

HTTP Random Access and Live Resources

IETF 100

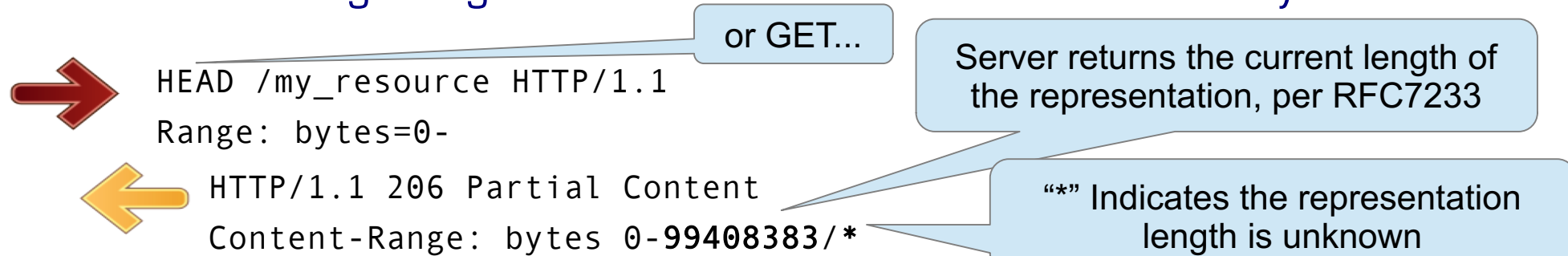
Barbara Stark
bs7652@att.com

Darshak Thakore
d.thakore@cablelabs.com

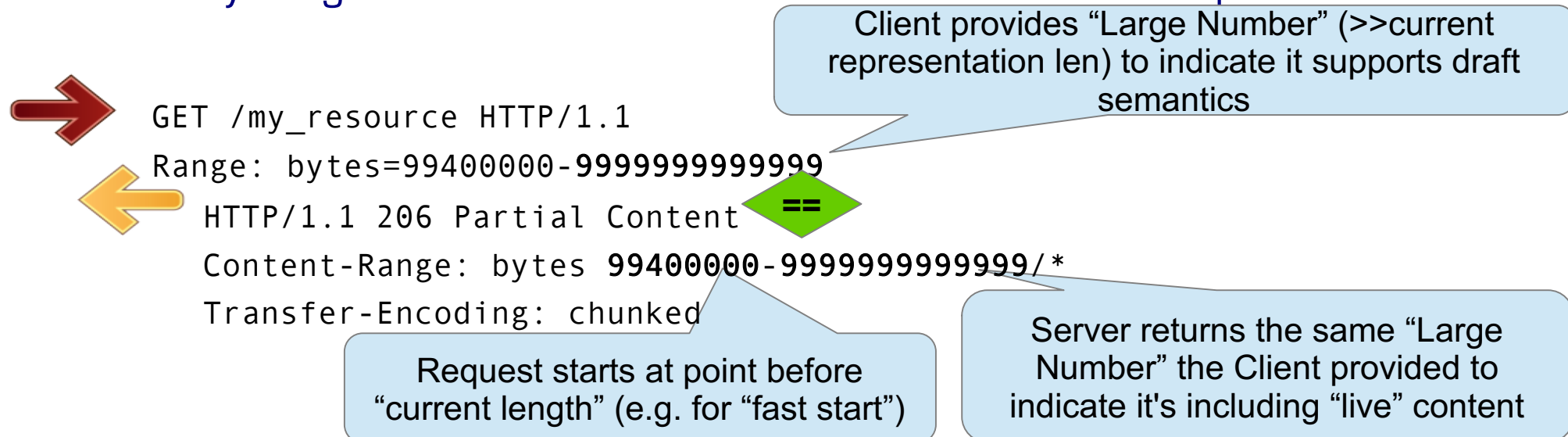
Craig Pratt
pratt@acm.org / craig@ecaspia.com

Use existing bytes Range Unit with “very large” numbers (draft-ietf-httpbis-rand-access-live)

- Clients use existing Range semantics to determine accessible bytes:



- and use Very Large numbers to indicate an indeterminate endpoint:



Interoperability with Servers/Proxies that do not support draft semantics

- Client uses existing semantics to determine accessible byte range (as before):

→ HEAD /my_resource HTTP/1.1
Range: bytes=0-

or GET...

