Multimedia in IP

The Web view

Interactive flows, real-time

- Audio-Video Flows
  - streaming audio/video
  - Use buffering at receiver
- Interactive real-time:
  - Buffering in receiver very limited
  - Delay <200ms
  - jitter <200ms
  - Keep low losses
- Loss impact:
  - depends on aplication, media, and user
- Áudio:
  - Humans tolerate “bad” audio for conversation
  - Humans require “good” audio for entertainment
- Vídeo:
  - Humans tolerate “low” video quality for business
  - Humans enjoy “good” video quality for entertainment
- Synchronizing audio/video:
  - Different flows?

© Rui L. Aguiar
(ruilaa@det.ua.pt)

universidade de aveiro

Audio

QoS Requirements

- Delay < 400ms:
  - Including jitter
- Low losses are prefereable:
  - some coding systems are tolerant to low losses
- Data rates:
  - Voice ≤ 64Kb/s
  - "good" music ≥ 128Kb/s

- Time-domain sampling
- Example – coded voice:
  - Coded 64Kb/s PCM
  - Sampling 8-bit
  - 8000 samples/sec
  - 40ms "time slices" for audio
  - 320 bytes (audio) per packet
  - 48 bytes overhead
    (20 bytes IP header)
    (8 bytes UDP header)
    (20 bytes RTP header)
  - 73.6Kb/s

© Rui L. Aguiar
(ruilaa@det.ua.pt)

universidade de aveiro

Voice coders taxonomy

- Voide coders
  - Wave coders
    - Time domain: PCM, ADPCM
    - Frequency domain: e.g. Sub-band coder, Adaptive transformer coders
  - Source coders
    - Linear Predictive coder
    - Vocoder

- Vocoders:
  - Analyze voice, extract parameters, transmit model parameters
    - Voice synthesized by parametric models
  - LPC-10: 2.4 kbps
- Wave coders: try to preserve the wave format, not voice specific.
  - PCM 64 kbps, ADPCM 32 kpbs, CVSDM 32 kbps
- Hybrid: Try to combine the best of both… Eg: CELP

Vídeo

universidade de aveiro

QoS Requirements

- Delay < 400ms:
  - including jitter
  - Equal to audio
  - Allows synchronization of flows
- Data rate – depends of:
  - Frame size
  - Color depth
  - frame rate
  - coding

- Usually processing on frequency domain:
  - discrete cosine transform (DCT) very common
- Losses need to be low

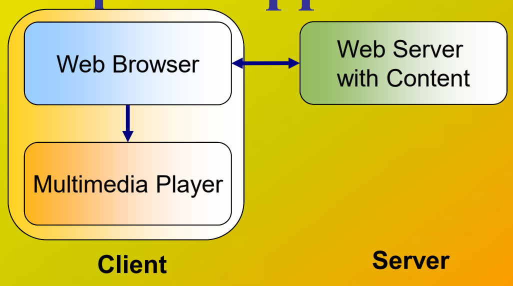
Internet Multimedia Support

universidade de aveiro

- Integrated services philosophy.
  - Requires dedicated links/channels with QoS requirements.
- Differentiated services philosophy.
  - Fewer changes to Internet infrastructure.
- Best effort.
  - No major changes.
  - More bandwidth when needed.
  - Application-level control and distribution.

Would require QoS
Only possible in private networks or operator infrastructure

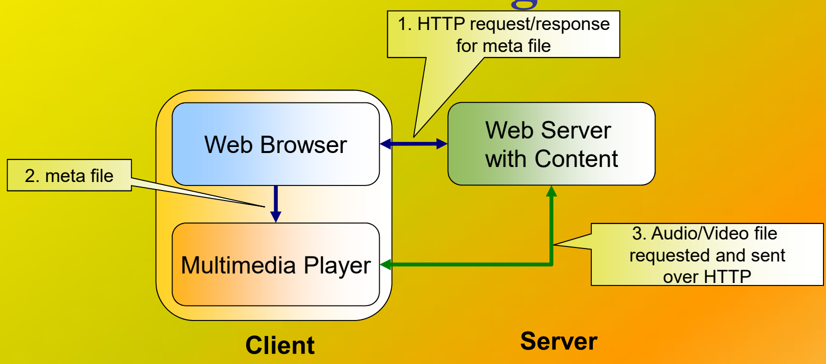
Internet Multimedia: Simplest Approach



- Audio or video stored in file.
- Files transferred as HTTP object (or using P2P).
  - Received in entirety at client as a file.
  - Then passed to default player in client.
- Audio&video is not streamed!
- No “pipelining”, long delays until playout!

36

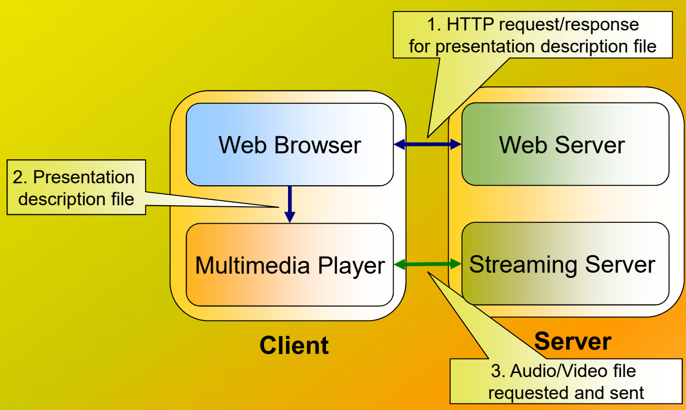
Internet Multimedia: Web Streaming



- Browser GETs metafile.
  - Content negotiation may happen.
- Browser launches player, passing metafile
- Player contacts server.
- Server streams audio/video to player.

37

Streaming from a streaming server

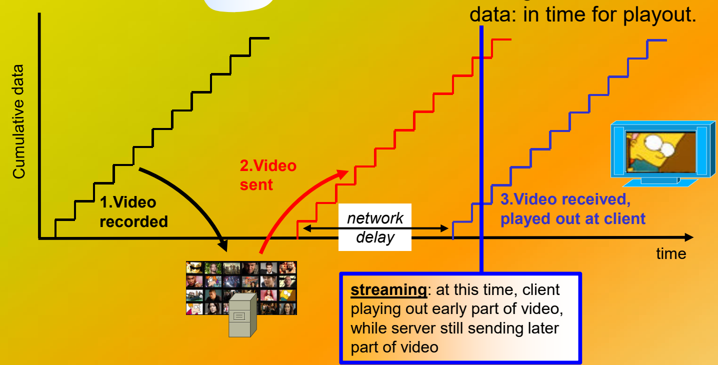


- This architecture allows for non-HTTP protocol between server and media player.
- Can use UDP or TCP transport.

38

Web Streaming: Stored Multimedia

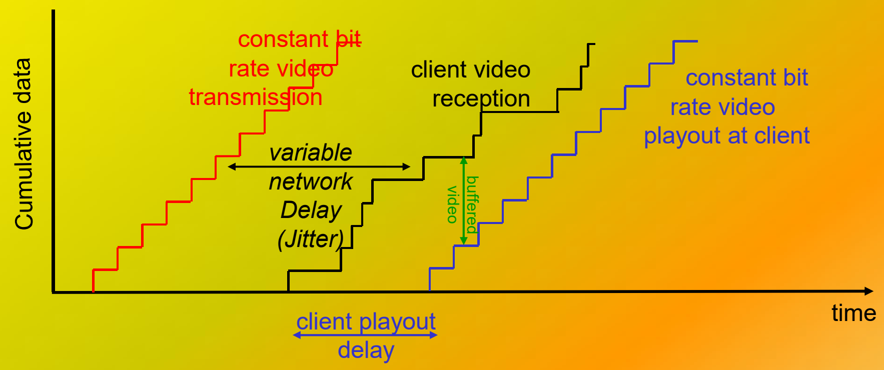
- Streaming:
  - Media stored at source is transmitted to client.
  - Client visualization begins before all data has arrived.
    - Timing constraint for transmitted data: in time for playout.



