MULTIMEDIA NETWORKED APPLICATIONS: STANDARDS, PROTOCOLS AND RESEARCH TRENDS

Maria Teresa Andrade
FEUP / INESC Porto
mandrade@fe.up.pt ; maria.andrade@inescporto.pt
http://www.fe.up.pt/~mandrade/ ; http://www.inescporto.pt

Presentation walkthrough

- Introduction:
  - current situation, types of services, alternatives for implementing those services, relevant characteristics, challenges
- Multimedia traffic
  - digital content sources, principles of media compression, standards (JPEG, MPEG, H264/AVC)
- IETF protocols
  - signaling (RTSP, SDP, SIP, SAP), transport (TCP, UDP, RTP +RTCP), QoS assurances (RSVP, DiffServ)
- Case study application: IPTV
- Research trends
  - context-aware content adaptation: adaptation alternatives, adaptation decision approaches, types of context, supportive standards (MPEG-21, W3C-CC/PP)

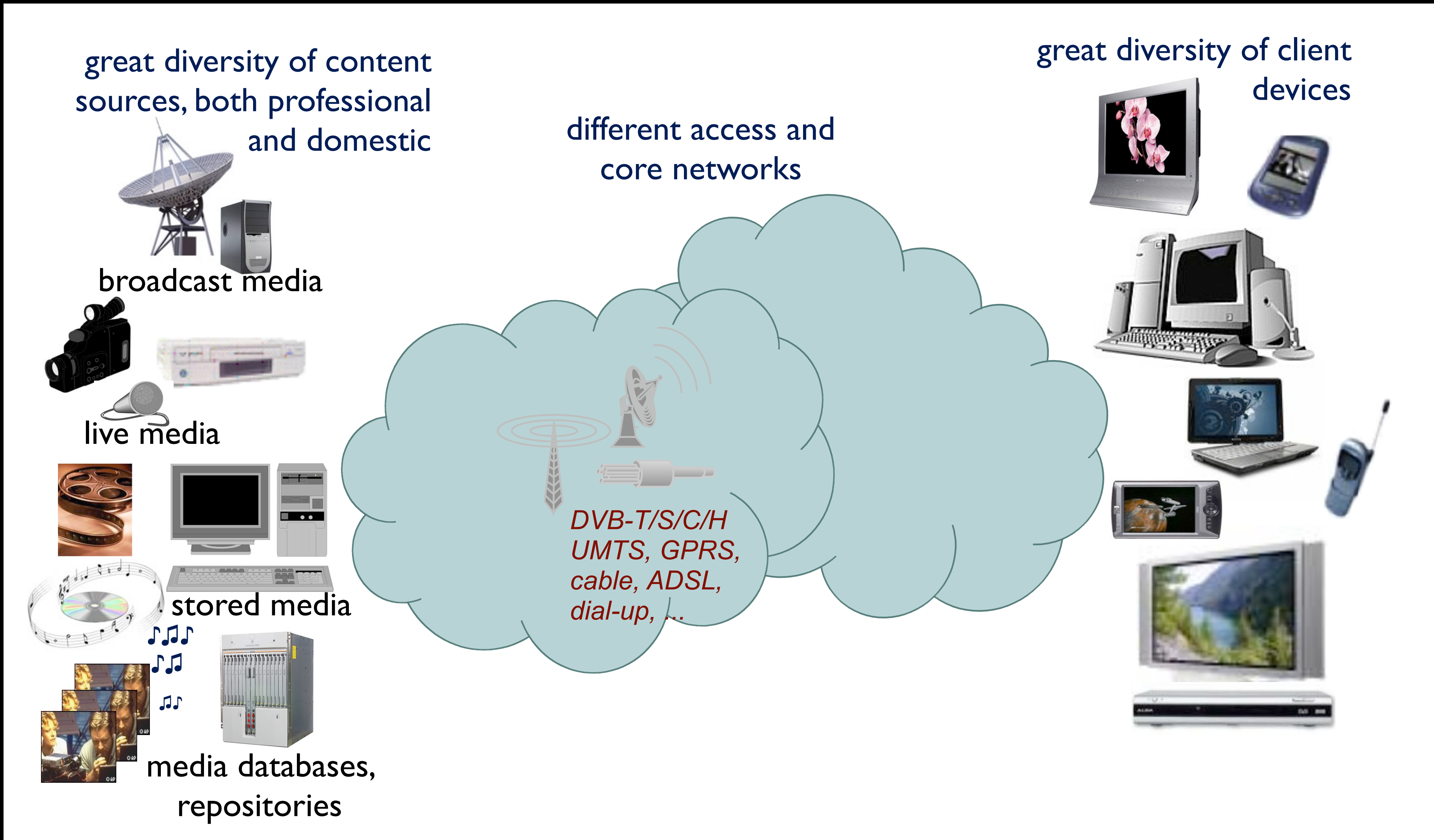
Introduction

- Introduction:
  - current situation
  - types of services
  - alternatives for implementing those services
  - relevant characteristics
  - challenges faced

