Avoiding zigzag quality switching in real content adaptive video streaming

Wassim Ramadan, Eugen Dedu and Julien Bourgeois

Laboratoire d'Informatique de l'Université de Franche-Comté
Montbéliard, France

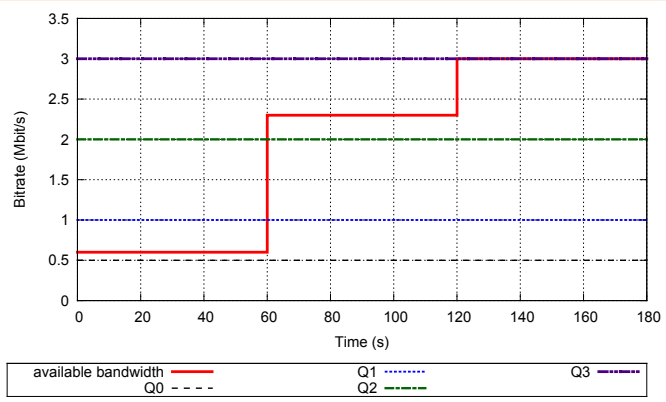
DICTAP
21–23 June 2011

Goals of zigzag avoiding

- Improve user video experience by minimizing unnecessary quality switching
- Optimise network resource usage by minimizing packet loss rate

Video adaptation and zigzag quality switching

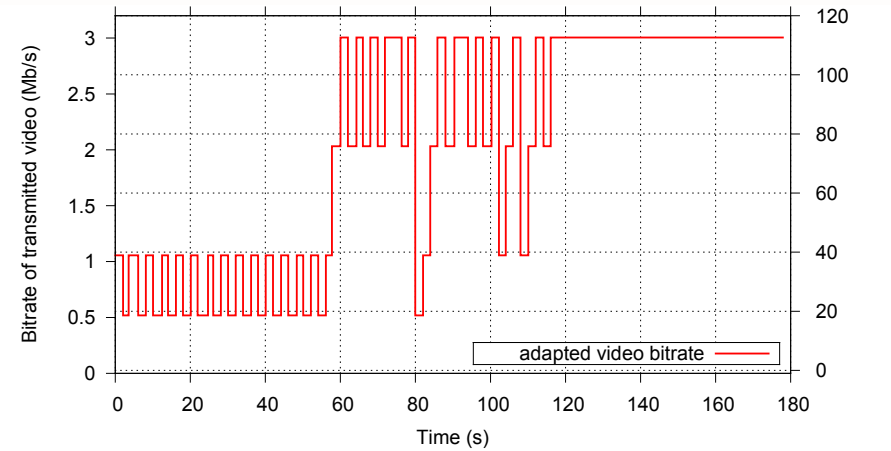
Dynamic bandwidth vs static bitrates



Ideal bitrate

- If bitrate<bw: non-optimised
- If bitrate>bw: lost packets
- Ideal: bitrate=bw

many zigzags occur



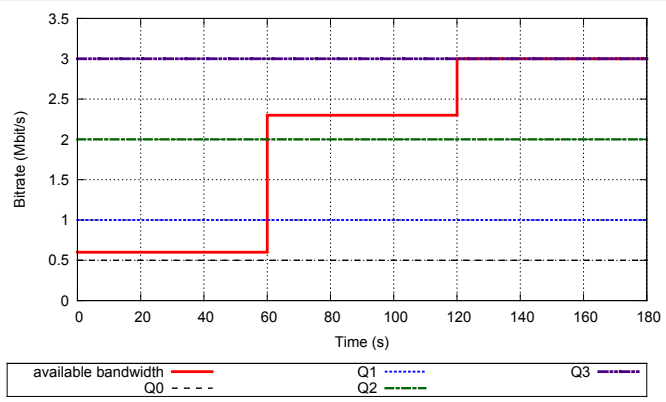
Zigzag problem

Adaptation leads to:

