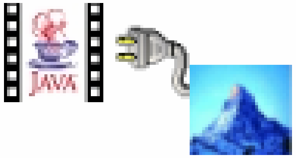
Media Adaptation Examples

Examples

❑ *WaveVideo Filtering*

- **Christian Kücherer**: Diploma thesis (University of Applied Sciences Mannheim - Prof. Bengel, in co-operation with NEC, 2001)
- Prototype in JMF (Sun – Java Media Framework)
- Real-time streaming using RTP/RTCP
- Integrate WaveVideo Codec into JMF by plugins:
  - Packetizer (at the sender)
  - Depacketizer (at the receiver)
  - QoS filter (for quality adaptation at the sender or in the network)

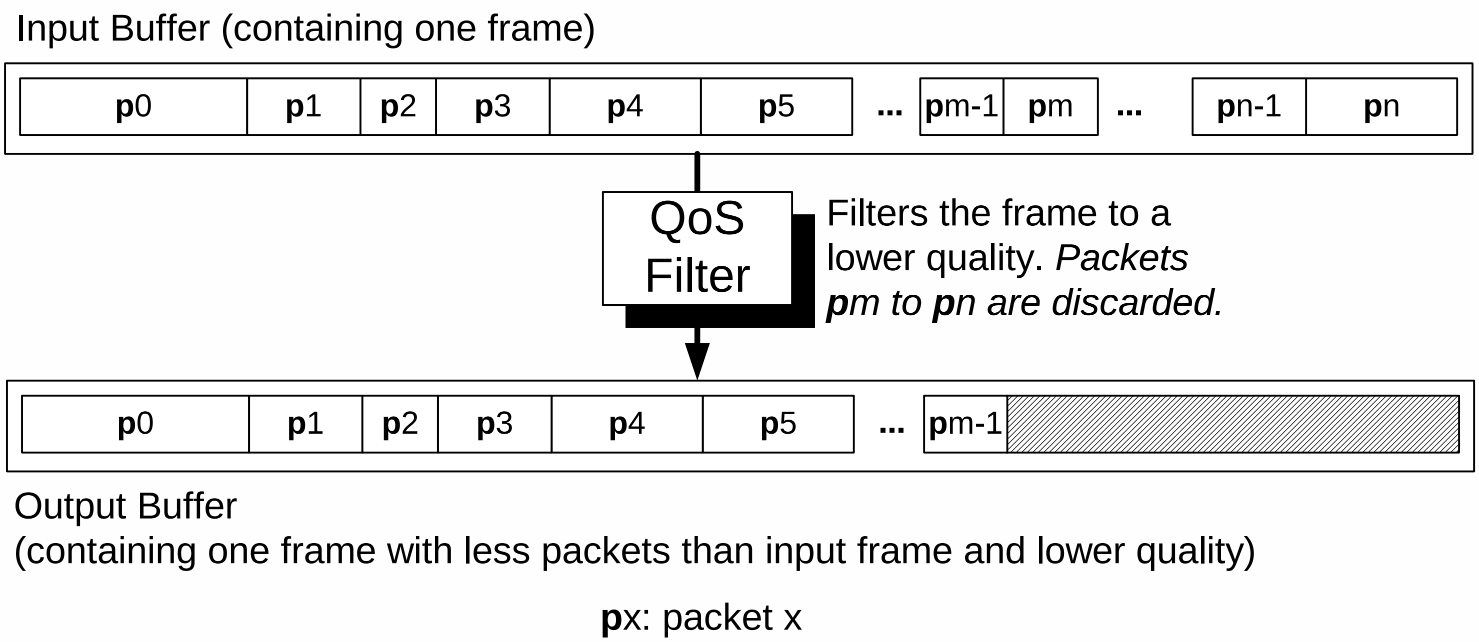


Christian Kücherer, Andreas Kassler, Andreas Schrader, Oliver Haase
End Device and Network Adaptation of WaveVideo Streams
Proceedings of the Conference on Advances in Infrastructure for Electronic Business, Science, and Education on the Internet
(SSGRR) L'Aquila, Italy, August 6-12, 2001

Andreas Kassler, Christian Kücherer and Andreas Schrader
Efficient Wavelet Video Filtering
2nd International Workshop on Quality of future Internet Services
(QofIS) Coimbra, Portugal, September 24-26, 2001

Examples

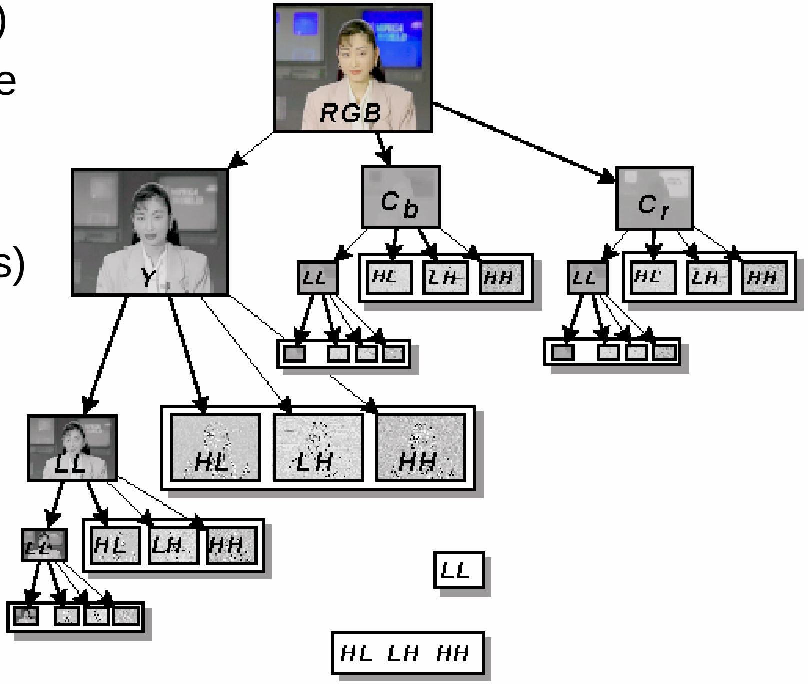
- ***WaveVideo Filtering***
  - Generic concept of filter



Examples

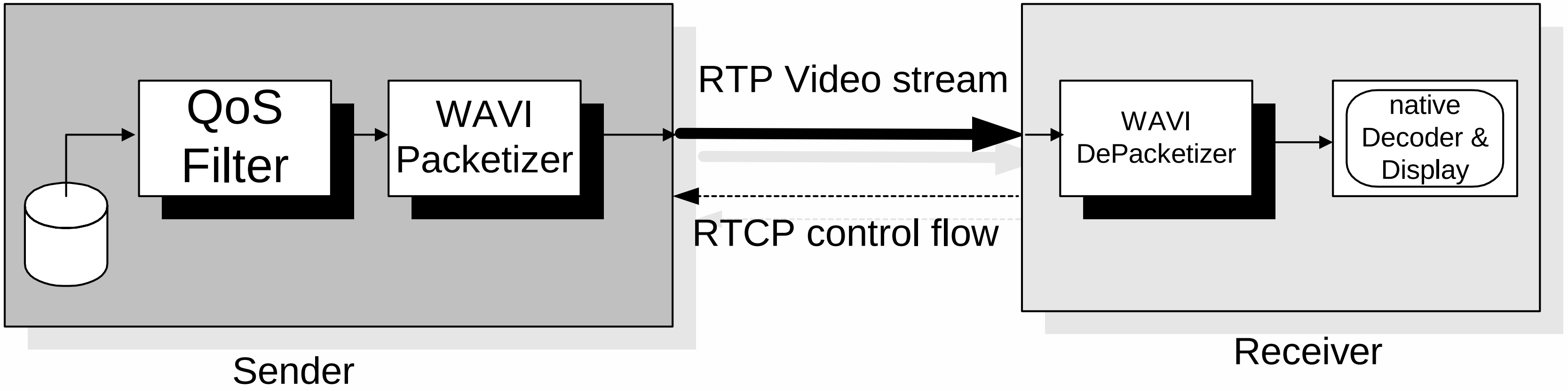
WaveVideo Filtering

- WaveVideo Codec developed at ETH Zurich
- Syntactical filtering
  - dropping decisions based on structural information (priority labels)
  - static and limited, but fast and simple
- Semantical filtering
  - dropping decisions based on content information (content analysis)
  - combined filter with luminance and chrominance resolution, image size & frame rate
  - effective and adjustable, but complex and slow
- WaveVideo is very effective but not standardized



Examples

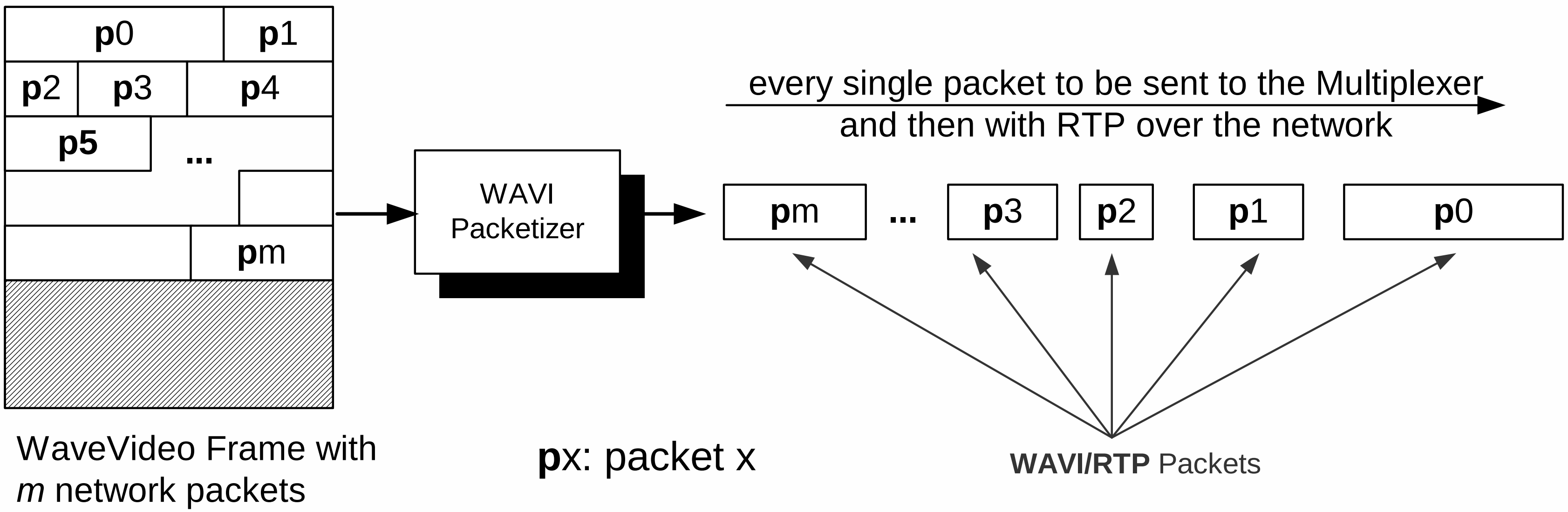
- **WaveVideo Filtering**
  - Overall Scenario



  - Sender
    - WaveVideo Packetizer
    - QoS Filter (using RTCP information for adaptation)
  - Receiver
    - WaveVideo Depacketizer

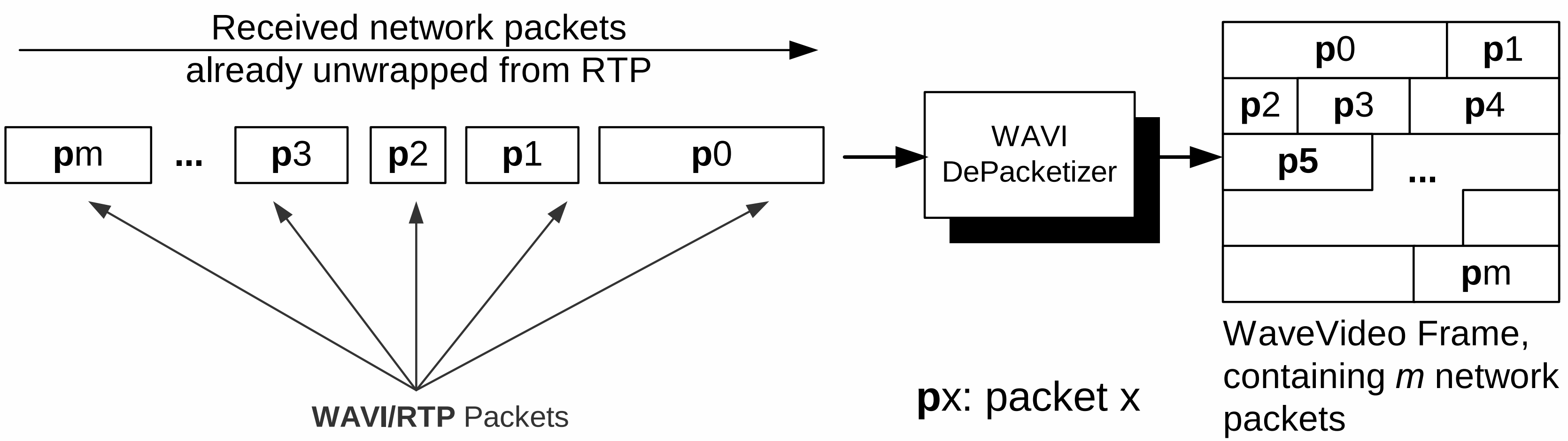
Examples

- **WaveVideo Filtering**
  - Packetizer
    - enables JMF to transmit WaveVideo encoded streams over the network
    - fragments a complete video frame into several small network packets which are then sent over the network using JMF's RTP stack



Examples

- ❑ ***WaveVideo Filtering***
  - Depacketizer
    - Used to enable JMF to receive encoded streams
    - Collects received network packets to build a complete frame and give it to the native decoder



Examples

- **WaveVideo Filtering**
  - Effort (Lines of Code)

Transmitter	
AVTransmit.java	507
GraphicsPanel.java	168
MeasuringDataPanel.java	121
WAVIFilterEffect.java	410
WAVIFilterMonitorControl.java	229
WAVIPacketizer.java	344
WAVIPacketizerMonitorControl.java	199
WAVIQualityControl.java	405
Sum	2383

Overall	5039
Project Part:	~3000

Receiver	
AVReceive.java	663
WAVIDePacketizer.java	388
Sum	1051

Wavi Filter Simulation	
WAVIFilterEffect.java	475
WAVIFilterSimulation.java	896
WAVIQualityControl.java	234
Sum	1605

Examples

- ***WaveVideo Filtering***
  - Random Dropping
    - Happens due to congestion in routers
    - Problem: Annoying artefacts due to randomness of dropping effect
    - Using parameter q as the percentage of packets coming through (e.g. Q=0.9 means 90% of the packets are forwarded, 10% are dropped)

Examples

- ***WaveVideo Filtering***
  - q=0.9

Examples

- ***WaveVideo Filtering***
  - q=0.8

Examples

- **WaveVideo Filtering**
  - q=0.7

Examples

- ❑ ***WaveVideo Filtering***
  - q=0.6

Examples

- **_WaveVideo Filtering_**
  - q=0.5

Examples

- ***WaveVideo Filtering***
  - q=0.4

Examples

- ***WaveVideo Filtering***
  - q=0.3

Examples

- ❑ ***WaveVideo Filtering***
  - q=0.25

Examples

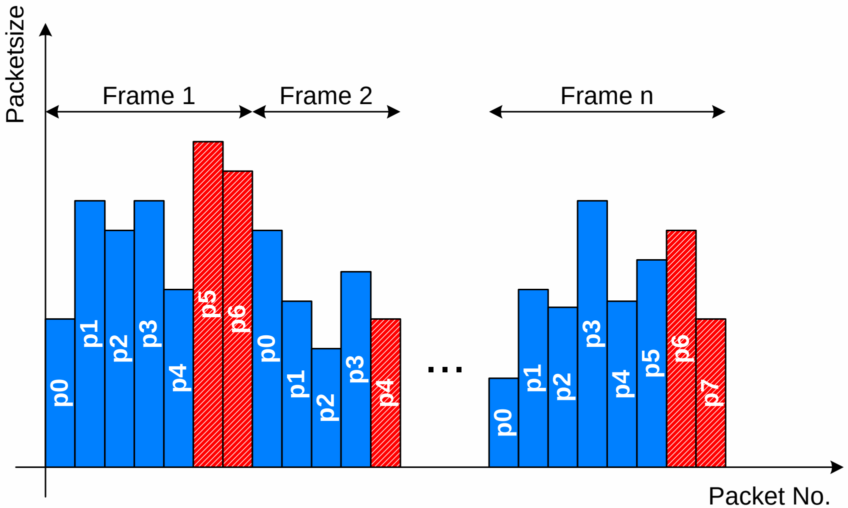
- ***WaveVideo Filtering***
  - Syntactical Filter
  - Priority-based dropping

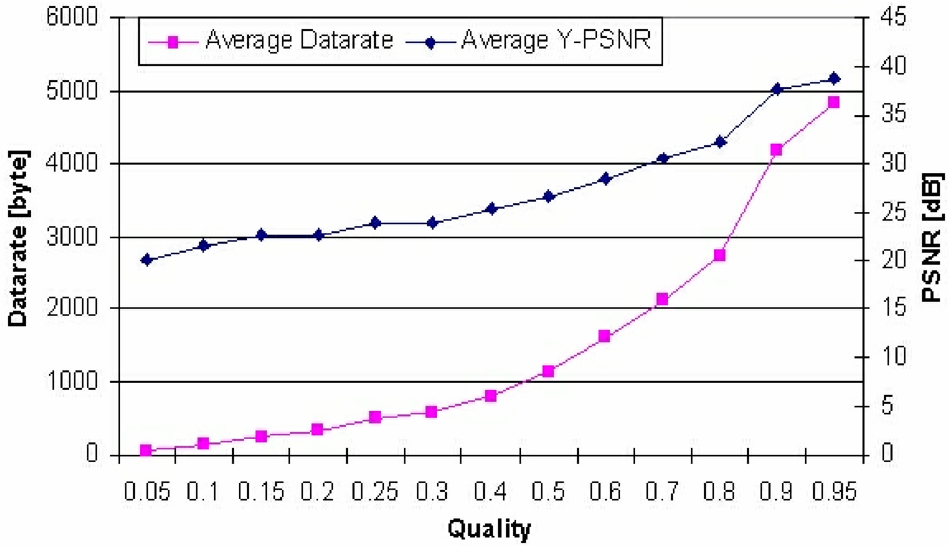
$$m = trunc(n * q)$$

n: number of WaveVideo packets of input frame

m: number of WaveVideo packets for output

q: quality factor in [0,1], 1 is the best quality





Frame 1 (q=0.8)

n = 7 packets (packet 0 to packet 6)

m = trunc (n * q) = trunc (7 * 0.8) = 5

⇒ packet 0 to packet 4 will pass the filter

⇒ packet 5 and 6 will be discarded

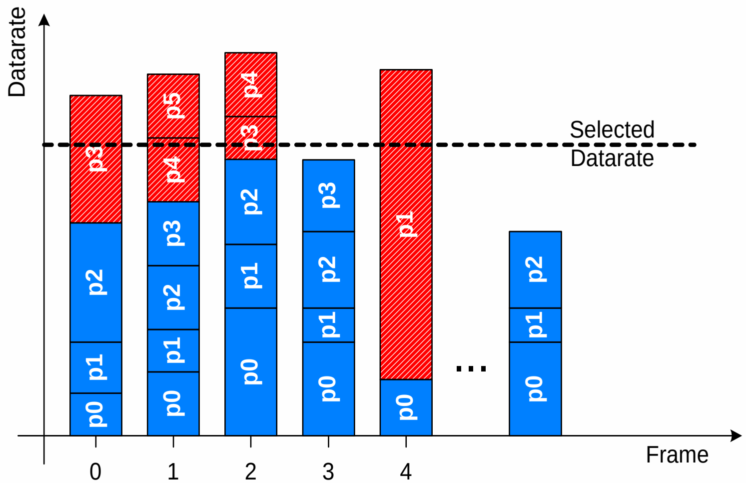
Examples

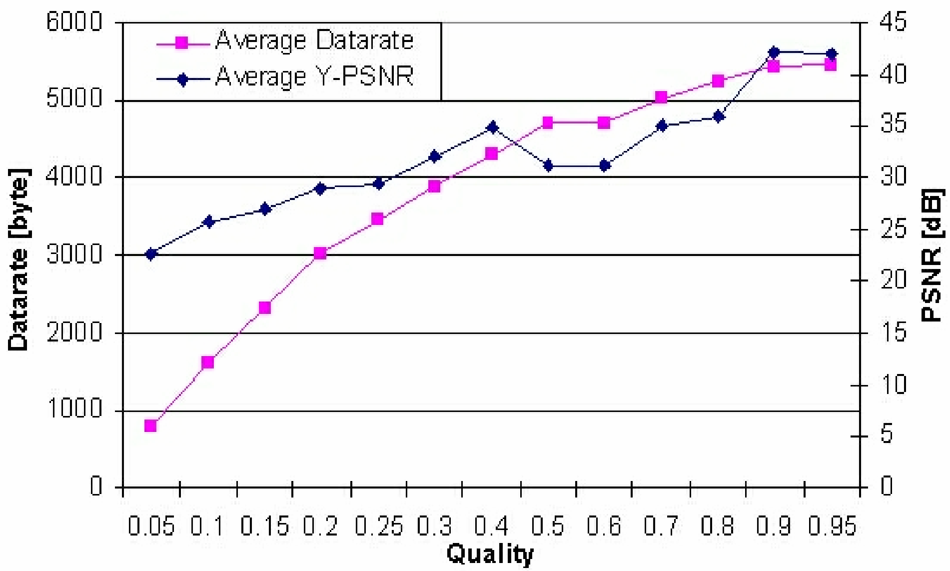
- **WaveVideo Filtering**
  - Syntactical Filter
  - Data-rate Shaper

$$m < numberOfPackets(frame_k)$$

and

$$\sum_{i=0}^{i=m} size(p_i) < datarate_{selected}$$





Examples

- ***WaveVideo Filtering***
  - q=0.9

Examples

- ***WaveVideo Filtering***
  - q=0.8

Examples

- ***WaveVideo Filtering***
  - q=0.7

Examples

- ***WaveVideo Filtering***
  - q=0.6

Examples

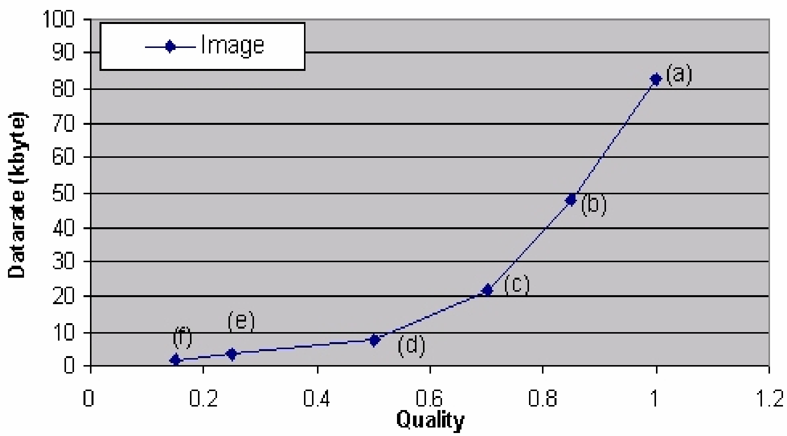
- ***WaveVideo Filtering***
  - q=0.5

Examples

- ***WaveVideo Filtering***
  - q=0.3

Examples

- **_WaveVideo Filtering_**
  - Visual Quality



Q-factor	Datarate (byte)	Compression factor
1.0	82800	1:1
0.85	47959	1:2
0.7	21775	1:4
0.5	7697	1:11
0.25	3455	1:24
0.15	1583	1:52

q=1.0

q=0.85

q=0.7

q=0.5

q=0.25

q=0.15

Examples

- ***WaveVideo Filtering***
  - Semantical Filter

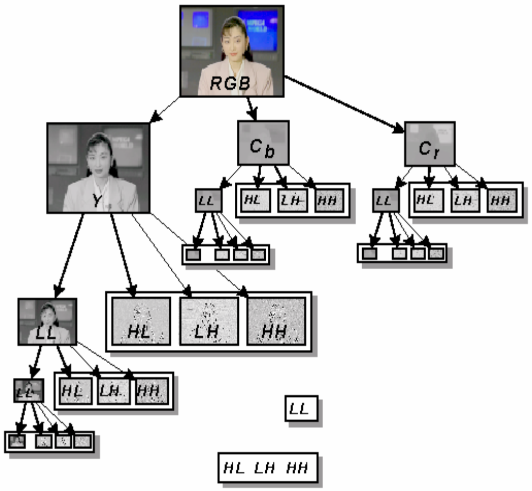
RTP	WaveVideo Header	WaveVideo Payload

Tag contains information about:

- **Quality Layer**
- **Colour Channel**
- **Recursion Depth**
- **Spatial Filtering**

Combi Filter allows adaptation of:

- **Frame rate**
- **Frame size**
- **Luminance quality**
- **Chrominance quality**



Examples

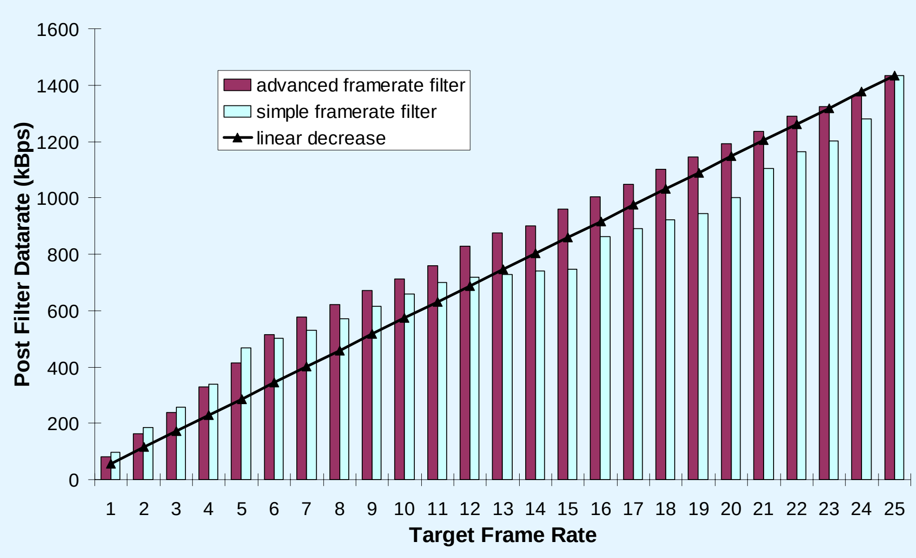
Simple Frame Rate Filter

- source rate specified in I-frame header
- $r := r_{src} / r_{dest}$
- for each frame f_i the next frame f_{fwd} to be forwarded
  - if f_i was forwarded, $f_{fwd} = f_i + r$
- all other frames are dropped
- adjust in all packets $r_{src} = r_{dest}$
- all Delta-frames referring to a dropped I-frame are dropped

Advanced Frame Rate Filter

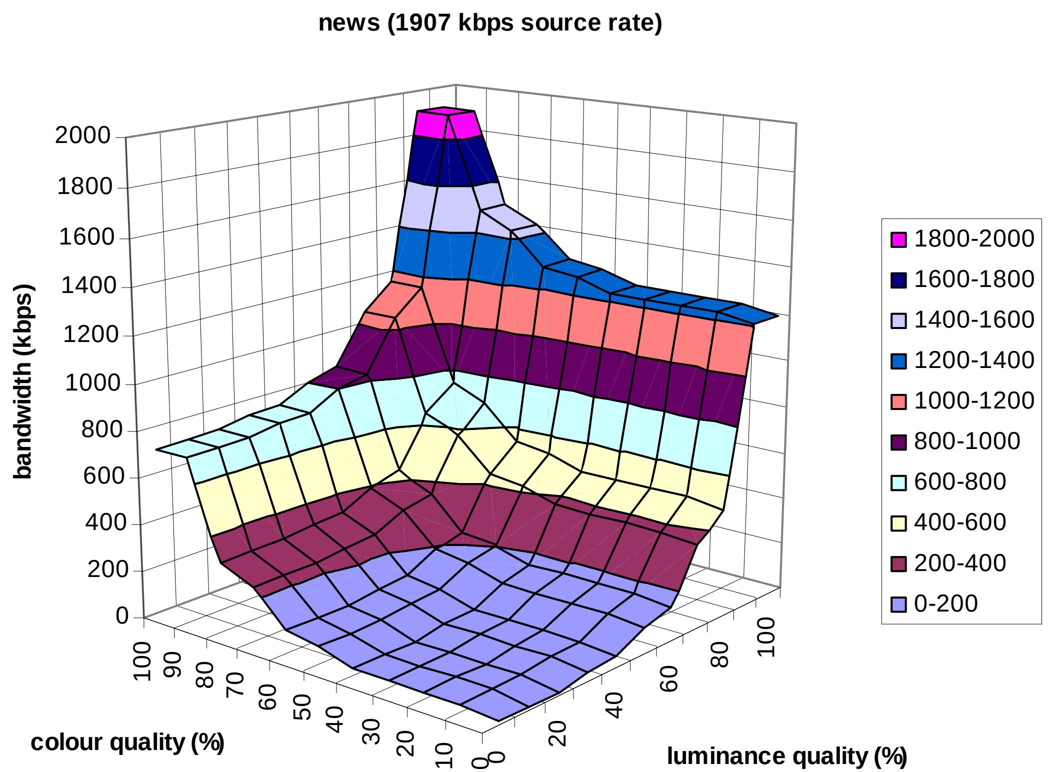
- Estimate r_{src} and r_{src_I}
  - $r_{src} = a\, r_{src} + (1-a)\, r_{\_max} / (frameNr\text{-}lastFrameNr)$
  - $r_{src\_I} = b\, r_{src\_I} + (1-b)\, r_{\_max} / (frameNr\text{-}last\_I\_FrameNr)$
- If $r_{src\_I} > r_{dest}$ the actual I-frame is forwarded only if the output frame rate does not exceed r_{dest}
- Otherwise, Delta-frames have to be forwarded, only if the post filtered frame rate does not exceed r_{dest} when forwarding the next expected I-Frame.

