A Hybrid Model of Adaptive Video Streaming Control Systems

G. Cofano, Luca De Cicco, S. Mascolo

Politecnico di Bari, Italy
55th IEEE Conference on Decision and Control - Las Vegas, USA

12-14 December 2016
Video will represent ~80% of Internet global traffic by 2018, up from ~60% in 2013 (source Cisco)

Users are moving from traditional TV broadcasting to Internet based video services

(Source statista.com)
The Challenge

- Provide an experience **at least** as good as traditional TVs on a heterogeneous set of devices (tablets, smartphones, smart TVs)
- **Playback interruptions** due to buffer depletion must be avoided (highly detrimental for QoE)
- Video bitrate should be as high as possible to increase visual quality
- Legacy systems (until \(\sim\) 2010) encoded video at **fixed bitrate**
- Modern video streaming systems can vary the video bitrate and resolution to adapt to:
  - Time-varying Internet available bandwidth
  - Device resolution
The video is encoded into a number of versions, the **video levels**, at different (nominal) bitrates $l_i$.

Each video level is temporally divided into segments of **fixed duration** $\hat{\tau}$.

$S_{i,s}$ is the actual size produced by the encoder for segment $s$ and level $i$. Actual bitrate is $S_{i,s}/\hat{\tau}$. 

---

**The mainstream Multi-Bitrate Approach**

- The video is encoded into a number of versions, the *video levels*, at different (nominal) bitrates $l_i$.
- Each video level is temporally divided into segments of **fixed duration** $\hat{\tau}$.
- $S_{i,s}$ is the actual size produced by the encoder for segment $s$ and level $i$. Actual bitrate is $S_{i,s}/\hat{\tau}$. 

---

L. De Cicco (luca.decicco@poliba.it)  A Hybrid Model of AVS Control Systems  IEEE CDC 2016 3 / 16
A video streaming control streaming in a nutshell

The client fetches video segments at a video bitrate $l_i$ decided by the controller from an HTTP server.
The controller’s output has two components:

1. The index \( i \in \{1, \ldots, N\} \) of the video level to be downloaded
2. The idle time \( w \) to be waited between two consecutive segments download

Control actions can be actuated only when a new segment download is triggered (event driven)
The Downloader

- Download time of segment $s$
- Idle time of segm. $s$

\[ t_{s+1} = t_s + \tau_s + W_s \]

- Download of segm. $s$ is started
- Download ends
- Downl. of segm. $s + 1$ is started

- The downloader serves as the actuator and fetches video segments from the HTTP server
- It pushed video segments as soon as they are downloaded (at time $t_s + \tau_s$)
Two possible states:

- **Pause**: video playback is paused (drain rate $d_r = 0$) to allow the buffer to be filled
- **Play**: the player drains the buffer at rate $d_r = 1$ and plays the video
The playout buffer

- It is **drained continuously** by the player at a rate $d_r$.
- It is **impulsively filled** by a fixed amount $\hat{\tau}$ (segm. duration) on completion of a segment download.
- Net increment (decrement) of queue length due to the download of a segment:

$$q(t_s + \tau_s) - q(t_s) = \hat{\tau} - \int_{t_s}^{t_s + \tau_s} d_r(\xi) d\xi$$
A hybrid system can be described by four elements [HDS]:

\[
\mathcal{H} : \begin{cases} 
\dot{x} = f(x, u, r) & (x, u, r) \in \mathcal{C}, \\
 x^+ = g(x, u, r) & (x, u, r) \in \mathcal{D} 
\end{cases}
\]

- **Flow set**: $\mathcal{C}$ where the state $x$ “flows” (evolves continuously)
- **Jump set**: $\mathcal{D}$ where the state $x$ “jumps” (time discrete evolution)
- **Flow map**: $f$ modelling time-continuous evolution
- **Jump map**: $g$ modelling time-discrete evolution

State, input, disturbance

**State**  \[ x = \begin{bmatrix} q \\ d_r \\ \Delta \\ \tau \\ \sigma \\ s \end{bmatrix} \]

- **Playout buffer length** (s) [PB]
- **Draining rate** [P]
- **Segment downloaded bytes** [D]
- **Idle timer (s)** [D]
- **Downloader state (1=ON,0=Idle)** [D]
- **Segment index** [D]

**Input**  \[ u = \begin{bmatrix} i \\ w \end{bmatrix} \]

- **Video level** \( i \in \{1, \ldots, N\} \)
- **Idle duration (s)**

**Disturbance**  \[ r \]

- **available bandwidth (bytes/s)**

---

L. De Cicco (luca.decicco@poliba.it)  
A Hybrid Model of AVS Control Systems  
IEEE CDC 2016 10 / 16
Jump set $\mathcal{D}$