- many zigzags
- poor quality

Video adaptation and zigzag quality switching

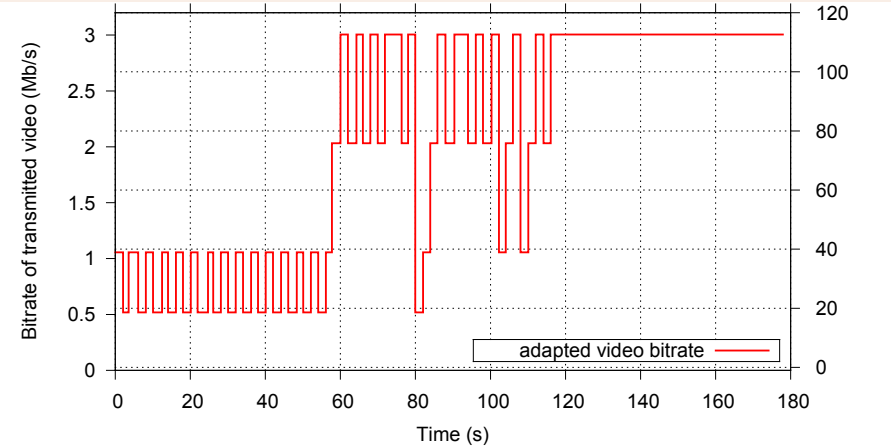
Dynamic bandwidth vs static bitrates



Ideal bitrate

- If bitrate<bw: non-optimised
- If bitrate>bw: lost packets
- Ideal: bitrate=bw

many zigzags occur



Zigzag problem

Adaptation leads to:

- many zigzags
- poor quality

Outline

Outline

Overview of ZAAL

ZAAL Zigzag Avoidance ALgorithm

ZAAL maintains a successfulness value for each bitrate which:

- is an arithmetic real value between 0 and 1
- indicates the successfulness of the bitrate if used
- uses a historic of the application last attempts to use the bitrate
- is calculated at the end of an adaptation period
- is obtained by applying an EWMA[a] algorithm on the bitrate successfulness historic