Comparison of Simple and Advanced Frame Rate Filter



Examples

- Combination of filters of course also possible
  - Parameters:
    - resolution r_l
    - colour resolution r_c
    - frame size r_s
    - for fixed frame rate



Examples

- ❑ ***WaveVideo Filtering***

Demo Video

Examples

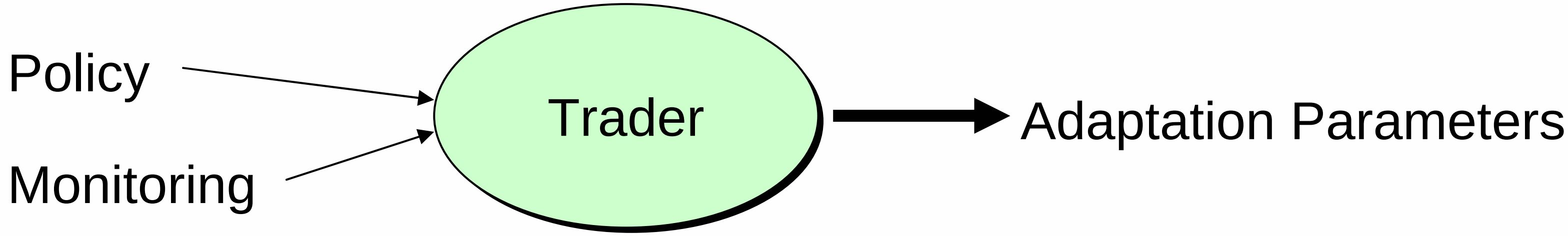
- ***WaveVideo Filtering***
  - Video-on-Demand Scenario

Beverly Hills Cop Movie (Scene)
352 x 288 pixels
25 fps

User QoS Policy:

Data rate <= 10 Mbit/s, Frame rate <= 10 f/s

A: Frame rate is more important than Frame quality
B: Frame rate is of equal importance then Frame quality



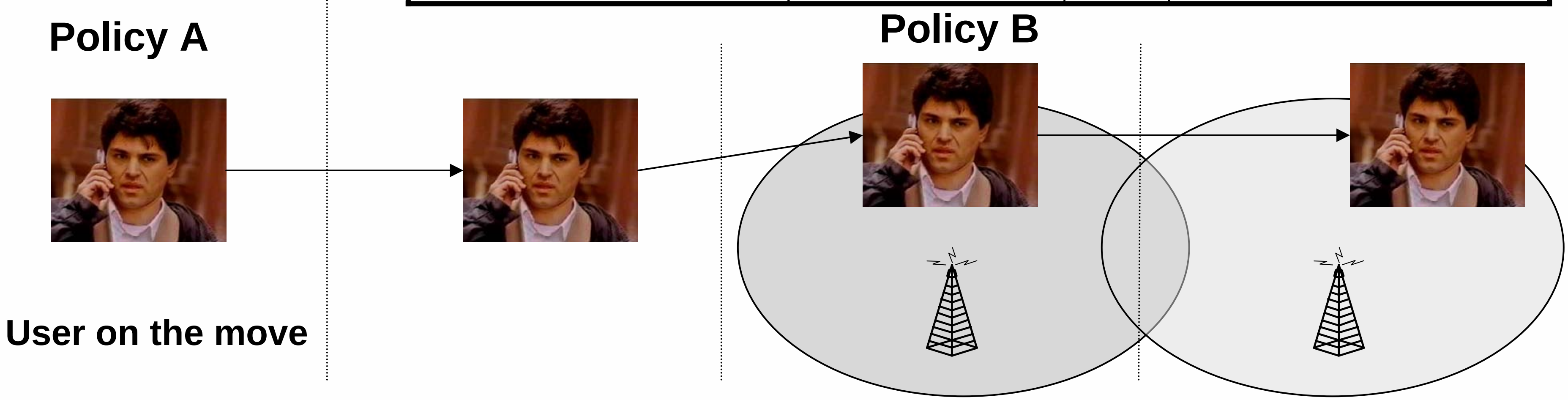
Examples

- ***WaveVideo Filtering***
  - Video-on-Demand Scenario

User QoS Policy:

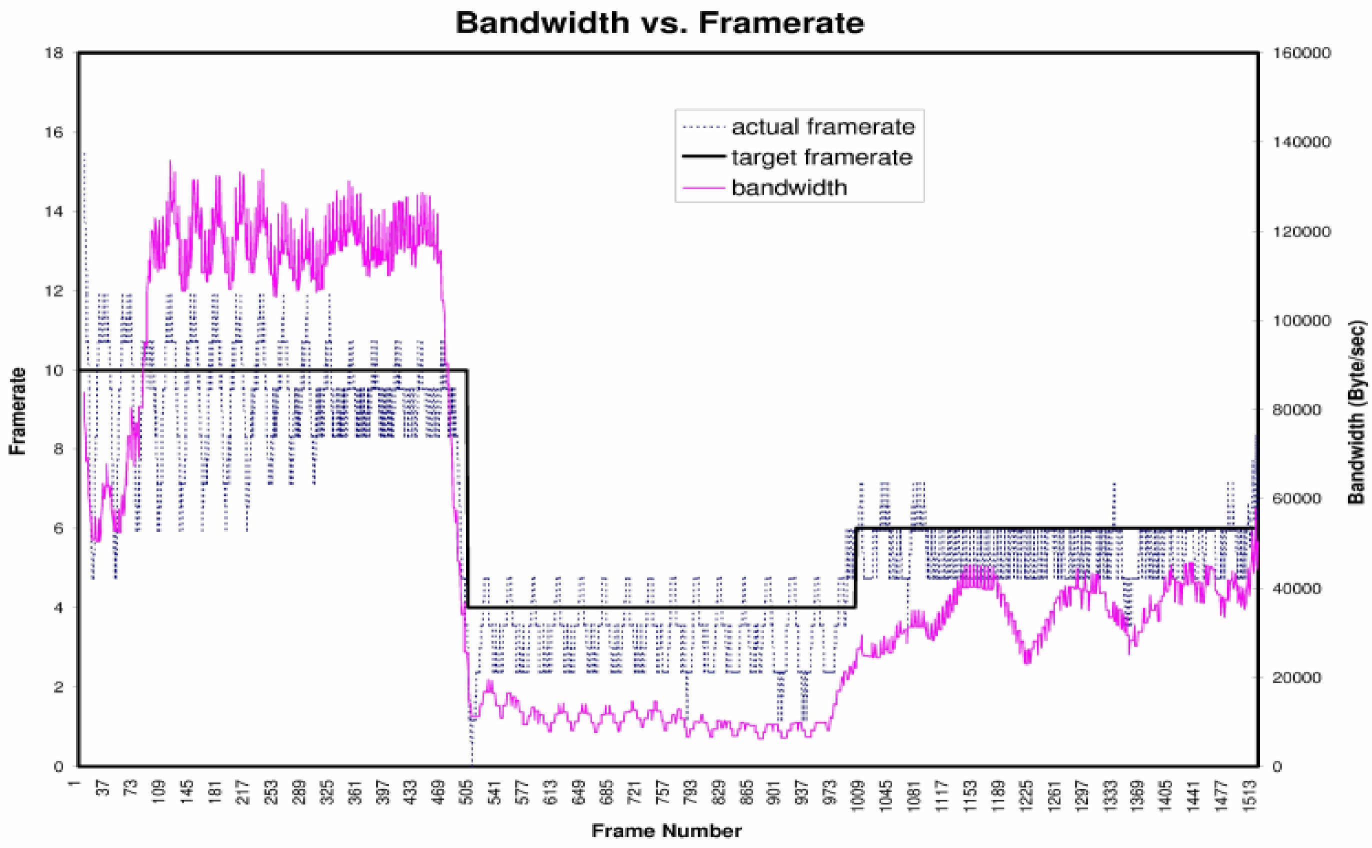
Data rate <= 10 Mbit/s, Frame rate <= 10 f/s
A: Frame rate is more important than frame quality
B: Frame rate is of equal importance then frame quality

<= 1 Mbit/s -> 10f/s, dropping only subbands of highest layer	<= 200 kbit/s -> 4 f/s, dropping a lot of subbands	<= 500 kbit/s -> 6 f/s, dropping less subbands
Ethernet	WaveLan 1 (bad Signal)	WaveLan 2 (better Signal)



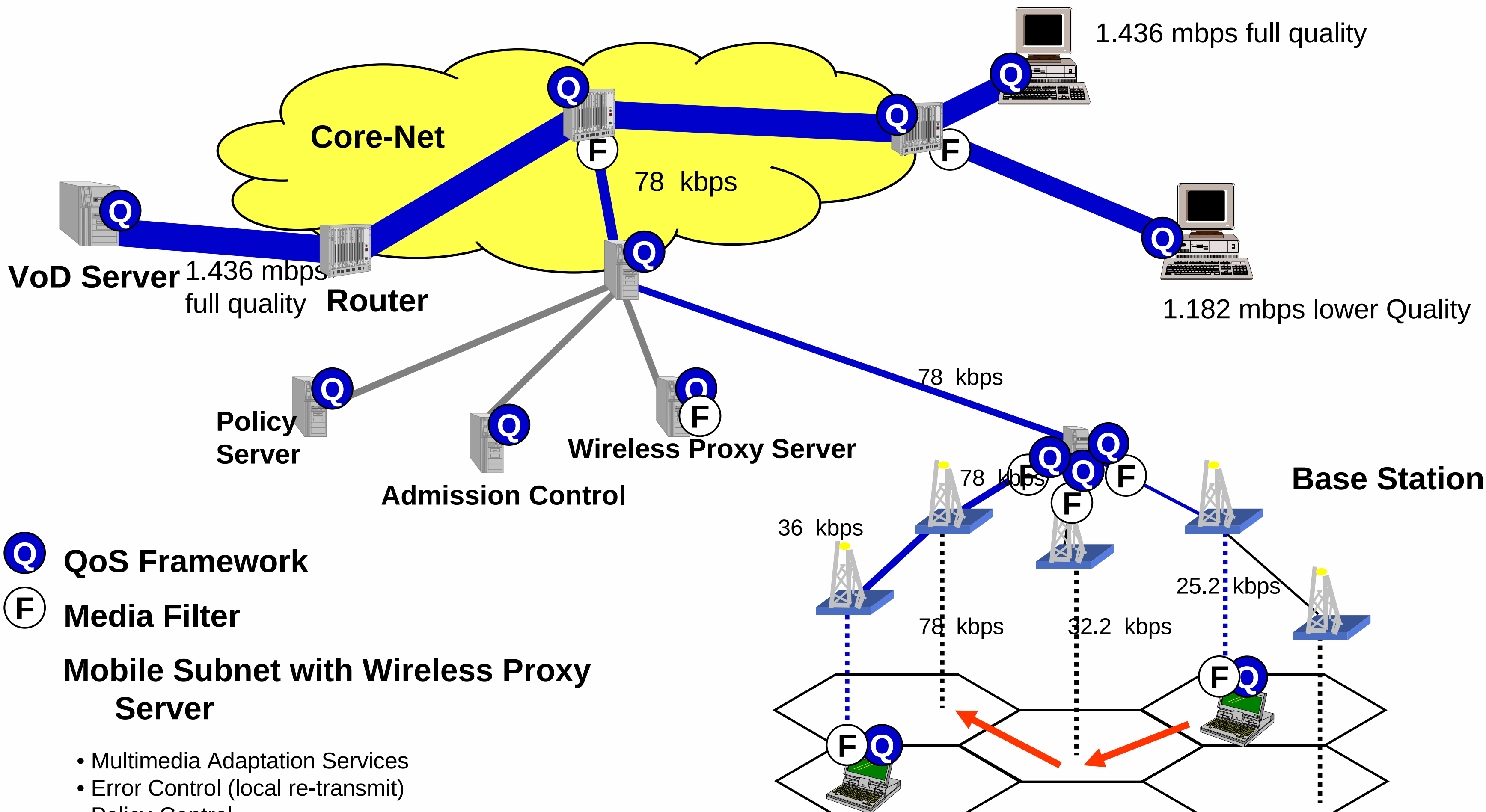
Examples

- ❑ ***WaveVideo Filtering***
  - Video-on-Demand Scenario



Examples

- Generic Usage Scenario for media filters



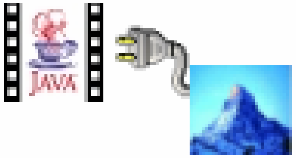
Mobile Subnet with Wireless Proxy Server

- Multimedia Adaptation Services
- Error Control (local re-transmit)
- Policy-Control

Examples

❑ *WaveVideo Filtering*

- **Christian Kücherer**: Diploma thesis (University of Applied Sciences Mannheim - Prof. Bengel, in co-operation with NEC, 2001)
- Prototype in JMF (Sun – Java Media Framework)
- Real-time streaming using RTP/RTCP
- Integrate WaveVideo Codec into JMF by plugins:
  - Packetizer (at the sender)
  - Depacketizer (at the receiver)
  - QoS filter (for quality adaptation at the sender or in the network)



Christian Kücherer, Andreas Kassler, Andreas Schrader, Oliver Haase
End Device and Network Adaptation of WaveVideo Streams
Proceedings of the Conference on Advances in Infrastructure for Electronic Business, Science, and Education on the Internet
(SSGRR) L'Aquila, Italy, August 6-12, 2001

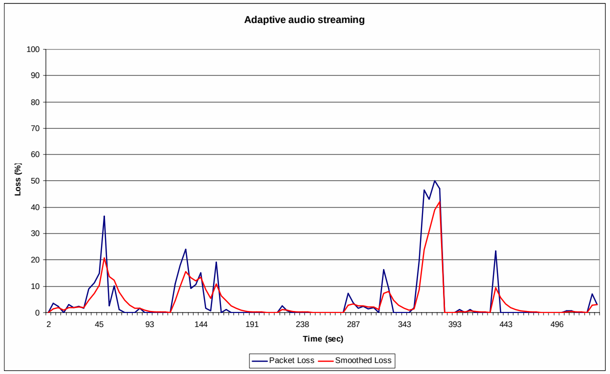
Andreas Kassler, Christian Kücherer and Andreas Schrader
Efficient Wavelet Video Filtering
2nd International Workshop on Quality of future Internet Services
(QofIS) Coimbra, Portugal, September 24-26, 2001

Examples

- ❑ ***Audio Adaptation***
  - **Hyung-Woo Kim**: Master thesis
    (University of Stuttgart - Prof. Kühn, in co-operation with NEC, 2001)
  - JMF-RTP/RTCP implementation (sender side)
  - Adaptation Strategy: don't lose packets

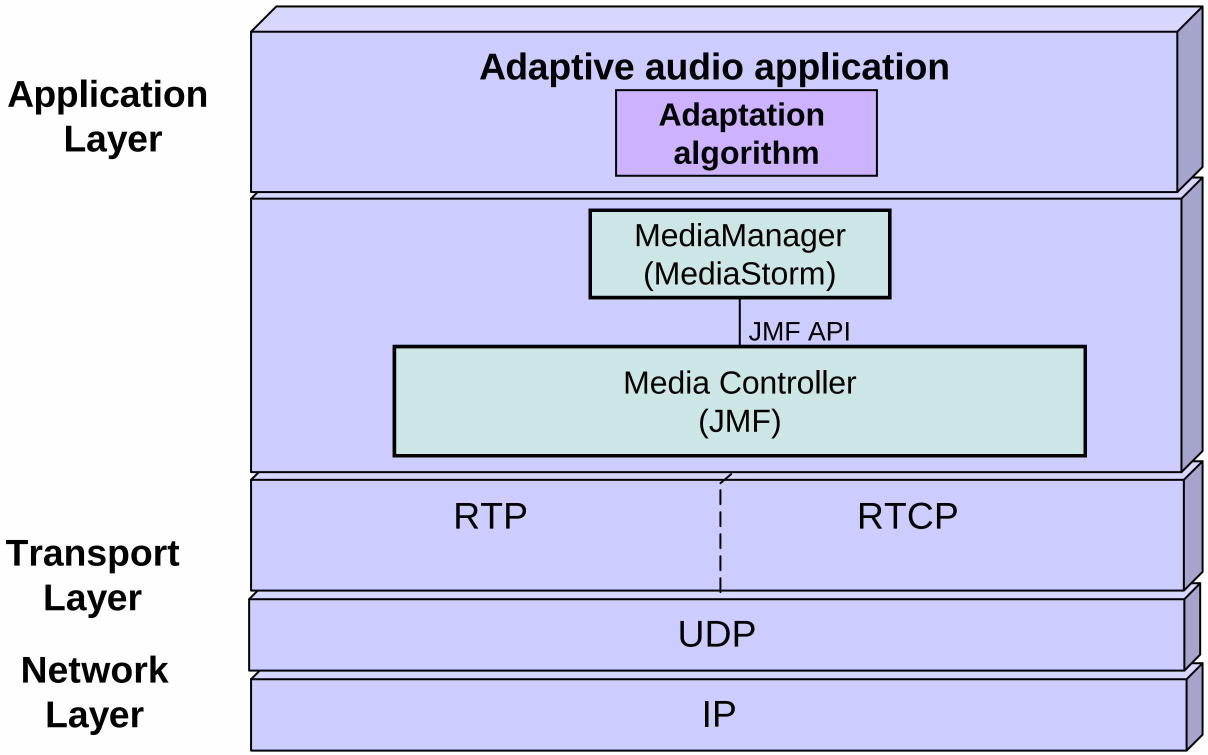
  - Algorithm:
    - Input:
      - RTCP Receiver Reports to indicate packet loss
      - Smoothed packet loss (λ): low pass filter over packet loss
        → Avoid adaptations with fast loss fluctuations
    - Output:
      - Commands to increase or decrease the data rate
      - $loss_{min}$,$loss_{max}$:
        thresholds fixed after several measurements
        → When exceeded, adaptation activated

$$\boxed{\lambda_i = (1-\alpha)\cdot\lambda_{i-1} + \alpha\cdot \mathrm{PacketLoss} \quad (0 < \alpha \leq 1)}$$



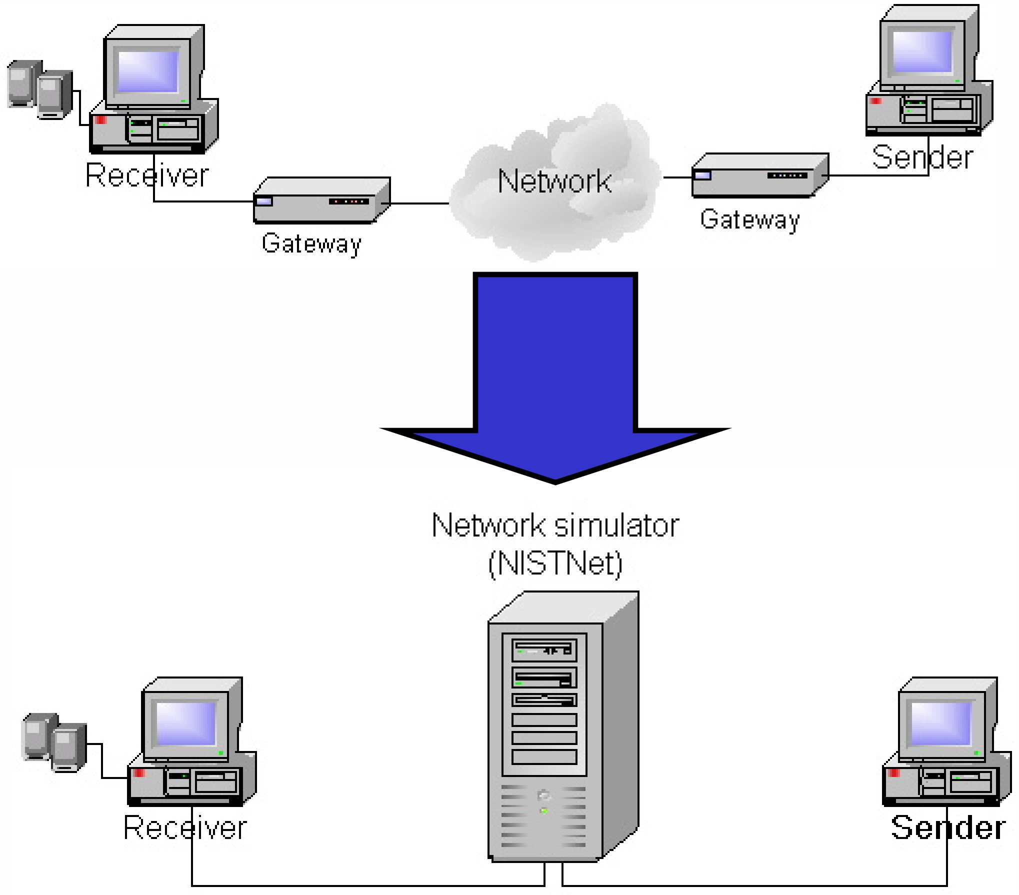
Examples

- **_Audio Adaptation_**
  - Architectural View



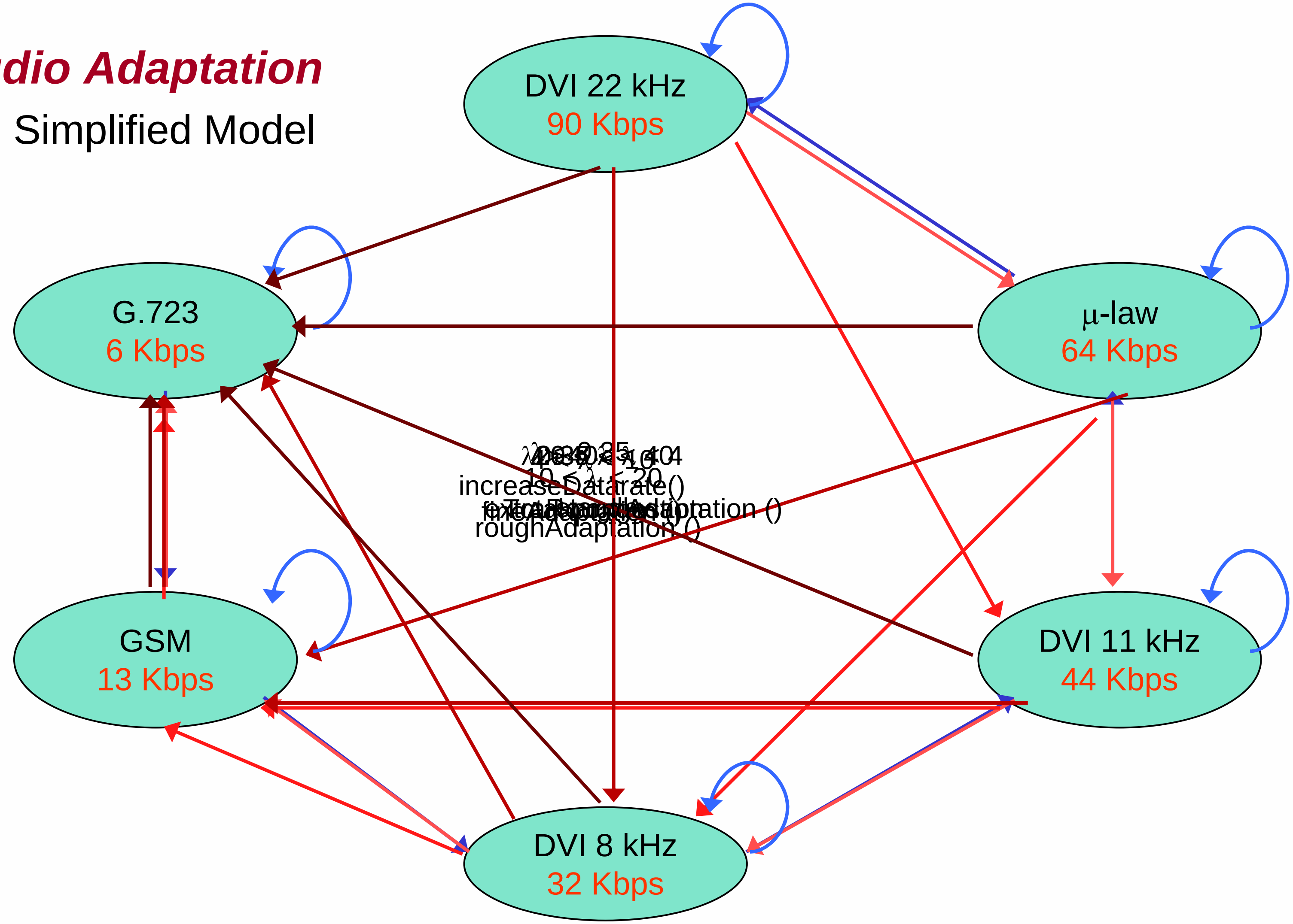
Examples

- **_Audio Adaptation_**
  - Hardware Setup



Examples

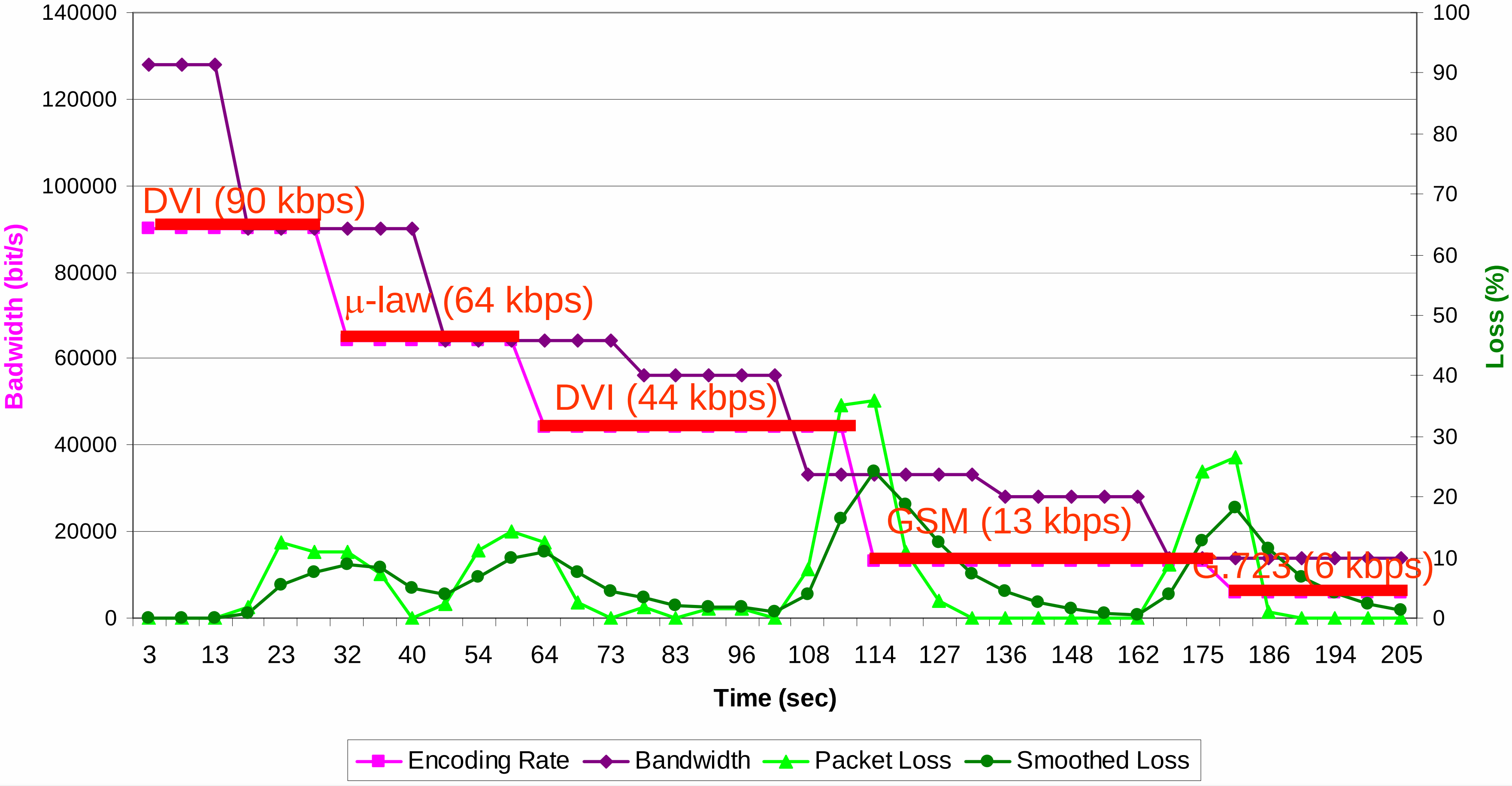
- ***Audio Adaptation***
  - Simplified Model



