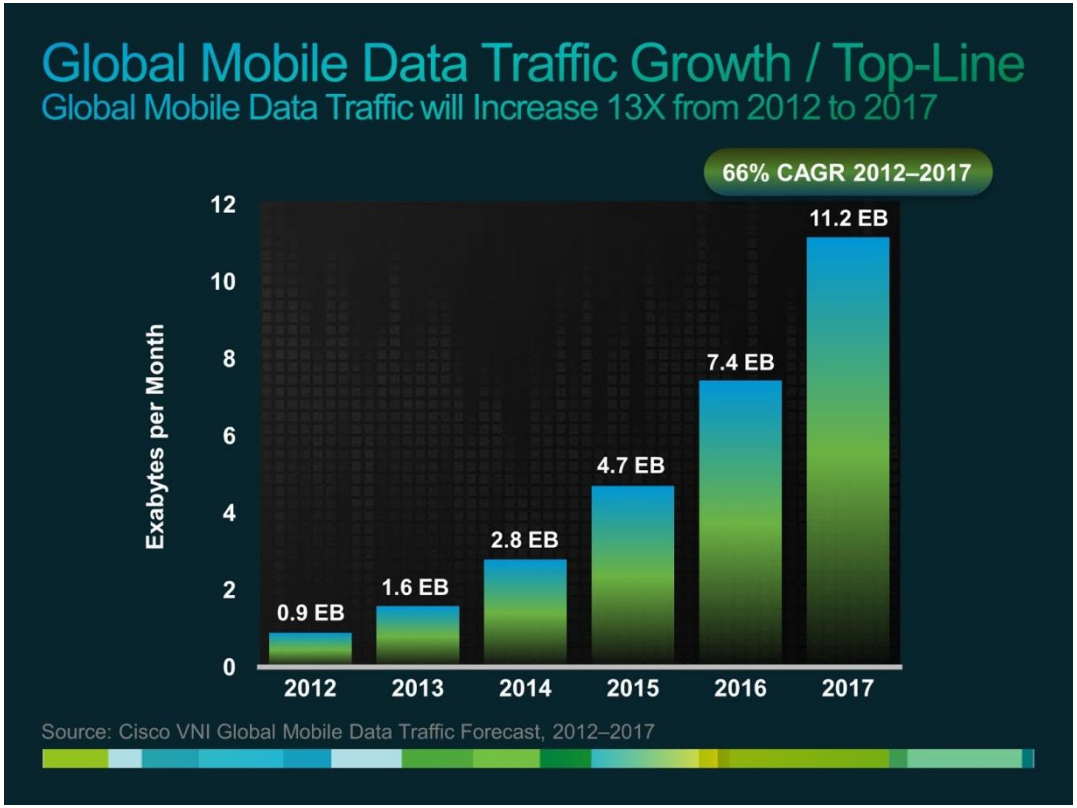
Improving caching efficiency and quality of experience with CF-Dash

Ali Gouta

Zied Aouini, Anne-Marie Kermarrec, Yannick Le Louédec,Mamadou Tourad DIALLO

Introduction to DASH



"Cisco reported that mobile data traffic will reach 11.2 exabytes per month by 2017".