39

universidade de aveiro

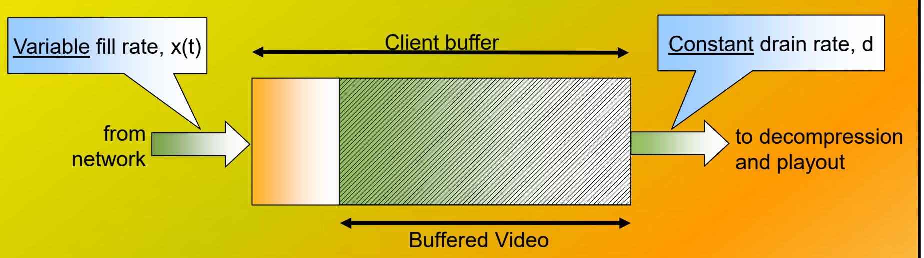
Streaming Multimedia: Client Buffering



- Client-side buffering, playout delay compensate for network-added delay, delay jitter.
  - Size of buffer/playout delay are parameters that can be adjusted dynamically
  - For VoIP, delay between 2 packets is about 20ms (plus *jitter*).

universidade de aveiro

Streaming Multimedia: Client Buffering



- Client-side buffering, playout delay compensate for network-added delay, delay jitter.
  - Size of buffer/playout delay are parameters that can be adjusted dynamically

Streaming Stored Multimedia

- Application-level streaming techniques for making the best out of best effort service:
  - Client side buffering.
  - Use of UDP versus TCP.
  - Multiple encodings of multimedia.
- Multimedia Player
  - Jitter removal,
  - Decompression,
  - Error concealment,
  - Graphical user interface with controls for interactivity.
- Network
  - Close to client content (multi-content) buffering for faster interactivity
  - Only viable in network operator proprietary services.

universidade de aveiro

Streaming Stored Multimedia: Interactivity

- VCR-like functionality: client can pause, rewind, fast-foward, push slider bar.
  - 10 sec initial delay OK.
  - 1-2 sec until command effect OK.
  - Timing constraint for still-to-be transmitted data: in time for playout.

universidade de aveiro

Streaming Live Multimedia

- Examples:
  - Internet TV/radio show.
  - Live sporting event.
- Streaming
  - Playback buffer.
  - Playback can lag tens of seconds after transmission.
  - Still have timing constraint.
- Interactivity
  - Fast forward impossible.
  - Rewind, pause possible!

45

universidade de aveiro

Interactive Real-Time Multimedia

- Applications:
  - IP telephony, video conference, online-game multimedia actions, distributed interactive worlds.
- End-end delay requirements:
  - Audio: < 150 msec good, < 400 msec OK
    - Includes application-level (packetization) and network delays.
    - Higher delays noticeable, impair interactivity.
- Requires session initialization
  - Advertise its IP address, port number, encoding algorithms, required contents, available contents

46

universidade de aveiro

UDP Streaming vs. TCP Streaming

- UDP
  - Server sends at rate appropriate for client .
    - Often send rate = encoding rate = constant rate.
    - Then, fill rate = constant rate - packet loss.
  - Short playout delay (2-5 seconds) to compensate for network delay jitter.
  - Error recover: time permitting.
- TCP
  - Send at maximum possible rate under TCP.
  - Fill rate fluctuates due to TCP congestion control.
  - Larger playout delay: smooth TCP delivery rate.
  - HTTP/TCP passes more easily through firewalls.

47

universidade de aveiro

HTTP/TCP Streaming

- Multiple versions with distinct/complementary characteristics are generated for the same content
  - With different bitrates, resolutions, frame rates.
- Each version is divided into time segments.
  - e.g., two seconds.
- Each segment is provided on a web server and can be retrieved through standard HTTP GET requests.
- Examples of protocols:
  - MPEG's Dynamic Adaptive Streaming over HTTP (DASH).
    - Standard ISO/IEC 23009-1. YouTube's default.
  - Adobe HTTP Dynamic Streaming (HDS).
  - Apple HTTP Live Streaming (HLS).
  - Microsoft Smooth Streaming (MSS).

48

User Control of Streaming Media: RTSP

- RTSP (Real Time Streaming Protocol): RFC 2326
  - Client-server application layer protocol.
  - For user to control display: rewind, fast forward, pause, resume, repositioning, etc…
- Does not define how audio/video is encapsulated for streaming over network.
- Does not restrict how streamed media is transported.
  - Can be transported over UDP or TCP.
- Does not specify how the media player buffers audio/video.
- RTSP messages are also sent out-of-band:
  - RTSP control messages use different port numbers than the media stream: out-of-band
    - Port 554
  - The media stream is considered “in-band”

49

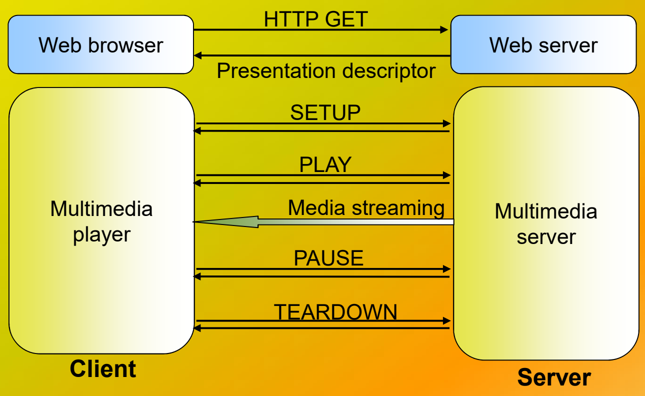
RTSP: out of band control

- FTP uses an “out-of-band” control channel:
  - A file is transferred over one TCP connection
  - Control information (directory changes, file deletion, file renaming, etc.) is sent over a separate TCP connection
  - The “out-of-band” and “in-band” channels use different port numbers
- RTSP messages are also sent out-of-band:
  - RTSP control messages use different port numbers than the media stream: out-of-band
  - Port 554
  - The media stream is considered “in-band”

50

universidade de aveiro

RTSP Operation



51

universidade de aveiro

Metafile Example

- <title>Twister</title>
- <session>
- <group language=en lipsync>
- <switch>
- <track type=audio
- e="PCMU/8000/1"
- src = "rtsp://audio.example.com/twister/audio.en/lofi">
- <track type=audio
- e="DVI4/16000/2" pt="90 DVI4/8000/1"
- src="rtsp://audio.example.com/twister/audio.en/hifi">
- </switch>
- <track type="video/jpeg"
- src="rtsp://video.example.com/twister/video">
- </group>
- </session>

52

RTSP Exchange Example

- C: SETUP rtsp://audio.example.com/twister/audio RTSP/1.0
- Transport: rtp/udp; compression; port=3056; mode=PLAY
- S: RTSP/1.0 200 1 OK
- Session 4231
- C: PLAY rtsp://audio.example.com/twister/audio.en/lofi RTSP/1.0
- Session: 4231
- Range: npt=0-
- C: PAUSE rtsp://audio.example.com/twister/audio.en/lofi RTSP/1.0
- Session: 4231
- Range: npt=37
- C: TEARDOWN rtsp://audio.example.com/twister/audio.en/lofi RTSP/1.0
- Session: 4231
- S: 200 3 OK

53

Streaming Media: RTSP

➔RTSP: RFC 2326
- Application layer protocol (client-server)
- Presentation control of streaming
  - rewind, fast forward, pause, resume, reposition, etc…

Limitations:
- Does not define how audio/video is encapsulated for streaming (RTP)
- Does not impose transport mechanisms (UDP or TCP)
- Does not describe how the audio/video is played (nor the type of buffering)

Out of band control:
- RTSP messages use different ports from the “media stream” (which is “in-band”)
  - Port 554

54

universidade de aveiro

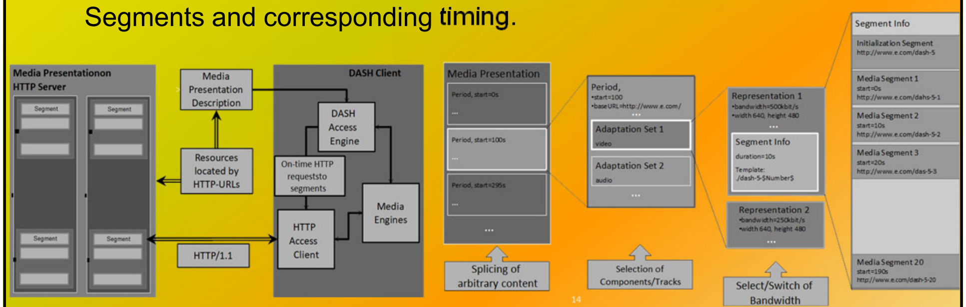
RTSP example

Client: SETUP rtsp://audio.example.com/twister/audio RTSP/1.0
  Transport: rtp/udp; compression; port=3056; mode=PLAY
Server: RTSP/1.0 200 1 OK
  Session 4231
Client: PLAY rtsp://audio.example.com/twister/audio.en/lofi RTSP/1.0
  Session: 4231
  Range: npt=0-
Client: PAUSE rtsp://audio.example.com/twister/audio.en/lofi RTSP/1.0
  Session: 4231
  Range: npt=37
Client: TEARDOWN rtsp://audio.example.com/twister/audio.en/lofi RTSP/1.0
  Session: 4231
Server: 200 3 OK

universidade de aveiro

Dynamic Adaptive Streaming over HTTP (DASH)

- Developed to be an Open Standard Delivery Format.
  - MPEG DASH ISO/IEC 23009-1.
- Video streaming solution where pieces of video streams/files are requested with HTTP and spliced together by the client.
  - Client entirely controls delivery.
- Media Presentation Description (MPD) describes accessible Segments and corresponding timing.



