Politecnico di Bari

Dipartimento di Ingegneria Elettrica e dell'Informazione

C3LAB

Control of Computing and Communication Systems Lab

ELASTIC: a Client-side Controller for Dynamic Adaptive Streaming over HTTP (DASH)

Packet Video Workshop - San Jose, CA, USA

12-13 December 2013

L. De Cicco, V. Caldaralo, V. Palmisano, S. Mascolo

Politecnico di Bari, Dipartimento di Ingegneria Elettrica e dell'Informazione

Adaptive Streaming today…

Video distribution platforms use adaptive video streaming, instead of progressive download streaming, to improve the QoE

Multi-bitrate (stream-switching) is the mainstream approach used to implement adaptive streaming over HTTP (MPEG-DASH, HLS)

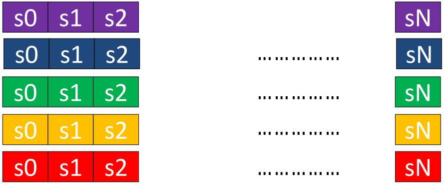
How does it work?

- Video is encoded at different bitrate and resolutions, the *video levels*

Level 4, 3500 kbps, 1280x720
Level 3, 2500 kbps, 1280x720
Level 2, 1500 kbps, 640x360
Level 1, 700 kbps, 640x360
Level 0, 300 kbps, 320x180

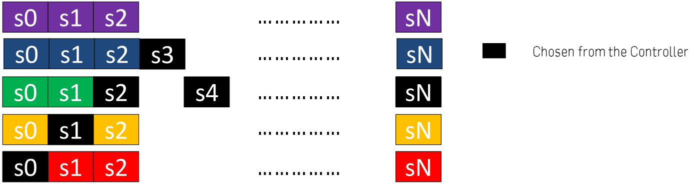
How does it work?

- Video is encoded at different bitrate and resolutions, the *video levels*
- Each video level is divided into segments



How does it work?

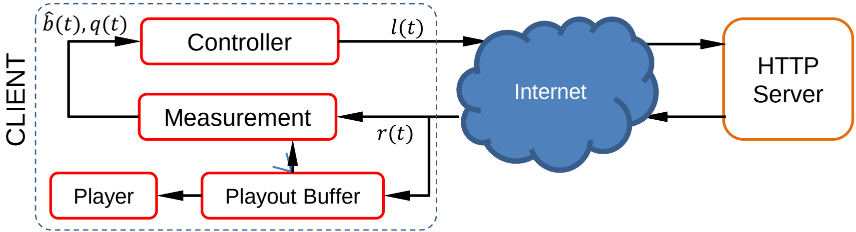
- Video is encoded at different bitrate and resolutions: the *video levels*
- Each video level is divided into segments
- For each video segment the stream-switching controller selects the video level according to channel conditions



Leading architectural approach (DASH, HLS)

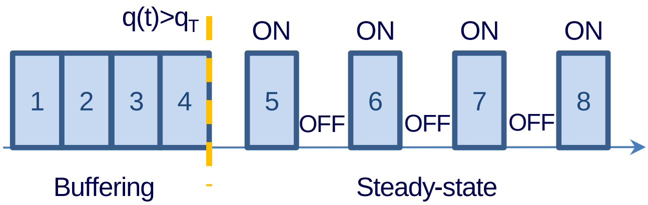
- The controller is placed at the client
- An HTTP server streams the video

Client-side Adaptive Video Streaming system



- The **HTTP server** sends video through an Internet connection with an end-to-end bandwidth $b(t)$
- The **Client** receives the video segments at a rate $r(t)$, and stores them in a **playout buffer**.
- The **measurement** module feeds the controller with the estimated bandwidth $\hat{b}(t)$ and the playout buffer level $q(t)$.
- The **controller** dynamically selects the video level $l(t)$ by sending a HTTP GET request to the HTTP Server.