➔ *Shift toward HTTP adaptive streaming (HAS) solutions*

NETFLIX

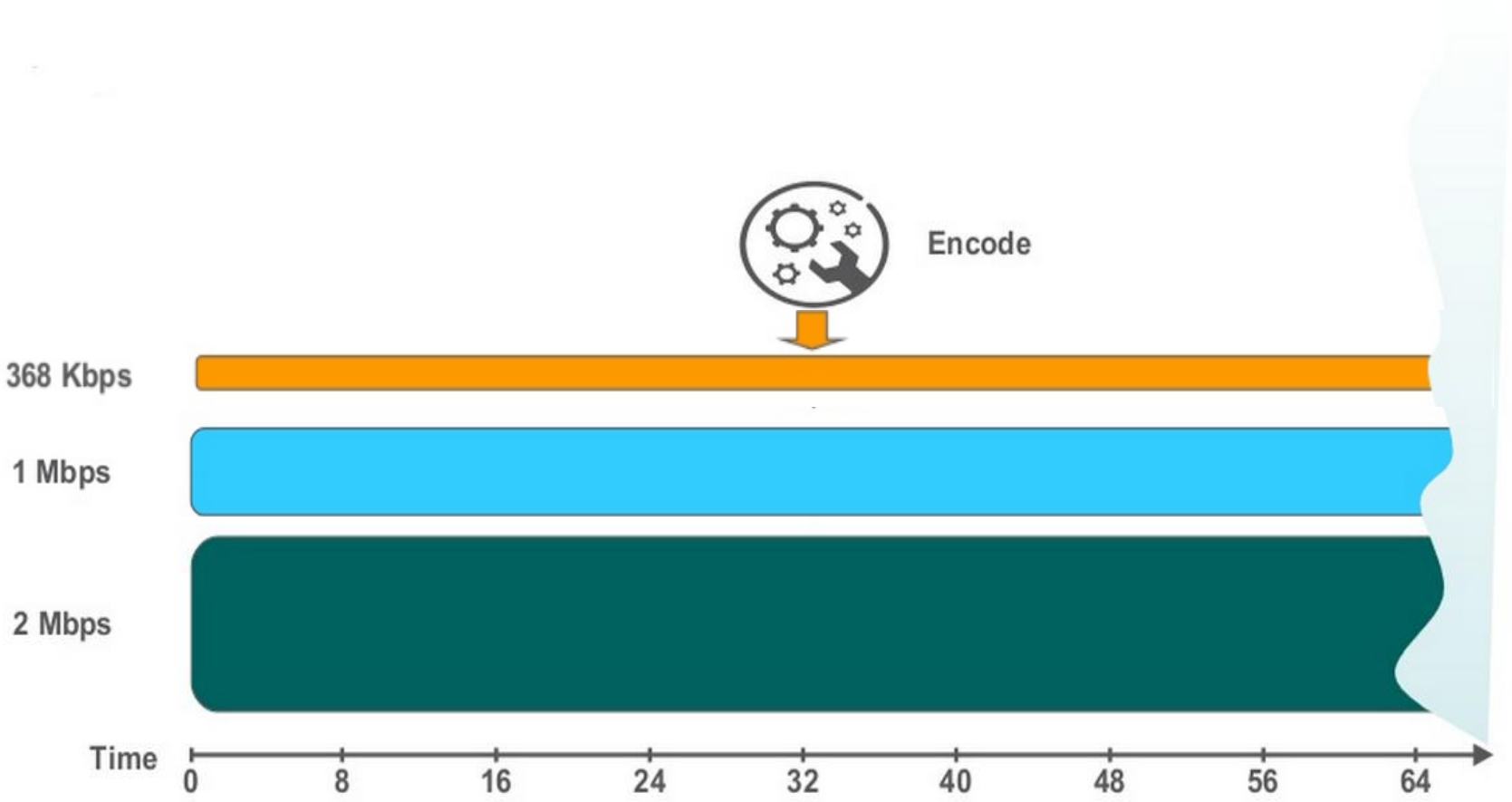
YouTube

france 2