Examples

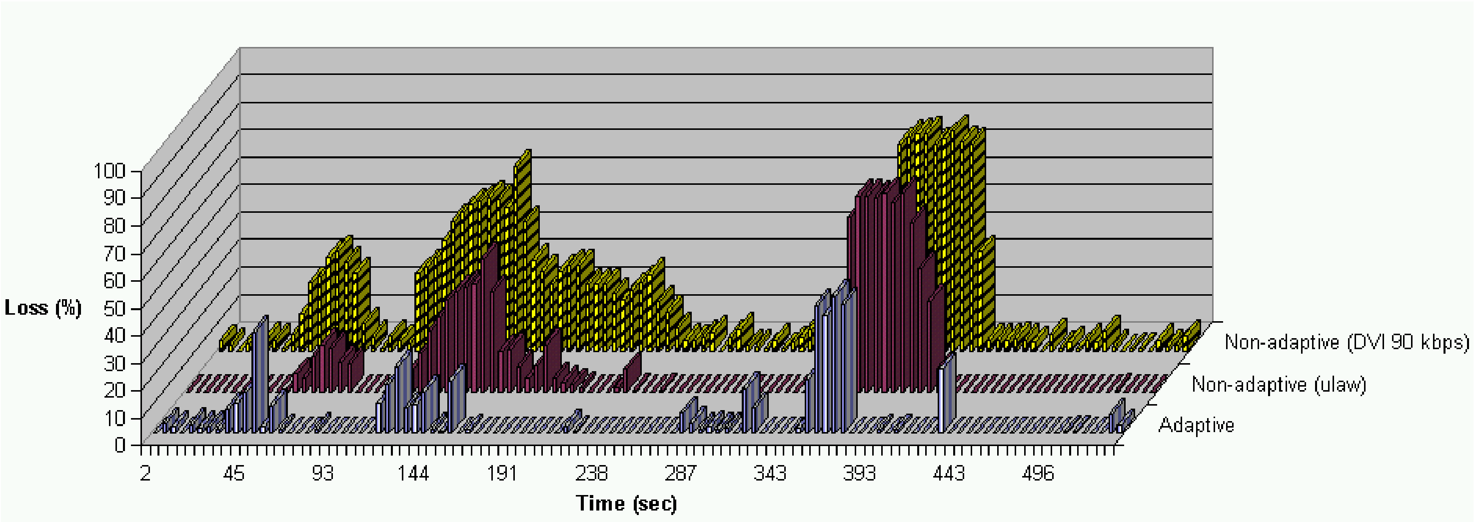
Audio Adaptation

- Experimental Results



Examples

- ❑ ***Audio Adaptation***
  - Results



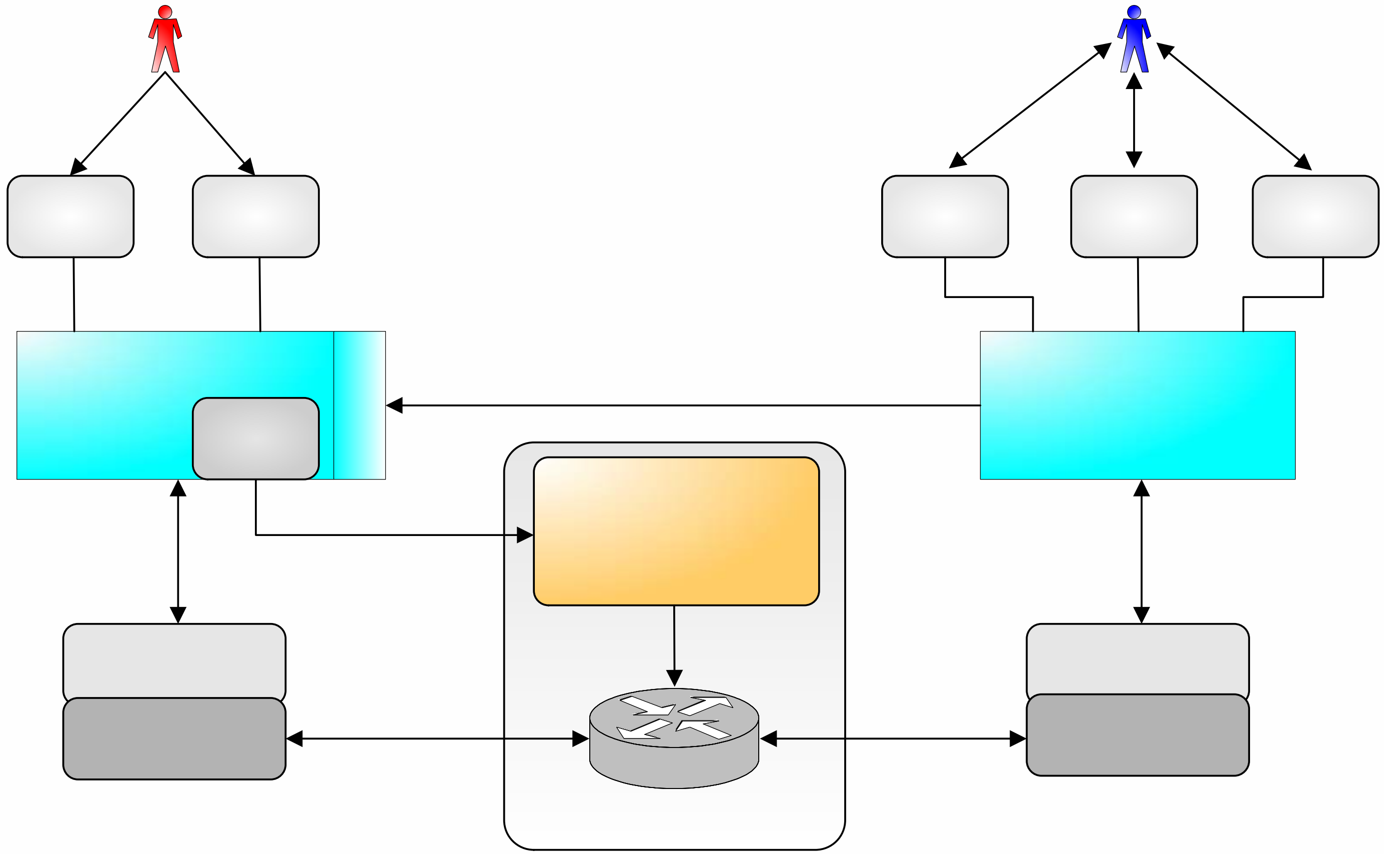
Streaming	Average packet loss (%)	Average throughput (kbit/s)
Adaptive	4.64	47
Non-adaptive (DVI 90 kbit/s)	10.76	56
Non-adaptive (ulaw)	20.63	70

Examples

- ***MPEG-4 Filtering***
  - Philipp Bostan: Master thesis (University of Applied Sciences Mannheim - Prof. Körner, in co-operation with NEC, 2002)
  - Integration of DivX4.12 codec for JMF
  - Implementation of RTP/RTCP (de-)packetizer
  - Extended measurements using NistNet network emulator
  - Adaptation algorithms for frame rate and quantization table
  - Decisions based on jitter, delay and loss

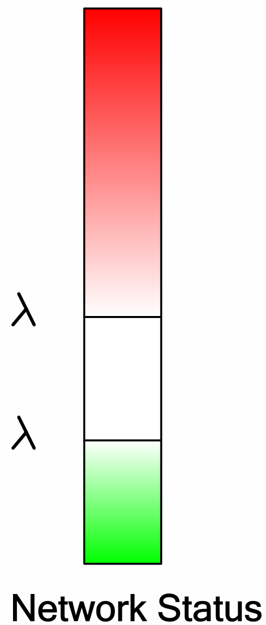
Examples

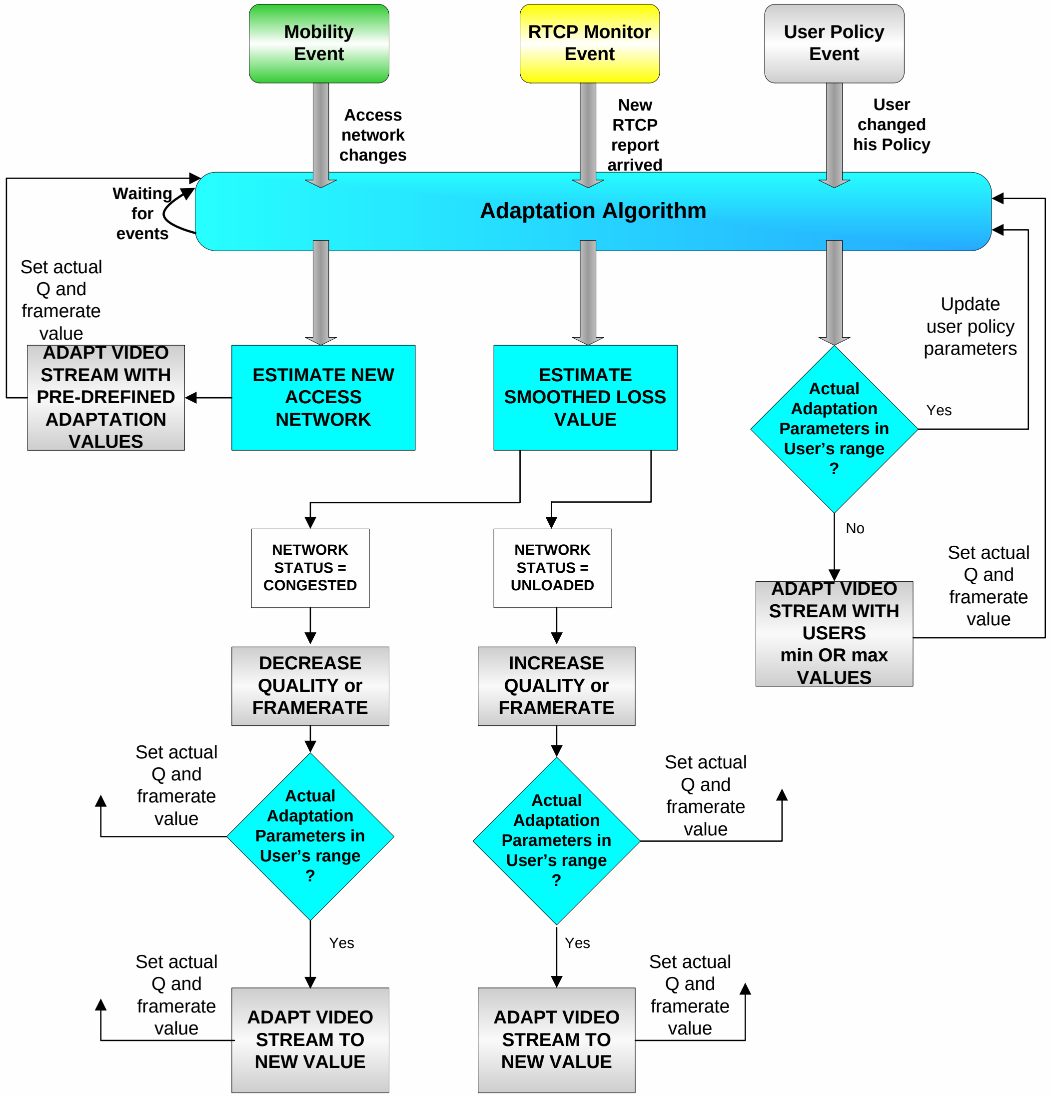
- **_MPEG-4 Filtering_**
  - System Architecture



Examples

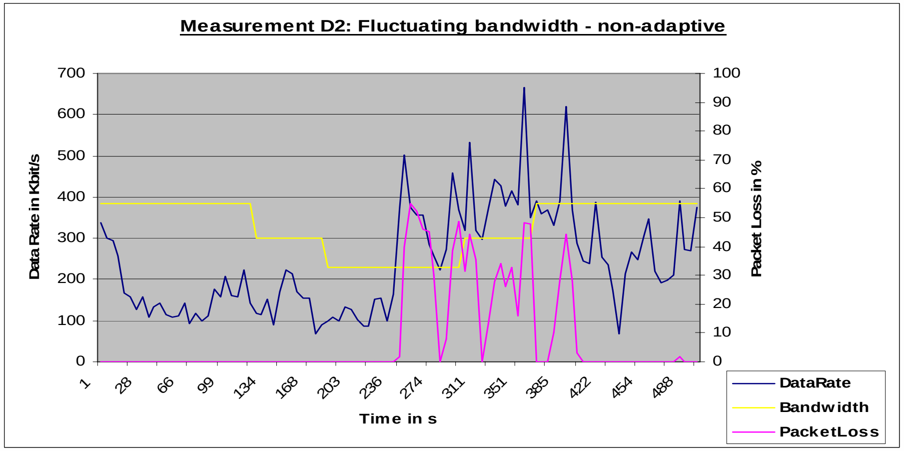
- **MPEG-4 Filtering**
  - Algorithm



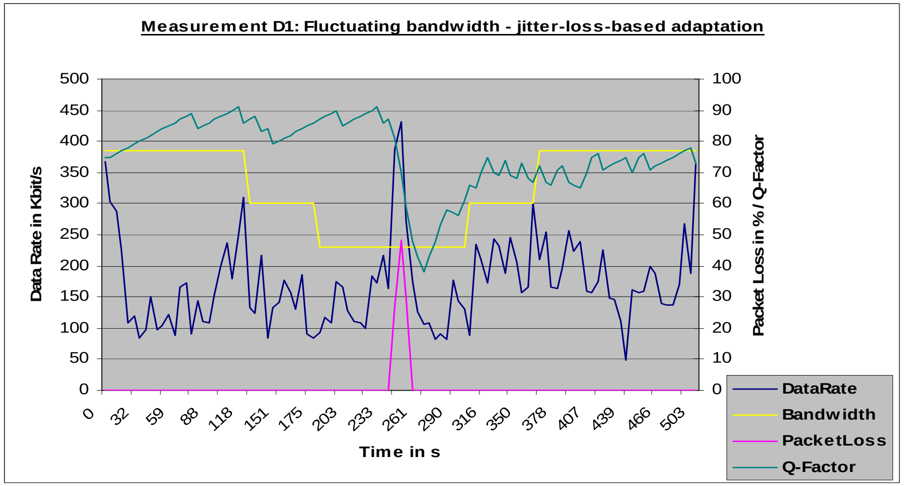


Examples

- **MPEG-4 Filtering**
  - Results



Adaptation off



Adaptation on

Examples

Seamless Codec Switching

- Schrader/Carlson, 2002
- Realized in Java (JMF/RTP/RTCP)
- Pluggable Adaptation Modules
  (Frame Filter, Quality, Datarate, Codec Switch)
- MPEG-4 Packetizer / Depacketizer / Frame Filter (DivX4.12)
- Results:
  - ***Gap time below 1 ms*** (measurement accuracy)
  - ***Zero packet loss*** (proved with packet sniffer)
  - ***Codec and media type independent***

Darren Carlson and Andreas Schrader
Seamless Media Adaptation with simultaneous Media Processing Chains
Proceedings of the ACM Conference on Multimedia
Juan-les-Pins, France, December 1-6, 2002

(International patent pending)

Examples

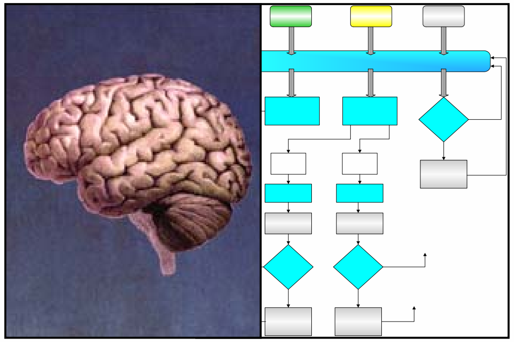
❑ *Automatic Media Adaptation*

User QoS Policy

Monitoring Results

Movement Detection

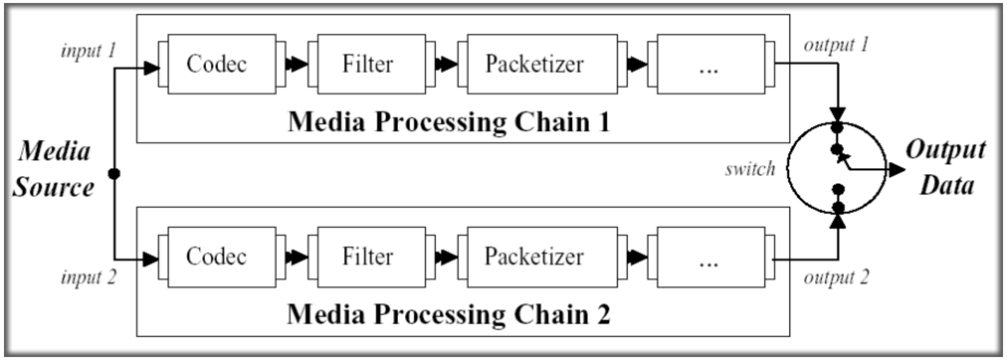
Resource Availability



MediaStorm
Java Multimedia Technology

- **QoS Parameters**
- **Codec Selections**
- **Filter Selections**

❑ *Seamless Chain Switching*

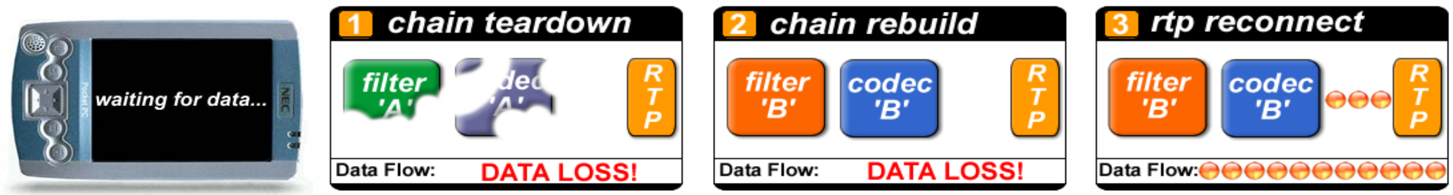


- **Media Processing Chains consist of Codecs, Filter, Packetizer etc.**
- **Switching between chains usually imposes additional degradations**
- **Our approach: seamless switching**

Examples

Traditional Adaptation Approach

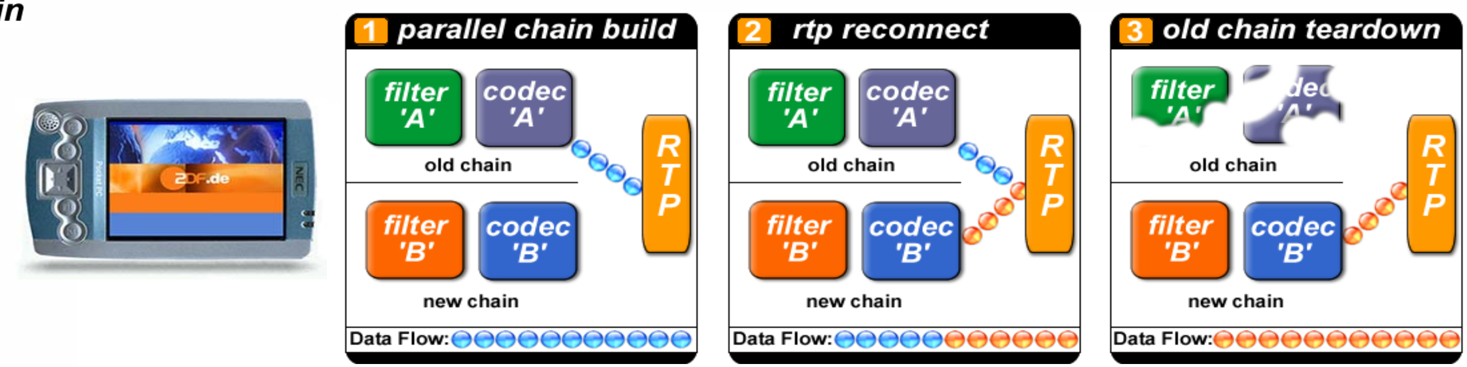
Original Media Chain



Problems:

- Data loss during chain teardown & rebuilding
- Long adaptation time interrupts stream

Seamless Adaptation Approach

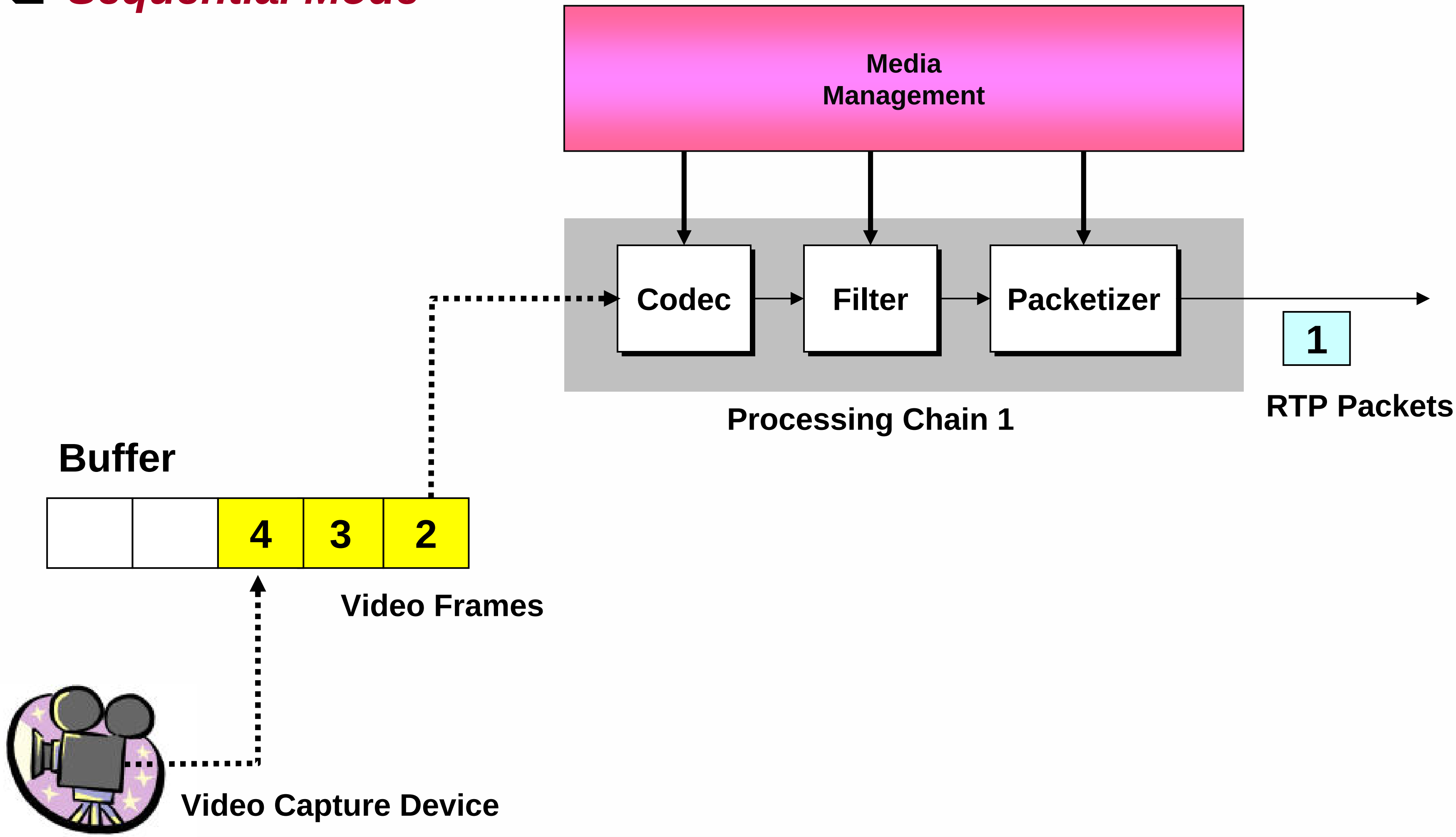


Advantages:

- No loss during chain reconstruction
- Reduced adaptation time

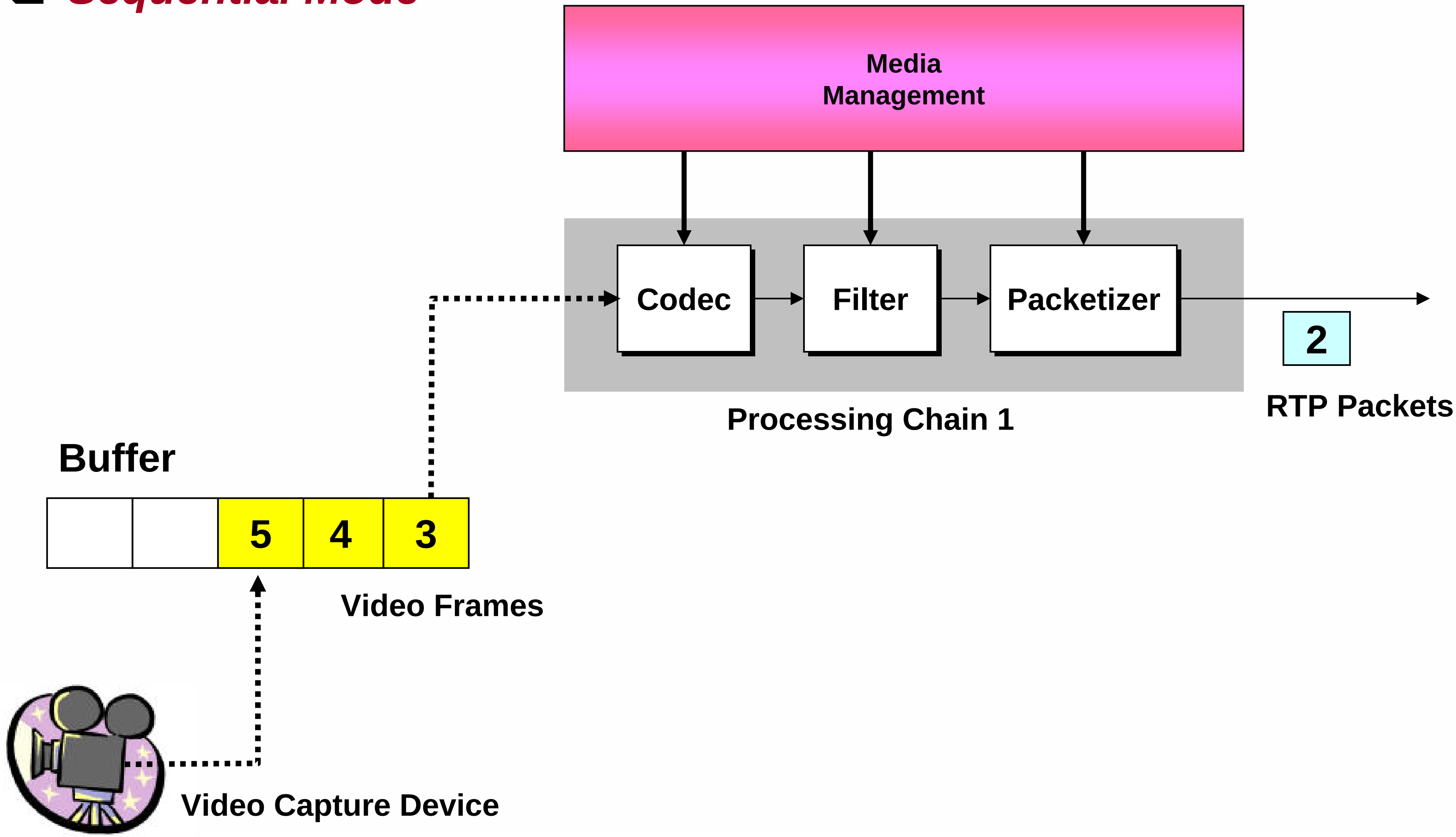
Codec Switching

- Sequential Mode



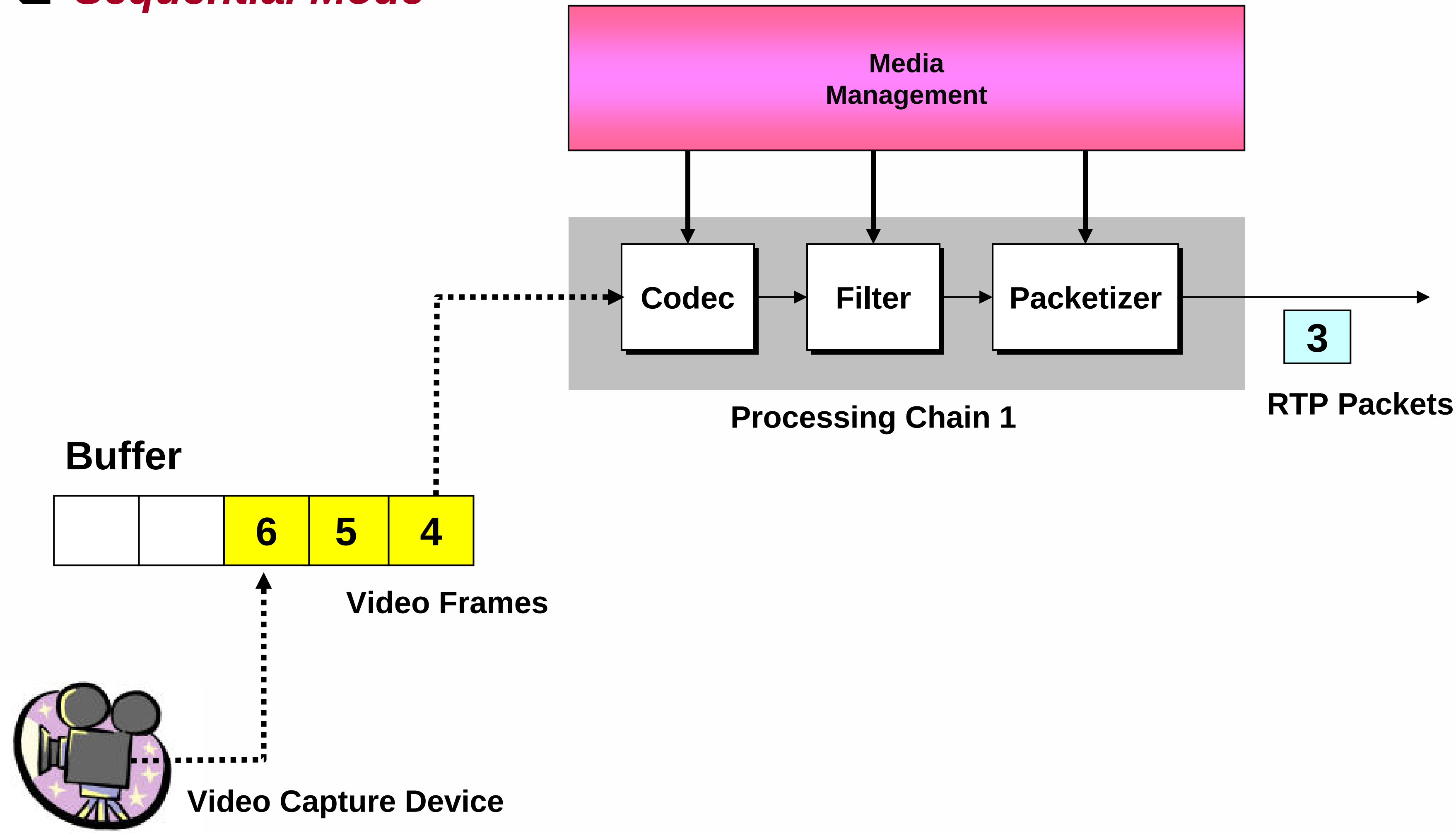
Codec Switching

- *Sequential Mode*



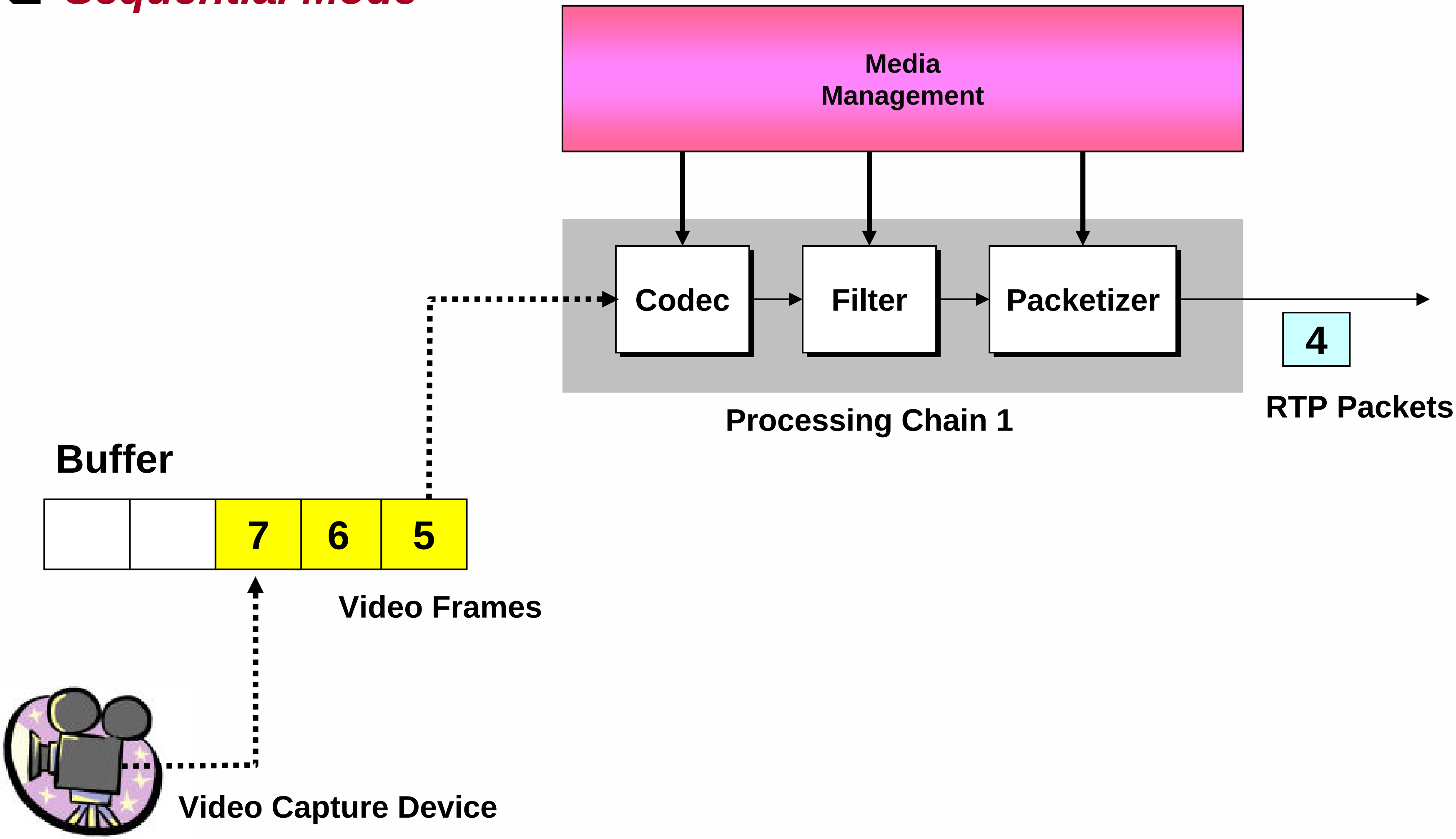
Codec Switching

- *Sequential Mode*



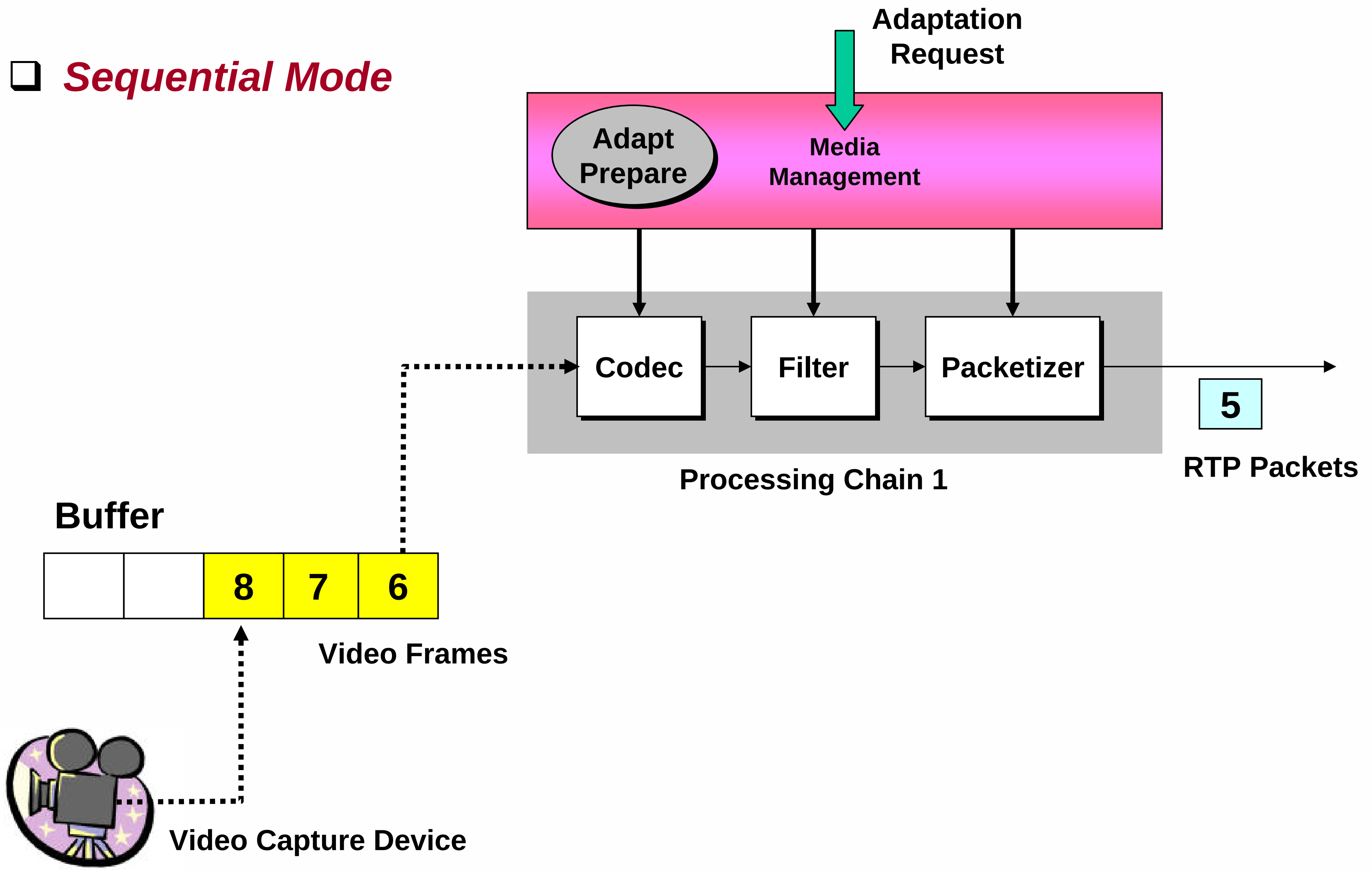
Codec Switching

- *Sequential Mode*



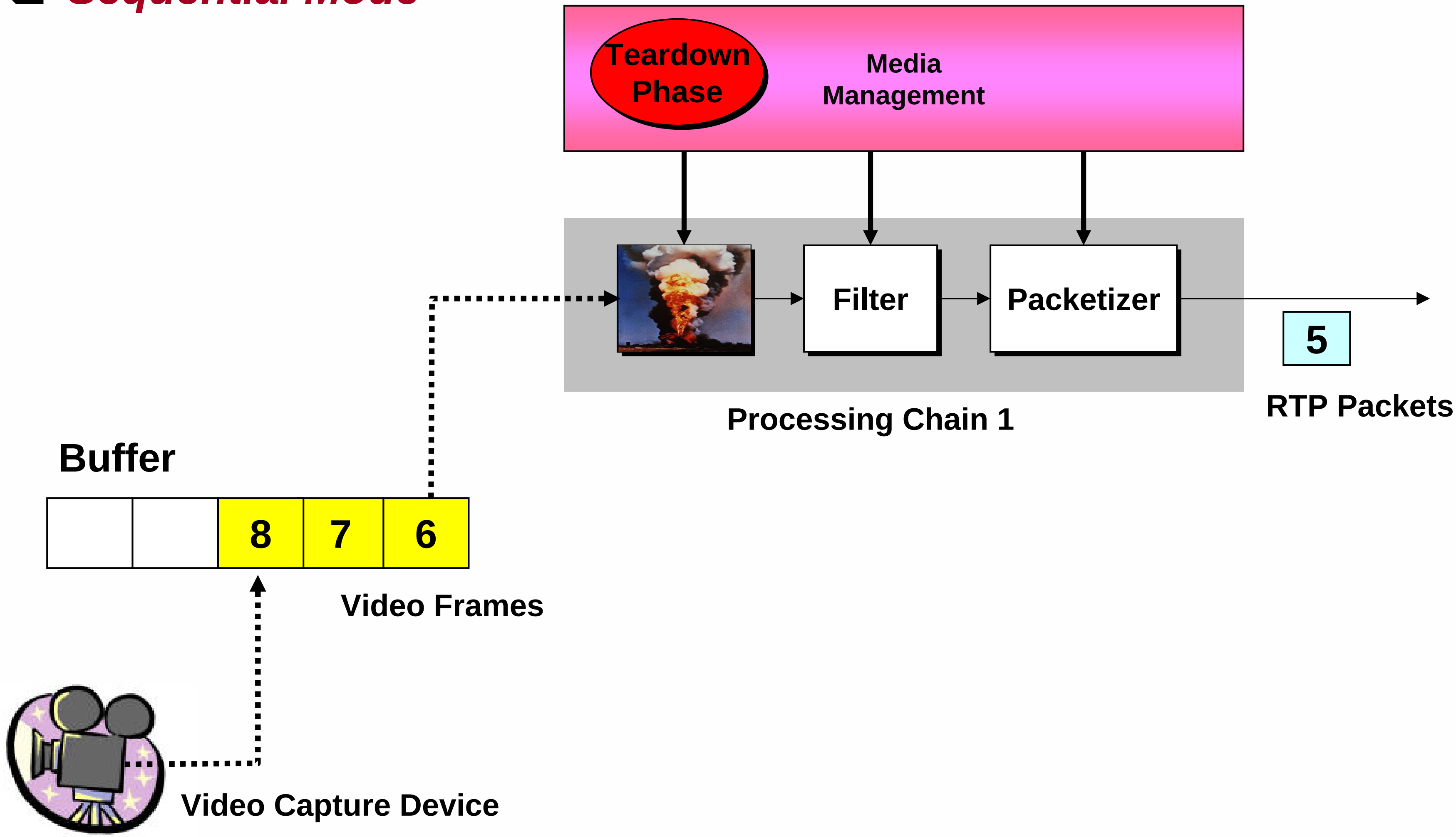
Codec Switching

- ***Sequential Mode***



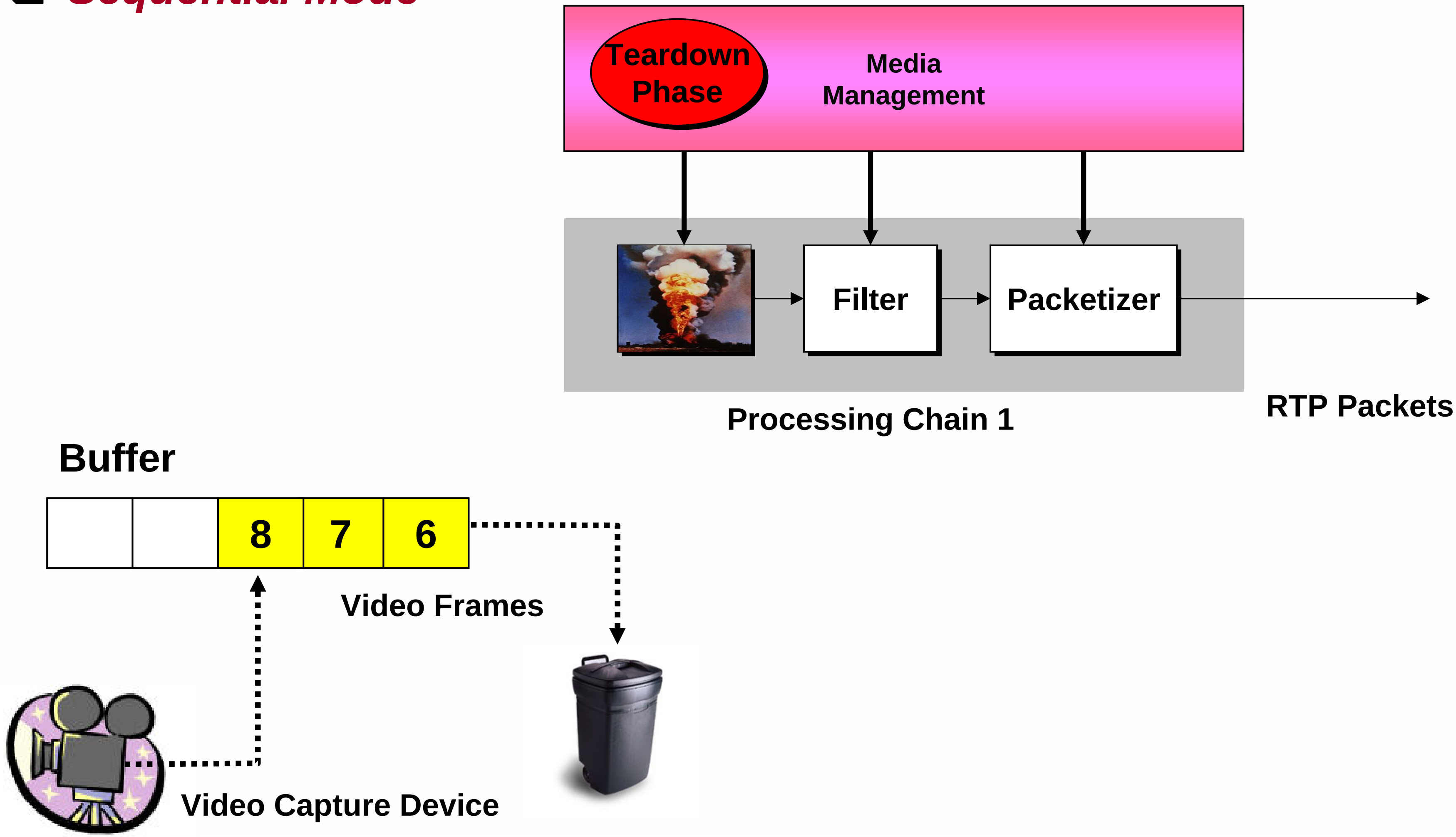
Codec Switching

- *Sequential Mode*



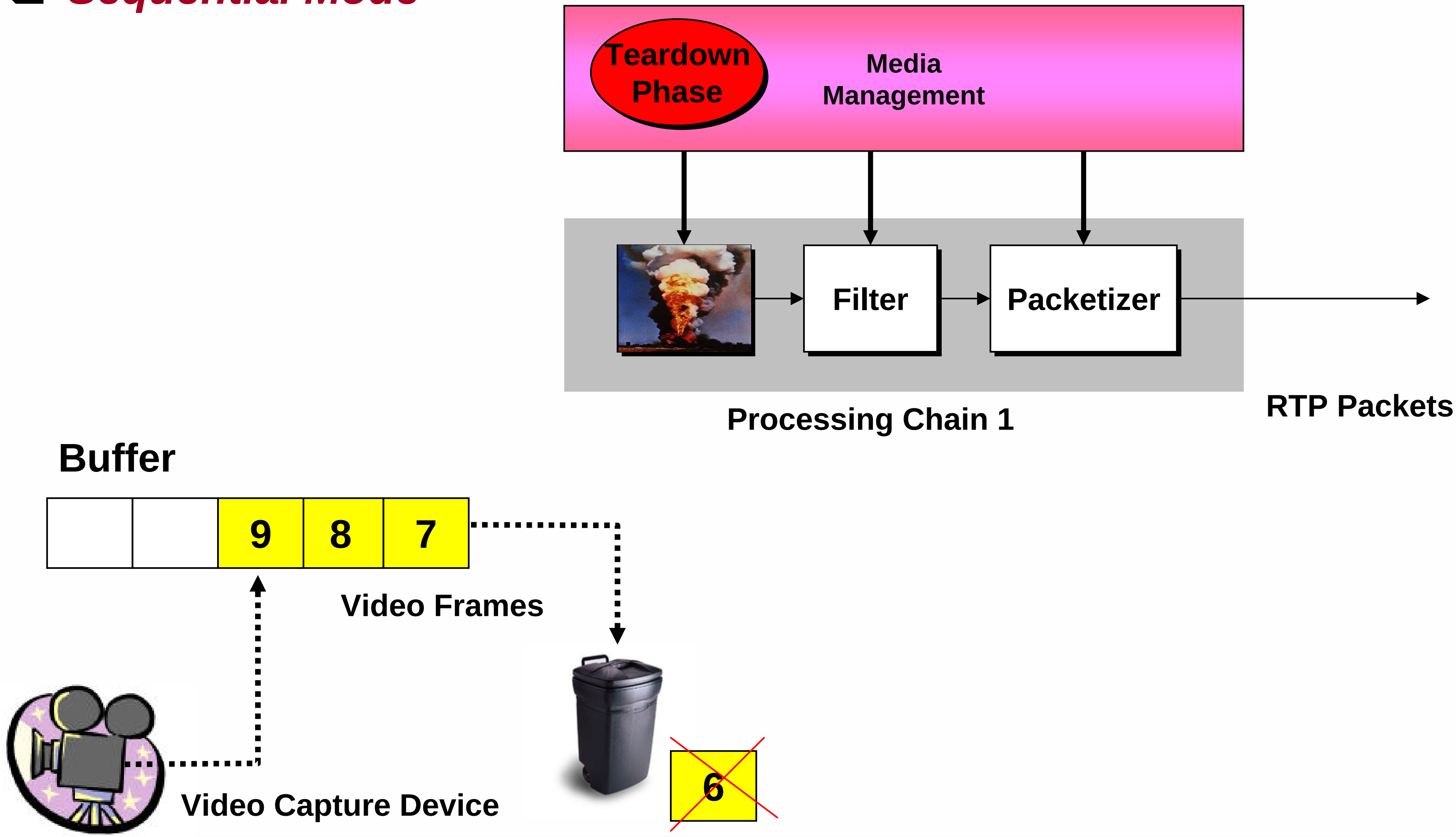
Codec Switching

- ❑ ***Sequential Mode***



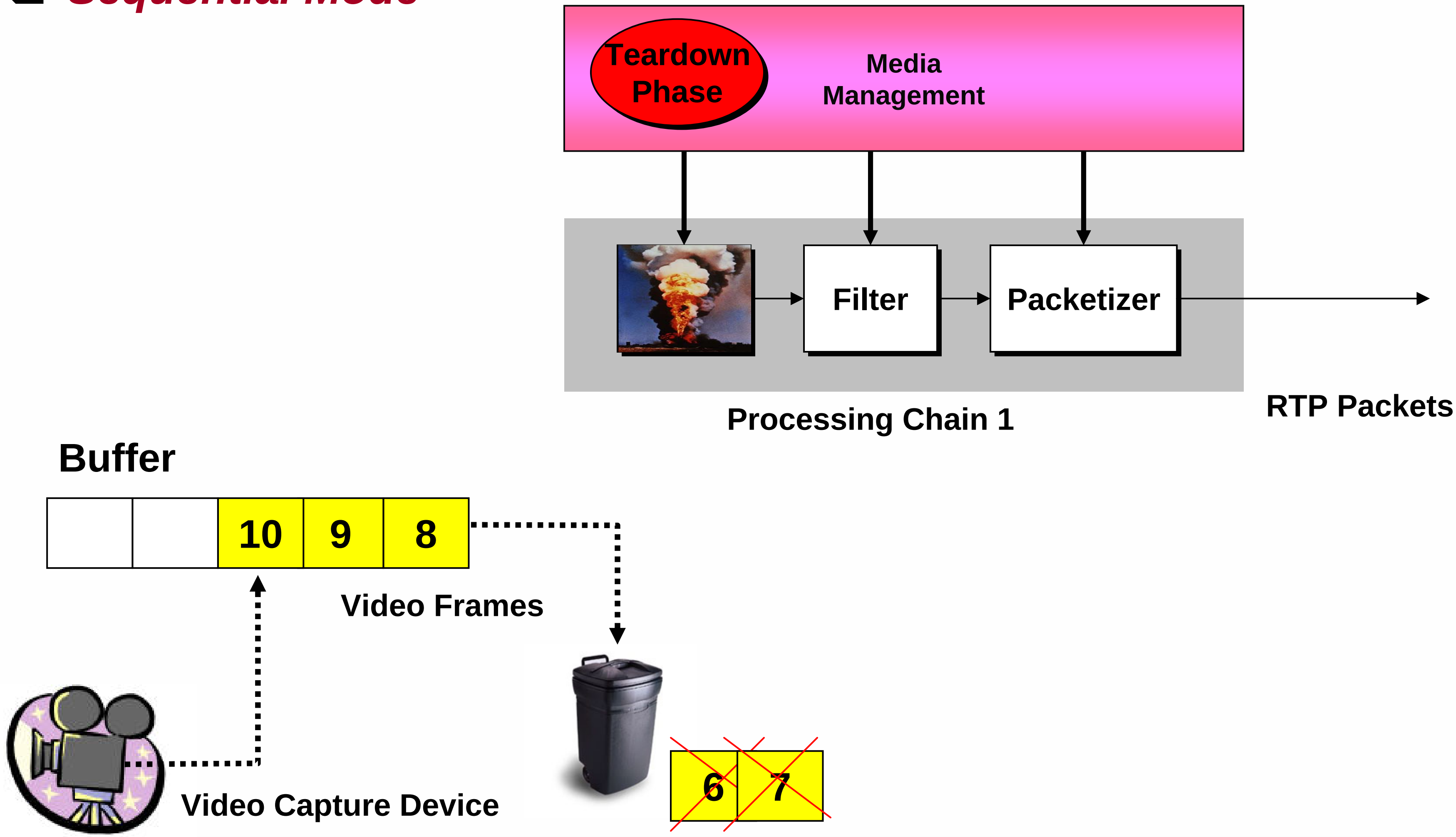
Codec Switching

- *Sequential Mode*



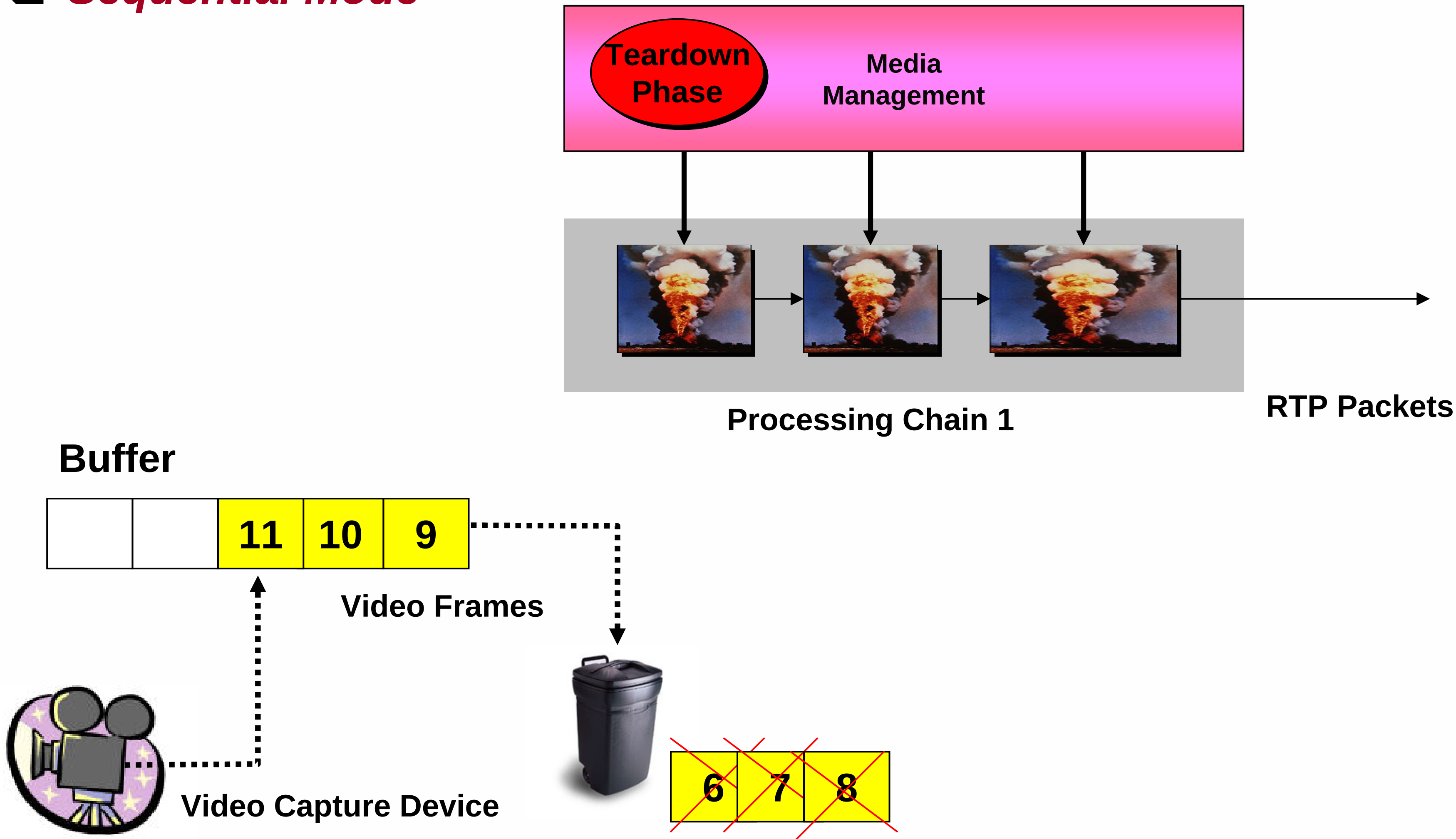
Codec Switching