Server returns the current length of the representation, per RFC7233

← HTTP/1.1 206 Partial Content
Content-Range: bytes 0-99408383/*

“*” Indicates the representation length is unknown

- Server instead returns the current length:

→ GET /my_resource HTTP/1.1
Range: bytes=99400000-999999999999
← HTTP/1.1 206 Partial Content
Content-Range: bytes 99400000-99410000/*
Transfer-Encoding: chunked

Client provides “Large Number” (>>current representation len) to indicate it supports draft semantics

Server returns the current representation length, indicating to the client that it doesn't support live range requests

Request starts at point before “current length” (e.g. for “fast start”)

Prototype Server

- Implemented a simple HTTP 1.1 Server compliant with the draft RFC hosting a continuously-buffering representation of a live NASA-TV stream
 - Live resource directly accessible via
 - <http://ietf100.ecaspia.com:8000/live/nasatv.ts>
 - Varnish reverse proxy (v4.1.1)
 - <http://ietf100.ecaspia.com:6081/live/nasatv.ts>
 - Squid reverse proxy (3.5.12)
 - <http://ietf100.ecaspia.com:3128/live/nasatv.ts>
 - Source jar:
 - <http://ietf100.ecaspia.com:8000/static/live-range-server.src.jar>
 - Runtime jar:
 - <http://ietf100.ecaspia.com:8000/static/live-range-server.jar>

Direct Access Example Session

```
$ curl -I -H "Range: bytes=0-" http://ietf100.ecaspia.com:8000/live/nasatv.ts
HTTP/1.1 206 Partial Content
Content-range: 0-15622047/*
Date: Thu, 16 Nov 2017 06:00:21 GMT
Content-type: video/mp2t
Accept-ranges: bytes
Cache-control: max-age=3600
```

"*" Indicates the representation length is unknown. Can check to see if the server supports live-range RFC

```
$ curl -v -H "Range: bytes=15600000-9999999999"
http://ietf100.ecaspia.com:8000/live/nasatv.ts > out.ts
< HTTP/1.1 206 Partial Content
< Content-range: 15600000-9999999999/*
< Date: Thu, 16 Nov 2017 06:00:50 GMT
< Transfer-encoding: chunked
< Content-type: video/mp2t
< Accept-ranges: bytes
< Cache-control: max-age=3600
```

Server returning the same large value (>>>15622047) indicates the server supports live-range requests on this resource

Direct Access Example Session (cont)

- Server feeds data as it's made available:

Response: 206

```
#####.....#####.....
.#####.....#####.....#####.....
.....#####.....#####.....
.#####.....#####.....#####
#####.....#####.....#####
.....#####.....#####.....#####
###.....#####.....#####.....
```

- Client (curl in this case) sees bitrate go up and down as data is aggregated:

100	2744k	0	2744k	0	0	222k	0	--:--:--	0:00:12	--:--:--	136k
100	3468k	0	3468k	0	0	141k	0	--:--:--	0:00:24	--:--:--	153k
100	4156k	0	4156k	0	0	113k	0	--:--:--	0:00:36	--:--:--	0
100	4848k	0	4848k	0	0	99284	0	--:--:--	0:00:50	--:--:--	0
100	6232k	0	6232k	0	0	97368	0	--:--:--	0:01:05	--:--:--	150k
100	6940k	0	6940k	0	0	91083	0	--:--:--	0:01:18	--:--:--	0
100	8328k	0	8328k	0	0	99803	0	--:--:--	0:01:25	--:--:--	314k

- Data will be continuously downloaded until the client cancels the transfer

Testing Matrix

	Status	Comments
Varnish	Works (kinda)	Does caching but as static content (limited caching)
Varnish (range request support)	Works	Works as a straight thru proxy, no caching
Squid (reverse proxy)	Works	Does not cache, always get a cache miss, goes to origin server every time
CloudFlare	-	Still testing

Varnish Reverse Proxy Results - Standard Config

- Default Varnish configuration will make a live server look like a static content server. But is byte-wise coherent. e.g.

```
$ curl -v -H "Range: bytes=0-9999999999" http://ietf100.ecaspia.com:6081/live/nasatv.ts
< HTTP/1.1 206 Partial Content
< Date: Thu, 16 Nov 2017 07:05:24 GMT
< Content-type: video/mp2t
< X-Varnish: 32772 3
< Age: 126
< Via: 1.1 varnish-v4
< Accept-Ranges: bytes
< Content-Range: bytes 0-24109496/24109497
< Content-Length: 24109497
< Connection: keep-alive
<
{ [1188 bytes data]
100 22.9M 100 22.9M 0 0 1089k 0 0:00:21 0:00:21 --:--:-- 1153k
```