The mainstream approach: Buffering – Steady state



The player can be in two different states:

- **Buffering phase**: Segments requests are performed back-to-back to quickly fill the playout buffer. This state is left when $q(t)>q_T$
- **Steady-state**: segment requests are spaced to keep the playout buffer level constant.
- This generates an **ON-OFF traffic pattern**:

ON
when the segments are downloaded

OFF
when the player is idle

Issues due to ON-OFF traffic

This ON-OFF traffic pattern causes three problems:

1. Unfair bandwidth utilization when many video share a bottleneck [Akhshabi et al, 2012]
2. Flickering of the requested video level [Akhshabi et al, 2012]
3. Video is not able to get the fair share when in the presence of long-lived TCP flows (*downward spiral effect*) [Huang et al, 2012]

ELASTIC

(fEedback Linearization Adaptive STreamIng Controller)

Conventional approach vs Elastic

Conventional approach

Two controllers are used:

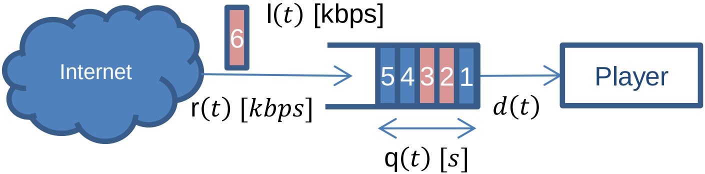
1) The first selects the video level to match the available bandwidth
2) The second controls the playout buffer length by regulating the idle period between the download of two segments (on-off traffic pattern).

Proposed approach

- Design one controller that throttles the video level $l(t)$ to drive the playout buffer length $q(t)$ to a set-point q_T.
- This eliminates the ON-OFF traffic pattern

The player is always in ON phase unless l(t) is the highest level and q > Q_{max} (>q_T)

Playout buffer model



Playout buffer model

$$\dot{q}(t) = \frac{r(t)}{l(t)} - d(t)$$

Filling rate Draining rate

Draining rate

$$d(t) = \begin{cases} 1 & \text{playing} \\ 0 & \text{paused or } q(t) = 0 \end{cases}$$

- Idea: Based on the playout buffer model, design a feedback control system that computes l(t) to steer q(t) to a threshold q_T
- Received rate r(t): considered as a (measurable) disturbance since it cannot be manipulated.
- l(t): output of the controller

ELASTIC: the control law

$$\dot{q}(t) = \frac{r(t)}{l(t)} - d(t) \qquad (1)$$ nonlinear system

We impose a linear second order dynamics for the queue

$$\begin{cases} \dot{q}(t) = -k_p q(t) - k_i q_I(t) & (2) \\ \dot{q}_I(t) = q(t) - q_T & (3) \end{cases}$$

integral error dynamics

The additional dynamics is added to make q converge to q_T at steady state

By equating (1) and (2) and solving for l(t) we get the following law for the stream-switching controller:

$$\boxed{l(t) = \frac{r(t)}{d(t) - k_p q(t) - k_i q_I(t)}}$$

ELASTIC: Implementation

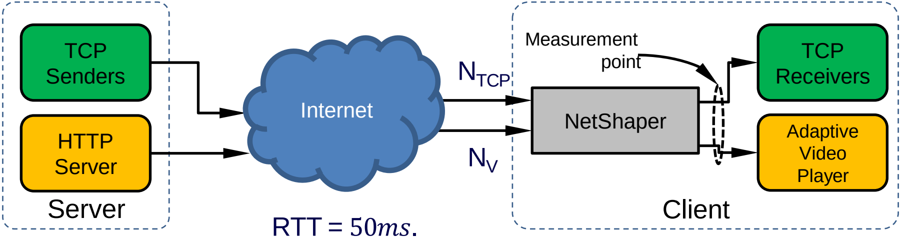
On segment download:

1. DT = getDownloadTime(); /* Downl. Time of last segm. (seconds) */
2. S = getSegmentSize(); /* Last downl. segment Size (bytes) */
3. q = getQueueLength(); /* current queue length (seconds) */
4. r = h(S/DT); /* Harmonic filter of last 5 received rate samples */
5. qI = qI + DT*(q-qT) /* Update integral error */
6. l = Q(r / (1 – kp*q – ki * qI)) /* Control law */

The quantizer Q(x) associates to x the highest video level *l* that is less than x

TESTBED

TESTBED



Server Host

- Apache as HTTP Server. Connections are persistent.

Client Host

- NetShaper: we developed this tool to set bandwidth and propagation delays
- AVP uses GStreamer libraries and supports HLS format. Implements several client-side algorithms as AVP plugins.
- IPerf is used to inject long-lived TCP flows

Experiments

Scenarios

S1. Two connections (1 video + 1 video or 1 video + 1 TCP) over a 4Mbps bottleneck

S2. A variable number of video and TCP connections over a 40Mbps bottleneck

Metrics

S1. received video level l(t), received rate r(t), channel utilization.