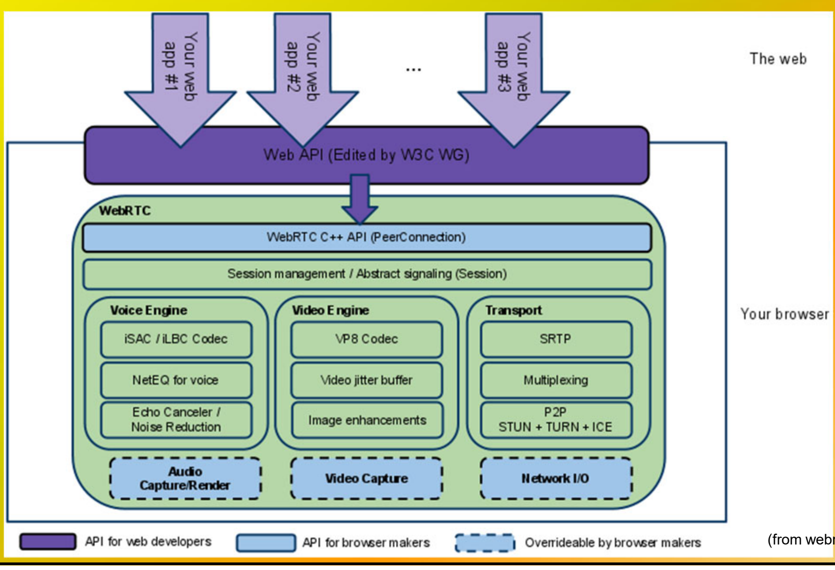
universidade de aveiro

WebRTC

- Peer-to-peer connections.
  - An instance allows an application to establish peer-to-peer communications with another instance in another browser, or to another endpoint implementing the required protocols.
- RTP Media.
  - Allow a web application to send and receive media stream over a peer-to-peer connection (discussed in a minute)
- Peer-to-peer Data
  - Allows a web application to send and receive generic application data over a peer-to-peer connection.
- Peer-to-peer DTMF.

universidade de aveiro

WebRTC Architecture



(from webrtc.org)

universidade de aveiro

Real Time Transport Protocol (RTP)

59

universidade de aveiro

Disadvantages of TCP

- Connection-oriented
  - Not appropriate to multicast
- Retains traffic (push)
- Retransmissions are not convenient to “soft” real time traffic (i.e. that accepts losses)
- Does not contain limitation on data length
- Does not provide timing information

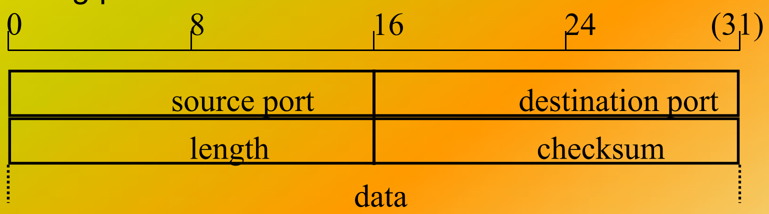
60

universidade de aveiro

Disadvantages of UDP

- *Connectionless* packet service, without guaranties, without order
- Without error control
- Without flow control
- Without congestion control
- Using ports

- Should be used to real-time data:
  - Congestion and flow control mechanisms in TCP are not adequated

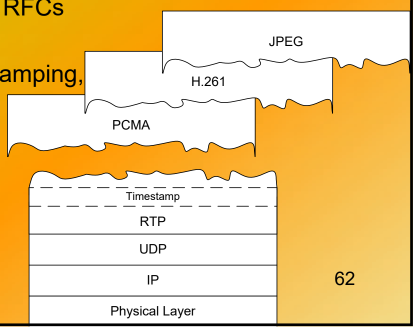


61

universidade de aveiro

RTP

- RTP specifies a packet structure for packets carrying audio and video data
- Includes two distinct protocols over UDP
  - *RTP:* streaming transmission at application level (simple signaling)
    - » Data channel
  - *RTCP:* control messages
    - » Control and information channel
- RFC1889: general format of messages
  - Formats to specific media types in other RFCs
- Provides services of
  - Packet identification, sequencing, timestamping, feedback
  - But no delivery guaranties or QoS
- RTP runs in the end systems.
- Interoperability:
  - if two Internet applications run RTP, then they **may** be able to work together

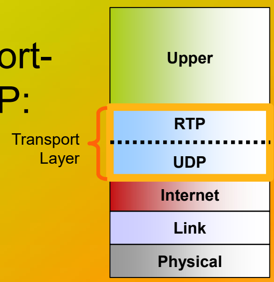


62

universidade de aveiro

RTP runs on top of UDP

- RTP libraries provide a transport-layer interface that extend UDP:
  - Port numbers, IP addresses
  - Payload type identification
  - Packet sequence numbering
  - Time-stamping



63

universidade de aveiro

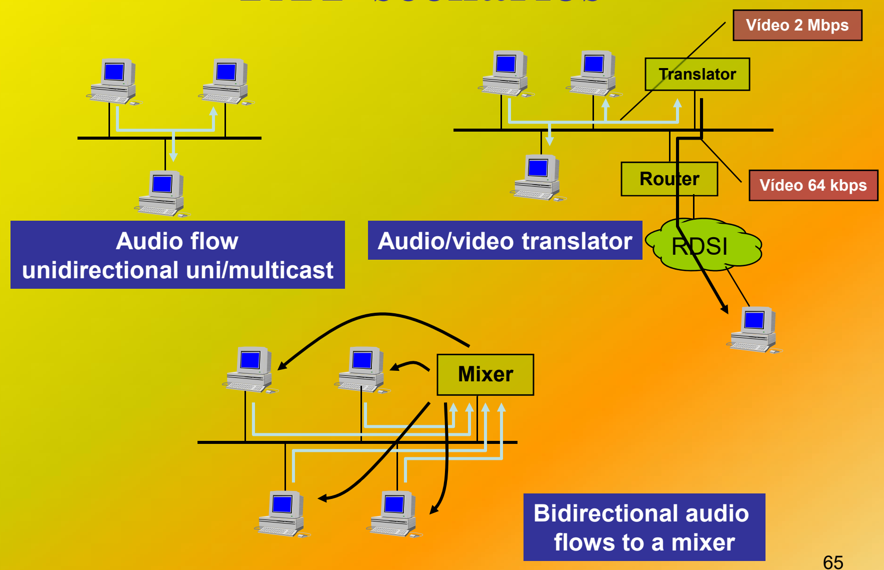
RTP and QoS

- RTP does not provide any mechanism to ensure timely delivery of data or provide other quality of service guarantees.
- RTP encapsulation is only seen at the end systems: it is not seen by intermediate routers.
  - Routers providing best-effort service do not make any special effort to ensure that RTP packets arrive at the destination in a timely matter.
  - Operators may create separate channels for specific services

64

universidade de aveiro

RTP scenarios



universidade de aveiro

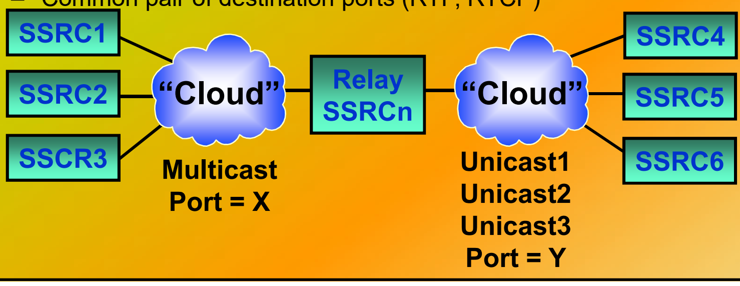
Types of RTP entities (Relays)

- Translators
  - Modify data format
  - Do not modify SSRC nor timestamp
  - Examples:
    - Multicast to unicast, coding, flow quality reduction
- Mixers (SSRC field)
  - Generate an output through several inputs (CSRC field)
  - Examples:
    - Audio mixer, video PiP, ...

universidade de aveiro

RTP session and cloud

- “session”: “clouds” interconnected by relays, in which each participant has a unique identifier (SSRC)
  – Usually an unique address (multicast) and a pair of ports to RTP and RTCP
- “Transport cloud”: sources, receivers and “relays” (translators/ mixers) of packet flows that share:
  – Direction of destination (1 multicast direction and 1 list of unicast directions)
  – Common pair of destination ports (RTP, RTCP)



67

universidade de aveiro

“Light” sessions

- There is no explicit control of group participation
  – Participants get together in groups
- There is no explicit conference control
- Members that have data to send, just send it
- Session packets quasi-periodical
  – Identity, reception reports, sincronization information
- Adequated to multicast models

68

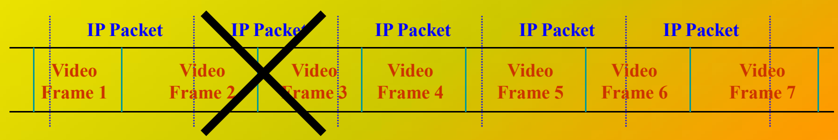
Concept: Application Level Framing (ALF)