$\mathcal{D}$ is the union of the sets:

$\mathcal{D}_{\text{feed}} = \{(x, u, r) \in \mathcal{X} \times \mathcal{U} \times \mathbb{R}_{\geq 0} : \Delta = S_{i,s}\}$

$\mathcal{D}_{\text{dwnl}} = \{(x, u, r) \in \mathcal{X} \times \mathcal{U} \times \mathbb{R}_{\geq 0} : \tau = w \land \sigma = 0\}$

$\mathcal{D}_{\text{empty}} = \{(x, u, r) \in \mathcal{X} \times \mathcal{U} \times \mathbb{R}_{\geq 0} : q = 0 \land d_r = 1\}$

$\mathcal{D}_{\text{play}} = \{(x, u, r) \in \mathcal{X} \times \mathcal{U} \times \mathbb{R}_{\geq 0} : q \geq q_{\text{min}} \land d_r = 0\}$

$\mathcal{D}_{\text{feed}}$ End of download of segment $s$ of level $l_i \Rightarrow$ The segment is fed to the buffer

$\mathcal{D}_{\text{dwnl}}$ End of an idle period $\Rightarrow$ Triggers the download of the next segment ($s + 1$)

$\mathcal{D}_{\text{empty}}$ Queue gets empty $\Rightarrow$ Playback is stopped ($d_r = 0$)

$\mathcal{D}_{\text{play}}$ Queue gets filled $\Rightarrow$ Playback is started ($d_r = 1$)
Continuous and discrete dynamics ($f$ and $g$)

**Continuous dynamics**

\[ \dot{q} = -d_r \]
\[ \dot{\Delta} = \sigma r \]
\[ \dot{\tau} = 1 \]

$(x, u, r) \in \mathcal{C}$

(other variables’ derivatives are 0)

**Discrete dynamics**

\[ d_r^+ = \begin{cases} 
1, & (x, u, r) \in \mathcal{D}_{\text{play}} \\
0, & (x, u, r) \in \mathcal{D}_{\text{empty}} \end{cases} \]

\[ q^+ = q + \hat{\tau} \]
\[ (x, u, r) \in \mathcal{D}_{\text{feed}} \]

\[ \Delta^+ = \begin{cases} 
0, & (x, u, r) \in \mathcal{D}_{\text{feed}} \end{cases} \]

\[ \sigma^+ = \begin{cases} 
0, & (x, u, r) \in \mathcal{D}_{\text{feed}} \\
1, & (x, u, r) \in \mathcal{D}_{\text{dwnl}} \end{cases} \]

\[ \tau^+ = 0 \]
\[ (x, u, r) \in \mathcal{D}_{\text{feed}} \cup \mathcal{D}_{\text{dwnl}} \]

\[ s^+ = s + 1 \]
\[ (x, u, r) \in \mathcal{D}_{\text{feed}} \]
Experimental Validation

- The model is implemented in Matlab through the Hybrid Equations (HyEq) Toolbox by Sanfelice
- Experimental data: gathered through tests on a controlled testbed in lab

- An Internet bottleneck link is emulated by a tool we developed allowing the bandwidth $r$ to be set
- An adaptive video streaming client is implemented using TAPAS (https://github.com/ldecicco/tapas)
**Employed controller** [ACC15]: a controller keeping the queue length within \([q_L, q_H]\) by throttling the video level between the closest ones \((l_i, l_{i+1})\) to the available bandwidth \(r\)

- We compare the dynamics of:
  - A fluid flow model of the queue length [ACC15]
  - The proposed Hybrid system model
  - The real system

- The video is the popular benchmark “Sintel” encoded at five nominal bitrates:
  \[ \mathcal{L} = \{240, 500, 900, 1400, 2600\} \text{kb/s} \]

Results: step response

Fluid Model

Proposed Model

Real System

L. De Cicco (luca.decicco@poliba.it)
Conclusions

- We have proposed a hybrid system modeling adaptive video streaming control systems.
- The model accurately predicts the dynamics of real video streaming systems.
- The model can be used to:
  - Design control laws and prove properties of the closed loop system (future work).
  - Perform simulations of video streaming systems to aid the design phase of controllers.
Thank you!

Luca De Cicco
luca.decicco@poliba.it
Results: step response

Fluid Model

Proposed Model

Real System