- ***Sequential Mode***



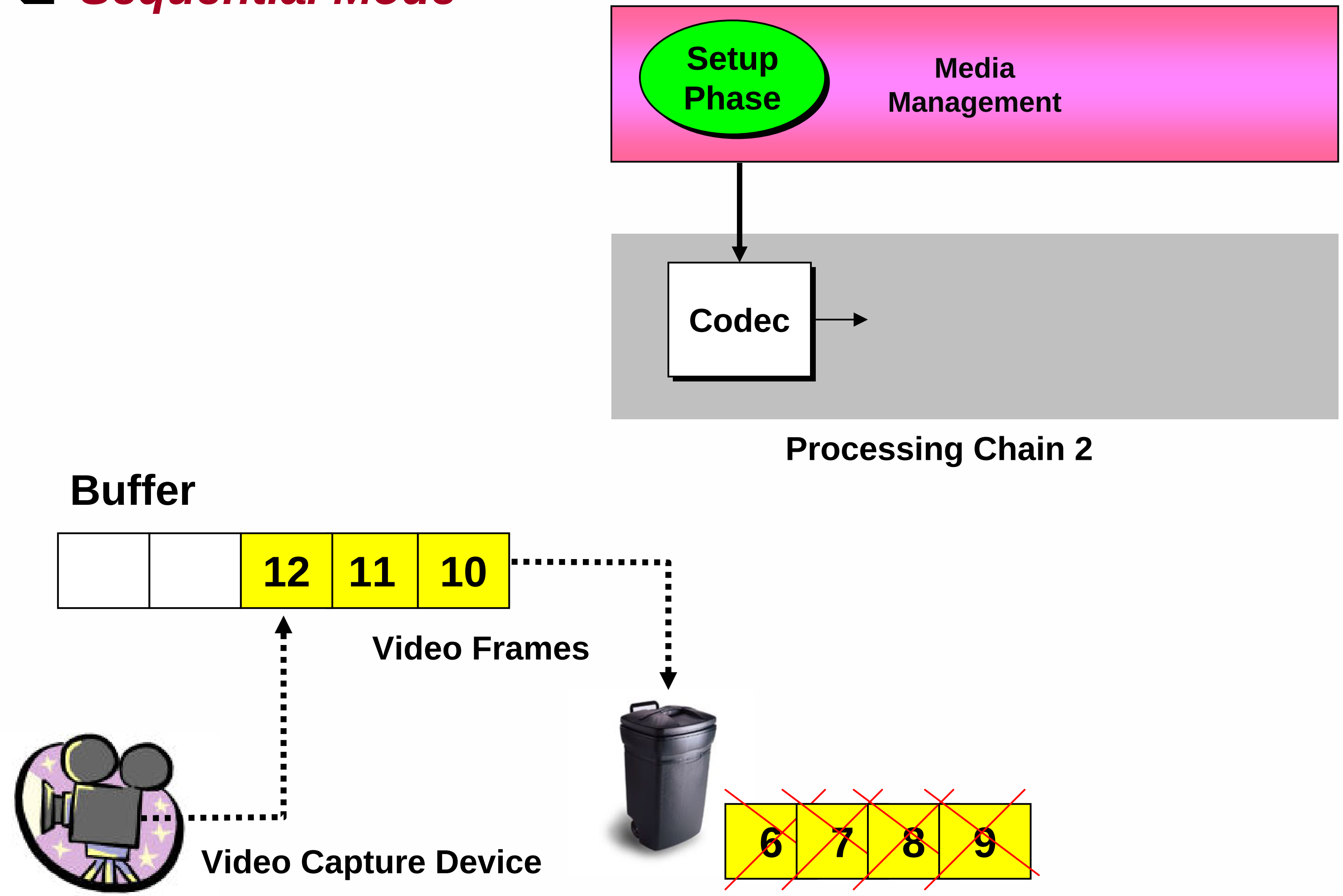
Codec Switching

- ❑ ***Sequential Mode***



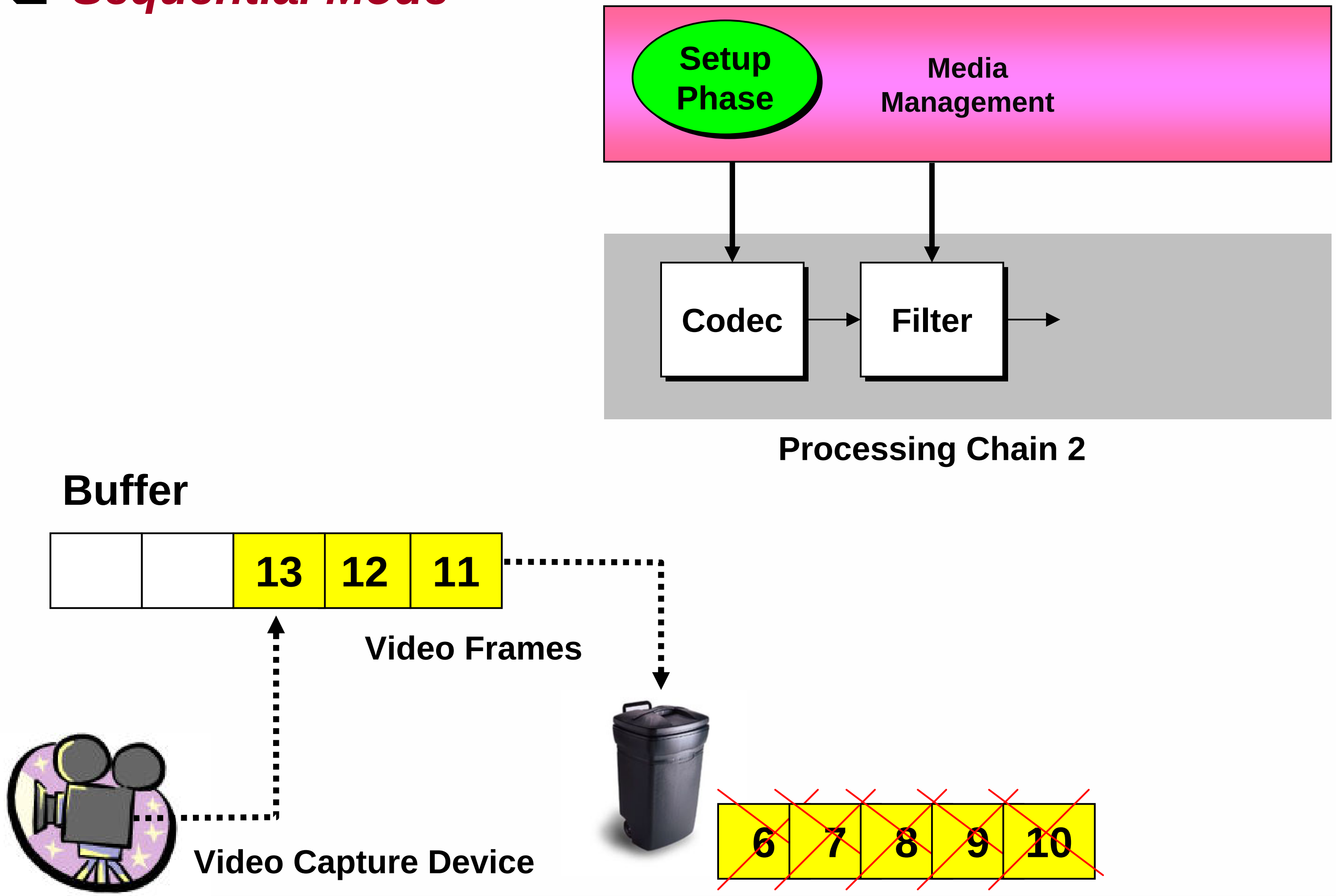
Codec Switching

- ***Sequential Mode***



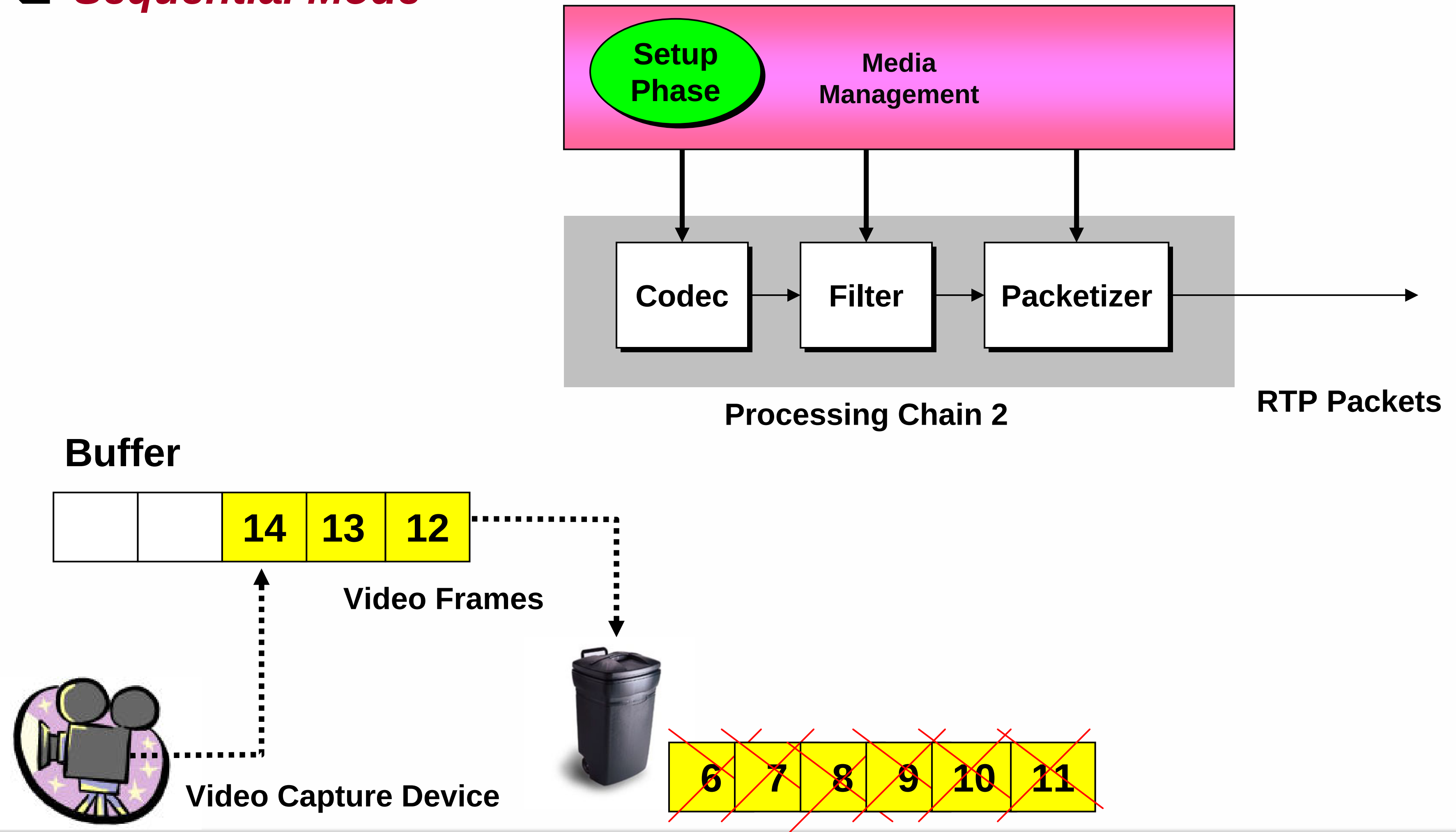
Codec Switching

- *Sequential Mode*



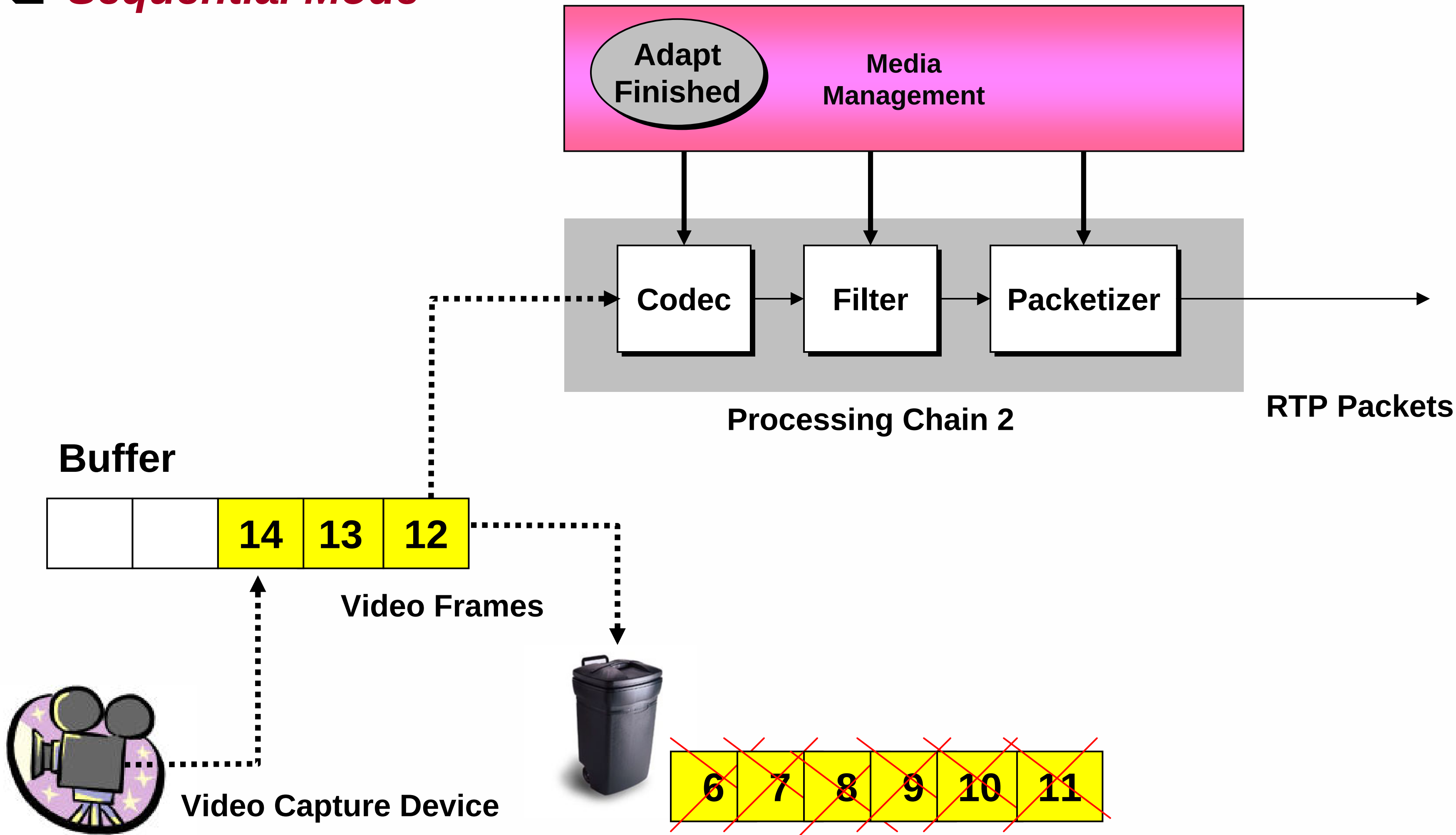
Codec Switching

- *Sequential Mode*



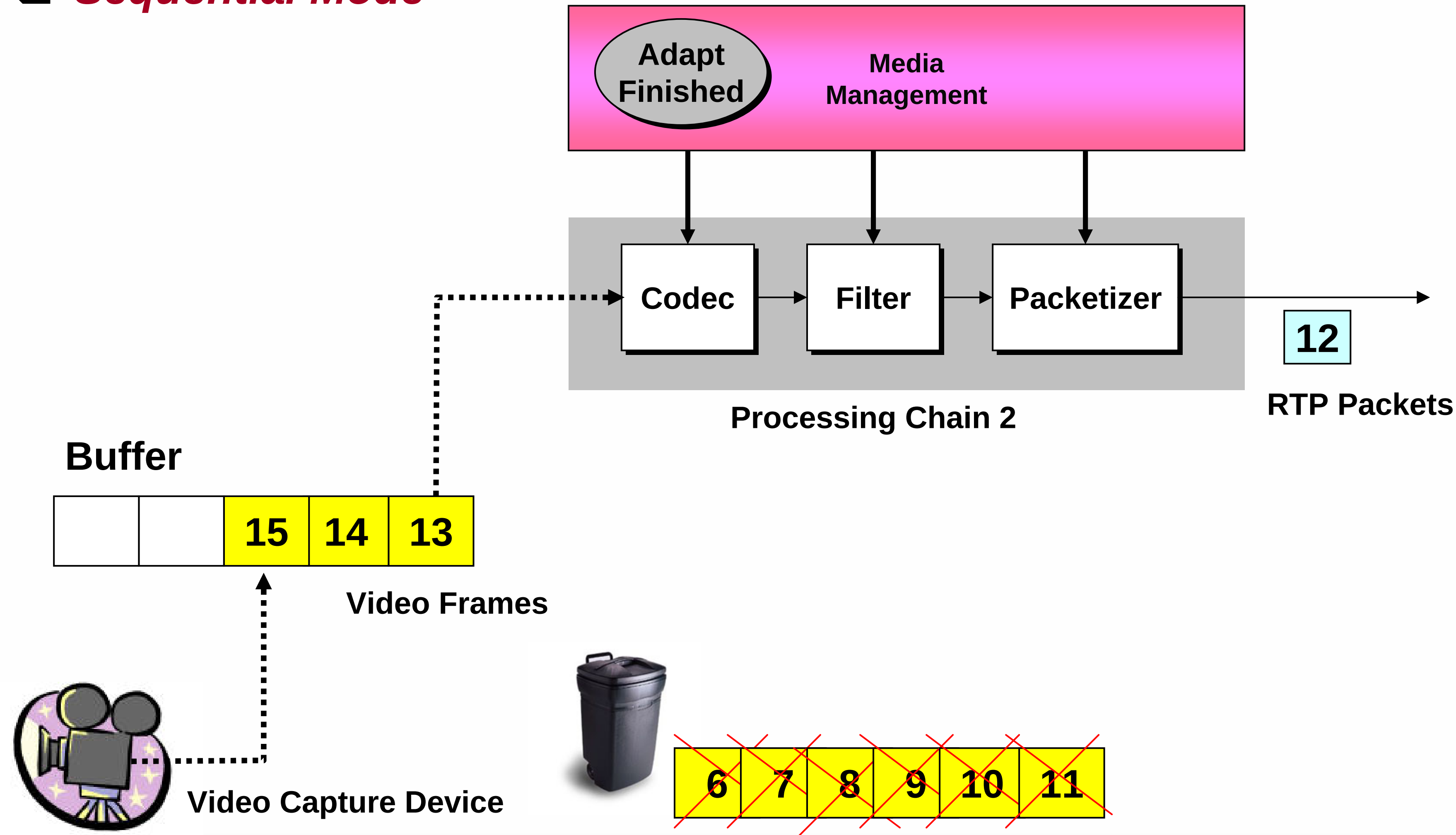
Codec Switching

- *Sequential Mode*



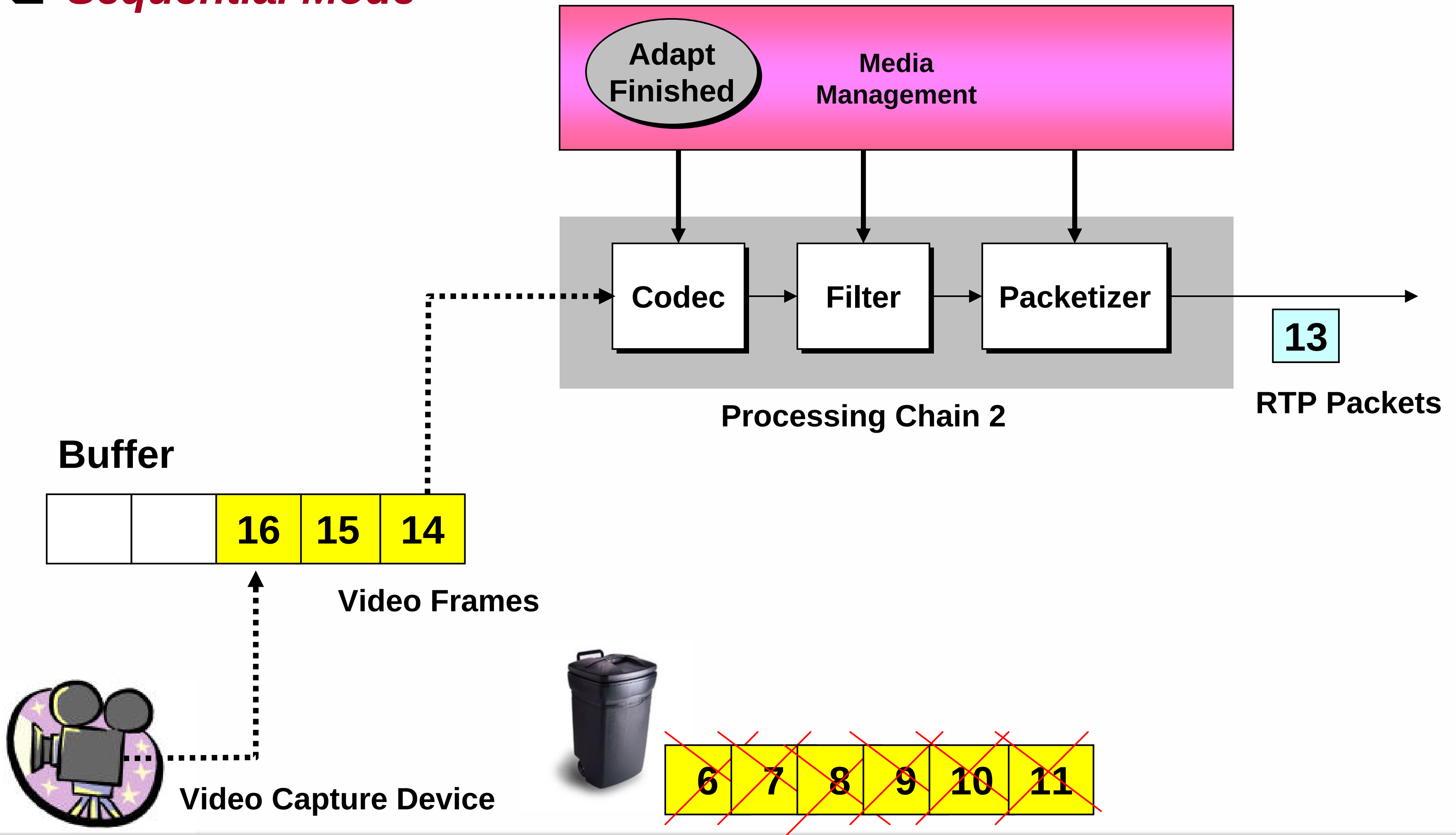
Codec Switching

- *Sequential Mode*



Codec Switching

- *Sequential Mode*



Codec Switching

- ***Sequential Mode - Conclusions***
  - Switching between processing chains usually imposes problems
  - Data loss between chain teardown and new chain setup
  - Long adaptation time interrupts media stream