Varnish Reverse Proxy Results + Range Support

- Server configured with Range support per <https://info.varnish-software.com/blog/caching-partial-objects-varnish> is fully functional with live ranges

```
$ curl -v -H "Range: bytes=0-9999999999" http://ietf100.ecaspia.com:6081/live/nasatv.ts
< HTTP/1.1 206 Partial Content
< Date: Thu, 16 Nov 2017 07:33:20 GMT
< Content-type: video/mp2t
< Cache-control: max-age=3600
< X-Varnish: 32770 3
< Age: 13
< Via: 1.1 varnish-v4
< Content-Range: 0-9999999999/*
< Transfer-Encoding: chunked
< Connection: keep-alive
```

Varnish Reverse Proxy Results + Range Support (cont)

- As is the case with direct (non-proxied) access, random access content is downloaded at higher bitrate, and reduces once the live point is hit

100	22.1M	0	22.1M	0	0	1464k	0	--:--:--	0:00:15	--:--:--	1532k
100	41.3M	0	41.3M	0	0	1391k	0	--:--:--	0:00:30	--:--:--	1702k
100	61.8M	0	61.8M	0	0	1392k	0	--:--:--	0:00:45	--:--:--	1303k
100	80.3M	0	80.3M	0	0	1360k	0	--:--:--	0:01:00	--:--:--	1214k
100	96.6M	0	96.6M	0	0	1311k	0	--:--:--	0:01:15	--:--:--	1040k
100	116M	0	116M	0	0	1315k	0	--:--:--	0:01:30	--:--:--	1807k
100	137M	0	137M	0	0	1336k	0	--:--:--	0:01:45	--:--:--	1427k
100	145M	0	145M	0	0	1230k	0	--:--:--	0:02:00	--:--:--	147k
100	145M	0	145M	0	0	1106k	0	--:--:--	0:02:15	--:--:--	0
100	147M	0	147M	0	0	1004k	0	--:--:--	0:02:30	--:--:--	132k
100	147M	0	147M	0	0	914k	0	--:--:--	0:02:45	--:--:--	0
100	149M	0	149M	0	0	846k	0	--:--:--	0:03:00	--:--:--	111k
100	149M	0	149M	0	0	836k	0	--:--:--	0:03:02	--:--:--	137k

- Data returned is byte-wise coherent with the origin server
- Varnish may cache too much data without additional configuration
 - It appears to continue trying to buffer the entire requested range - probably needs to be bounded...

Squid Reverse Proxy Results

Standard Disk Caching Config

- Seems to support the live range semantics

```
$ curl -v -H "Range: bytes=0-9999999999" http://ietf100.ecaspia.com:6081/live/nasatv.ts
< HTTP/1.1 206 Partial Content
< Date: Thu, 16 Nov 2017 07:33:20 GMT
< Content-type: video/mp2t
< Cache-control: max-age=3600
< X-Varnish: 32770 3
< Age: 13
< Via: 1.1 varnish-v4
< Content-Range: 0-9999999999/*
< Transfer-Encoding: chunked
< Connection: keep-alive
```

Squid Reverse Proxy Results

Standard Disk Caching Config (cont)

- And supports live range transfer:

100	22.1M	0	22.1M	0	0	1464k	0	--:--:--	0:00:15	--:--:--	1532k
100	41.3M	0	41.3M	0	0	1391k	0	--:--:--	0:00:30	--:--:--	1702k
100	61.8M	0	61.8M	0	0	1392k	0	--:--:--	0:00:45	--:--:--	1303k
100	80.3M	0	80.3M	0	0	1360k	0	--:--:--	0:01:00	--:--:--	1214k
100	96.6M	0	96.6M	0	0	1311k	0	--:--:--	0:01:15	--:--:--	1040k
100	116M	0	116M	0	0	1315k	0	--:--:--	0:01:30	--:--:--	1807k
100	137M	0	137M	0	0	1336k	0	--:--:--	0:01:45	--:--:--	1427k
100	145M	0	145M	0	0	1230k	0	--:--:--	0:02:00	--:--:--	147k
100	145M	0	145M	0	0	1106k	0	--:--:--	0:02:15	--:--:--	0
100	147M	0	147M	0	0	1004k	0	--:--:--	0:02:30	--:--:--	132k
100	147M	0	147M	0	0	914k	0	--:--:--	0:02:45	--:--:--	0
100	149M	0	149M	0	0	846k	0	--:--:--	0:03:00	--:--:--	111k
100	149M	0	149M	0	0	836k	0	--:--:--	0:03:02	--:--:--	137k

- Caching and coalescing of byte ranges doesn't appear functional. But the data returned is byte-wise coherent with the origin server

Questions

Comments

Discussion