Current scenario

- Continuous progress of technology
  - great number of consumer multimedia-enabled devices
    - though with different capabilities to process and present multimedia content → heterogeneity!
  - Convergence of media and roles
    - content producers make available their contents through multiple platforms
    - consumers are also producers (“prosumers”)
  - Convergence of networks
    - all-IP distribution

Current landscape



Consumer expectations

- Mobility
- Personalization
- More quality and more choice
- Interoperability and transparency
- better search results
- access anywhere, from any terminal and at anytime the desired content

↓

Universal Multimedia Access (UMA)

Existing services

- Entertainment / home
  - media clips download and/or streaming (YouTube, ...)
  - mp3 file sharing (iTunes, BearShare, LimeWire, ...)
  - images/photos sharing (Flickr, ...)
  - Internet radio, Internet TV
  - on-line newspapers (OhmyNews, Google News, ...)
  - blogs, RSS feeds
  - IPTV, VoD
- Businesses, education
  - file sharing
  - live seminars, videoconferencing (Mbone, ...)
  - streaming of educational content (classes, additional video material)
  - VoIP
  - collaborative work

Implementation alternatives

- unicast, broadcast, multicast
- unicast or point-to-point
  - one-to-one communication, ex., videotelephony, media streaming on the Internet, VoD
  - requires a copy of the content for each client
  - can be customized to the client needs, preferences or context restrictions
  - usually requires the existence of a return channel between the client and the source

Implementation alternatives

- broadcast
  - communication one-to-many (goal "one-to-all")
  - most well-known example, regular TV broadcasting
  - efficient way to deliver popular content to a large audience simultaneously
  - a broadcast system must be designed for each customer in the same way regardless of their requirements, terminal devices or connection conditions
    - the system must be dimensioned for the worst case if all customers are to be served
    - the system has limited possibilities to adapt to changing usage conditions
  - given the great number of terminal devices, usually it is not practical to establish feedback channels for every consumer

Implementation alternatives

- multicast
  - one-to-many communication but not one-to-all as in broadcast
    - receiving terminals must take te action to connect to a multicast IP address
    - in broadcast the signal arrives in fact to every terminal, regardless on whether the terminal is connected or not
  - it is more efficient than unicast to reach simultaneously several terminals
    - rather than having to establish one end-to-end connection between each consumer and the source, replicating the number of streams as much as the number of connected users
    - it offers the same advantages as broadcast
  - however, current IP networks have limited support for IP multicast
    - other aproaches are being developed, working at the application level - overlay multicast

Aspects to consider

- real-time
  - real-time coding versus stored, pre-coded content
  - real-time transmission and presentation
- interactive applications versus non-interactive
- static channels versus dynamic
- compression schemes
  - constant bit rate (CBR) or Variable bit rate (VBR)
  - features/capabilities of compression schemes
- download versus streaming
- required level of quality of service (QoS)
- content/applications accessible (on-line) and easily found

Aspects to consider: real-time

- interactive applications (videoconference, VoIP, etc) and live events (sports, seminar, keynote speech, etc) require real-time encoding and transmission
  - if data arrives too late, it no longer can be used
  - necessary to impose a limit to the end-to-end latency (maximum delay of coding+transmission+decoding+presentation) - typically 150 ms
- Video on Demand or video-clips streaming can be pre-encoded and stored but require real-time transmission
  - pre-coding can be performed in non-real time, enabling the use of less expensive coders
  - it also allows multiple-step encoding, resulting in better compromise between quality and bit rate (current practice in DVD)
  - delay can be absorbed using a receiving buffer

Aspects to consider: channels

- static versus dynamic channels
  - static channels offer a fixed bandwidth and usually fixed delays (small jitter) and small loss rate
    - examples include ISDN (px64 kbit/s), cable or DVD
  - dynamic channels present a higher degree of variability and usually higher bit and loss rate
    - pose more challenges to the implementation of high-quality multimedia services
    - examples include radio channels of wireless networks (such as UMTS) but also wired best-effort networks (such as Ethernet, shared media)

Aspects to consider: compression

- different compression schemes provide different results in terms of compromise quality-bit rate-complexity
- some are more popular and used than others