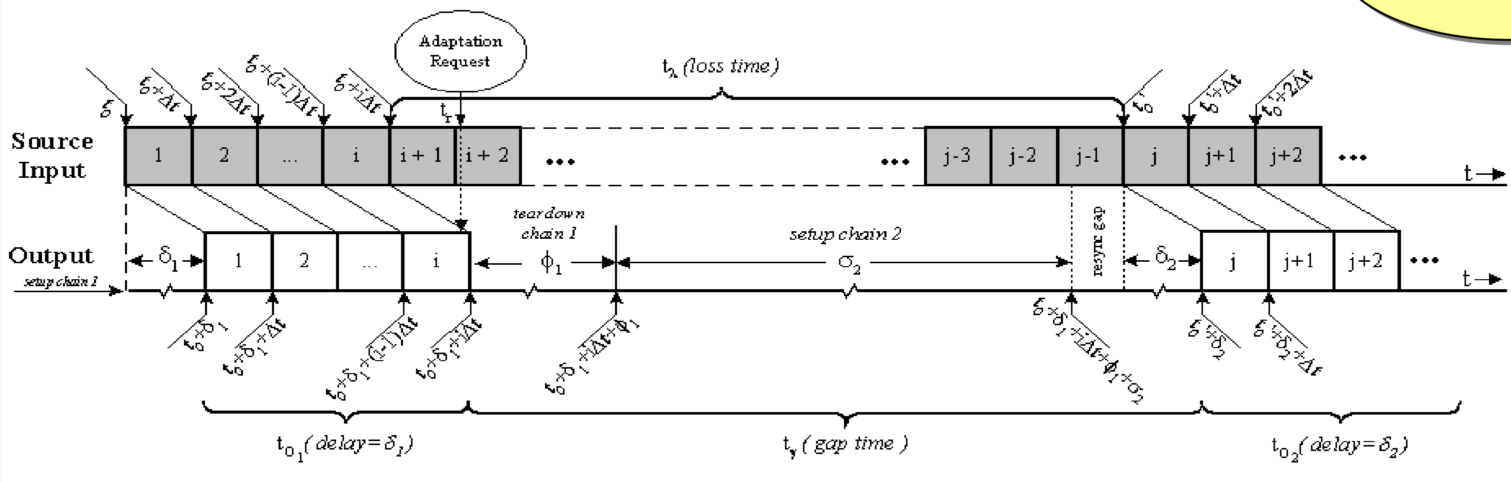
Gap Time [ms]	DVI	GSM	G.711	G.723	MP3
DVI	174.4	586.8	667.0	583.0	140.0
GSM	178.2	586.6	600.6	589.0	140.4
G.711	188.2	598.8	678.8	556.6	142.2
G.723	182.4	550.4	671.0	588.8	148.8
MP3	234.2	585.0	654.8	603.0	148.2

JMF CatureDataSource reading audio from microphone, average of 10 adaptation cycles
(Athlon Thunderbird 750MHz, 256MByte Memory, Windows 2000)

Examples

Conventional Approach

Seamless Coding Switching



Conventional media adaptation through sequential chain rebuilding

Loss Time $$t_\lambda = \left\lceil \frac{\delta_1 + \phi_1 + \sigma_2}{\Delta_t} \right\rceil \cdot \Delta_t$$

Lost Frames $$\lambda = \left\lceil \frac{\delta_1 + \phi_1 + \sigma_2}{\Delta_t} \right\rceil$$

