Applications taxonomy

Andrea Bianco
Telecommunication Network Group
firstname.lastname@polito.it
http://www.telematica.polito.it/

Applications

- Data
  - Generated by single users, by servers, by data centers, by enterprise networks, by P2P architectures, by computing app (e.g. Mapreduce)
  - E-mail, web, messaging, remote login, file transfer, grid computing, ….
- Voice
  - Phone calls, IP calls, skype, …
- Audio
  - Music
- Video
- Multimedia
  - Streaming, videoconferencing

Applications taxonomy

- From the bit rate requirements point of view
  - Elastic applications (opportunistic)
    - If resources are available, elastic applications try to exploit them
    - If resources become scarce, elastic applications may reduce their rate (file transfer)
  - Non-elastic applications (multimedia mostly belong to this category)
    - Require a minimum amount of resources
      - If available, the application works properly
      - If not available, the application is unable to work properly
    - May become slightly elastic if changing the coding scheme

Applications taxonomy

- From the data loss point of view
  - Loss tolerant
    - Uncompressed audio, video, voice
  - Loss intolerant
    - File transfer, e-mail, web, grid computing, compressed audio, video, voice
- From the time sensitivity point of view
  - Not sensitive
    - File transfer, e-mail, web, grid computing
  - Very sensitive (100ms)
    - Phone
  - Sensitive (few s)
    - Streaming

Elastic applications

- Consider a file transfer
- Small end-to-end delay preferred (not fundamental)
- Required bit rate: the higher the better but it may be low
- Packet losses recovered by the transport protocol through retransmission (less often through error correction)
  - End-to-end delay increases

Real-time multimedia and streaming

- Real time multimedia applications
- Two users interact (in real time)
- Low delay fundamental (a delayed packet is equivalent to a lost packet)
- Required bit-rate **may** be significant depending on whether video is involved
- May be robust to (limited) packet losses depending on the compression level

- Multimedia streaming applications
- No real time requirements
- May tolerate packet delays if initial delay large (buffering)
- Required bit-rate **may** be significant depending on whether video is involved or not
- May be robust to (limited) packet losses depending on the compression level

Example of real time multimedia application

- IP telephony: three different problems
  - Establish multimedia connection, find IP addresses (possibly multicast), negotiate the type of coding and/or compression scheme, possibly inter-operate with the telephone network
  - Once the connection has been established, transfer audio packets
  - Periodically send feedback information to the transmitter (and to receivers) to indicate the quality of the (possibly multimedia) connection

H.323
SIP

RTP

RTCP

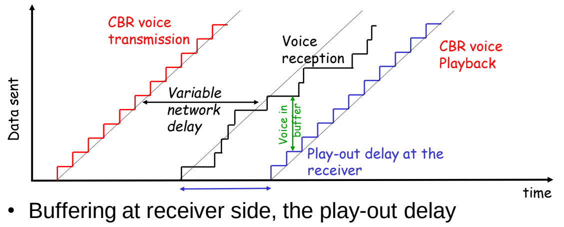
Real-time multimedia: Internet Phone

- Voice as input: sounds and silence period alternate
- Packets generated at a constant rate or when the source emitting power is above a given threshold :
  - E.g.: 20 ms of voice sample at 8kb/s
- Packets are delayed (should be compensated) and lost:
  - Network losses, due to congestion
    - max tolerable may be 10%
  - Losses due to excessive delays (IP datagram received too late for playout)
    - Max tolerable is roughly 400 ms
- Compensation techniques
  - At the transmitter (adaptive coding)
  - At the receiver (buffering)

Reaction to losses, delay and jitter

- Use of a variable bit-rate coder
  - Send small size packets when congestion is detected and the experienced delay is high
  - Send large size packets if the network is lightly loaded
- Quality of reception estimate mechanisms are needed
- The transmitter bit rate should be controlled according to:
  - Instantaneous and/or average loss rate
  - Absolute or relative delay
  - Delay jitter

Delay and jitter compensation: buffering

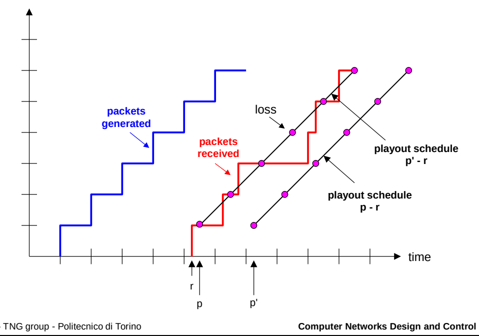


- Buffering at receiver side, the play-out delay compensate the network delay and jitter
  - Fixed play-out delay
  - Adaptive play-out delay

Fixed playout delay

- The receiver plays out each voice sample exactly *q* seconds after the sample generation
  - If the sample has timestamp *t*, it is played out at *t+q*
  - If the sample is received after *t+q*, it is considered as lost
  - Coding scheme may compensate for losses
- The value assumed by *q*:
  - if *q is* large: less packets are lost, higher delay, more buffering needed
  - if *q* is small: improved interactivity

Fixed playout delay



Adaptive playout delay

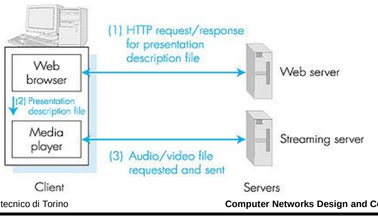
- Objective: minimize play-out delay while keeping low the loss rate
  - Estimate the network delay, to determine the play-out delay at speech startup
  - Compress or extend the silence periods
  - Samples always reproduced after 20ms during activity periods

Multimedia streaming

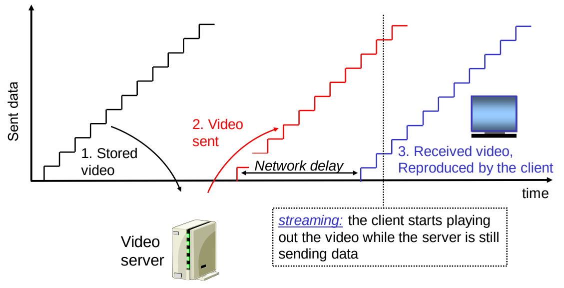
- Streaming
  - Multimedia file stored at the source
  - Sent to the client
  - File play-out starts when the file transfer is under way
  - Constraint: missing data should reach the receiver before the play-out ends
  - Alternative to file download to playback it later (file transfer!)

Multimedia: streaming approach

- The client browser receives the metafile containing the multimedia streaming file description
- The browser passes the metafile to the player
- The player contacts the streaming server
- The server sends data in streaming



Multimedia streaming



Multimedia streaming with buffering at the client side

- Tradeoff betweeen initial delay (buffer size) and tolerance to network jitter