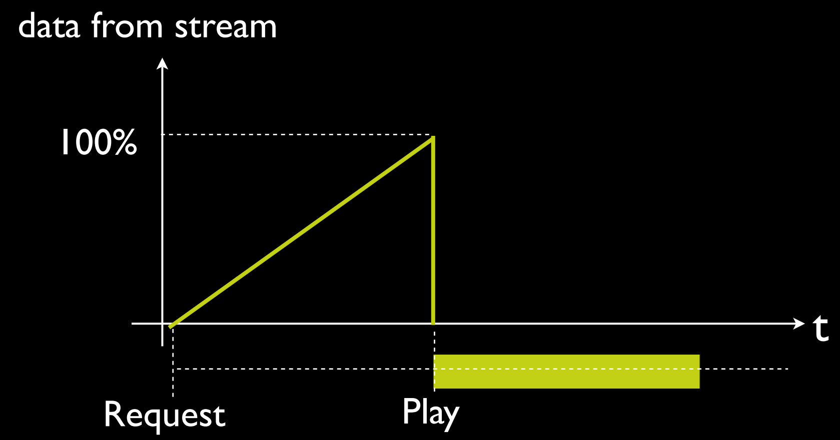
scheme	area of application	bit rate
H.261	videoconference	piecewise CBR, p*64 Kbit/s
MPEG1	storage and retrieval, streaming	CBR, up to 1.15 Mbit/s
MPEG2	digital TV	CBR and VBR, from 1.5 up to 20Mbit/s
H.263	videoconference	CBR and VBR, from 33.6 Kbit/s
MPEG4-2	object-based coding, synthetic coding, ...	CBR and VBR
AVC/MPEG4-10/H.264	video streaming in mobile networks, HDTV, ...	CBR and VBR, from 10 Kbit/s up to several Mbit/s

Aspects to consider: download versus streaming

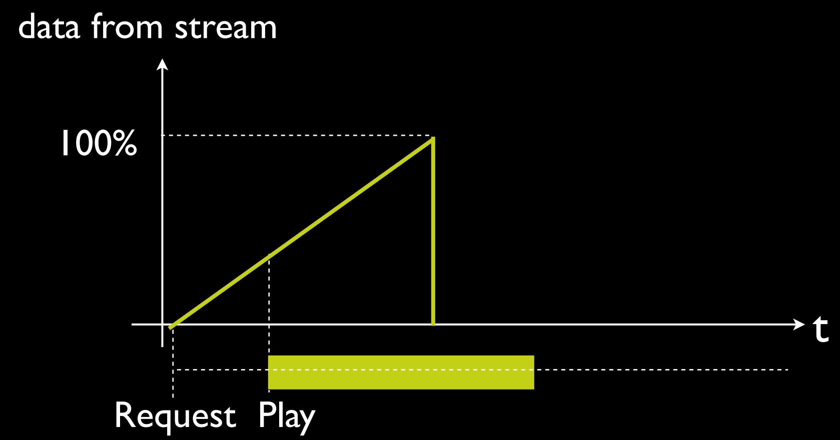
- download requires that the user waits until completion of the file
  - larger initial delay for the user
  - necessity of storage capacity in the receiver
  - limited flexibility as complete download must be made even when the user is not sure
- streaming requires only a small initial delay (pre-roll)
  - reduced storage capacity at the receiver and small wait for the user
  - content is segmented in packets, successively being sent out to the receiver; each packet may be presented as received and decoded
  - conceptually, streaming is a sequences of the following steps:
    - segmentation of the content at the sender side into packets
    - sequential transmission of those packets
    - decoding and presentation of each packet as received