Gap Time $$t_\gamma = t_\lambda + (\delta_2 - \delta_1)$$

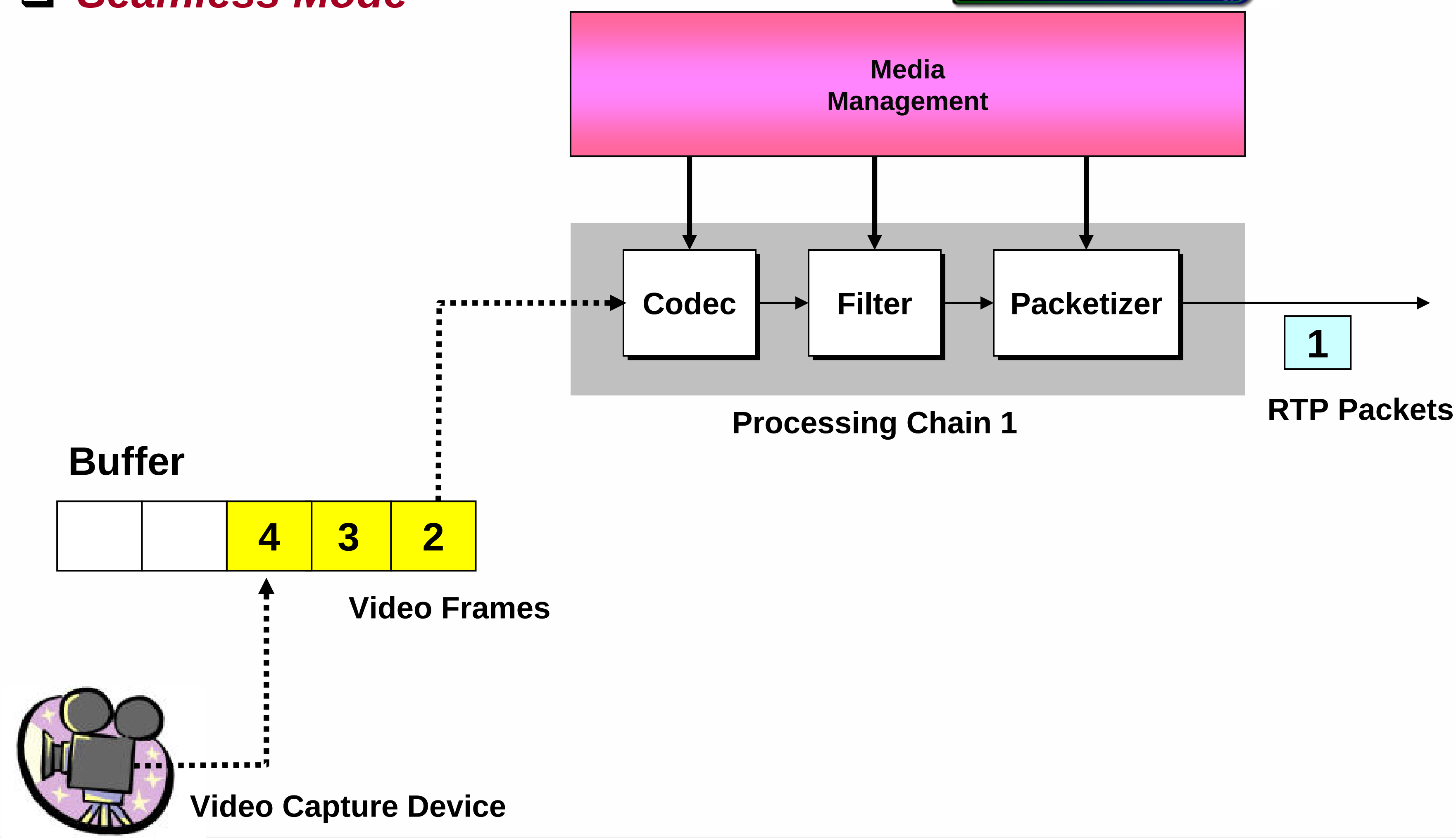
σ_2 : New Chain Set-up time

ϕ_1 : Old Chain Tear-down time

δ_1 δ_2 : Intrinsic Chain Delays

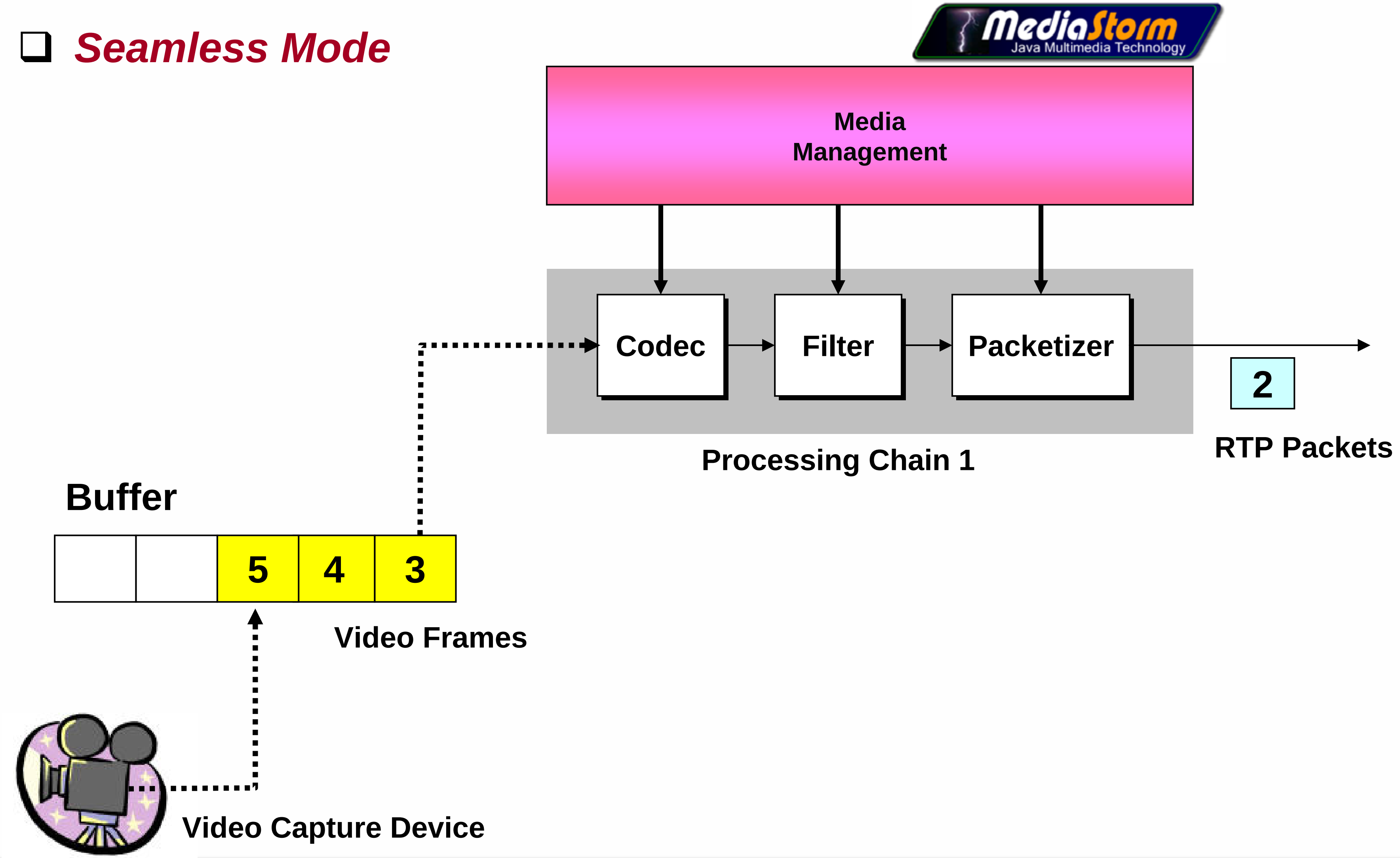
Codec Switching

- Seamless Mode



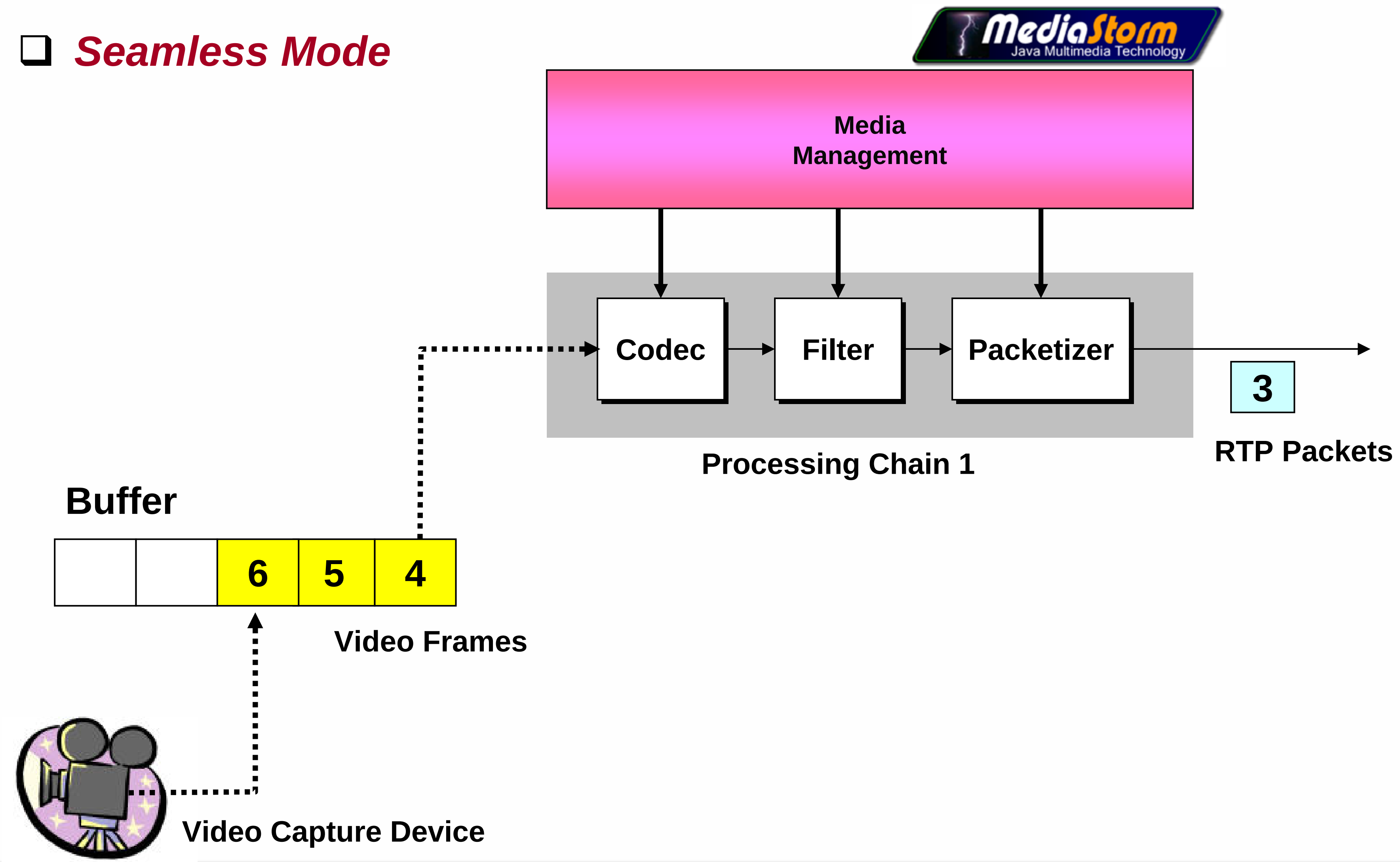
Codec Switching

- *Seamless Mode*



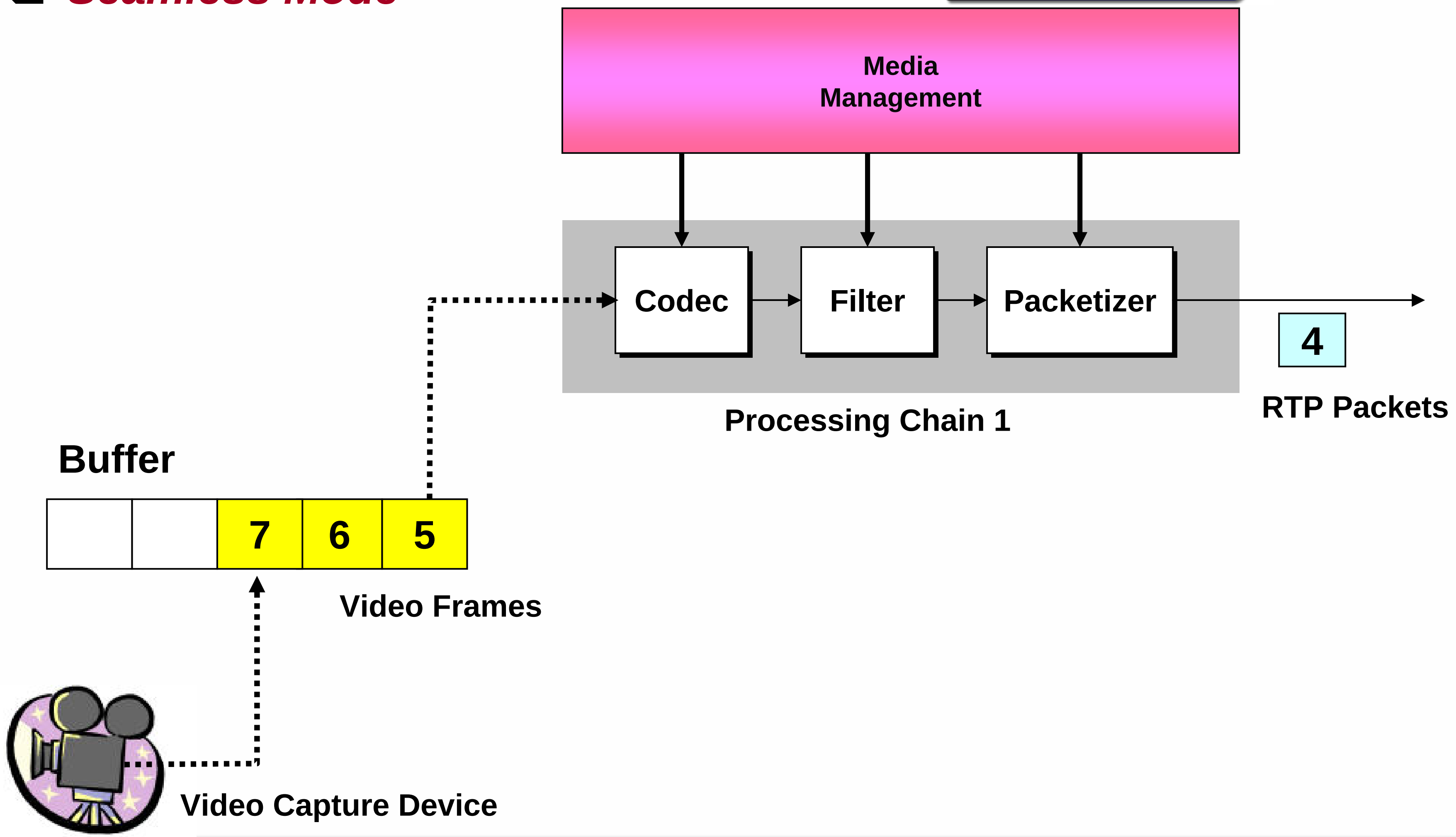
Codec Switching

- *Seamless Mode*



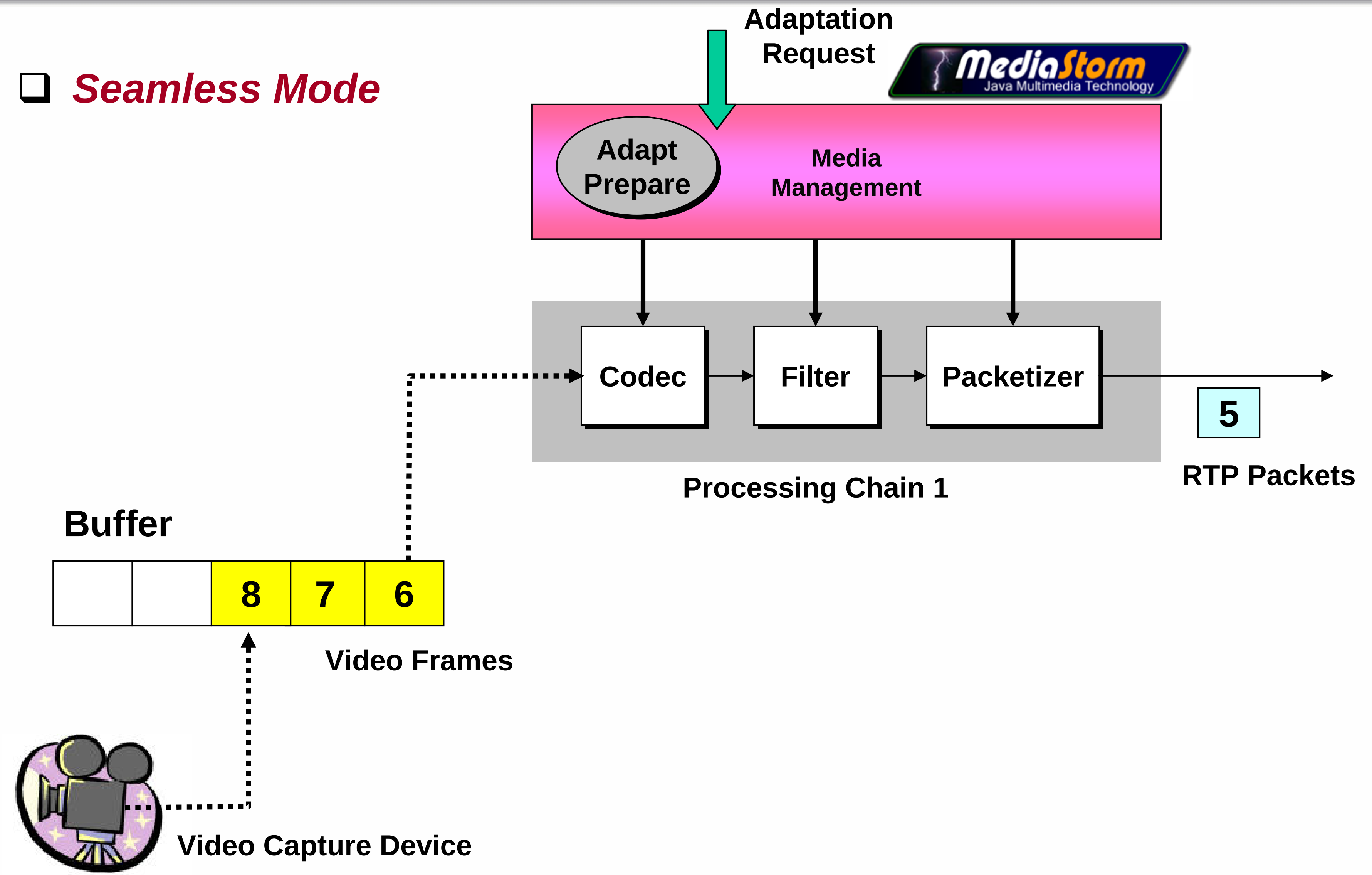
Codec Switching

- ❑ ***Seamless Mode***



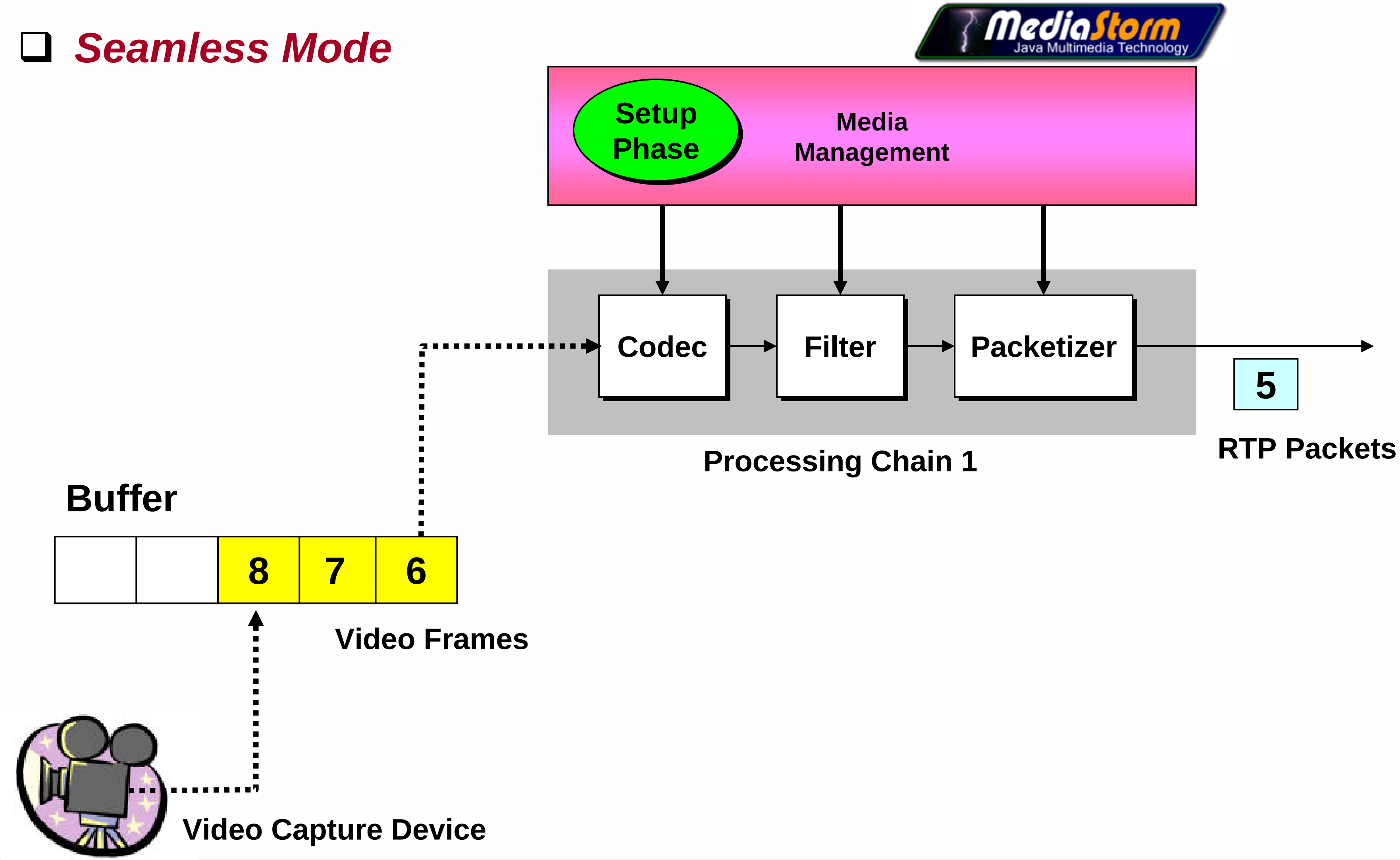
Codec Switching

- Seamless Mode



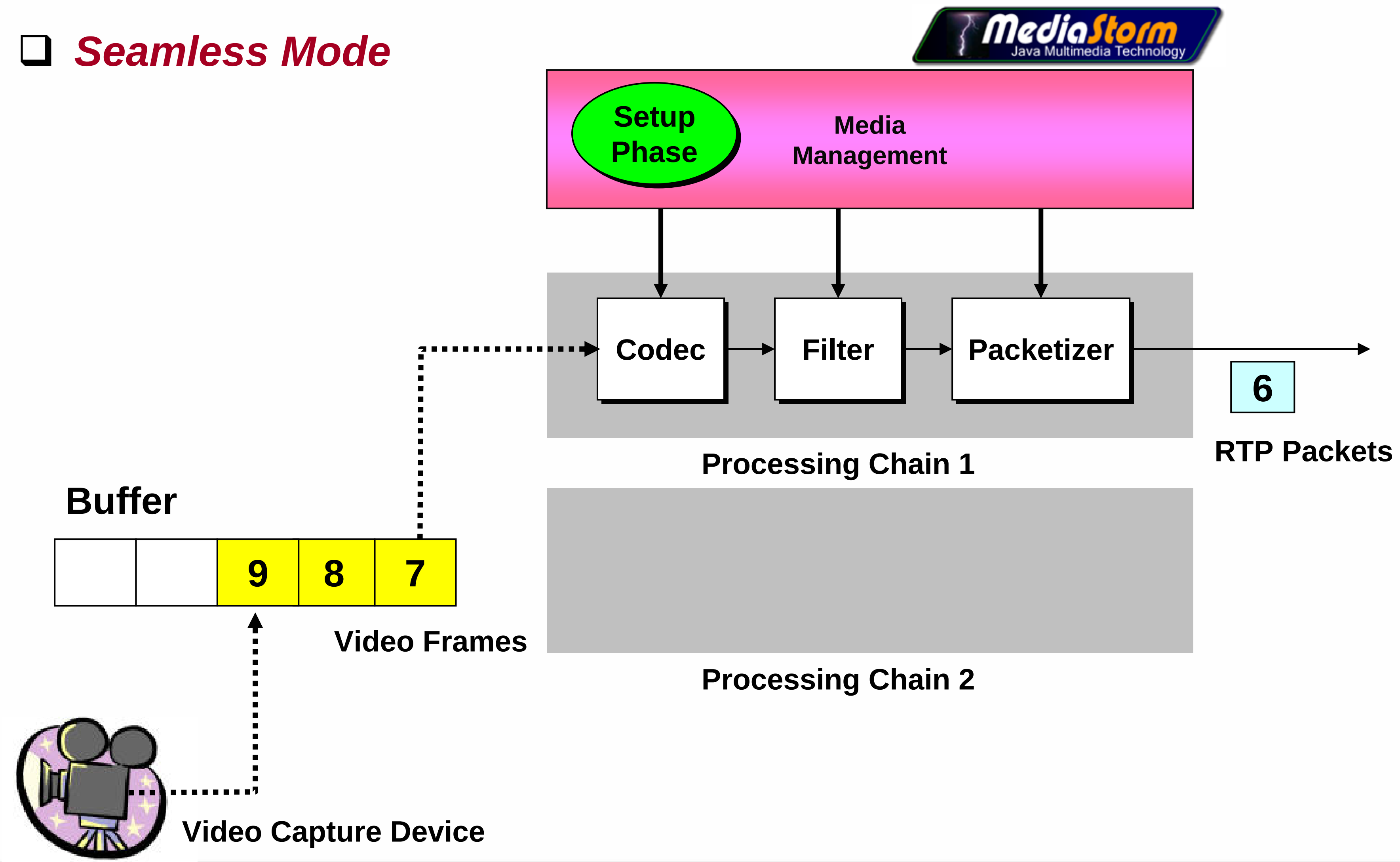
Codec Switching

- ***Seamless Mode***



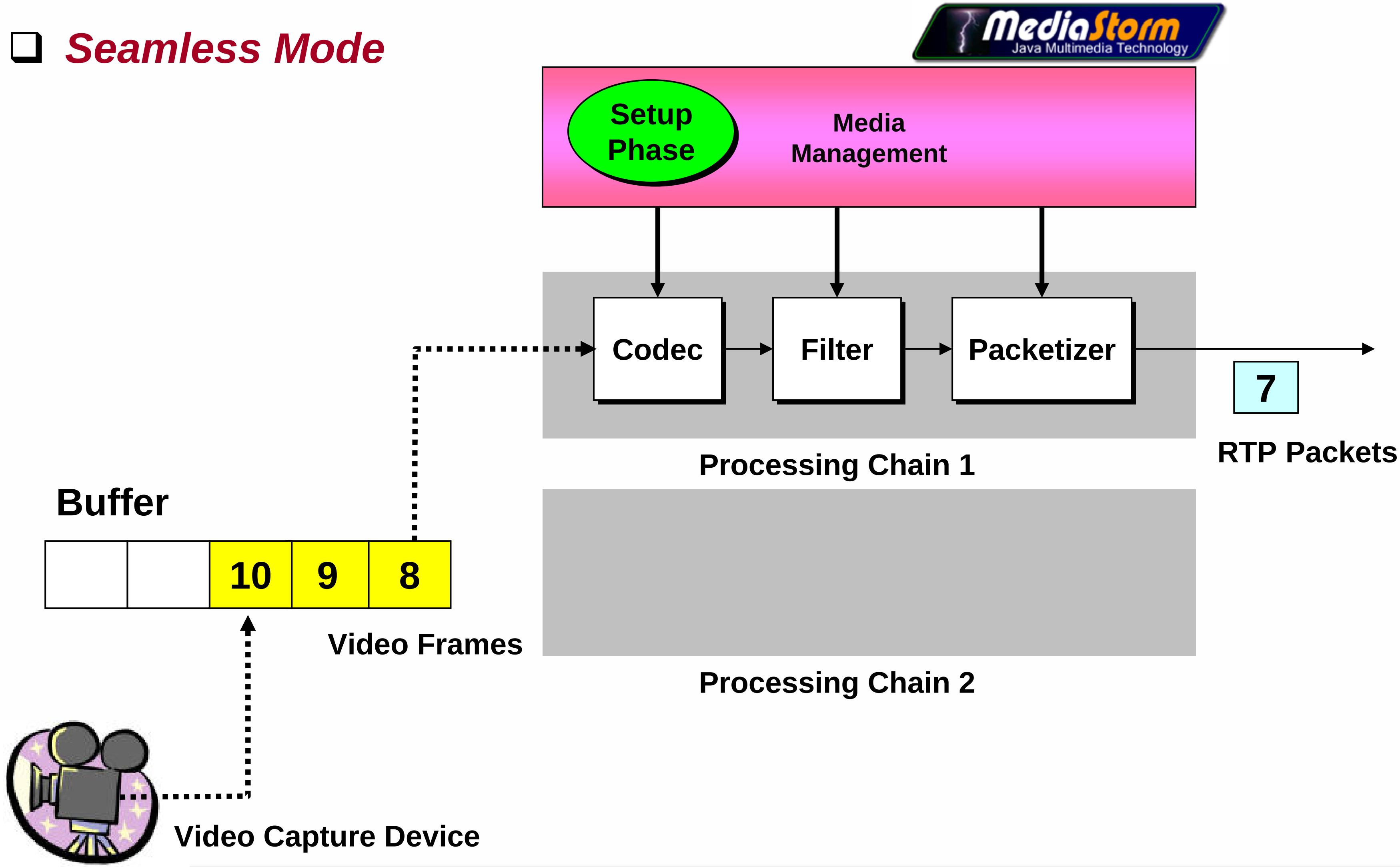
Codec Switching

- ***Seamless Mode***



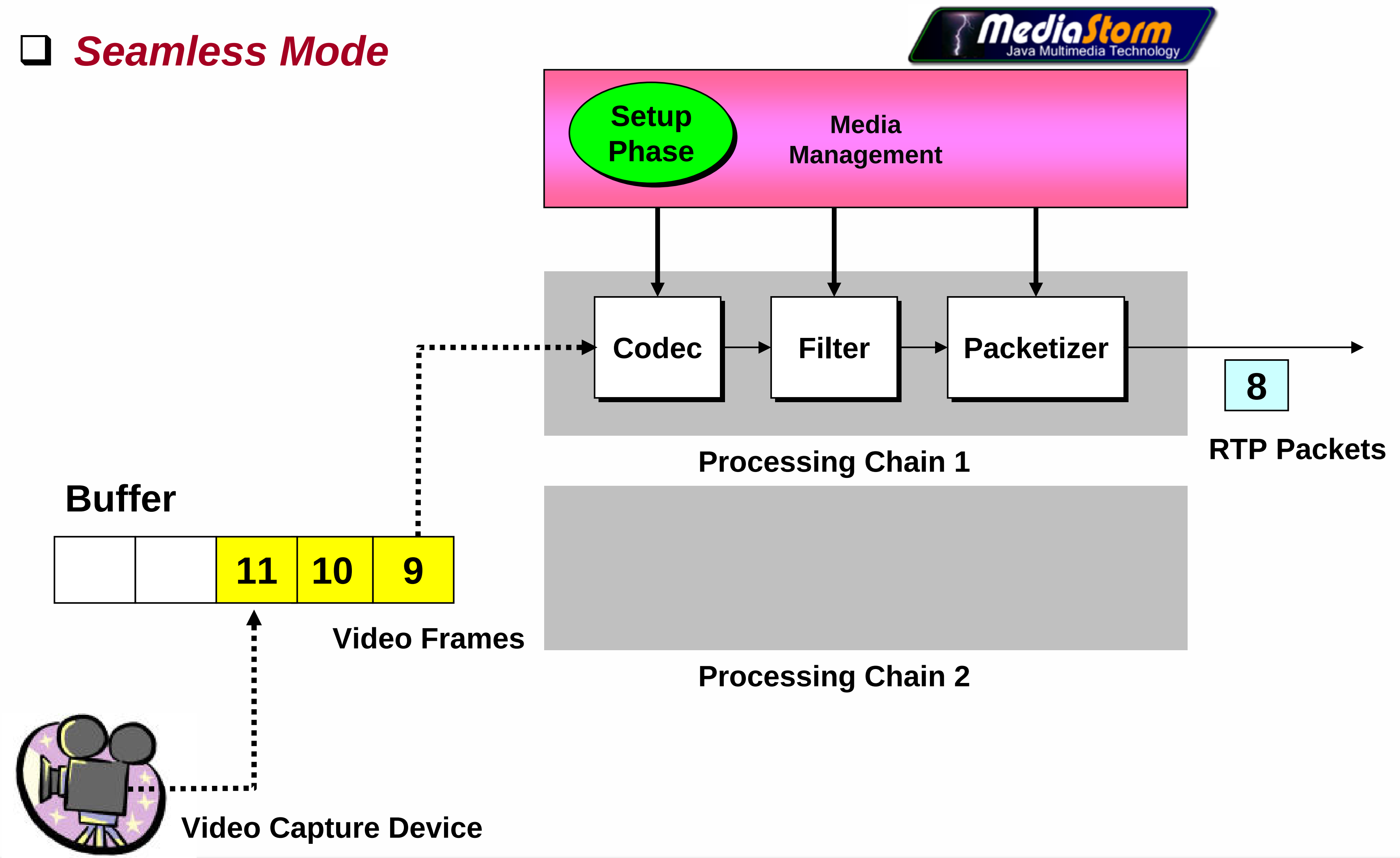
Codec Switching

- *Seamless Mode*



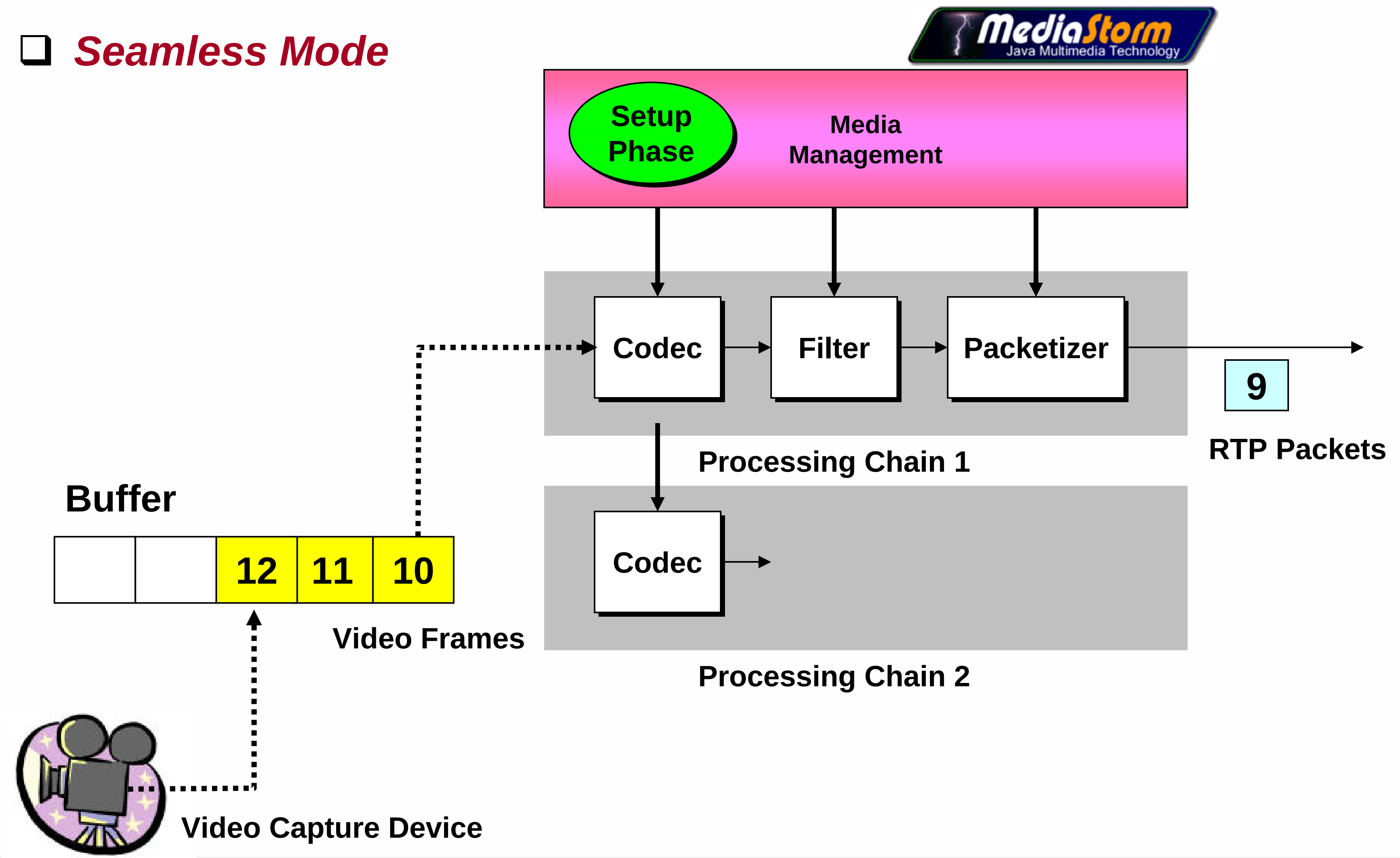
Codec Switching

- ***Seamless Mode***



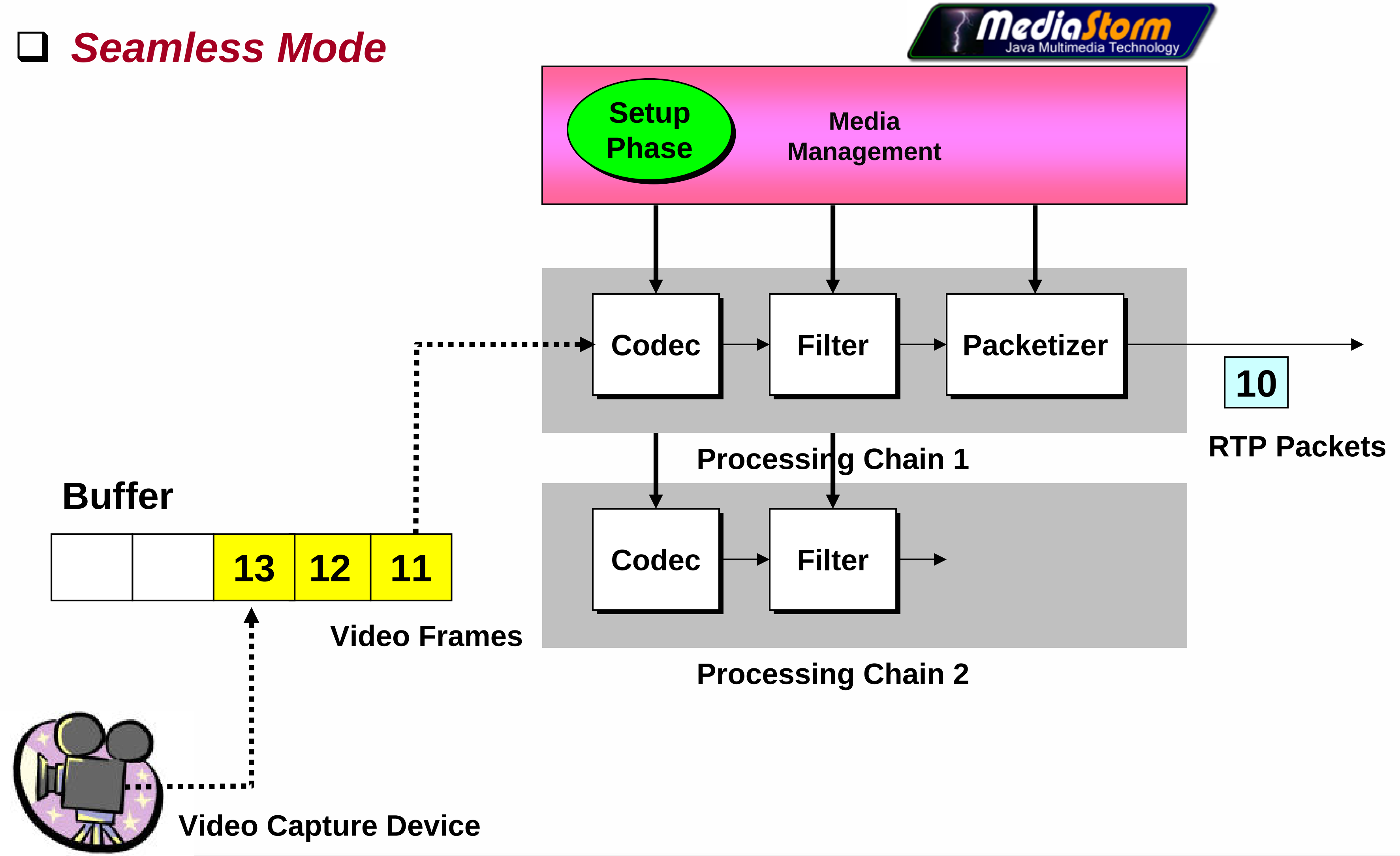
Codec Switching

- *Seamless Mode*



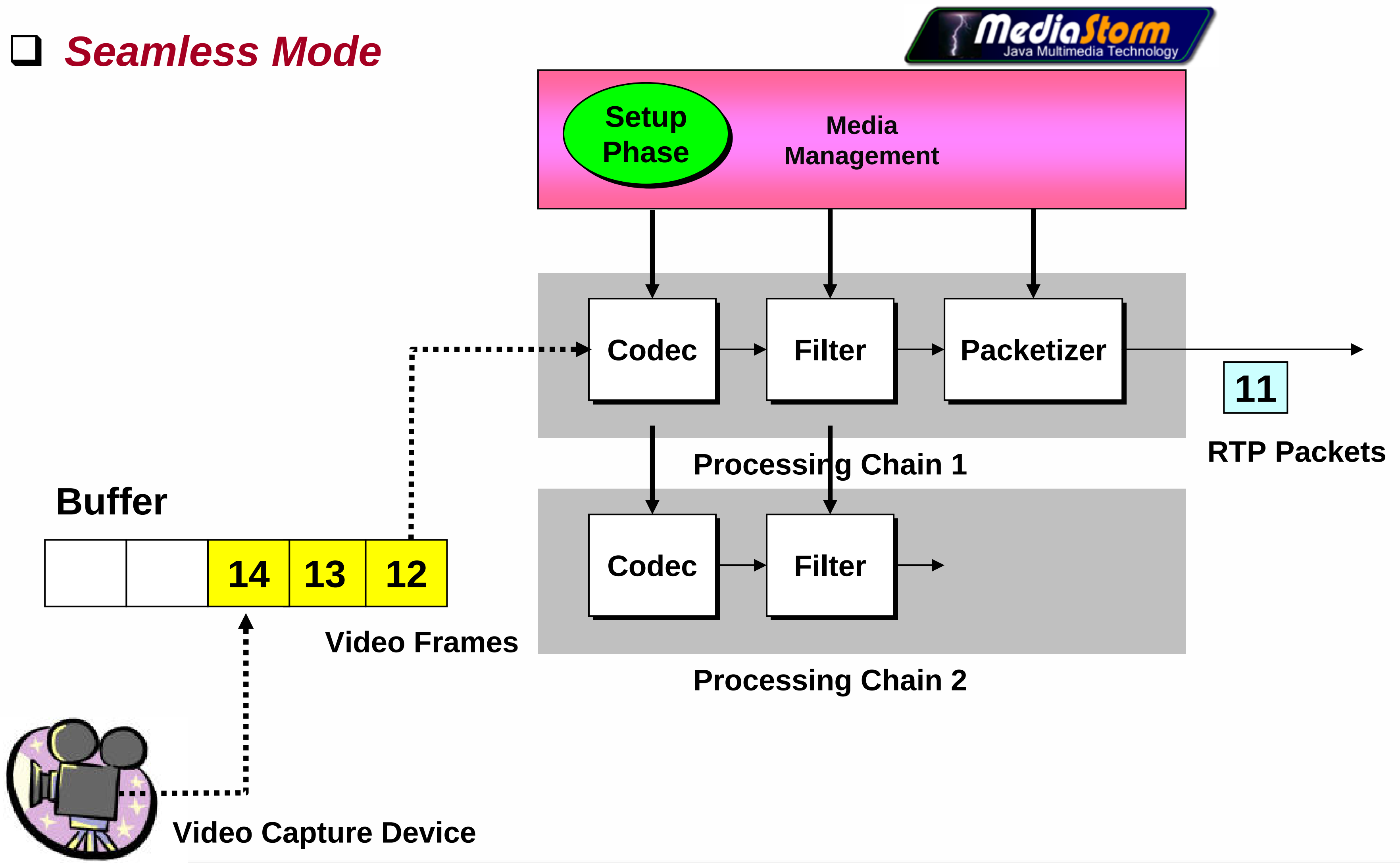
Codec Switching

- *Seamless Mode*



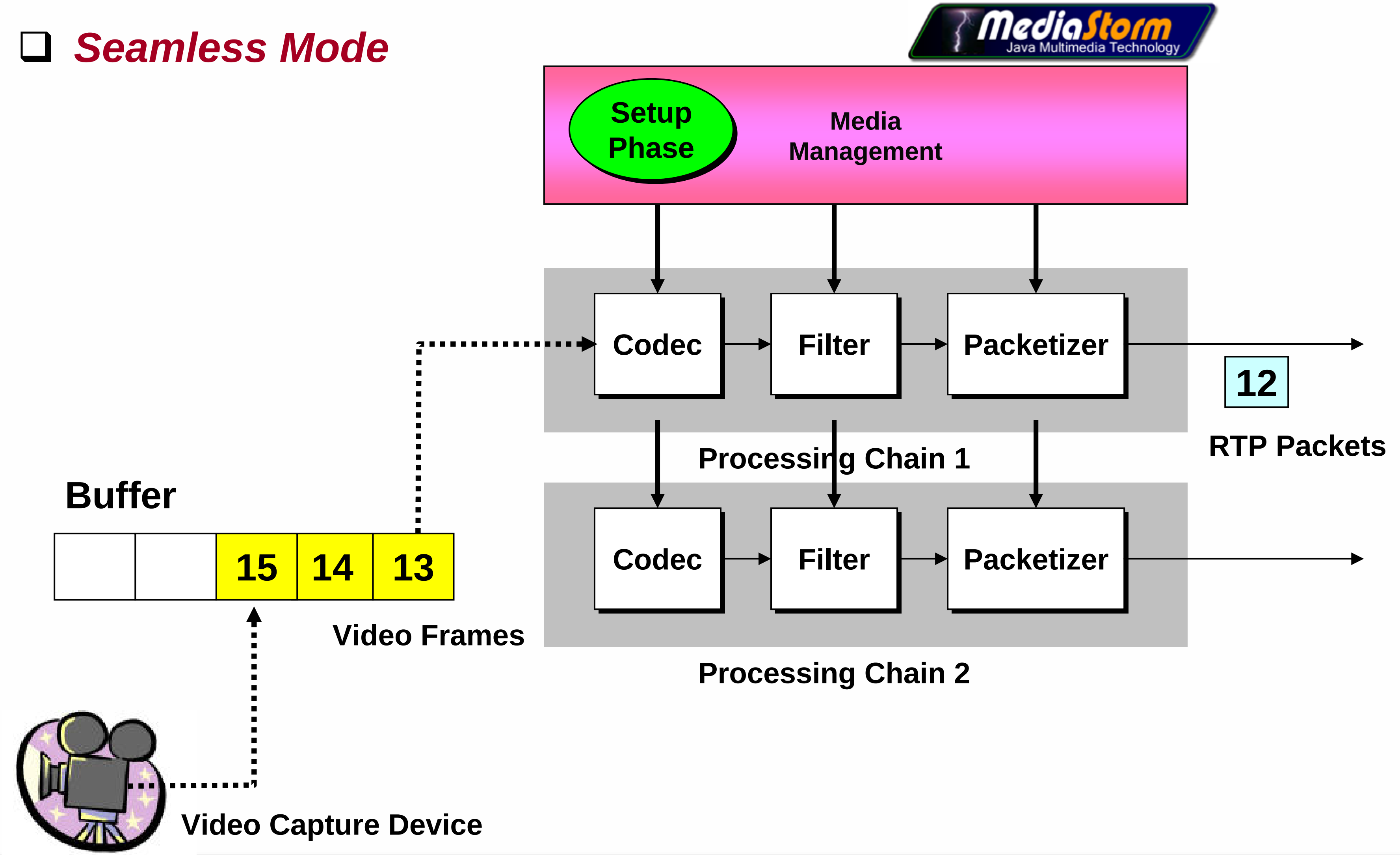
Codec Switching

- **_Seamless Mode_**



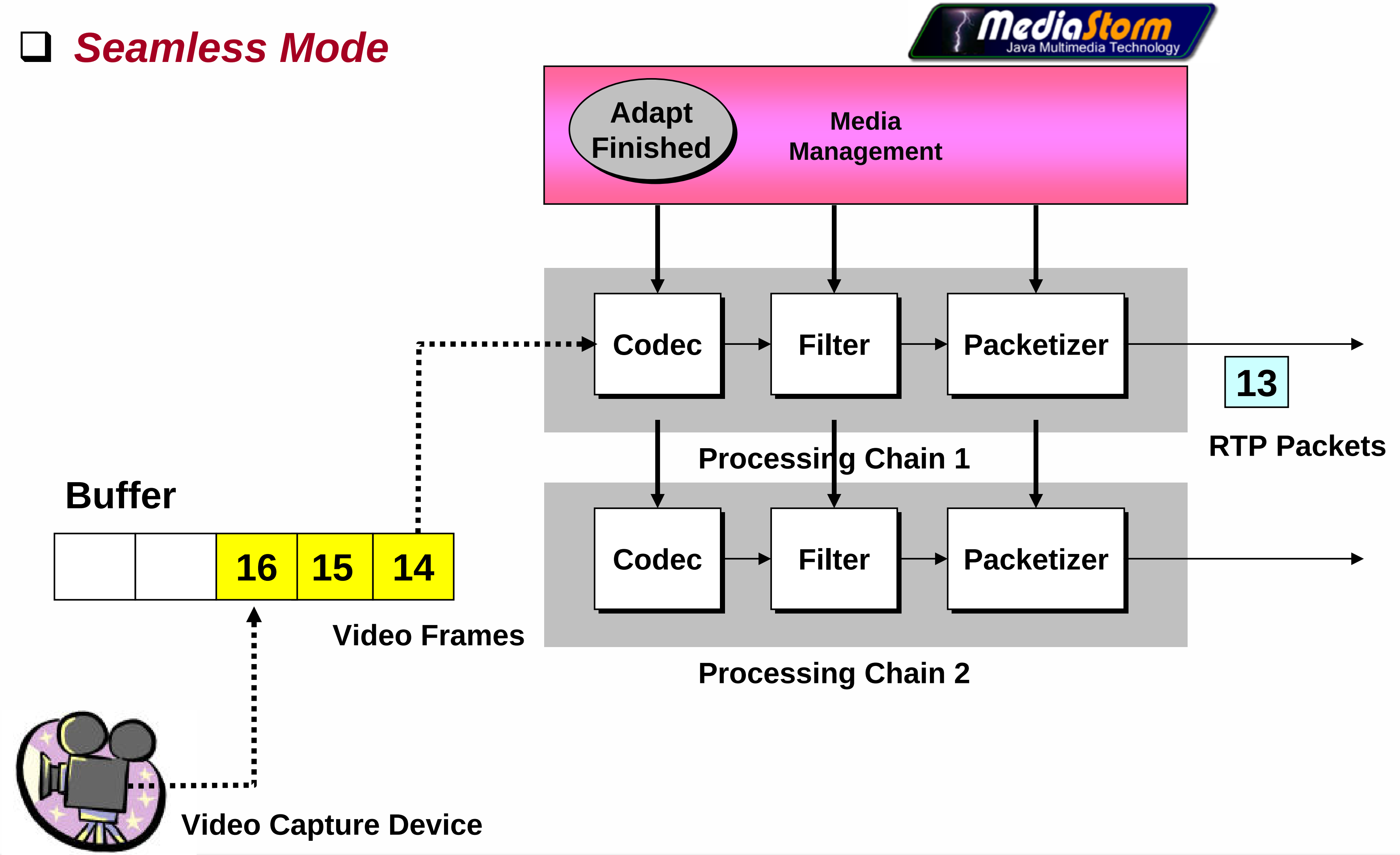
Codec Switching

- *Seamless Mode*



