DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

Dynamic Adaptive Streaming over HTTP

DASH

Luca Mazzon 1151499

Luca Vallerini 1110975

19 September 2017

Computer Networks - Academic Year 2016/2017

Contents

We organized our chapter in two parts:

1. MPEG-DASH protocol breakdown and design principles, with references to other protocols
2. Analysis of the Adaptive Bit Rate algorithm SARA

What is DASH

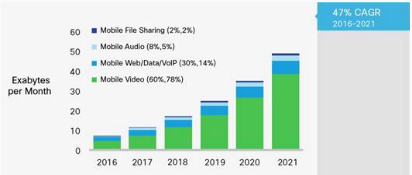
Dynamic Adaptive Streaming over HTTP (DASH) is a video streaming protocol which is capable of dynamically adapt to the conditions of the system in order to provide seamless video playback.

Different protocols with the same conceptual level:

- A video is taken from a source and it’s encoded with different bitrates;
- Encoded videos are hosted on HTTP servers and fragmented;
- A client progressively downloads and plays back the segments choosing dynamically the best bitrate for the network condition.

Why is DASH important?

Cisco Visual Networking Index Mobile Data Forecast for 2021 expect that video data will account for 78% of the mobile data traffic (more than 38 Exabytes per month).



An increase in demand and in quality expectation from users require appropriate advanced technologies like DASH.

Different protocols, same idea

Company	Protocol
MPEG	MPEG-DASH (ISO/IEC23009)
3GPP	Adaptive HTTP Streaming (AHS) (TS 26.233 ...)
Apple	HTTP Live Streaming (Internet-Draft, Informational)
Microsoft	Smooth Streaming Protocol (MS-SSTR)
Adobe	HTTP Dynamic Streaming

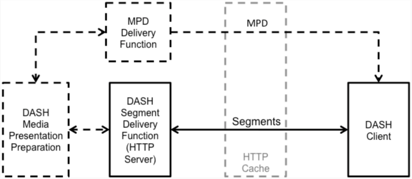
MPEG-DASH Data Model

Three core elements:

- **Media Presentation Description**: XML formatted file which describes the Media Presentation, which is a bounded or unbounded list of HTTP URLs to the Segments which compose the video to be streamed;
- **Segment**: content response of an HTTP GET or partial HTTP GET. A Segment can contain both media data and metadata.
- **Period**: media content period during which the encoded versions of the media are consistent.

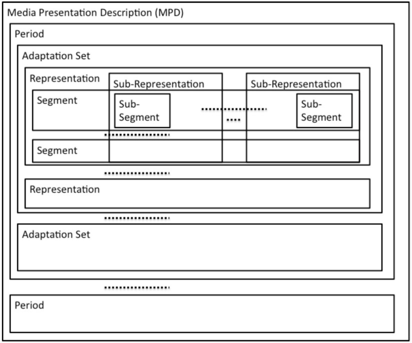
The content within a Period is arranged in an **Adaptation Set**, which contains different **Representations** of the media. Representations are deliverable encoded versions of one or several media content components, such as different resolutions, subtitles, languages and captions. Representations are then divided into Segments.

MPEG-DASH MPD structure and streaming Schema

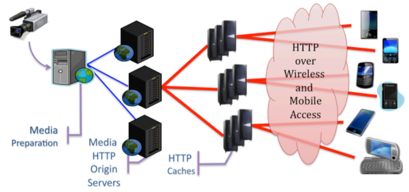

MPD Delivery Function
MPD
DASH Media Presentation Preparation
DASH Segment Delivery Function (HTTP Server)
Segments
HTTP Cache
DASH Client

The client first requests the MPD and then progressively requests the Segments to be played back, until the broadcast ends or the user interrupts the process. The MPD is periodically refreshed to update the list of URLs.

MPD Structure



Why HTTP?



HTTP over Wireless and Mobile Access
Media Preparation
Media HTTP Origin Servers
HTTP Caches

HTTP is the most used protocol for content delivery and communication over WWW, so preexisting CDNs can be exploited and there aren't any firewall issues.

1. Offload the central server with peripheral servers;
2. Faster media access with HTTP Caches;
3. Stateless connection: no extra space on server and data overhead for maintaining state, the client can decide autonomously initial bitrate and variations without having to negotiate with the server (faster adaptation).

ABR algorithm and SARA

Adaptive Bit Rate (ABR) algorithms are designed with three goals in mind:

- Maximize efficiency
- Minimize re-buffering
- Stability

Segment Aware Rate Adaptation (SARA) proposes to enhance the MPD file with the size of each segment and then estimates the throughput based also on this information.