Aspects to consider: download versus streaming

Download

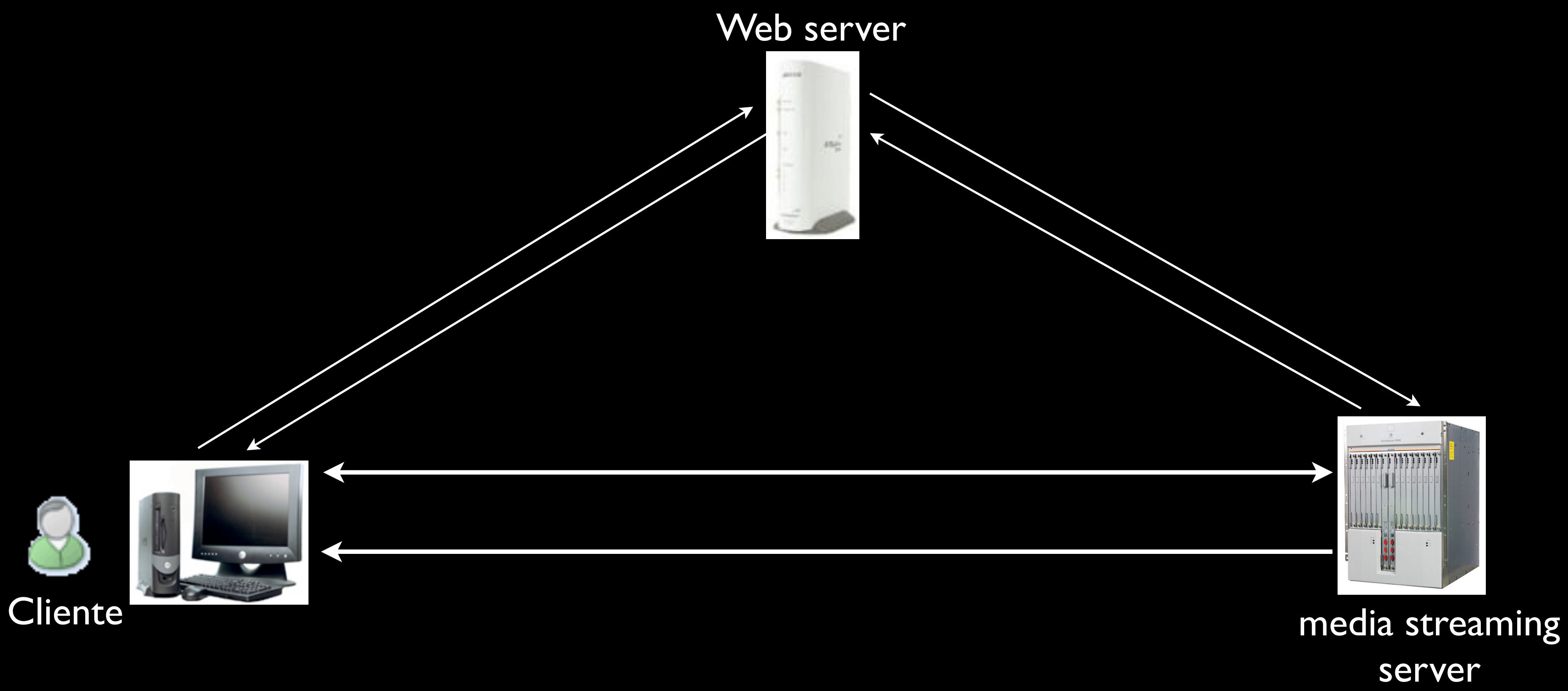


Streaming

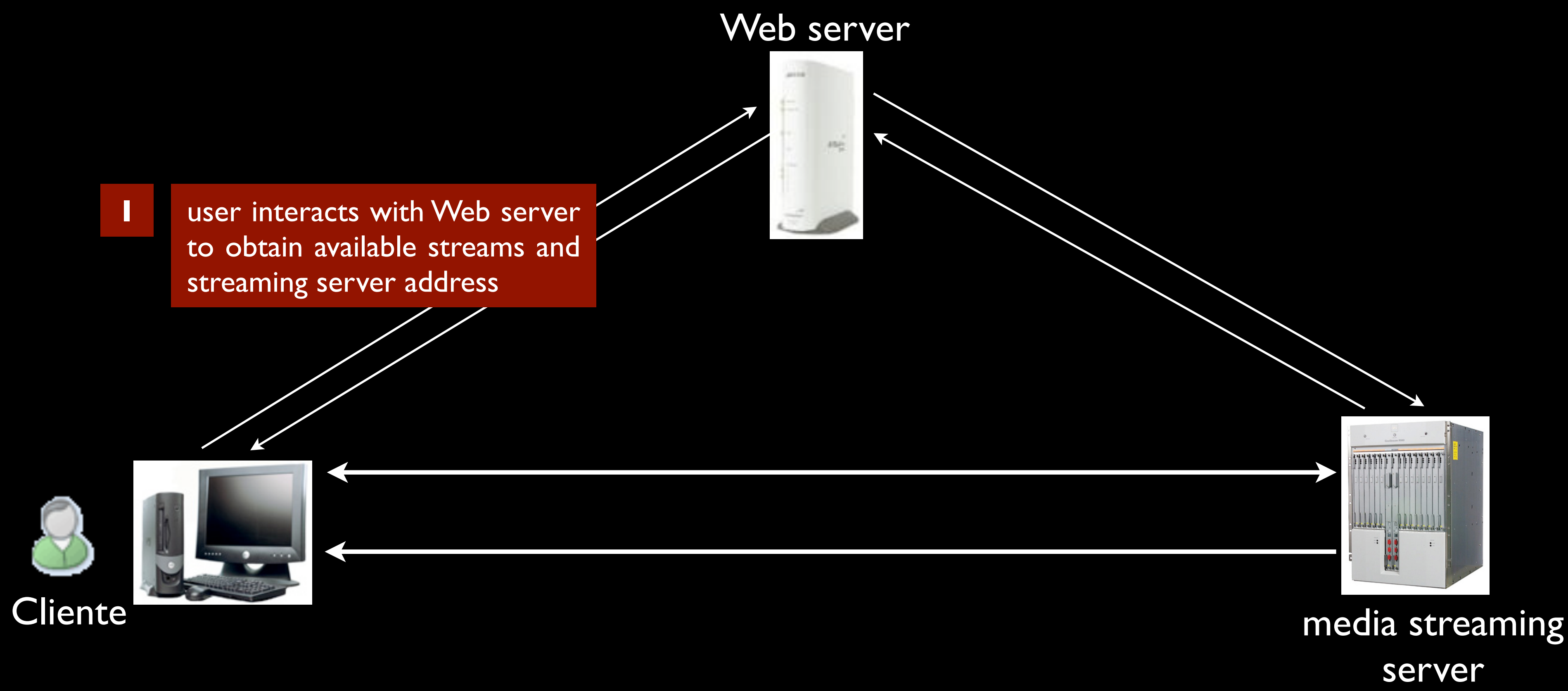