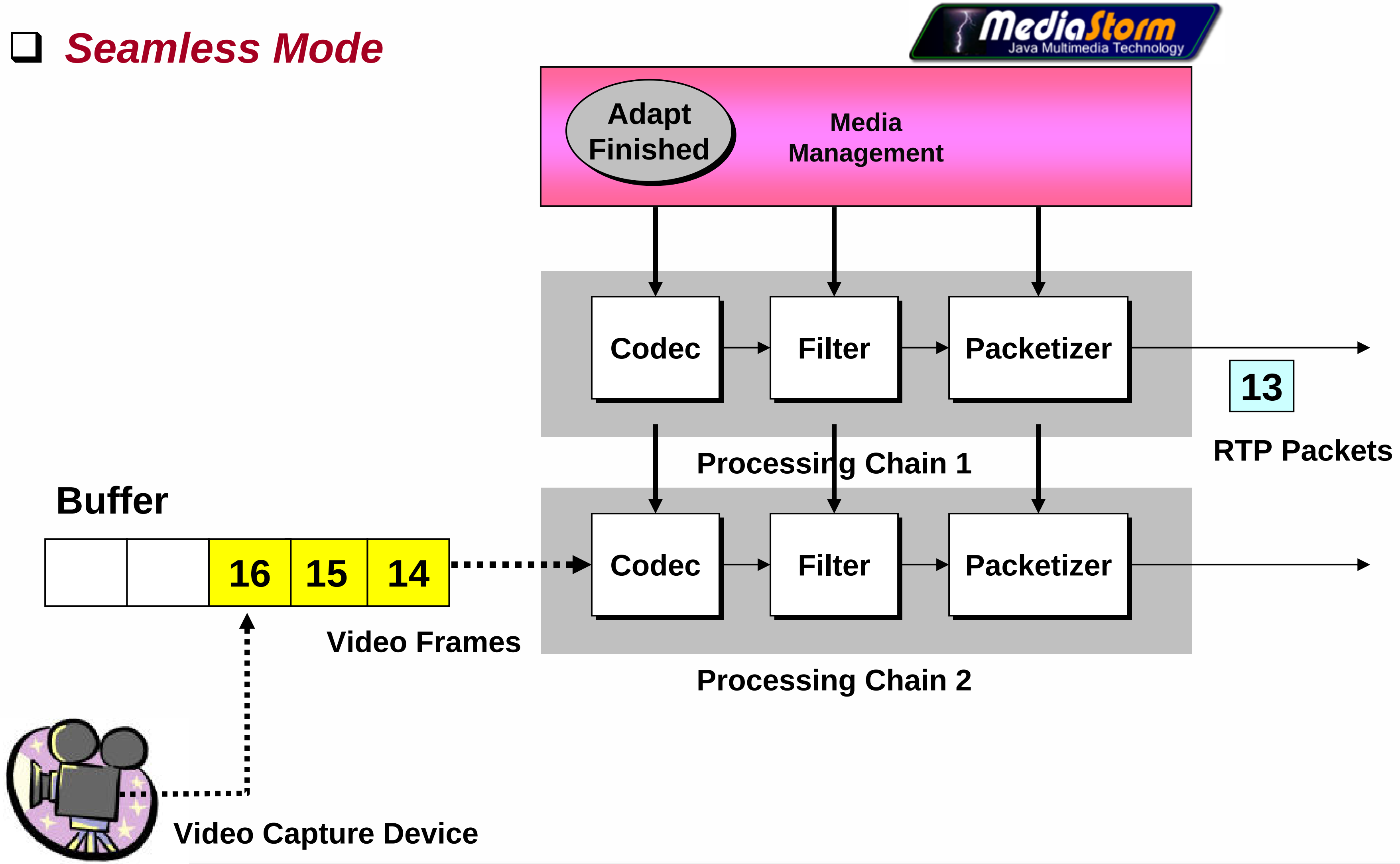
Codec Switching

- *Seamless Mode*



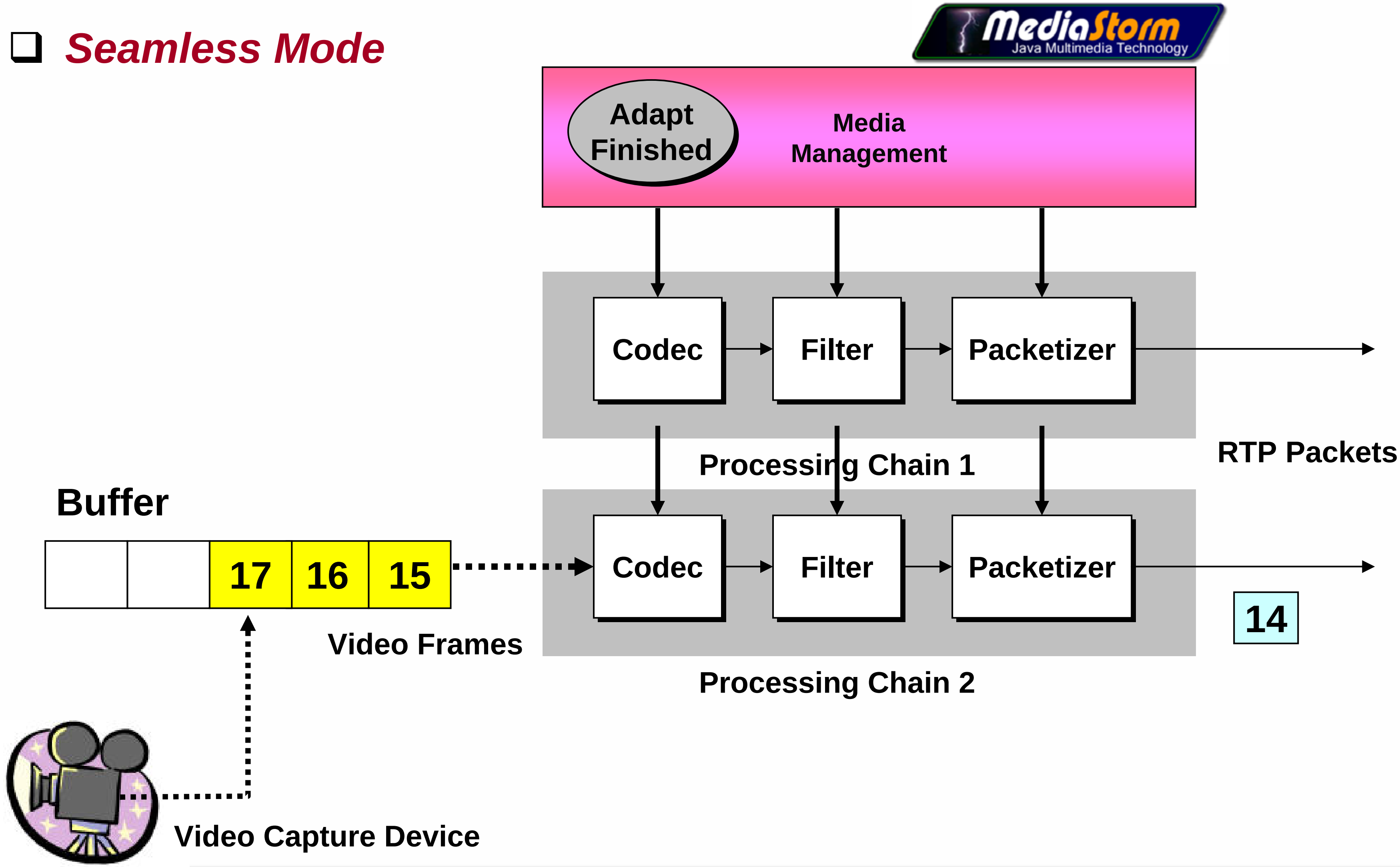
Codec Switching

- *Seamless Mode*



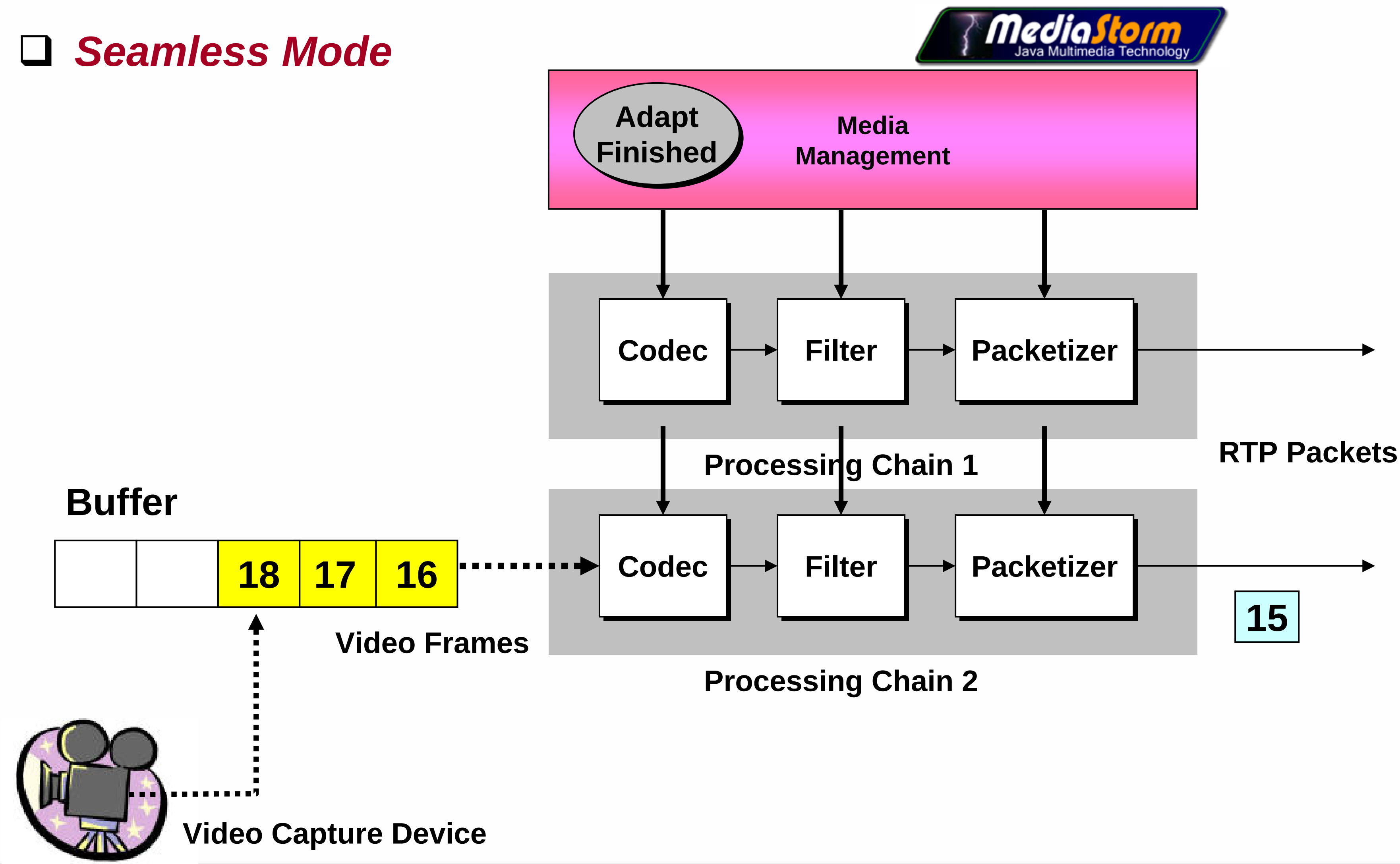
Codec Switching

- *Seamless Mode*



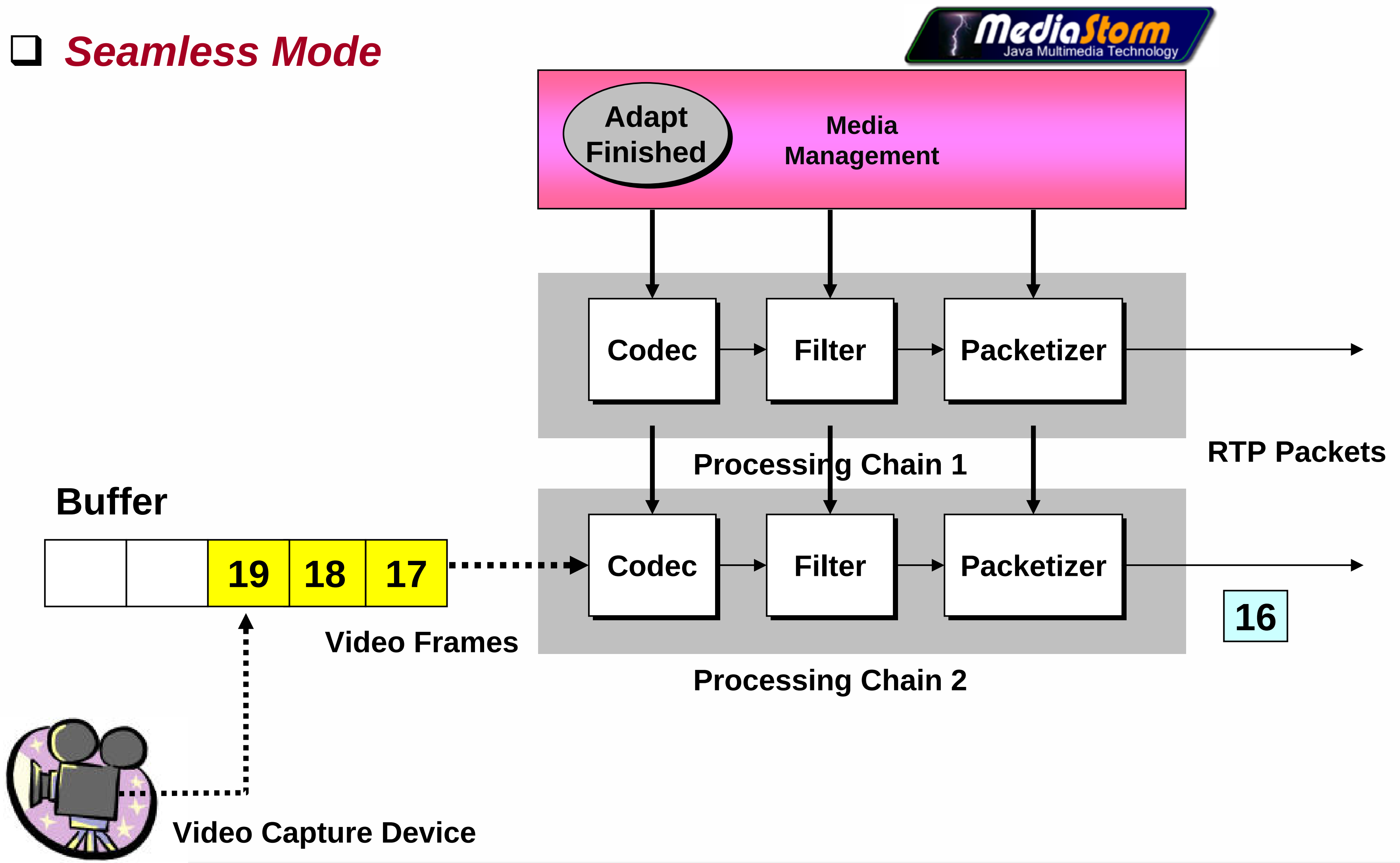
Codec Switching

- *Seamless Mode*



Codec Switching

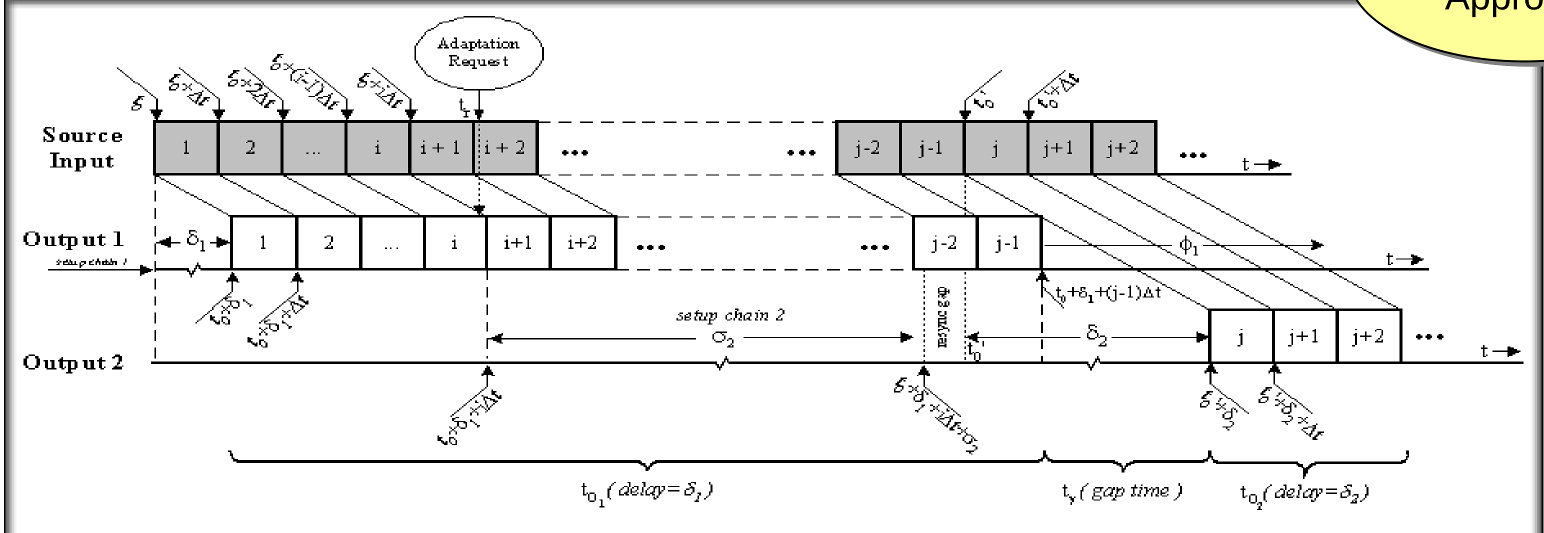
- ❑ ***Seamless Mode***



Codec Switching

- **Seamless Codec Switching**

Seamless Approach



Seamless media adaptation through parallel chain rebuilding

Loss Time	$t_\lambda = 0$
Lost Frames	$\lambda = 0$

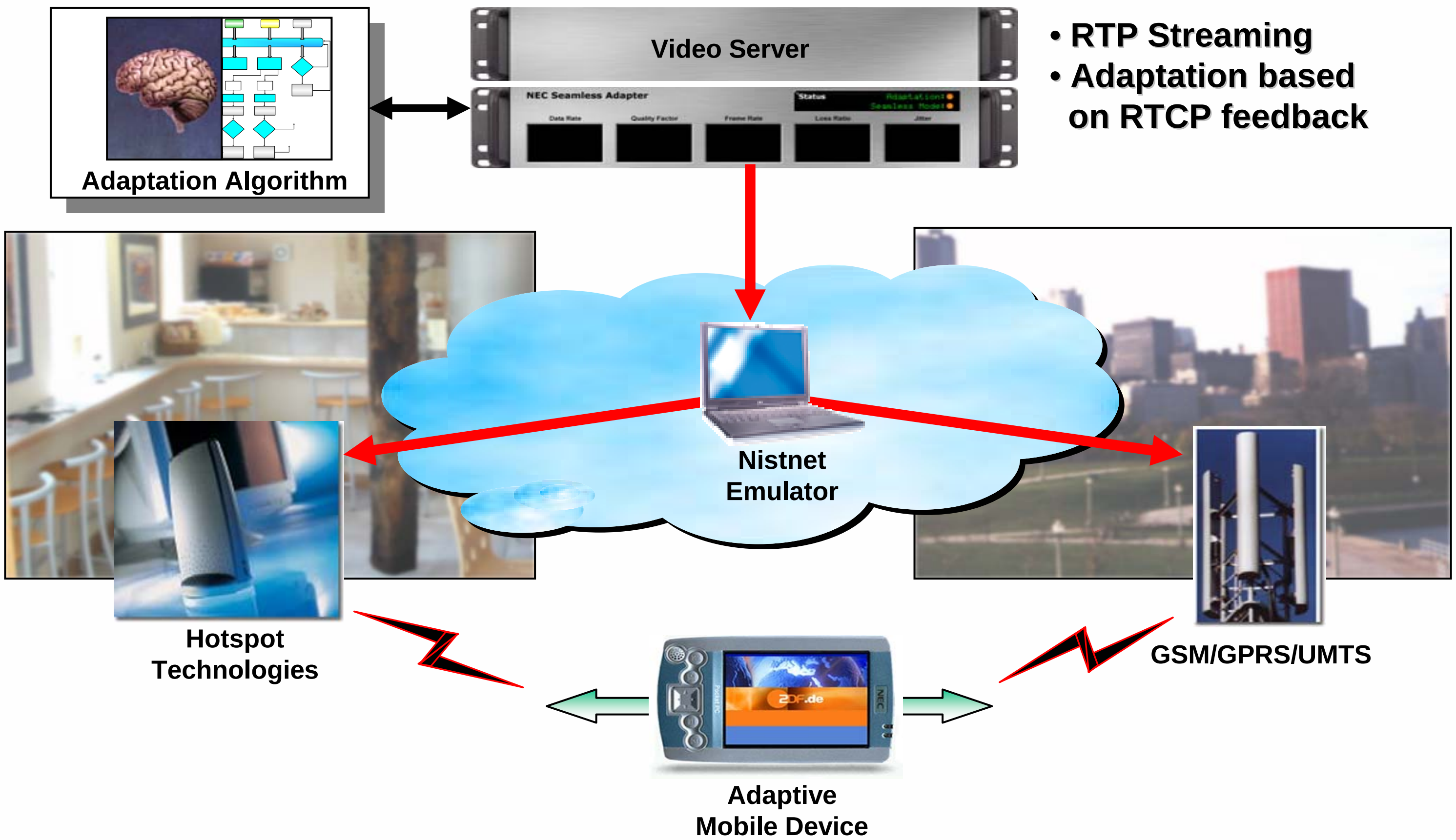
No loss at all!

Gap Time $t_\gamma = \delta_2 - \delta_1$

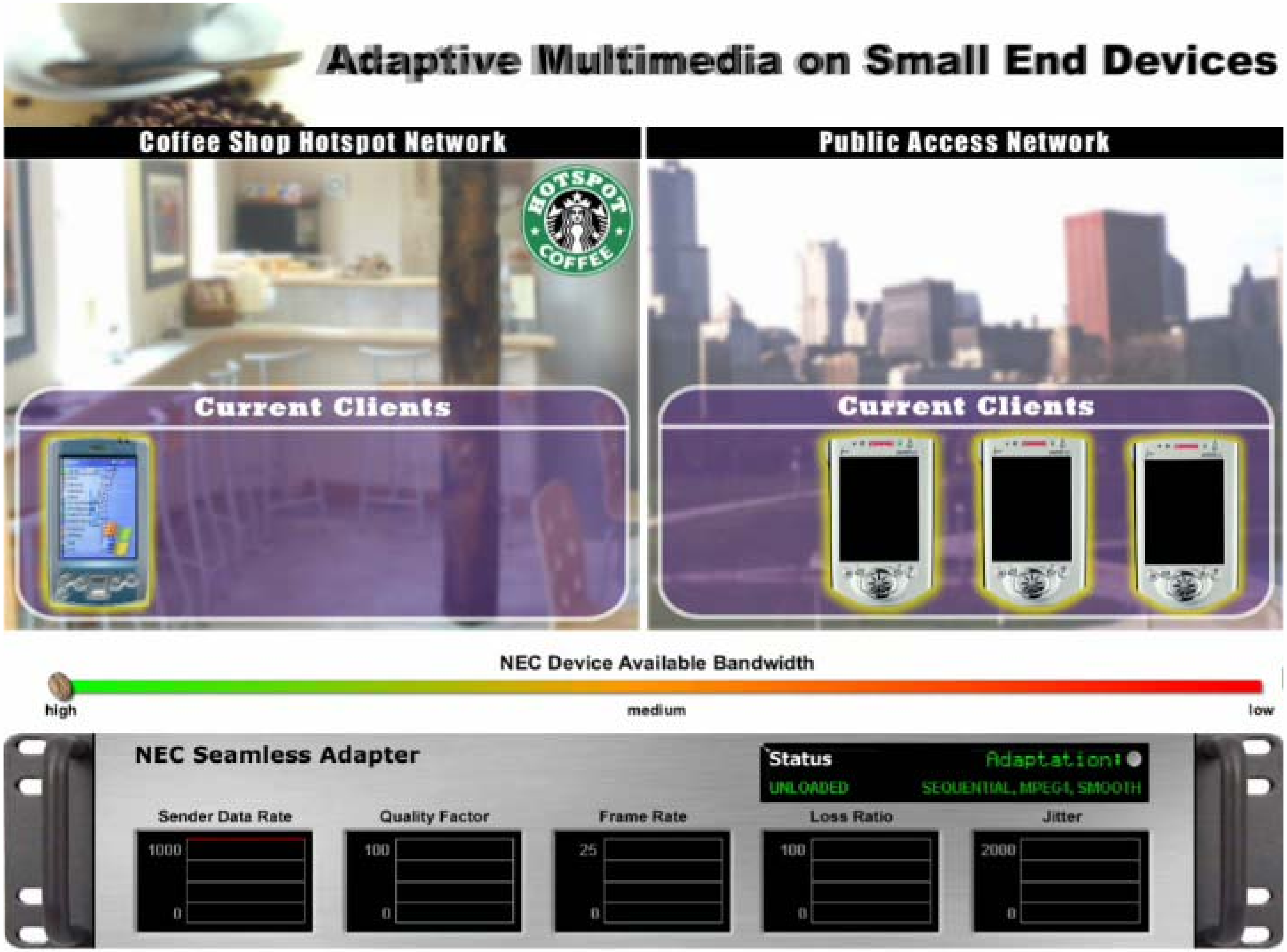
Just codec-intrinsic delay!
Overall adaptation time also reduced!

Examples

- ***Seamless Codec Switching***
  - Demo Scenario



Examples



Adaptive Multimedia on Small End Devices
Coffee Shop Hotspot Network
Public Access Network
HOTSPOT
COFFEE
Current Clients
Current Clients
NEC Device Available Bandwidth
high
medium
low
NEC Seamless Adapter
Status
Adaptation:
UNLOADED
SEQUENTIAL, MPEG4, SMOOTH
Sender Data Rate
Quality Factor
Frame Rate
Loss Ratio
Jitter
1000
0
100
0
25
0
100
0
2000
0