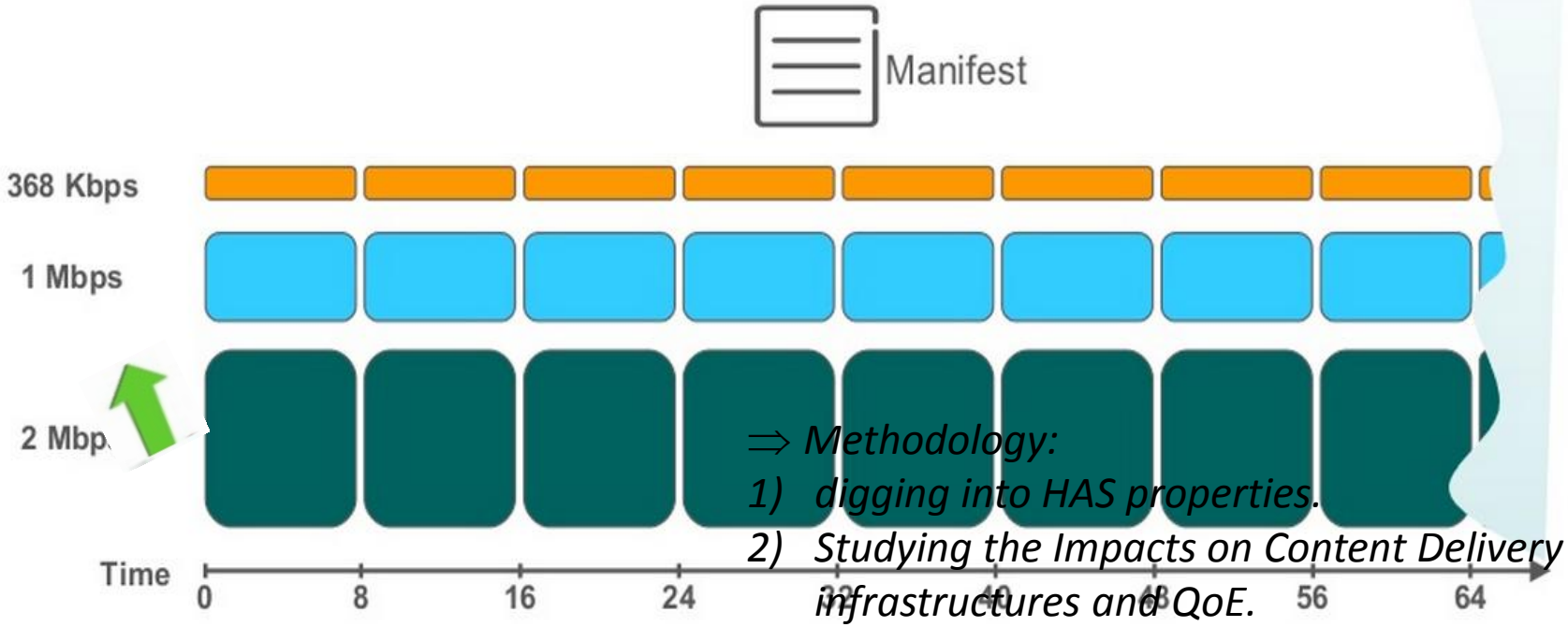
M6

TF1

❖HTTP adaptive streaming



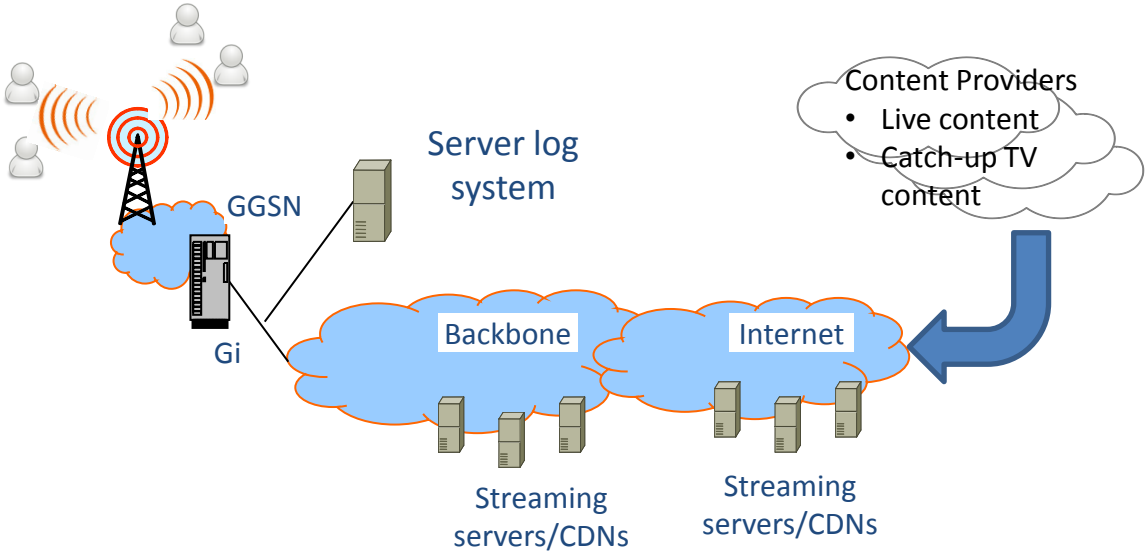
❖HTTP adaptive streaming



⇒ Methodology:

1) *digging into HAS properties.*
2) *Studying the Impacts on Content Delivery infrastructures and QoE.*
3) *Leveraging these findings to design efficient caching strategies.*
4) *Evaluation.*

Dataset overview



- 246.913 unique active clients during the measurement period.
- Collection from November 7*th* 2012 until January 9*th* 2013, involving mainly Apple HTTP Live Streaming (HLS) and Microsoft smooth streaming sessions (HSS).
- 1.763.516 adaptive streaming sessions (92.595.115 HTTP GET requests).
- Type of the contents: *Live* and *catch-up.*

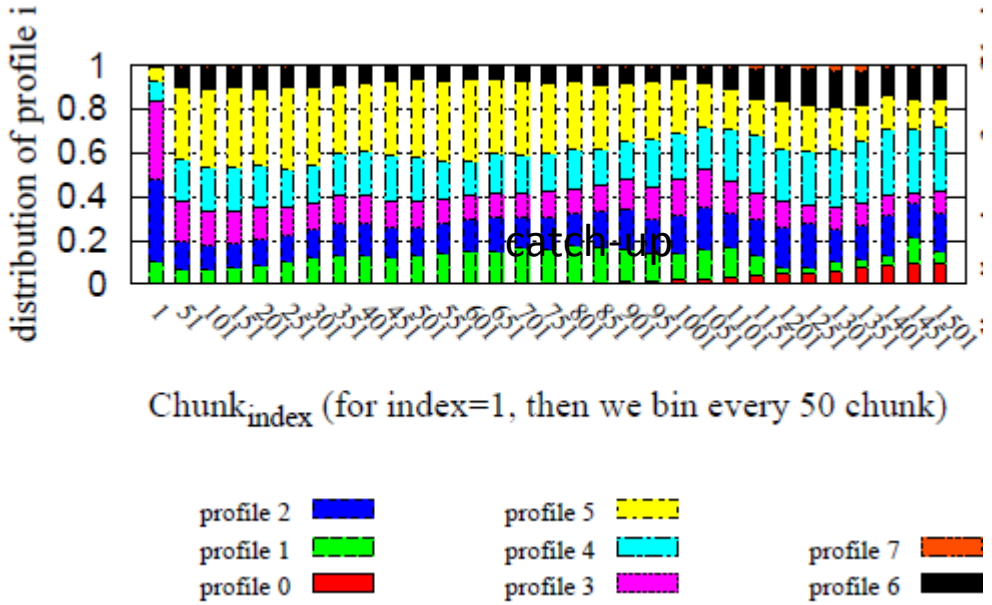
Switching behavior of DASH users
&
Cache-Friendly Dash