Adapted from L. Farinetti - Politecnico di Torino

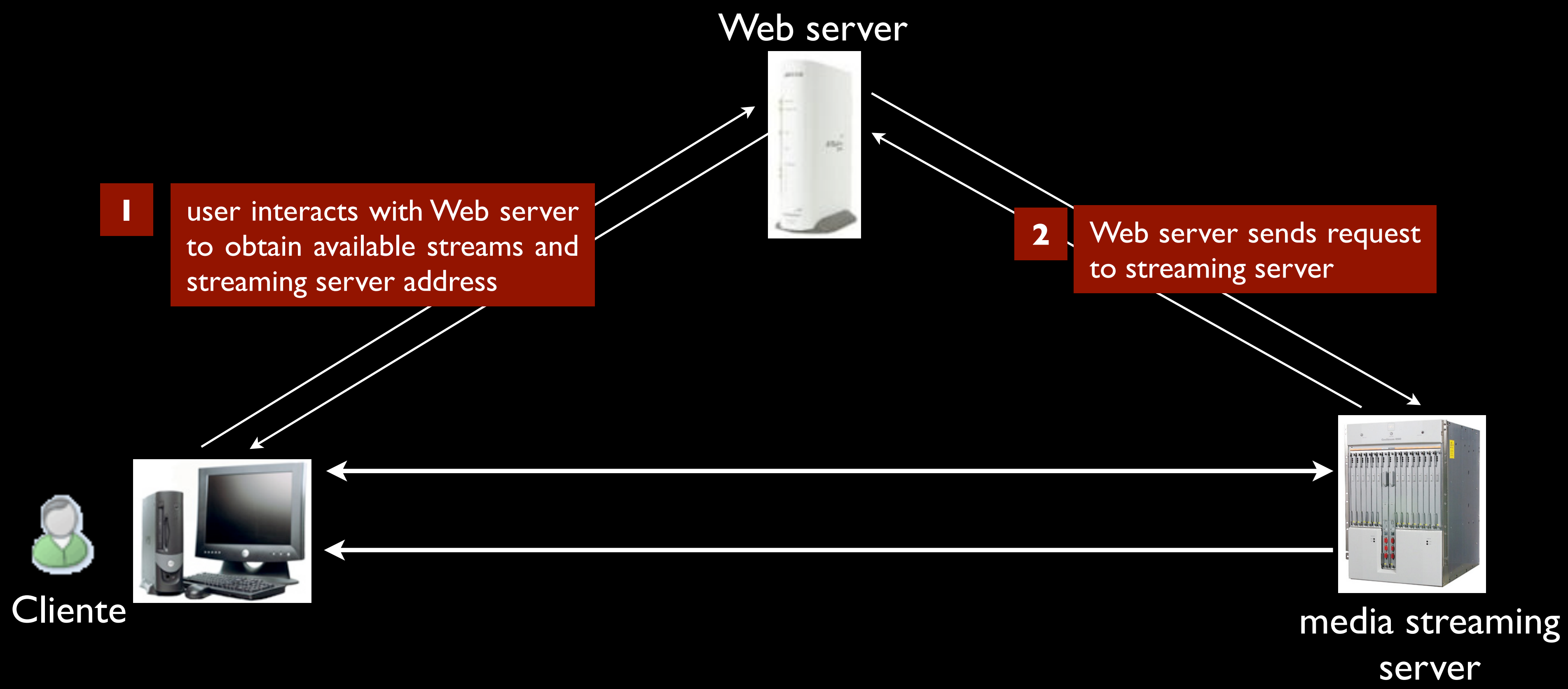
Streaming: how it works (overview)