- Application semantics should be reflected in the communication protocol
- Application to control the packing of information in packets
- It creates Application Data Units (ADUs)
  - Each ADU can be independently processed
- Associate ADUs in a single network packet (if possible)

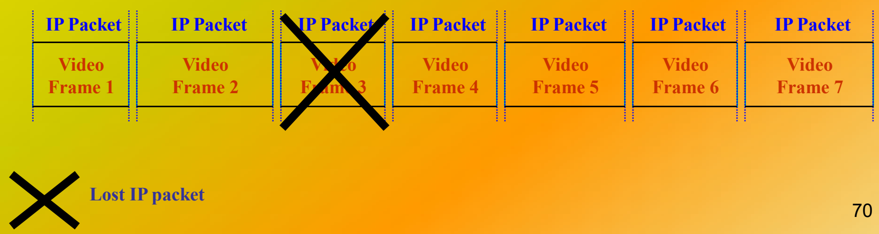
69

Example ALF:ADU

Application ignores the framing aspects (e.g. TCP)



Application frames are mapped in network layer packets (e.g. UDP without fragmentation)

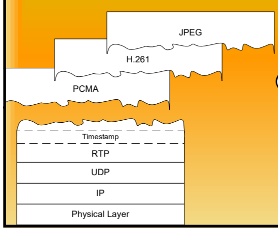


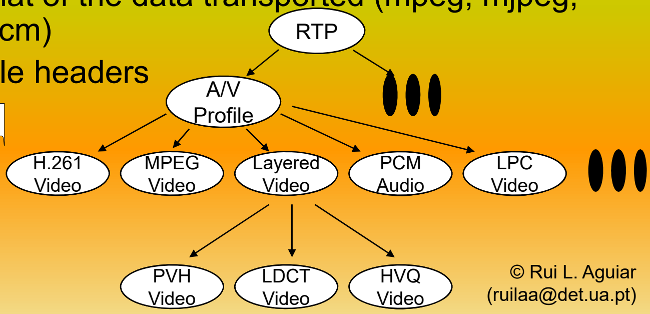
70

universidade
de aveiro

Back to RTP...

- Objective:
  - Global framework for packet delivery...
- A.L.F. principle appears in the RTP hierarchical structure (because of multimedia formats)
  - Requires profiles (e.g. Audio video)
  - And format of the data transported (mpeg, mjpeg, H.261, pcm)
  - Extensible headers





universidade
de aveiro

Joint source and channel coding

- Principle
  - Corollary for A.L.F.
    - Need to consider the transmission channel when coding the data
- The source coding algorithm becomes sensitive to the network
  - Algorithms are modified for producing self-sufficient information.
  - Packet loss has low impact
- Example: H.261

  Next slides…

universidade de aveiro

Standard H.261 algorithm

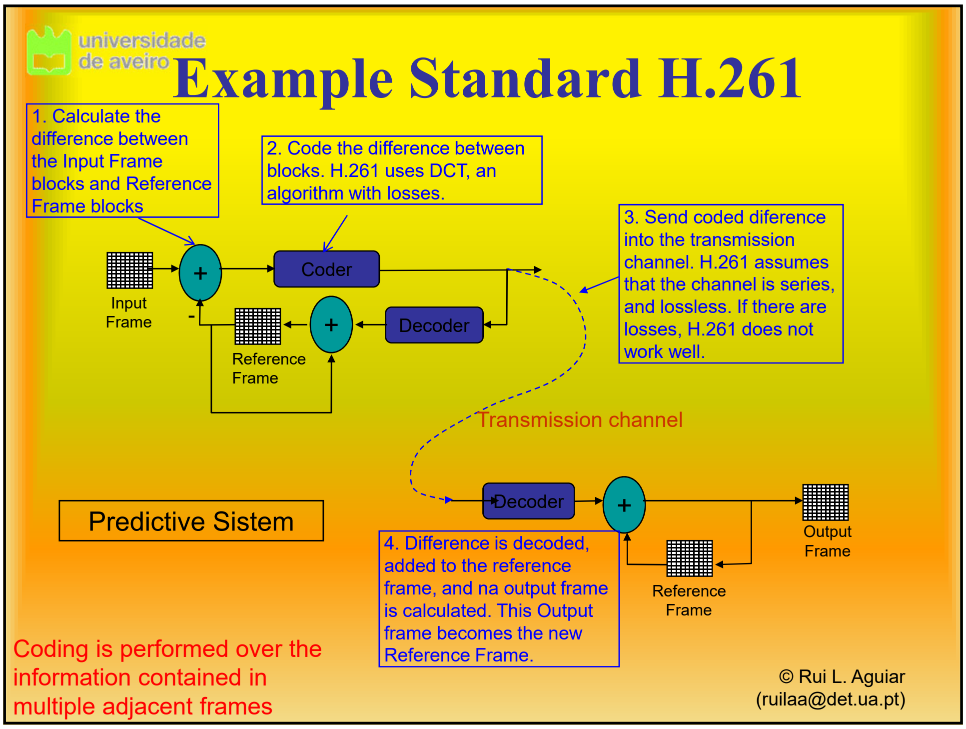
- Video coding similar to MPEG.
- Predictive mechanism
  - "predicted image"
- Time compression
  - differential coding between frame N and frame N-1
- Assumes "lossless" channel
- If there are data loss...
  - Resynchronizes with the next Group of Blocks (GOB)
  - Reconstruction errors remain in the decoder.

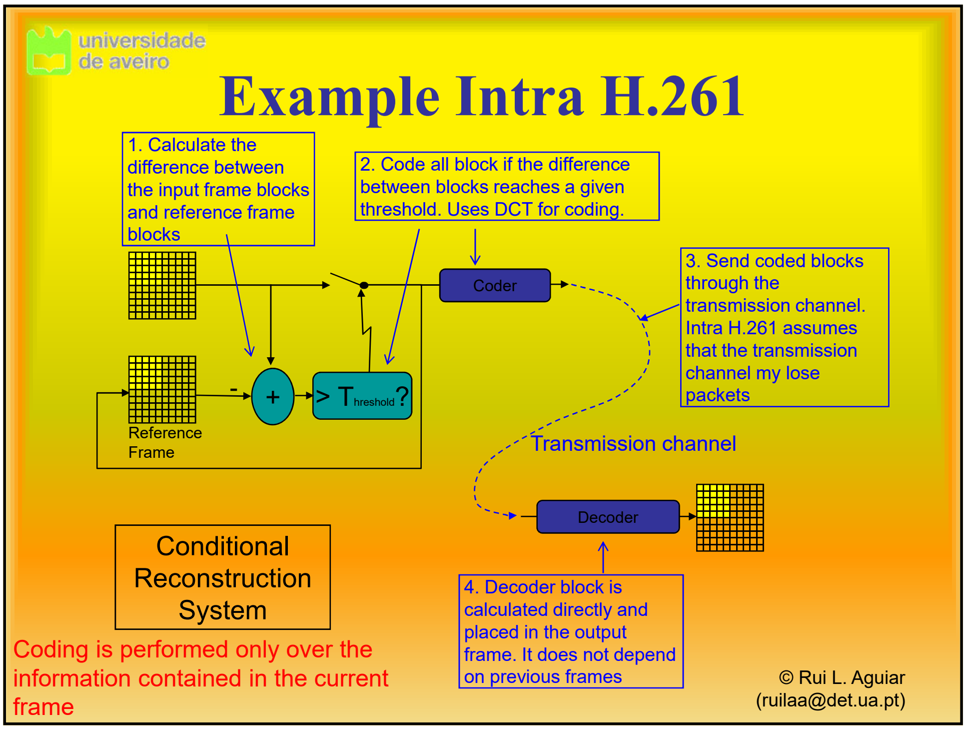
universidade de aveiro

Intra H.261 Algorithm

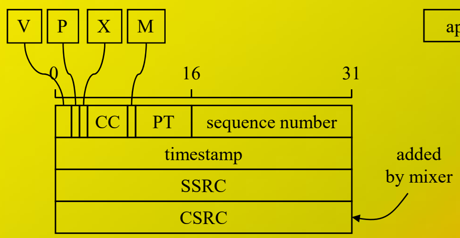
- Reacts to IP network features
- Subset of H.261
- Conditional image reconstruction
- No differential reconstruction of frames
- Macroblocks become ADUs (application data units)