S2. We consider:

- r: average received rate
- U: channel utilization
- JFI: Jain Fairness Index
- RB: Rebuffering-ratio
- n_s: number of video level switches

Video specifics

- Video encoded at five different bitrates (like Akamai does):

Video level	l_0	l_1	l_2	l_3	l_4
Bitrate (kbps)	300	700	1500	2500	3500
Resolution	320x180	640x360	640x360	1280x720	1280x720

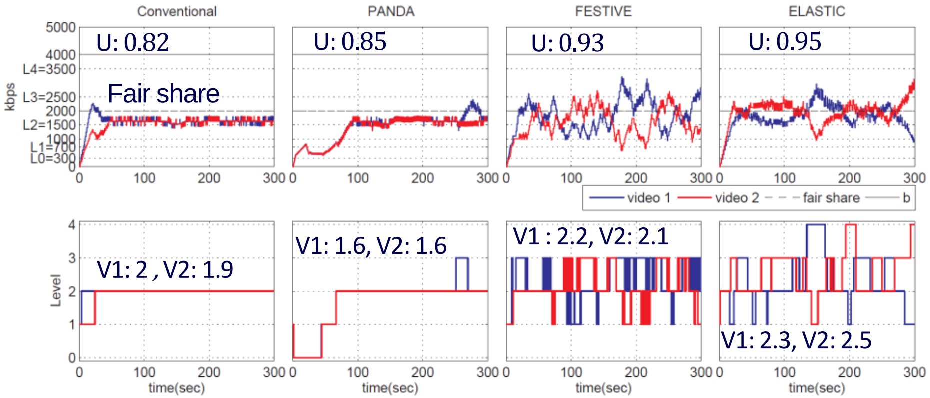
- Fragment duration of 2s.

ALGORITHMS

- ELASTIC: kp = 1/100, ki = 1/1000
- FESTIVE: specifically designed to counteract unfairness in multi-client scenario. Implemented according to the Conext 2012 paper
- PANDA: proposed by Li et al, dynamically computes the duration of OFF phases based on a control law. We employ the parameters suggested in the paper
- Conventional: implemented as described in PANDA paper, it selects the video level as a function of the estimated bandwidth

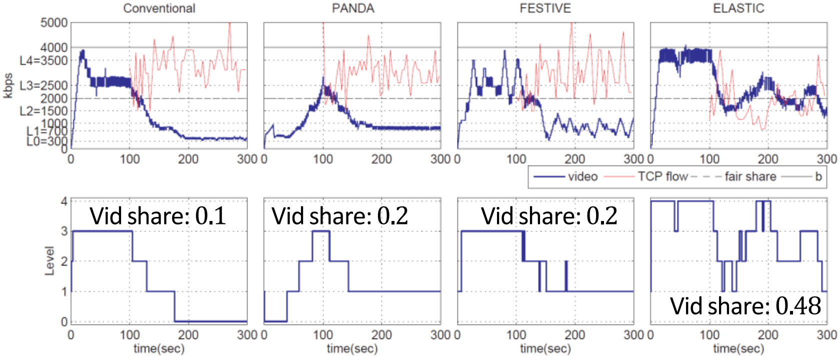
RESULTS

TWO VIDEOS SHARING A 4 Mbps BOTTLENECK



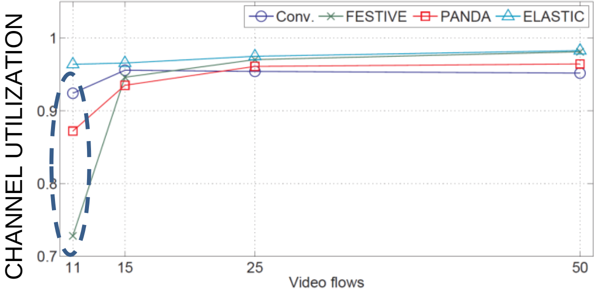
- Conventional and PANDA: the two videos fairly share the channel with some link underutilization
- ELASTIC and FESTIVE provide a received video rate that oscillates around the fair share, with an increased number of video level switches.

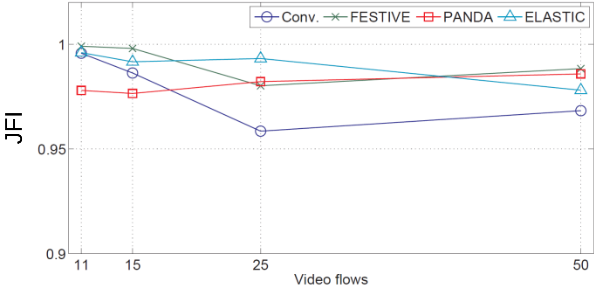
ONE VIDEO AND ONE TCP FLOW SHARING A 4 Mbps BOTTLENECK