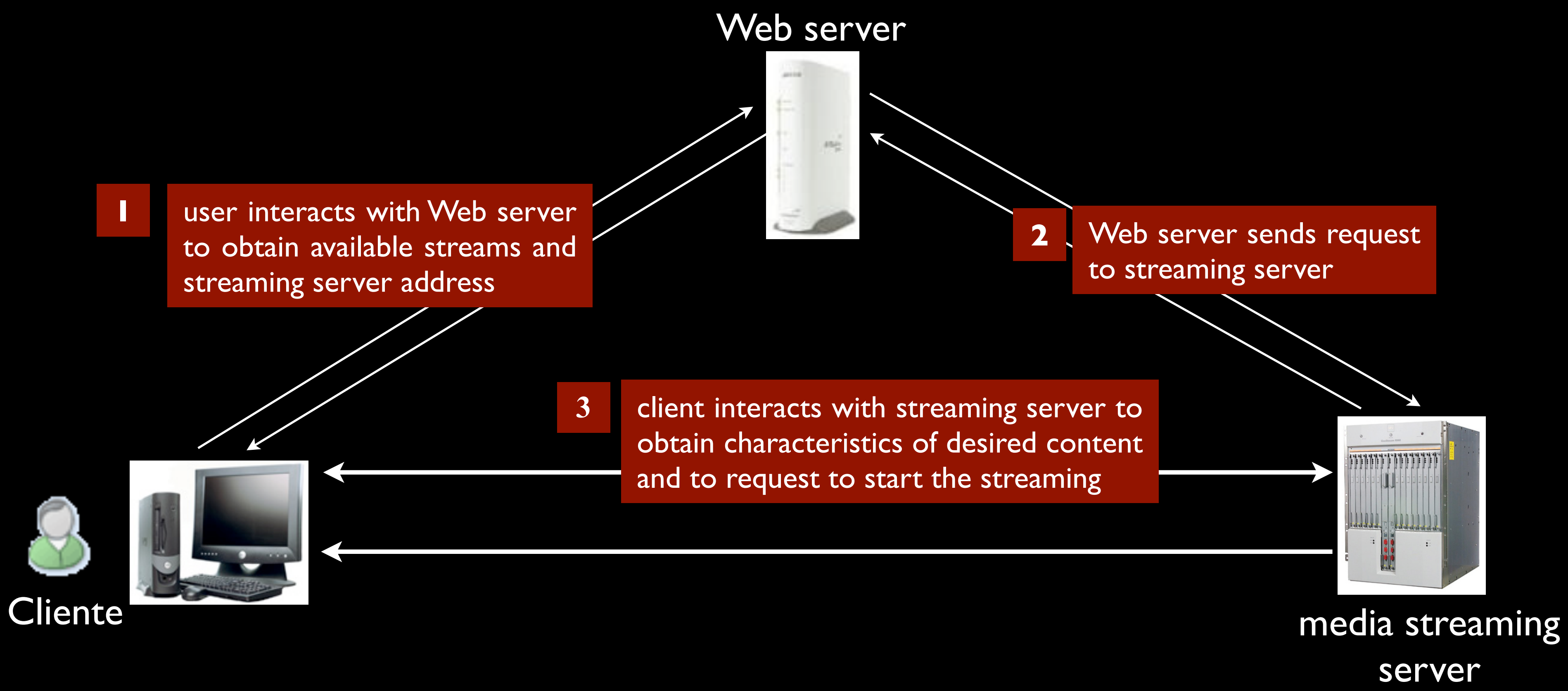
Streaming: how it works (overview)



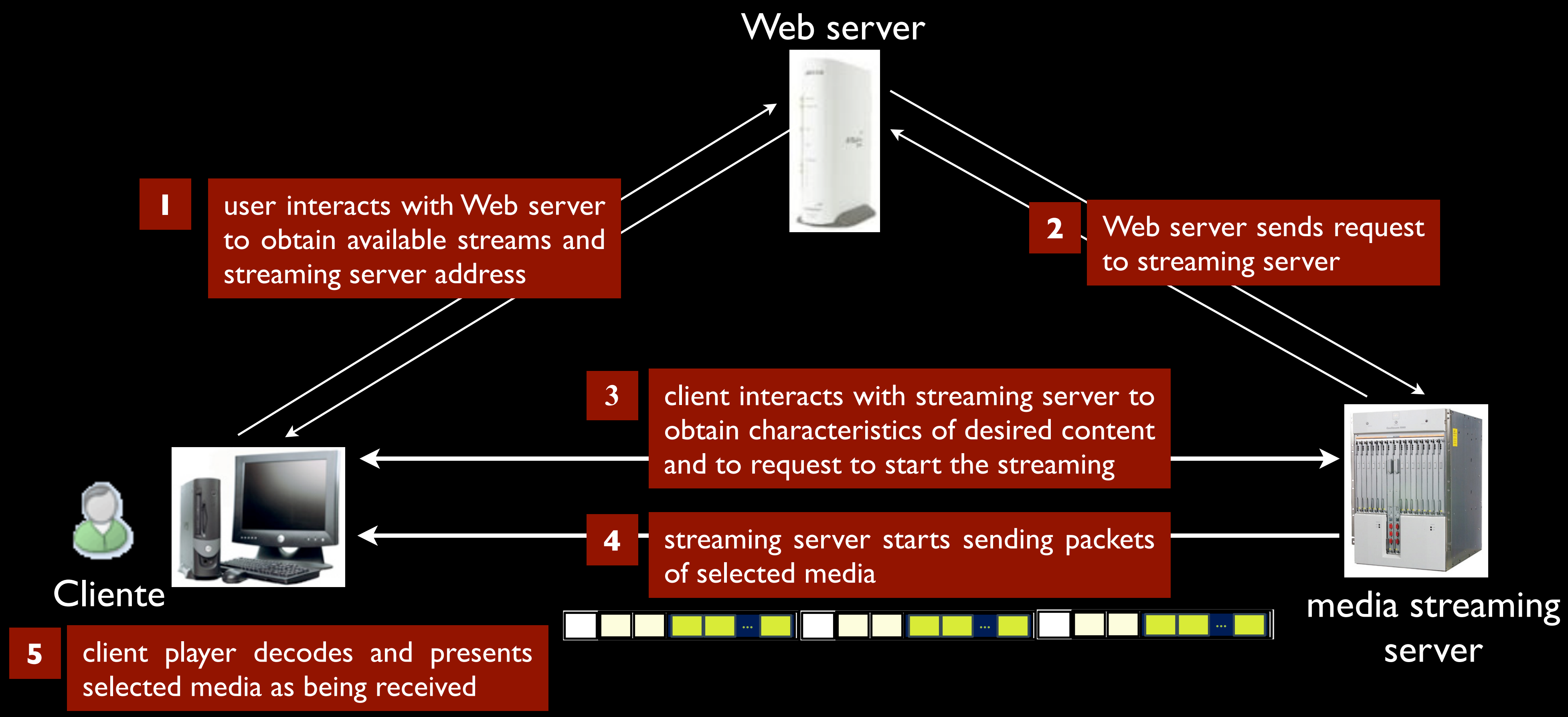
Streaming: how it works (overview)



