Maybe useful
- 502 Bad Gateway: The multicast was being fed by a gateway that failed
- 503 Service Unavailable

7.3. Probably Inapplicable

- 201 Created
- 202 Accepted
- 204 No Content
- 205 Reset Content
- 304 Not Modified
- 305 Use Proxy
- 400 Bad Request
- 401 Unauthorized
- 402 Payment Required
- 403 Forbidden
- 405 Method Not Allowed
- 406 Not Acceptable
- 407 Proxy Authentication Required
- 408 Request Time-out

- 409 Conflict
- 411 Length Required
- 412 Precondition Failed
- 413 Request Entity Too Large
- 414 Request-URI Too Large
- 415 Unsupported Media Type
- 416 Requested range not satisfiable
- 417 Expectation Failed
- 500 Internal Server Error
- 501 Not Implemented
- 504 Gateway Time-out
- 505 HTTP Version not supported

8. Operation of the URL Handler

When the client-side URL handler gets the first URL for a given session, it would 'tune in that session' and (with luck) start receiving files and metainformation. On the receipt of 'special files' (e.g., an SDP) it can expand its knowledge of the session. Other files not corresponding to the immediate request in hand should be cached, observing the cache control headers. When the indicated file (or at least the requested byte-range of the indicated file) is available, it is returned. If a 404, 410, or 3xx response is received for the indicated file, then an appropriate error is returned, as indeed it is if the URL specifies that the multicast is only available over a given time range, and the request is not or cannot be satisfied in that time range.

9. Security Considerations

TBC.

10. IANA Considerations

This section contains the registration information for the "fcast" URI scheme (in accordance with Section 5.4 of [RFC4395]).

Editor's note: The registration template will be provided in a later revision.

11. References

11.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC6968] Roca, V. and B. Adamson, "FCAST: Object Delivery for the Asynchronous Layered Coding (ALC) and NACK-Oriented Reliable Multicast (NORM) Protocols", RFC 6968, July 2013.

[RFC4395] Hansen, T., Hardie, T., and L. Masinter, "Guidelines and Registration Procedures for New URI Schemes", BCP 35, RFC 4395, February 2006.

11.2. Informative References

[RFC5775] Luby, M., Watson, M., and L. Vicisano, "Asynchronous Layered Coding (ALC) Protocol Instantiation", RFC 5775, April 2010.

[RFC5651] Luby, M., Watson, M., and L. Vicisano, "Layered Coding Transport (LCT) Building Block", RFC 5651, October 2009.

[RFC6726] Paila, T., Walsh, R., Luby, M., Roca, V., and R. Lehtonen, "FLUTE - File Delivery over Unidirectional Transport", RFC 6726, November 2012.

[RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, July 2006.

Authors' Addresses

Dave Singer
Apple, Inc.
1 Infinite Loop
Cupertino 95014
USA

EMail: singer@apple.com

Ali Begen
Cisco
181 Bay Street
Toronto, ON M5J 2T3
Canada

EMail: abegen@cisco.com