- Conventional, PANDA and FESTIVE are not able to obtain the fair share when coexisting with a TCP flow (*downward spiral effect*)
- ELASTIC avoids the downward spiral effect since it does not produce ON-OFF traffic and behaves as the long-lived TCP flow

N_V VIDEOS SHARING A 40 Mbps BOTTLENECK (NO TCP flows)





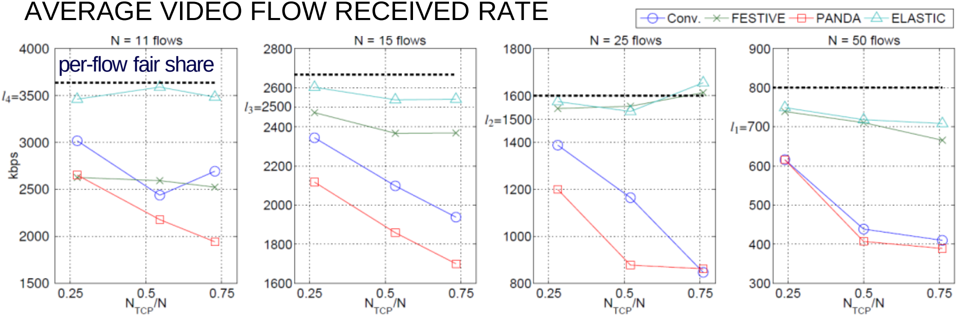
CU for ELASTIC is independent of N_v and around 0.96.

Other algorithms exhibit a lower channel utilization for $N_V = 11$.

JFI is in the range $[0.96, 1]$ for all considered algorithms.

N_V VIDEOS SHARING A 40 Mbps LINK WITH N_{TCP} TCP FLOWS

AVERAGE VIDEO FLOW RECEIVED RATE

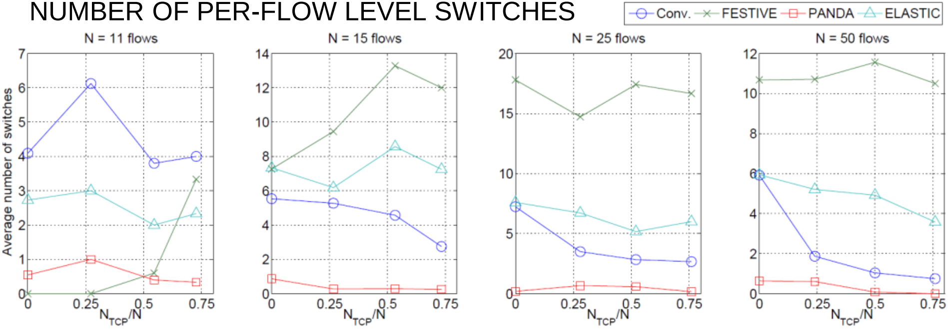


- ELASTIC is able to get the fair share regardless of the total number of flows and of the fraction of TCP flows
- The per-flow average level obtained by PANDA and Conventional decreases when the number of concurrent flow increases

The elimination of the ON-OFF pattern avoids the «downward spiral effect»

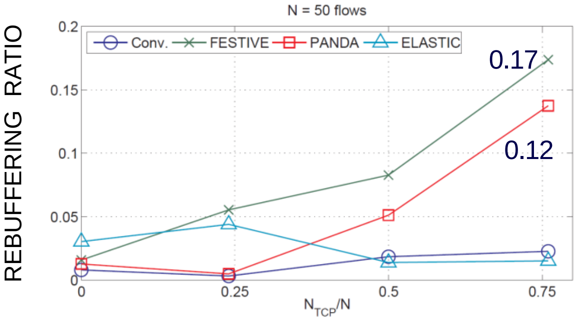
N_V VIDEOS SHARING A 40 Mbps LINK WITH N_{TCP} TCP FLOWS

NUMBER OF PER-FLOW LEVEL SWITCHES



- ELASTIC performance is comparable to that of Conventional (slightly worse)
- PANDA provides the best results (under 2 switches) but at the expense of a reduced received video bitrate

50 VIDEOS SHARING A 40 Mbps LINK WITH N_{TCP} TCP FLOWS



- For N<50 flows we measured RB ratio less than 0.01
- ELASTIC and Conventional provide RB ratios < 0.05
- FESTIVE and PANDA exhibit increasing RB ratios.

- We have proposed ELASTIC a client-side controller which does not generate an ON-OFF traffic pattern
- An experimental evaluation has shown that ELASTIC is able to get the fair share when competing with TCP long-lived flows