Streaming: how it works (overview)



Streaming: how it works (overview)



Requirements of streaming

- video streaming can be expressed by a set of timing requirements
  - time interval Δ between images presentation (usually 40 ms) must be preserved
    - each image must be sent and decoded within its presentation time
- the sequence of images must observe the following rules:
  - image n transmitted and decoded up to instant Tn
  - image (n+1) transmitted and decoded up to instant Tn + Δ
  - image (n+2) transmitted and decoded up to instant Tn + 2Δ
  - ...
- data that arrives after the indicated instant can no longer be used except to assist the decoding of other images
  - it depends on the compression scheme

Challenges faced

- current networks exhibit, in more or less degree
  - delays, delay jitter, error rate, losses, congestion
- this has implications on
  - support for real-time constraints
    - how to cope with delay and jitter
  - QoS assurances
    - how to deal with bandwidth variability, packet loss, bit-error-rate ...
    - applications must be error robust
  - inter-media synchronization
    - jitter (variation on the delay) prevents lip-sync (audio and video) or synchronization of data (ex., powerpoint presentation with video in a seminar/class)
- number of simultaneous clients
  - costs of media servers, replication of content, use of CDNs, caching

Challenges faced - bandwidth variability

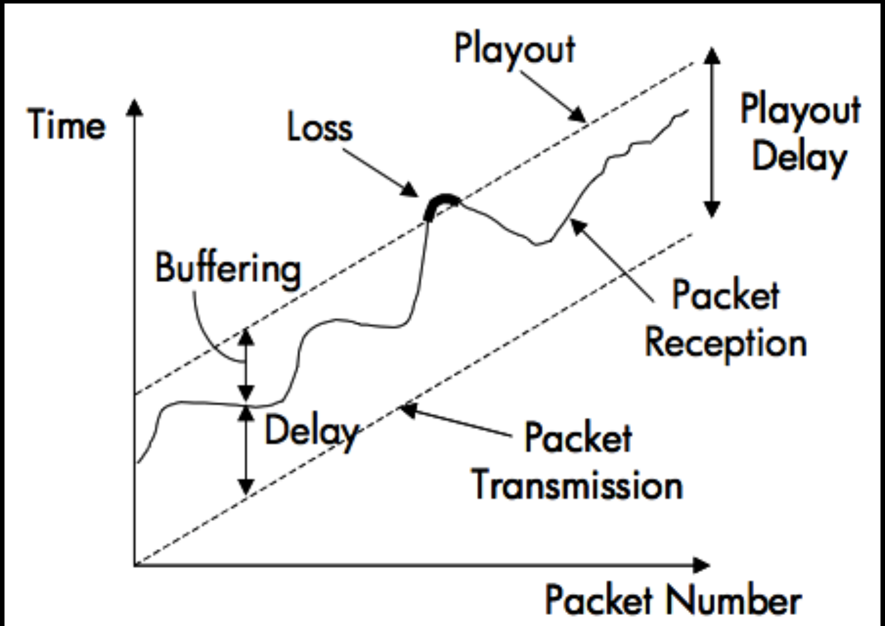
- in best-effort shared networks, available bandwidth between two peers is usually unknown and variable
  - it is not possible to predict congestion periods
  - if the server sends packets at a rate higher than the availability, congestion will occur, packets will be lost and consequently quality will drop
  - if the server sends packets at a rate lower than the availability, then it is not maximizing the compromise quality-bit rate
- to overcome this problem, systems should be able to instantaneously measure bandwidth availability and even infer future behaviors
  - and then be able to dynamically adapt the source bit rate to meet the constraints

Challenges faced - delay and jitter

- end-to-end delay may vary from packet to packet - jitter
  - real problem, as the rate of presentation of images should always be the same (after the pre-roll delay where initial data is buffered)
    - it may happen the decoder needing information from an image and that data not being yet arrived
      - buffer underflow
    - or the terminal receiving more data than the amount it can actually process
      - buffer overflow
  - this causes a picture degradation referred to as "jerkiness"
    - the movement in the video sequence is "ruined"
    - the decoder may have to skip an image or to delay the presentation of an image (thus "freezing" the previous image)

Challenges faced - delay and jitter

- How to compensate jitter?
  - jitter may be attenuated through the use of a playout buffer
    - it compensates the variations of the delay (i.e., the jitter) but it introduces an additional delay
  - if a maximum bound of the jitter is assured by the network, then a conveniently sized buffer will solve the problem



    - but many times this is not the case ...