Distribution of the requested profiles

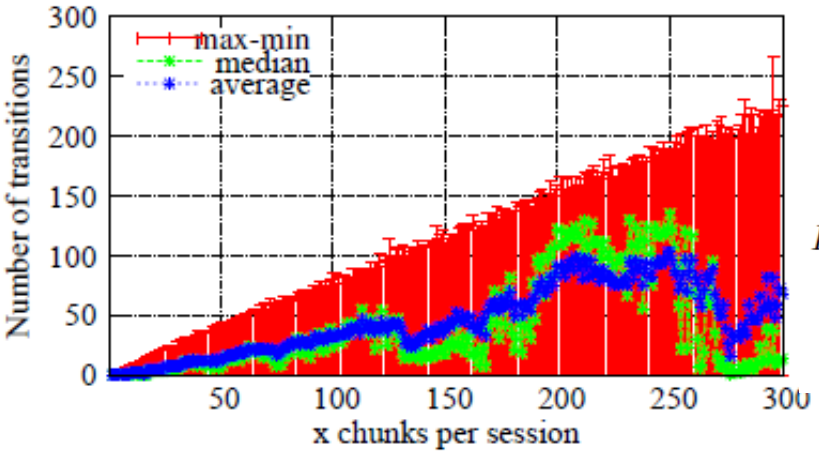
Profile i	Encoding bitrate (kbps)
Profile 0 (P_0)	< 50
Profile 1 (P_1)	[50-150)
Profile 2 (P_2)	[150-280)
Profile 3 (P_3)	[280-420)
Profile 4 (P_4)	[420-600)
Profile 5 (P_5)	[600-1000)
Profile 6 (P_6)	[1000-2000)
Profile 7 (P_7)	$\geq$ 2000

- For catch-up contents:

Clients request mostly profiles: 5, 4 and 3.



Frequency of bitrate-switching



Number of transitions during HAS sessions

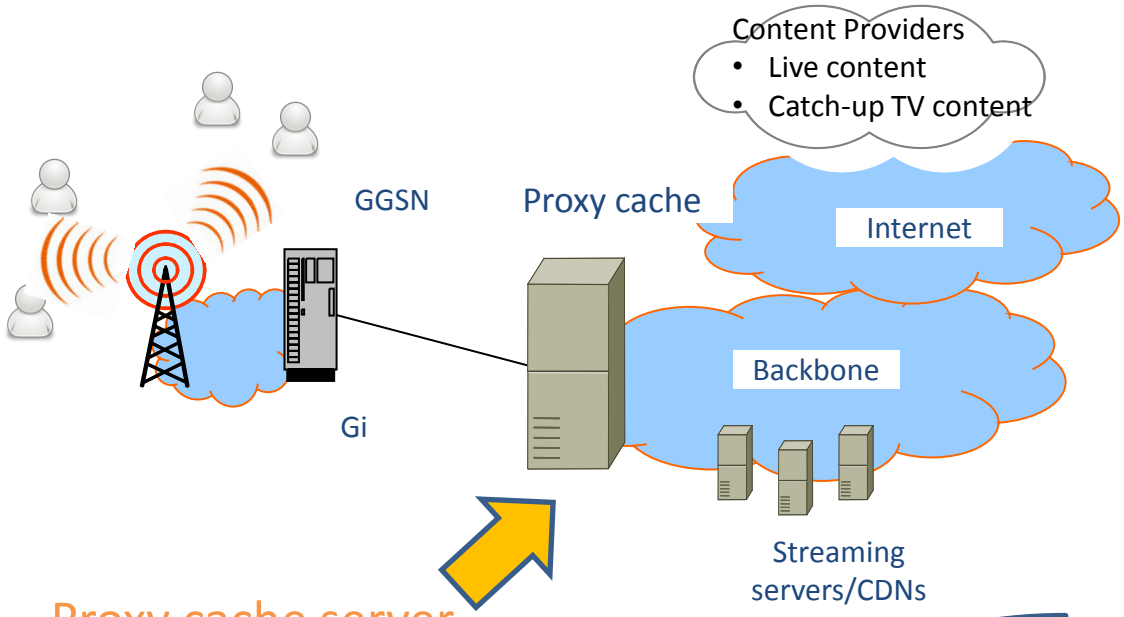
$$P = \begin{pmatrix} 0 & 0.0997 & 0.2650 & 0.2685 & 0.1784 & 0.1358 & 0.0460 & 0.0066 \\ 0.0053 & 0 & 0.1533 & 0.3482 & 0.2683 & 0.1753 & 0.0464 & 0.0029 \\ 0.0036 & 0.0405 & 0 & 0.4237 & 0.28 & 0.2034 & 0.0456 & 0.0028 \\ 0.0021 & 0.0410 & 0.2496 & 0 & 0.4294 & 0.2486 & 0.0339 & 0.0041 \\ 0.0007 & 0.0192 & 0.1271 & 0.3867 & 0 & 0.4080 & 0.0500 & 0.0079 \\ 0.0009 & 0.0099 & 0.0817 & 0.2005 & 0.4790 & 0 & 0.2016 & 0.0260 \\ 0.001 & 0.0055 & 0.0303 & 0.0739 & 0.1617 & 0.6060 & 0 & 0.1213 \\ 0.0004 & 0.0018 & 0.0078 & 0.0160 & 0.0441 & 0.2266 & 0.7031 & 0 \end{pmatrix}$$