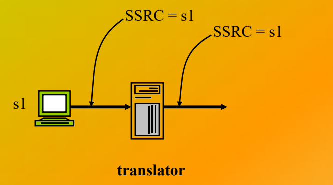
© Rui L. Aguiar
(ruilaa@det.ua.pt)


universidade de aveiro
Example Standard H.261
1. Calculate the difference between the Input Frame blocks and Reference Frame blocks
2. Code the difference between blocks. H.261 uses DCT, an algorithm with losses.
3. Send coded diference into the transmission channel. H.261 assumes that the channel is series, and lossless. If there are losses, H.261 does not work well.
Coder
Decoder
+
-
Input Frame
Reference Frame
Transmission channel
Decoder
+
Output Frame
Reference Frame
Predictive Sistem
4. Difference is decoded, added to the reference frame, and na output frame is calculated. This Output frame becomes the new Reference Frame.
Coding is performed over the information contained in multiple adjacent frames
© Rui L. Aguiar (ruilaa@det.ua.pt)


universidade de aveiro
Example Intra H.261
1. Calculate the difference between the input frame blocks and reference frame blocks
2. Code all block if the difference between blocks reaches a given threshold. Uses DCT for coding.
3. Send coded blocks through the transmission channel. Intra H.261 assumes that the transmission channel my lose packets
Coder
+
-
> Threshold ?
Reference Frame
Transmission channel
Decoder
Conditional Reconstruction System
4. Decoder block is calculated directly and placed in the output frame. It does not depend on previous frames
Coding is performed only over the information contained in the current frame
© Rui L. Aguiar (ruilaa@det.ua.pt)

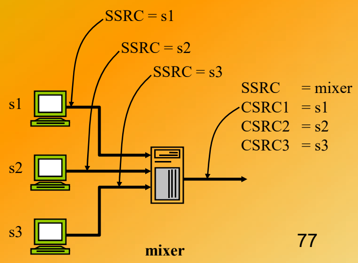
RTP: overview





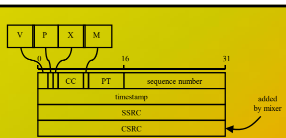
V 2-bits, version number (=2)
P 1-bit, indicates padding
X 1-bit, indicates extension header present
CC 4-bits, number of CSRCs (CSRC count)
M 1-bit, profile specific marker (defined elsewhere)
PT 7-bits, payload type, profile specific (defined elsewhere)
SSRC synchronisation source
CSRC contributing source

timestamp has profile/flow-specific units



77

RTP Header



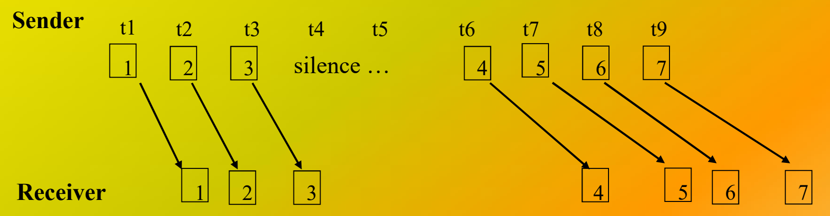
Flags	payload type	sequence number	timestamp	synchronization source identifier	misc. fields

- Flags:
  - **RC Count (CC)** – Number of CSRC identifiers (mixed)
  - **Marker (M)** – Announces frame limits: first packet of a burst in voice; end of packet sequence in video
- Payload Type (7 bits)
  - Indicates type of encoding currently being used. If sender changes encoding in middle of conference, sender informs the receiver through this payload type field
    - Payload type 0: PCM mu-law, 64 kbps
    - Payload type 3, GSM, 13 kbps
    - Payload type 7, LPC, 2.4 kbps
    - Payload type 26, Motion JPEG
    - Payload type 31. H.261
    - Payload type 33, MPEG2 video
- Sequence Number (16 bits)
  - Increments by one for each RTP packet sent, and may be used to detect packet loss and to restore packet sequence
- Timestamp field (32 bytes long)
  - Reflects the sampling instant of the first byte in the RTP data packet, in terms of sampling measurements
    - Audio: timestamp usually increases by 1 every sampling period (e.g., each 125 msecs to sampling at 8 KHz)
- SSRC field (32 bits long)
  - Identifies the source of the RTP stream. Each stream in a RTP session should have a distinct SSRC
- CSRC (*Contributing source identifiers*) – List of sources that contributed to the packet payload
  - Used when a mixer combines different packet sources
  - Receiver can then identify original sender

78

Timestamp vs sequence number

- Silence suppression and detection
- Variable length packets
- Jitter



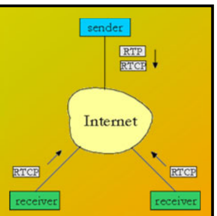
"play" time vs packet loss detection

RTP Example

- Consider sending 64 kbps PCM-encoded voice over RTP
- Application collects the encoded data in chunks, e.g., every 20 msec = 160 bytes in a chunk
- The audio chunk along with the RTP header form the RTP packet, which is encapsulated into a UDP segment
- RTP header indicates type of audio encoding in each packet
  - Sender can change encoding during a conference
- RTP header also contains sequence numbers and timestamps

universidade de aveiro

Real-Time Control Protocol (RTCP)



- Works associated to RTP, to obtain feedback information that can lead to behavior change
- Each participant in RTP session periodically transmits RTCP control packets to all other participants
- Sends all session in multicast: a multicast address per session, shared by RTP and RTCP packets
  - In different ports
  - RTCP traffic per participant is variable with time
- Each RTCP packet contains sender and/or receiver reports
  - Report statistics useful to application including number of packets sent, number of packets lost, interarrival jitter, etc...
- Essential to multicast
  - Diagnosis tool
  - Feedback control can lead to change in the sender transmission rate

83

universidade de aveiro

RTCP Protocol

- Provides information about reception quality
  - To senders and receivers
  - QoS information to the flow
    - packet info: loss, delay, jitter
    - end-system info: user info
    - application-specific or flow-specific info
- Identifies each participant
- Calculates the number of sources
- Minimum session control
  - Information about participants
  - Session leave, ...
  - Minimum synchronization
- Protocol “Announce-Listen”, *soft-state*
  - Good for scalability

84

RTCP Scalability

- Uses 5% of the session bandwidth
  - Shared by all participants
  - 25% for senders
  - 75% for receivers
- Characteristics for scalability
  - Reports periodically sent with a variable delay
    - Information sent with different periodicity (e.g. name vs email)
  - Period based on the number of participants
    - Counting of the participants number, through the source identifier

85

Types of RTCP packets

- Sender report (SR): sending of statistics by senders
  - SSRC of the RTP stream, the current time, the number of packets sent, and the number of bytes sent.
- Receiver Report (RR): sending of statistics by receivers
  - Fraction of packets lost, last sequence number, average interarrival jitter
- Source Description (SDES): CNAME, NAME, e-mail, ...
  - Sender e-mail address, sender's name, SSRC of associated RTP stream.
  - Provide mapping between the SSRC and the user/host name.
- BYE: Leaving the session
- APP: Specific for each application
- It is common the concatenation of PDUs: at least two should be sent in each UDP message
  - Mixers and translators also concatenate packets

86

universidade de aveiro

Sources (streams) synchronization

- RTP used to synchronize different streams
  - Consider videoconferencing application for which each sender generates one RTP stream for video and one for audio
    - Timestamps in RTP packets tied to the video and audio sampling clocks
    - Not tied to the wall-clock time
- RTCP reports have:
  - Timestamp in the last RTP packet
  - Time (wall clock) of the packet generation
- Receivers can use this information to synchronize different media sources (audio/video)
  - RTP timestamp value is centered in the sampling rates and not in the transmission time

92

universidade de aveiro

RTP limitations

- RTP standardizes and makes easier the transmission of continuous audio and video flows, but:
  - Does not reserve resources
  - Does not have QoS guaranties
  - Does not support congestion control
  - Does not support reliability
  - ...
- ...should work together with other protocols (RSVP) and networks (ATM) for QoS guaranties
  - An essential aspect for these flows
- Routers do not “see” RTP
  - They cannot provide priviledge services
- Scalability problems:
  - When many receivers get in the sessions simultaneously (many reports, not aggregated)

93

VoIP
Voice (and Video and …) over IP

94

Overview recall: Voice over IP

- Network loss: IP datagram lost due to network congestion (router buffer overflow).
- Delay loss: IP datagram arrives too late for playout at receiver.
  - Delays: processing, queueing in network; end-system (sender, receiver) delays.
  - Typical maximum tolerable delay: 400 ms.
- Loss tolerance: depending on voice encoding, packet loss rates between 1% and 10% can be tolerated.
- Speaker's audio: alternating talk/speech with silent periods.
  - 64 kbps during talk/speech.
  - Packets generated only during talk/speech.
    - 20 msec chunks at 8 Kbytes/sec: 160 bytes data.
- Requires session establishment.
- VoIP protocols/frameworks:
  - Session Initiation Protocol (SIP)
    - Session Description Protocol (SDP)
  - H.323
- VoIP and PSTN interoperability in large/ISP scalable scenarios require complex control frameworks:
  - Media Gateway Controller Protocol (MGCP);
  - H.248/Megaco.