Challenges faced - errors and losses

- depending on the type of network, different errors and losses may occur
  - in wired networks such as Ethernet, it is usual that entire packets are lost
  - in wireless networks, there are single bit errors or bursts of errors
- losses have a direct negative impact on the quality of the recovered pictures
  - to fight this impact, the application may implement error control mechanisms (channel coding and source coding approaches)

    (1) forward error correction (FEC)

    (2) retransmissions

    (3) error concealment

    (4) error-resilient video coding

Challenges faced - errors and losses

- how to fight errors and losses?
- FEC adds redundancy to the bit stream, the additional bits being used to recover the errors
  - for ex., Read-Solomon (RS) block codes
    - for each block of K packets it generates N packets (N>K)
    - the N-K packets are redundant packets
    - if at least K packets are correctly received, then the code is able to recover all the errors in the remaining N-K packets
    - increases the bit rate by a factor of N/K

Challenges faced - errors and losses

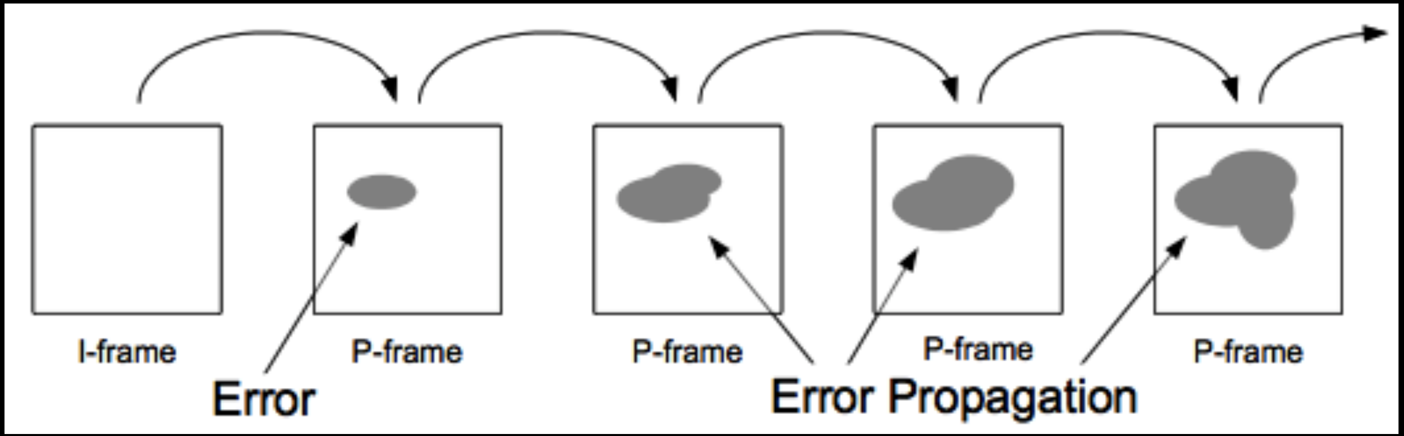
- how to fight errors and losses?
- error concealment estimates the amount of lost information to conceal or hide the error
  - it takes advantage of the spatial and temporal redundancy that exists in the video signal
  - part of the existing correlation is explored during the encoding process to achieve compression
  - un-explored correlation can be used to predict the loss of data
  - using spatial and/or temporal interpolation or extrapolation it makes an estimation of the lost data
    - when an image sample or a block of samples are missing due to transmission errors, the decoder can try to estimate them based on surrounding received samples, by making use of inherent correlation among spatially and temporally adjacent samples

Challenges faced - errors and losses

- how to fight errors and losses?
- error resilient video coding incorporates in the coding algorithm itself, mechanisms that are robust to errors
    - most popular compression schemes are based on
        - prediction with motion compensation, DCT or other spatial transform, entropy coding with variable length codes
    - in this kind of schemes the most common errors are

      1) loss of synchronism along the bitstream

      2) incorrect state or propagation of errors



Challenges faced - errors and losses

- error resilient video coding schemes can limit the extent of error propagation by
  - carefully designing both the predictive coding loop and the variable length coder
    - encoders become less efficient in terms of compromise quality-bit rate
    - but the decoding process becomes less susceptible to errors
      - erroneous or missing bits in a compressed stream will not have a disastrous effect in the reconstructed video quality
  - adding information that can be used by the decoder in case of missing data (at the research level)
    - "Spare pictures"
      - when the motion compensation reference image is lost, decoders may use a spare image that resembles the actual reference picture
    - Distributed video coding with side information available at the decoder

next: Multimedia traffic

- Multimedia traffic
  - digital content sources
    - principles of media compression
  - media compression standards
    - JPEG
    - MPEG2, MPEG4
    - H264/AVC)