[a] EWMA: Exponential Weighted Moving Average

Details of ZAAL

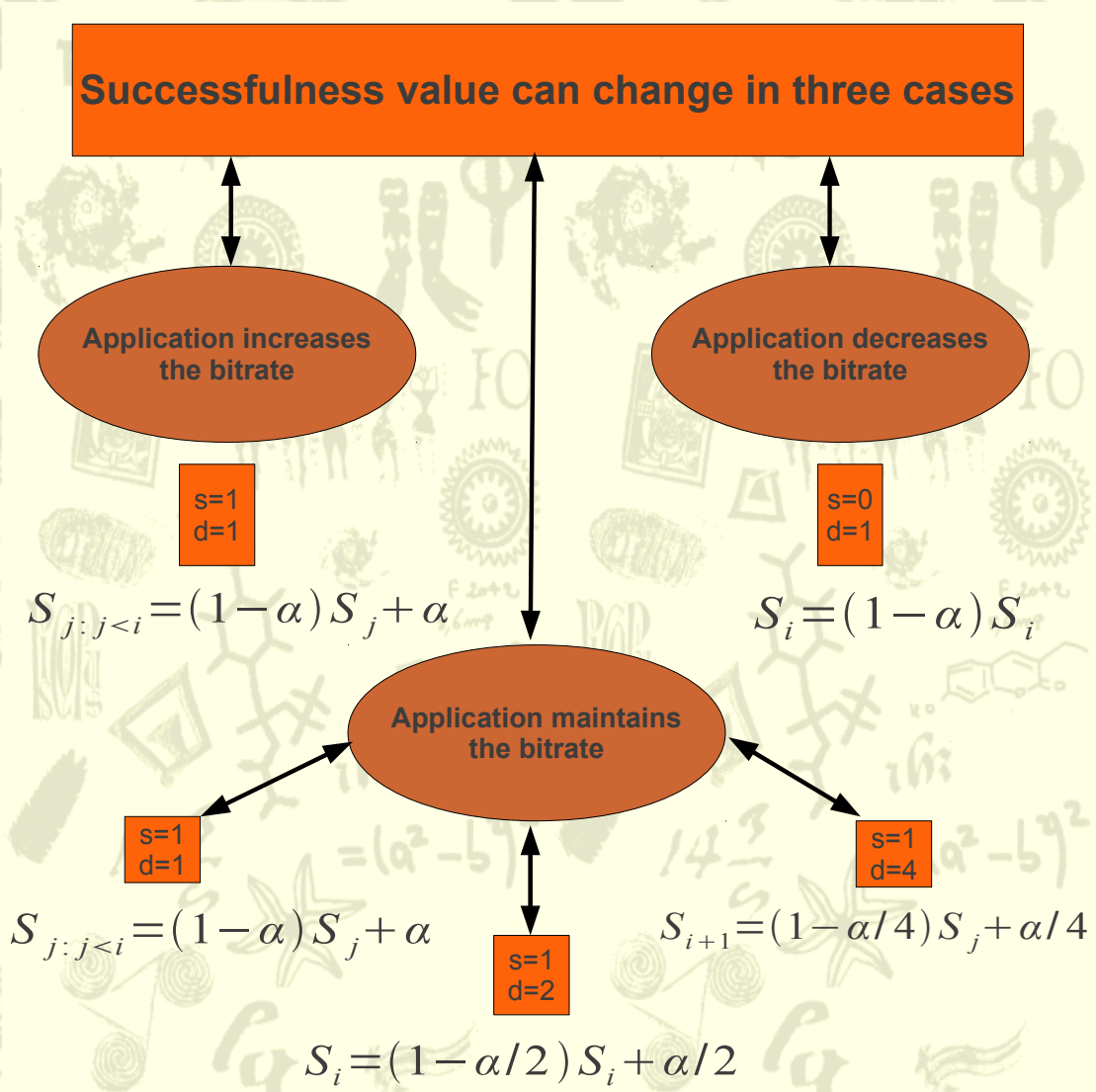
ZAAL works as follows

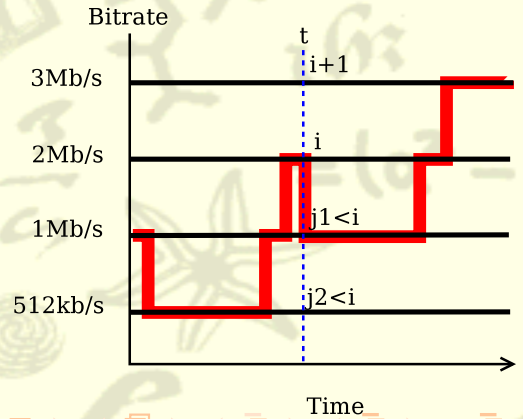
- at the beginning of a video transmission, all successfulness values S_i[a] are set to 1
- the application switches to a higher quality $i+1$ only if $S_i > \beta$ [b]
- ZAAL uses the following general formula: $S_i = (1 - \alpha/d)S_i + s(\alpha/d)$ [c] [d] [e]

---

[a] i: bitrate index
[b] $\beta = 0.7$ switching threshold
[c] if the bitrate succeeds $s = 1$ else $s = 0$
[d] $\alpha = 0.3$ degree of EWMA weighting increase/decrease
[e] and $d = 1, 2$ or 4 division factor