96

SIP vs H.323

- SIP comes from IETF: Borrows much of its concepts from HTTP.
- H.323 is another signaling protocol for real-time, interactive.
  - Comes from the ITU (telephony).
- SIP has a Web flavor, whereas H.323 has a telephony flavor.
- SIP is a single component. Works with RTP, but it can be combined with other protocols and services.
- H.323 is a complete, vertically integrated suite of protocols for multimedia conferencing: signaling, registration, admission control, transport and codecs.

97

Learning outcomes

- Understand the scope of VoIP models
- Describe RTP operation
- Understand the SIP and H.323 protocols
- Describe architectures for interconnecting POTS and the Internet.

98

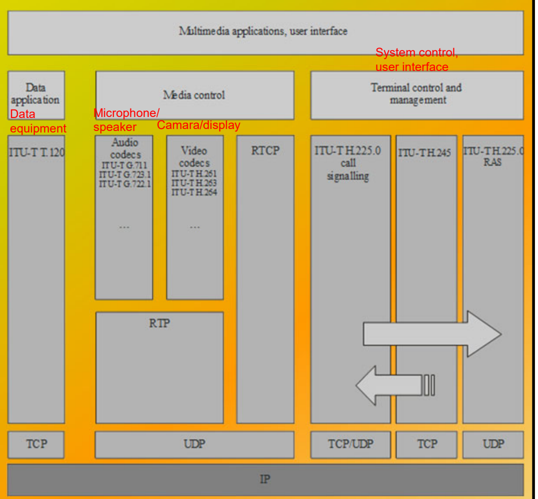
universidade de aveiro

H.323

universidade de aveiro

H.323

- H.323 is a set of recommendations from the International Telecommunication Union (ITU).
  - Contains several standards (signaling, control, transport, etc...).
- Consists of family of protocols that are
  - used for call set-up, call termination, registration, authentication and other functions.
  - transported over TCP or UDP protocols.

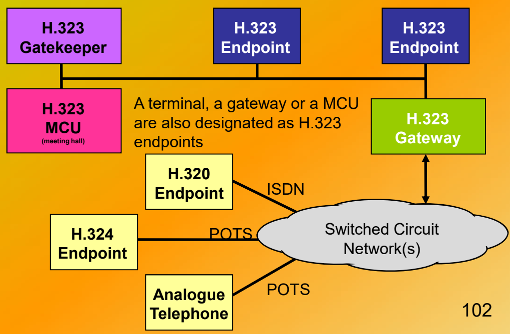


100

universidade de aveiro

H.323

- Is a set of protocols for multimedia communications
- Can be used in any IP environment
- Contains several standards (signaling, control, transport, etc…)
- Defines everything, from codecs to the control of a remote camera

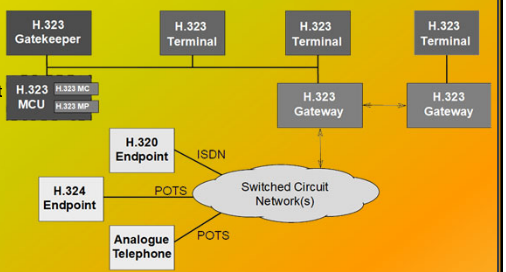


102

universidade de aveiro

H.323 Elements

- Terminal
  - Endpoint on the network which provides for real-time, two-way communications with another H.323 terminal, Gateway, or Multipoint Control Unit.
  - This communication consists of control, indications, audio, video, and/or data between the two endpoints.
- Gateway (GW)
  - Endpoint on the network which provides for real-time, two-way communications between Terminals on the packet-based network and other Terminals on a switched circuit network or to another H.323 Gateway.
- Gatekeeper (GK)
  - Entity on the network that provides address translation and controls access to the network for H.323 terminals, Gateways and MCUs.
  - The Gatekeeper may also provide other services to the terminals, Gateways and MCUs such as bandwidth management and locating Gateways.

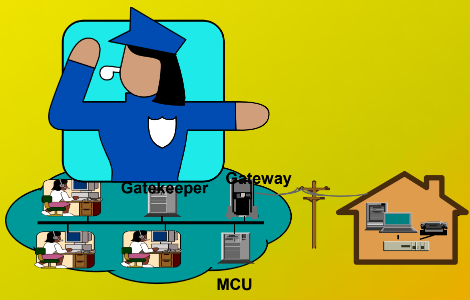


- Multipoint Control Unit (MCU)
  - Endpoint on the network which provides the capability for three or more terminals and Gateways to participate in a multipoint conference.
  - Consists of two parts: a mandatory Multipoint Controller (MC) and optional Multipoint Processors (MP).
  - Multipoint Controller (MC)
    - Entity on the network which provides for the control of three or more terminals participating in a multipoint conference.
  - Multipoint Processor (MP)
    - Entity on the network that provides centralized processing of audio, video and/or data streams in a multipoint conference.

103

universidade de aveiro

Gatekeeper



- Main point of network management (in a zone)
  - Address translation (IP/telephone)
  - Access control of network resources to terminals, gateways and MCU's
  - Call management
  - Bandwidth control
- Every endpoints of an H.323 network should be registered in the respective gatekeeper

105

universidade de aveiro

Gatekeeper in an H.323 system

- Gatekeeper is optional
  - When present, can provide a set of functionalities
    - Routing of call signalling (better control, intelligent routing decisions, load balacing of gateways)
    - However, these messages can be sent directly between users
- H.323 networks with IP/PSTN gateways should contain a gatekeeper to make address translation
- Mandatory functions
  - Address translation, admission and bandwidth control, zone management
- Optional functions
  - Call control signalling, call authorization and management

106

universidade de aveiro

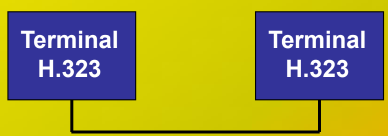
Multipoint Control Unit

- Supports the required functionalities for three or more terminals and gateways to participate in a multi-point session
  - Multipoint Controller (MC)
    - Signalling and session control
  - Multipoint Processor (MP)
    - Processing (multiplexing and sending) of multimedia flows.
- MC e MP
  - Centralized multi-point session
    - Signalling, control, and multimedia data information traverse the MCU
- Only MC
  - Descentralized multi-point session
    - Only signalling and control information traverse the MCU

107

universidade de aveiro

H.323 Terminals



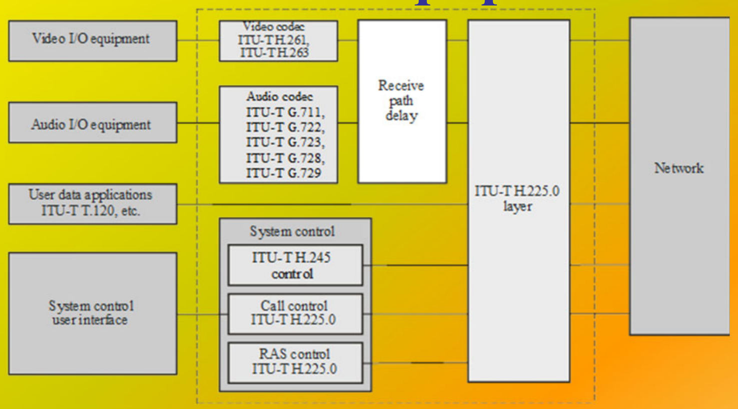
•An H.323 terminal is a network endpoint that can have bidirectional real time communications with another H.323 terminal, gateway or MCU
- •Signaling information, control, audio, video and/or data
- •Hardware or software that implements H.323 terminal functionalities.

H.323 terminals support
- G.711 (PCM audio codec, 64Kbps)
- H.245 (media identification and control, such as voice and video)
- Q.931 (signaling for call establishment)
- Registration/Admission/Status (RAS) Channel (data channel to communicate with the Gatekeeper – it can or not exist)
- RTP/RTCP

108

universidade de aveiro

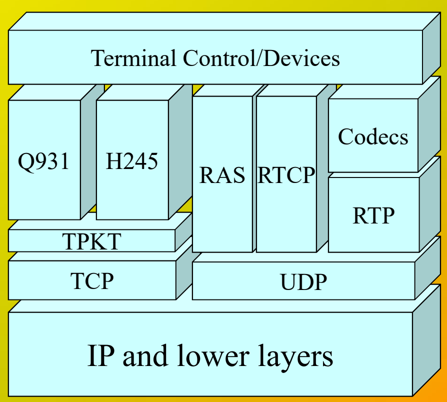
H.323 Terminal Equipment stack



- H.225
  - Registration, Admission and Status (RAS), which is used between an H.323 endpoint and a Gatekeeper to provide address resolution and admission control services.
  - Call Signaling, which is used between any two H.323 entities in order to establish communication (based on Q.931/Q.932).
- H.245
  - Control protocol for multimedia communication, which describes the messages and procedures used for capability exchange, opening and closing logical channels for audio, video and data, control and indications.

universidade de aveiro

H.323 protocol stack



- **Q931**
  - Call signalling
- **H245**
  - Creation of channels and exchange of capacities
- **RAS (registration, admission and status), H225.**
  - Of terminals+gateways to gatekeepers
  - User registration
- **Codecs:**
  - Audio (G.711, G722,23.1,28,29)
  - Video (H.261, H.263)
- **Data:**
  - T.12x

H.323 operation (protocols)

- Obtain gatekeeper permission ***(RAS Admission Request)***
- Find the address of the user to call ***(RAS Address resolution)***
- Press the number (call) ***(Q931 call setup)***
- Tell the partners what languages it understands/talks ***(H245 capability negotiation – Set, Ack, Reject)***
- Wait for the communication of its capabilities ***(H245 capability negotiation – Set, Ack, Reject)***
- Inform what languages will be used during the conversation ***(H245 Logical channel signaling; languages=codecs)***
- Start talking (and listening) ***(Data transfer with RTP/RTCP)***
- Upon termination, say Bye ***(H245 end session)***
- Disconnect ***(Q931 call termination, release complete)***
- Inform the gatekeeper that the call ended ***(RAS Disengage Request)***

111

H.323 operation (more...)