Switching from $P_i to\ P_j$

In average, the number of transitions during a HAS session is bounded between [1/6; 1/2] of the total requested chunks per session.

$\Rightarrow$ *Implications on caching efficiency:*
This reduces the performance of the cache in term of hit-ratio.

Simulation Scenario



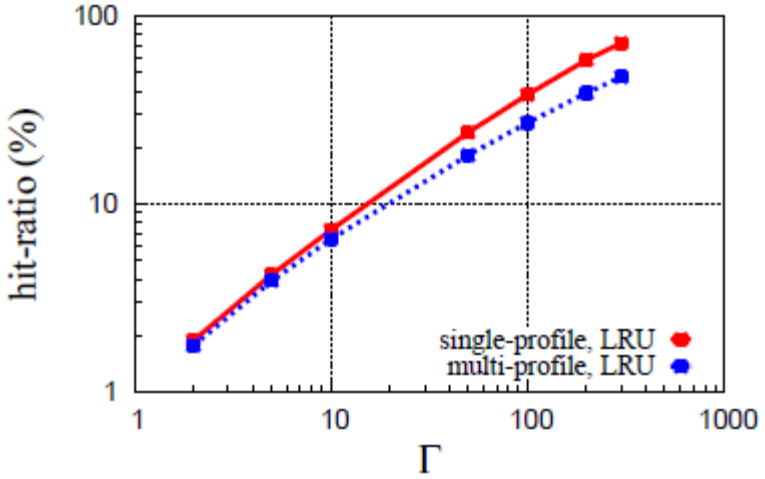
- Proxy cache server
- LRU caching logic

Assumptions

- Single profile: 640kbps
- Multi-profile:

[40, 64, 240, 360, 440, 640, 1840, 2540]

- All chunks are 10 second length.
- We only consider the Catch-up TV sessions.
- LRU
- Clients do not make any jump forward/backward during the video session.



❖QoE evaluation

- Subjective quality evaluation: Film, Sport, News.



Profile	Video resolution	Bitrate (kbps)	User perception (MOS)
Profile 1	176*144	100	Bad(1)
Profile 2	280*160	210	Bad(1.2)
Profile 3	320*180	250	medium (2.2)
Profile 4	400*224	510	good (3.3)
Profile 5	480*270	900	good (3.8)
Profile 6	640*360	1500	Excellent (4)
Profile 7	1024*576	3500	Excellent (4.5)

❖ CF-Dash (Cache Friendly-Dash):

Goal is 2-fold

➔ sustain the quality of experience of mobile clients: Prevent clients to turn systematically to the highest profiles even though they experience a high bandwidth. This increases the probability to other clients to download chunks from the cache.
⇒ Fairness.