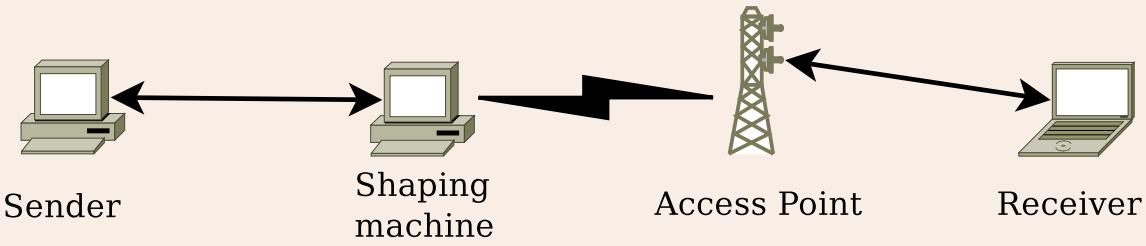
Formula of ZAAL





Outline

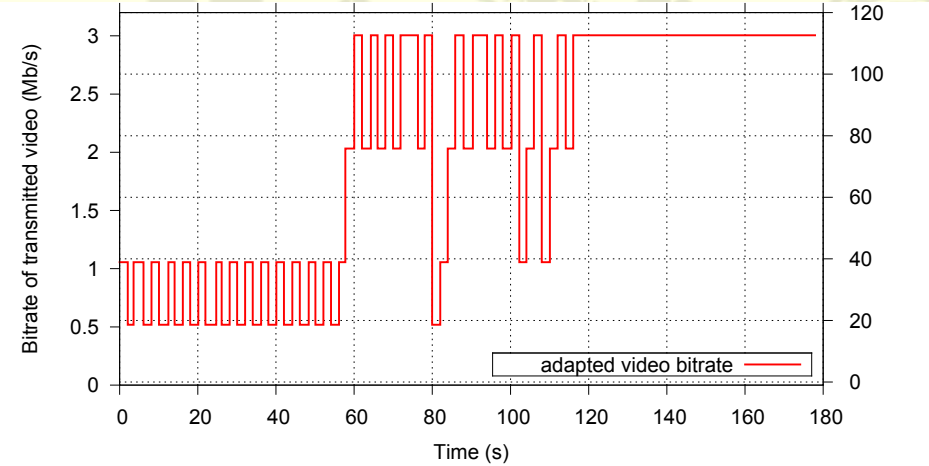
Network used in experiments



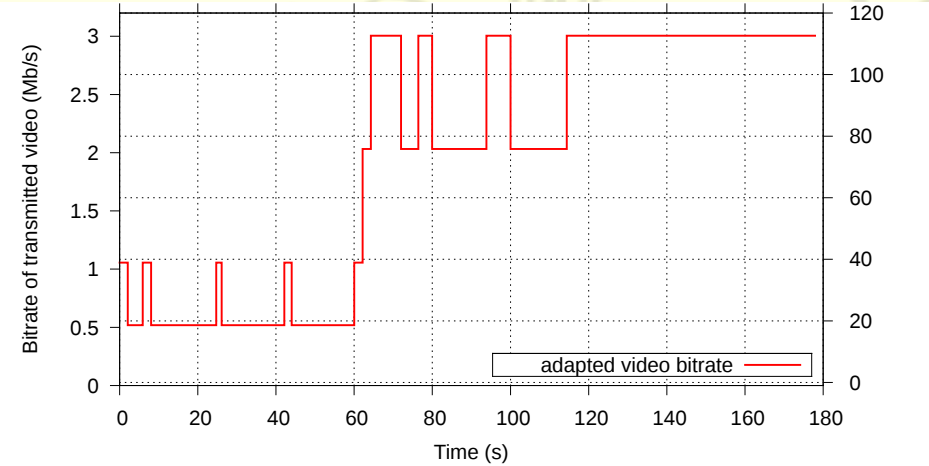
Network parameters

Parameter name	Parameter value
Experiments place	In building
Packet size	1024 bytes
PC1 (sender): Wired card, 100Mb/s PC2 (shaper machine): Wireless card, 802.11b/g 54Mb/s PC2 (shaper machine): Wired card, 100Mb/s PC3 (receiver): Wired card, 100Mb/s PC1,2&3 OS PC1,2&3 OS kernel	 Linux (Ubuntu 64bits) 2.6.35 generic
DCCP	Included in the kernel
Distance (AP $\leftrightarrow$ PC2)	50cm

Zigzag avoidance, ex. of one flow in (case of) traffic shaping



(c) without ZAAL: many zigzags occur



(d) with ZAAL: few zigzag occur

Performance comparison

Two experiments: one flow and ten concurrent flows

	Traffic shaping			*10 concurrent flows*		
Method	*Sent pkts*	*Received pkts*	*Lost pkts*	*Sent pkts*	*Received pkts*	*Lost pkts*
Without ZAAL	47795	42043	5752 (12%)	41191	32307	8884 (21%)
With ZAAL	43548	39865	3683 (8%)	36713	32477	4236 (11%)

Number of sent and received packets (average of all flows) with and without ZAAL

- in the first experiment, ZAAL has less sent and received packets than without ZAAL, but using ZAAL is more useful because it avoids the zigzag and leads to 30% fewer packet losses
- in the second experiment, ZAAL is better in terms of sent and received packets, avoiding the zigzag in the same time

Outline

Conclusions

- ZAAL is a general solution, since it can be integrated to any adaptation method
- ZAAL avoids the undesirable constant switching in quality (the zigzag problem)

Perspectives

- working on a hybrid solution, which uses a bandwidth estimation method together with ZAAL

Conclusions

- ZAAL is a general solution, since it can be integrated to any adaptation method
- ZAAL avoids the undesirable constant switching in quality (the zigzag problem)

Perspectives

- working on a hybrid solution, which uses a bandwidth estimation method together with ZAAL

Questions?