- Multiples languages can be used during the communication
- The language can be changed during conversation, as long as the other partner understands it
  - An explicit announcement has to be done
- Say Bye before terminating is optional….
- ...

112

H.225 RAS Messages

gatekeeper discovery and registration

- Gatekeeper discovery:
  - **Gatekeeper Request (GRQ), Gatekeeper Confirm (GCF)** and **Gatekeeper Reject (GRJ)**
  - If one gatekeeper answers positively, the endpoint should select which one to use.
- Endpoints registration:
  - **Registration Request (RRQ)** and **Unregistration Request (URQ)**
- Endpoints location:
  - **Location Request (LRQ), Location Confirm (LCF)** and **Location Reject (LRJ)**
  - Through the alias of another endpoint, it can obtain contact information of that endpoint.
- Admission to participate in a session:
  - **Admission Request (ARQ), Admission Confirmation (ACF) and Admission Reject (ARJ)**
- Change of bandwidth by an endpoint or gatekeeper
  - **Bandwidth Request (BRQ), Bandwidth Confirm (BCF)** and **Bandwidth Request (BRJ)**
- State information of an endpoint:
  - **Information Request (IRQ)** and **Information Request Response (IRR)**
- Session leave:
  - **Disengage Request (DRQ), Disengage Confirm (DCF**) and **Disengage Reject (DRJ)**
- Communication of available resources - gateways should inform gatekeepers about its capacities:
  - **Resource Available Indicate (RAI)** and **Resource Available Confirmation (RAC)**

113

H.225 Call Signaling Q.931 Messages

- Call establishment messages:
  - **Setup, Setup Acknowledge, Alerting**, **Call Proceeding**, **Connect**, **Connect Acknowledge**, and **Progress**.
- Call Clearing messages:
  - **Disconnect**, **Release,** and **Release Complete**.
- Call Information Phase messages:
  - **Resume**, **Resume Acknowledge**, **Resume Reject**, **Suspend**, **Suspend Acknowledge**, **Suspend Reject**, and **User Information.**
- Miscellaneous messages:
  - **Congestion Control**, **Information**, **Notify**, **Status**, and **Status Inquiry**.
- Q.932/H.450 messages:
  - **Facility**, **Hold**, **Hold Acknowledge**, **Hold Reject**, **Retrieve**, **Retrieve Acknowledge**, and **Retrieve Reject**.

114

universidade
de aveiro

H.225 Call Signaling (most common)

- **Setup** - Establish a session between endpoints.
- **Call Proceeding** (optional) - answer to a setup indicating that it received the establishment process of the running session.
- **Alerting** - message sent by a callee to indicate that the user was already notified (corresponds to the phone ringing).
- **Progress** - optional message sent by a gateway to indicate that the session is in progress.
- **Connect** - message sent by a callee that indicates session acceptation.
- **Release Complete** - message sent by an endpoint to terminate a session.
- **Facility** - message sent by an endpoint to another one to inform where to redirect the session (other information can be sent)
- **Notify** - optional message used by any H.323 entity to send information to another one.
- **Status Inquiry** - message used by an endpoint during a session lifetime to ask another one about its status.
- **Status** - message used to answer to a status inquiry message.

115

universidade
de aveiro

Q.931 Call Signaling – establish, control and terminate connections

- *Setup* – Establish a session between endpoints
- *call proceeding* (optional) – answer to a setup indicating that it received the establishment process of the running session
- *Alerting* – message sent by a callee to indicate that the user was already notified (corresponds to the phone ringing)
- *progress* – optional message sent by a gateway to indicate that the session is in progress
- *connect* – message sent by a callee that indicates session acceptation
- *release complete* – message sent by an endpoint to terminate a session
- *facility* - message sent by an endpoint to another one to inform where to redirect the session (other information can be sent)
- *notify* - optional message used by any H.323 entity to send information to another one
- *status inquiry* – message used by an endpoint during a session lifetime to ask another one about its status
- *status* – message used to answer to a *status inquiry* message

116

universidade de aveiro

H.245 Control Messages

- Capacities and preferences negotiation of each participant entity
- Signalling of logical channels used for data communication
- Used after the exchange of **Setup** and **Connect** messages to open an H.245 control channel.
- Capacities negotiation (supported formats for sending and reception):
  - **terminalCapabilitySet, terminalCapabilitySetAck, terminalCapabilitySetReject**
- Master/slave determination to solve conflicts that may appear during a session lifetime:
  - **masterSlaveDetermination, masterSlaveDeterminationAck, masterSlaveDeterminationReject**
- Opening of logical channels for several flows:
  - **openLogicalChannel, openLogicalChannelAck, openLogicalChannelConfirm, openLogicalChannelReject**
- Closing of logical channels:
  - **closeLogicalChannel, closeLogicalChannelAck, requestChannelClose, requestLogicalChannelAck, requestLogicalChannelReject**
- When all logical channels are closed, the session can be terminated:
  - **endSession**

117

universidade de aveiro

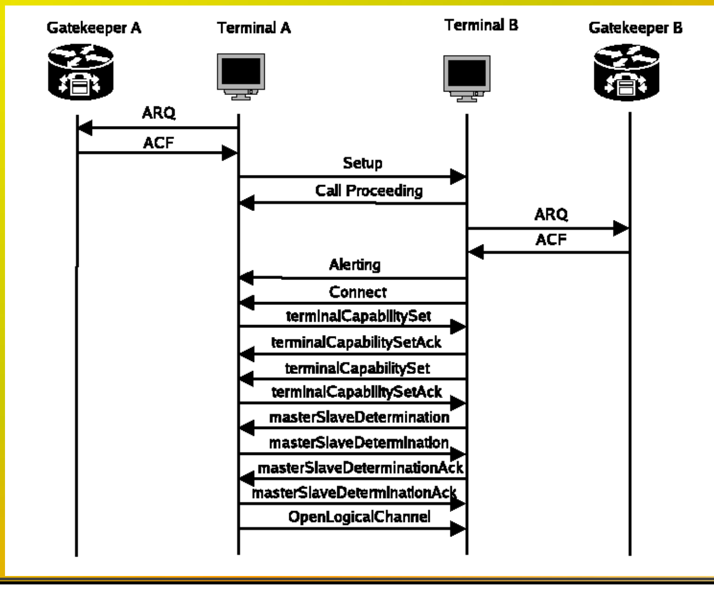
ITU Recommendations

	H.320	H.321	H.322	H.323v1 H323v2	H.324
Network	Narrowband ISDN	Broadband ISDN/ATM	Guaranteed B/W	Non-Guaranteed B/W	PSTN/POTS
Approval	1990	1995	1995	1996/1998v2	1996
Audio	G.711 G.722 G.728	G.711 G.722 G.728	G.711 G.722 G.728	G.711 G.722 G.728 G.723.1 G.729, 729A	G.723
Video	H.261 H.263	H.261 H.263	H.261 H.263	H.261 H.263	H.261 H.263
Data	T.120	T.120	T.120	T.120	T.120
Control	H.230 H.243	H.242	H.230 H.242	H.245	H.245

118

universidade de aveiro

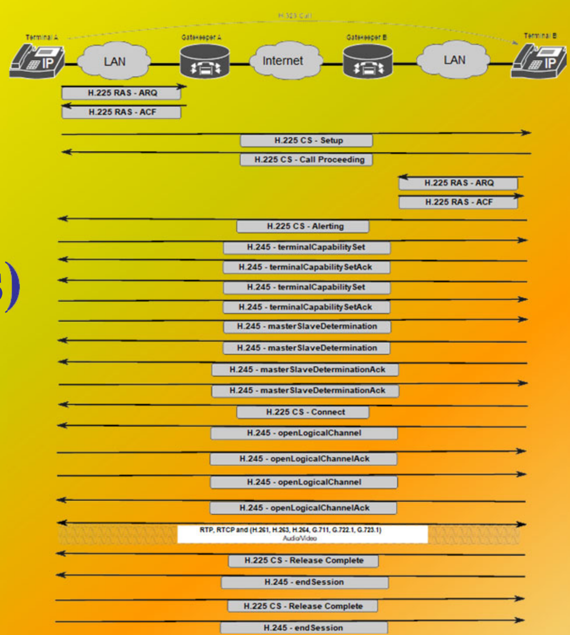
Establishment of an H.323 connection



119

universidade de aveiro

H.323 Call (with Gatekeepers)

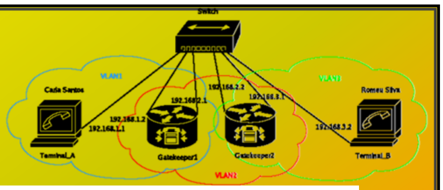


- Multiple messages may be transported in the same IP packet.