SARA: throughput estimate

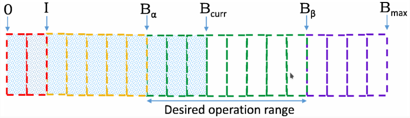
To better estimate the throughput SARA uses the Harmonic mean. For a generic segment i a weight w_i proportional to the segment size is assigned. Call d_i the download rate of segment i. Then, the weighted harmonic mean download rate of the first n segments is

$$H_n = \frac{\sum_{i=1}^{n} w_i}{\sum_{i=1}^{n} \frac{w_i}{d_i}}$$

SARA can then estimate the download time of the next segment as $\hat{d}_{n+1} = w_{n+1}/H_n$.

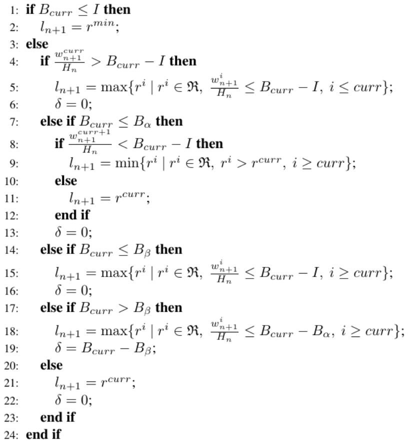
SARA: buffer

The buffer B stores each downloaded segment and feeds the video player with them.



SARA identifies four thresholds (I, B_α, B_β and B_{max}) and it uses them to decide which representation to download for the next segment. The idea is to keep the buffer in the $[B_\alpha, B_\beta]$ range.

SARA: algorithm



```
if $B_{curr} \le I$ then
    $l_{n+1} = r^{min}$;
else
    if $\frac{w_{n+1}^{curr}}{H_n} > B_{curr} - I$ then
        $l_{n+1} = \max\{r^i \mid r^i \in \mathfrak{R},\ \frac{w_{n+1}^i}{H_n} \le B_{curr} - I,\ i \le curr\}$;
        $\delta = 0$;
    else if $B_{curr} \le B_\alpha$ then
        if $\frac{w_{n+1}^{curr+1}}{H_n} < B_{curr} - I$ then
            $l_{n+1} = \min\{r^i \mid r^i \in \mathfrak{R},\ r^i > r^{curr},\ i \ge curr\}$;
        else
            $l_{n+1} = r^{curr}$;
        end if
        $\delta = 0$;
    else if $B_{curr} \le B_\beta$ then
        $l_{n+1} = \max\{r^i \mid r^i \in \mathfrak{R},\ \frac{w_{n+1}^i}{H_n} \le B_{curr} - I,\ i \ge curr\}$;
        $\delta = 0$;
    else if $B_{curr} > B_\beta$ then
        $l_{n+1} = \max\{r^i \mid r^i \in \mathfrak{R},\ \frac{w_{n+1}^i}{H_n} \le B_{curr} - B_\alpha,\ i \ge curr\}$;
        $\delta = B_{curr} - B_\beta$;
    else
        $l_{n+1} = r^{curr}$;
        $\delta = 0$;
    end if
end if
```

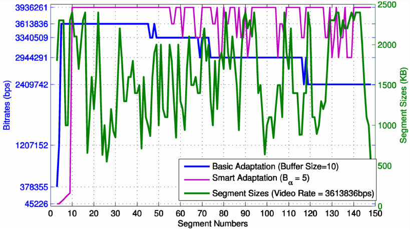
Fast Start ($B_{curr} \le I$) The algorithm selects the lowest bit rate.

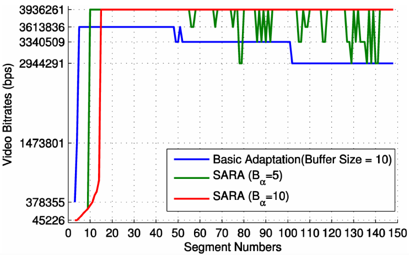
Additive Increase ($I < B_{curr} \le B_\alpha$) The algorithm starts to increase the bit rate in small steps.

Aggressive Switching ($B_\alpha < B_{curr} \le B_\beta$) The algorithm selects the most suitable bit rate.

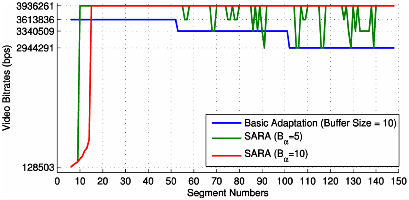
Delayed Download ($B_\beta < B_{curr} \le B_{max}$) Same as *Aggressive Switching* but the request of the next segment is delayed.

SARA: performance





Three examples that compares SARA with a basic adaptation algorithm with different bandwidths: 1 Mbps (above), 4 Mbps (right, top) and 8 Mbps (right, bottom).



Thank you for your attention.