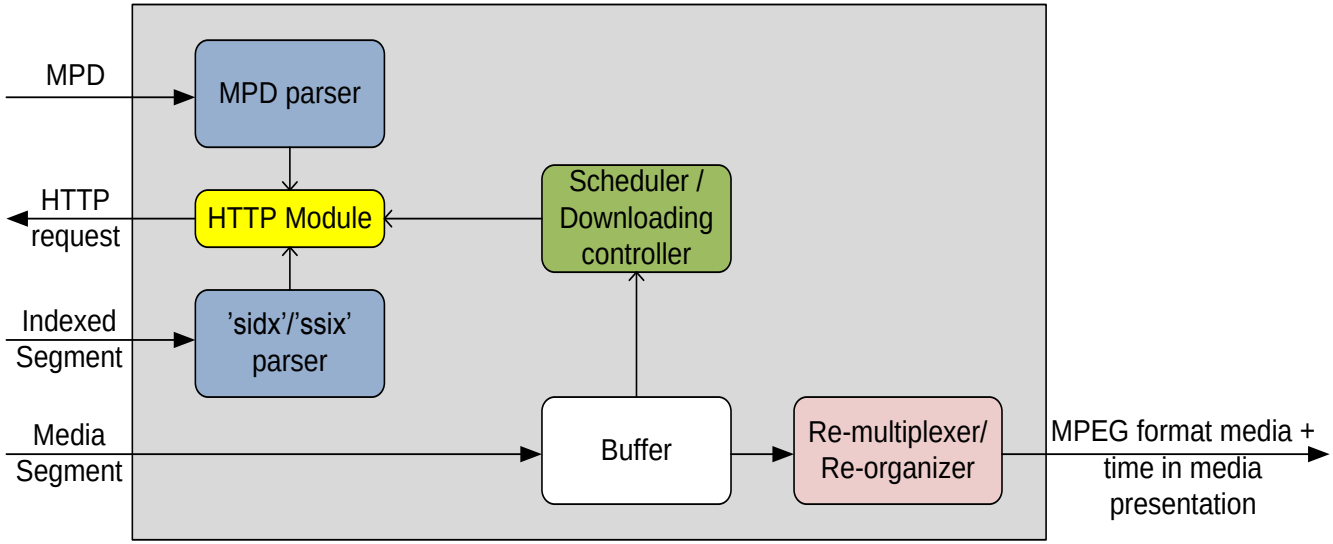
➔ improving caching efficiency: Reduce the number of switching between qualities.
⇒ Stability.

❖ Rational:

=> Defining a ***profile-limit (PL),*** where this profile should afford a good user-experience **and** improves the caching efficiency.
=> Clients do not scales systematically above the PL. If desired they have to manually fix this profile.

Testbed experiments

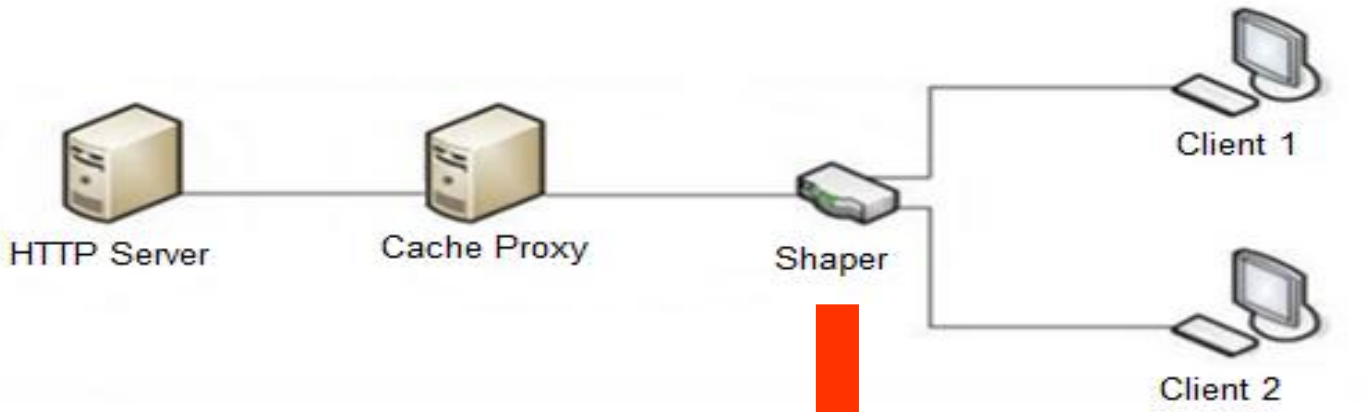
❖MPEG-DASH architecture



❖ CF-Dash implementation

- **HTTP Module**: Exchange of messages between the client-player and the cache to learn about the *profile-limit.*
- **MPD Parser and Downloading controller**: to hinder the client-player from switching systematically to profiles above the *profile-limit.*