120

universidade de aveiro

H.323 Session



No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.1.1	255.255.255.2	H.225.	RAS: gatekeeperRequest
2	0.001943	192.168.1.2	192.168.1.1	H.225.	RAS: gatekeeperConfirm
3	0.011375	192.168.1.1	192.168.1.2	H.225.	RAS: registrationRequest
4	0.014833	192.168.1.2	192.168.1.1	H.225.	RAS: registrationConfirm
5	24.33812	192.168.1.1	192.168.1.2	H.225.	RAS: admissionRequest
6	24.34116	192.168.1.2	192.168.1.1	H.225.	RAS: admissionConfirm
7	24.34617	192.168.1.1	192.168.1.2	H.225.	CS: setup
8	24.36137	192.168.1.2	192.168.1.1	H.225.	CS: callProceeding
9	24.39984	192.168.1.1	192.168.1.2	H.225.	RAS: infoRequestResponse
10	24.90758	192.168.1.2	192.168.1.1	H.225.	CS: connect
11	24.92385	192.168.1.1	192.168.1.2	H.245	TerminalCapabilitySet
12	24.93220	192.168.1.1	192.168.1.2	H.245	MasterSlaveDetermination
13	25.00825	192.168.1.2	192.168.1.1	H.245	TerminalCapabilitySet
14	25.01673	192.168.1.2	192.168.1.1	H.245	MasterSlaveDetermination
15	25.02929	192.168.1.2	192.168.1.1	H.245	TerminalCapabilitySetAck
16	25.03250	192.168.1.2	192.168.1.1	H.245	MasterSlaveDeterminationAck
17	25.04127	192.168.1.1	192.168.1.2	H.245	TerminalCapabilitySetAck
18	25.04409	192.168.1.1	192.168.1.2	H.245	MasterSlaveDeterminationAck
19	25.06737	192.168.1.1	192.168.1.2	H.245	OpenLogicalChannel (gsmFullRate)
20	25.27152	192.168.1.2	192.168.1.1	H.245	[TCP ACKed lost segment] OpenLogicalChannel
21	25.29772	192.168.1.1	192.168.1.2	H.245	OpenLogicalChannelAck
22	25.30046	192.168.1.2	192.168.1.1	H.245	OpenLogicalChannelAck
23	25.40158	192.168.1.1	192.168.3.2	RTP	Payload type=GSM 06.10, SSRC=3780823517, Se
24	25.48155	192.168.1.1	192.168.3.2	RTP	Payload type=GSM 06.10, SSRC=3780823517, Se
25	25.60718	192.168.3.2	192.168.1.1	RTP	Payload type=ITU-T G.711 PCMU, SSRC=3097049
26	25.60762	192.168.3.2	192.168.1.1	RTP	Payload type=ITU-T G.721, SSRC=3097049058,
27	31.57993	192.168.1.1	192.168.1.2	H.245	[TCP ACKed lost segment] EndSessionCommand
28	31.58027	192.168.1.1	192.168.1.2	H.225.	CS: releaseComplete
29	31.58559	192.168.1.2	192.168.1.1	H.245	EndSessionCommand
30	31.68600	192.168.1.1	192.168.1.2	H.225.	RAS: disengageRequest
31	31.68737	192.168.1.2	192.168.1.1	H.225.	RAS: disengageConfirm

21

universidade de aveiro

H.323 Direct Call

- Multiple messages may be transported in the same IP packet.

122

universidade de aveiro

H.323 Direct Call

Source	Destination	Protocol	Length	Info
192.168.56.1	192.168.56.101	H.225.0/H	155	masterSlaveDetermination terminalCapabilitySet CS: setup OpenLogicalChannel
192.168.56.101	192.168.56.1	H.225.0	181	CS: callProceeding
192.168.56.101	192.168.56.1	H.225.0/H	460	masterSlaveDeterminationAck terminalCapabilitySetAck terminalCapabilitySet CS: empty
192.168.56.101	192.168.56.1	H.225.0	181	CS: alerting
192.168.56.1	192.168.56.101	H.225.0/H	109	masterSlaveDeterminationAck terminalCapabilitySetAck CS: empty
192.168.56.101	192.168.56.1	H.225.0/H	106	roundTripDelayRequest CS: empty
192.168.56.1	192.168.56.101	H.225.0/H	106	roundTripDelayResponse CS: empty
192.168.56.101	192.168.56.1	H.225.0	360	CS: connect OpenLogicalChannel
192.168.56.101	192.168.56.1	H.225.0/H	131	endSessionCommand CS: releaseComplete
192.168.56.1	192.168.56.101	H.225.0/H	131	endSessionCommand CS: releaseComplete

Source	Destination	Protocol	Length	Info
192.168.56.101	192.168.56.1	H.261	1023	H.261 message
192.168.56.101	192.168.56.1	H.261	1021	H.261 message
192.168.56.101	192.168.56.1	H.261	353	H.261 message
192.168.56.1	192.168.56.101	RTP	106	PT=DynamicRTP-Type-112, SSRC=0x8A1B4D04, Seq=48411, Time=0, Mark
192.168.56.1	192.168.56.101	RTP	106	PT=DynamicRTP-Type-112, SSRC=0x8A1B4D04, Seq=48412, Time=320
192.168.56.1	192.168.56.101	RTP	106	PT=DynamicRTP-Type-112, SSRC=0x8A1B4D04, Seq=48413, Time=640
192.168.56.101	192.168.56.1	H.261	336	H.261 message
192.168.56.101	192.168.56.1	RTP	214	PT=ITU-T G.722, SSRC=0x56D0B555, Seq=6053, Time=0, Mark
192.168.56.101	192.168.56.1	RTP	214	PT=ITU-T G.722, SSRC=0x56D0B555, Seq=6054, Time=160
192.168.56.101	192.168.56.1	RTP	214	PT=ITU-T G.722, SSRC=0x56D0B555, Seq=6055, Time=320
192.168.56.1	192.168.56.101	RTP	106	PT=DynamicRTP-Type-112, SSRC=0x8A1B4D04, Seq=48414, Time=960
192.168.56.101	192.168.56.1	RTP	214	PT=ITU-T G.722, SSRC=0x56D0B555, Seq=6056, Time=480
192.168.56.101	192.168.56.1	H.261	386	H.261 message
192.168.56.1	192.168.56.101	H.261	1023	H.261 message
192.168.56.1	192.168.56.101	RTP	106	PT=DynamicRTP-Type-112, SSRC=0x8A1B4D04, Seq=48415, Time=1280
192.168.56.101	192.168.56.1	RTP	214	PT=ITU-T G.722, SSRC=0x56D0B555, Seq=6057, Time=640
192.168.56.101	192.168.56.1	RTP	214	PT=ITU-T G.722, SSRC=0x56D0B555, Seq=6058, Time=800
192.168.56.1	192.168.56.101	RTP	106	PT=DynamicRTP-Type-112, SSRC=0x8A1B4D04, Seq=48416, Time=1600
192.168.56.101	192.168.56.1	H.261	346	H.261 message
192.168.56.101	192.168.56.1	RTP	214	PT=ITU-T G.722, SSRC=0x56D0B555, Seq=6059, Time=960
192.168.56.1	192.168.56.101	H.261	1021	H.261 message
192.168.56.1	192.168.56.101	RTP	106	PT=DynamicRTP-Type-112, SSRC=0x8A1B4D04, Seq=48417, Time=1920
192.168.56.101	192.168.56.1	RTP	214	PT=ITU-T G.722, SSRC=0x56D0B555, Seq=6060, Time=1120

universidade de aveiro

Advantages and disadvantages

+ Works!
+ There are many implementations... some are free...
+ Supports many languages (codecs).
+ Interoperates with other languages: H320 (ISDN), H324 (POTS)
+ Good role in a specific transition period of coexistence

- Very complicated…
- Many protocols can be combined in the H.323 environment... It implies technology redundancy….
- Problems when Gatekeeper is overloaded and MCU is full!
- Firewalls ? ... Difficult to develop (multiple ports have to be managed in a conversation).