❖ Testbed implementation (PoC)



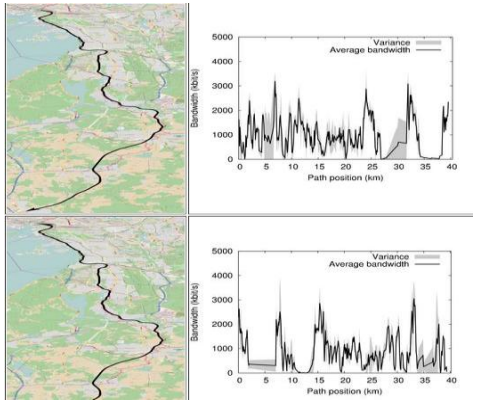
- HTTP server: Catalogue of

20 videos (Zipf popularity)

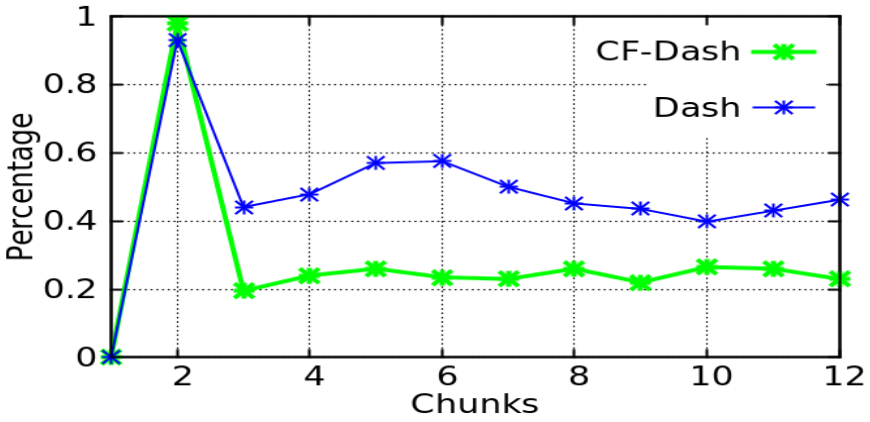
- Gpac framework installed

on both clients:

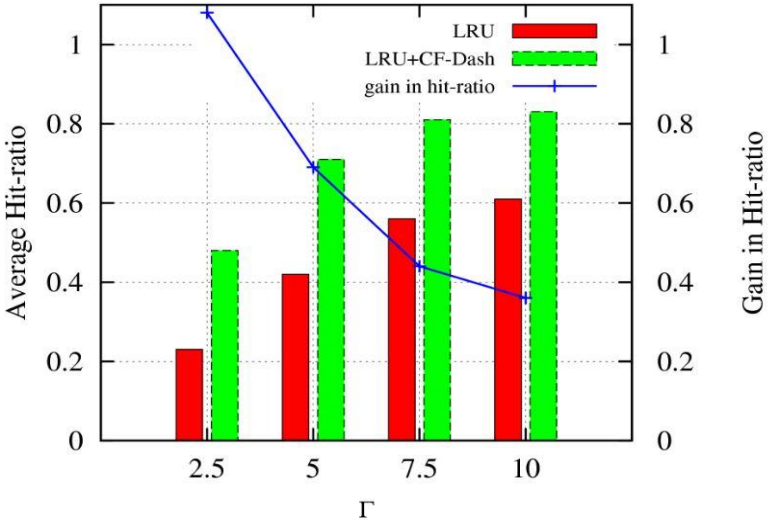
200 simultaneous sessions.



- Swithcing between qualities:



- Gain in Hit-Ratio:



Conclusion

- CF-Dash aims to give the network delivery actors

(CDNs, operators,…) to assist the client-player to select the video quality that both clients' players and content delivery actors find it convenient to serve. This is in line with dash2.0 (SAND).

- Futur works:

we will further investigate the ideal profile to be cached and define incentive strategies to encourage clients requesting the same encoding profiles.

Any question?

Thank you