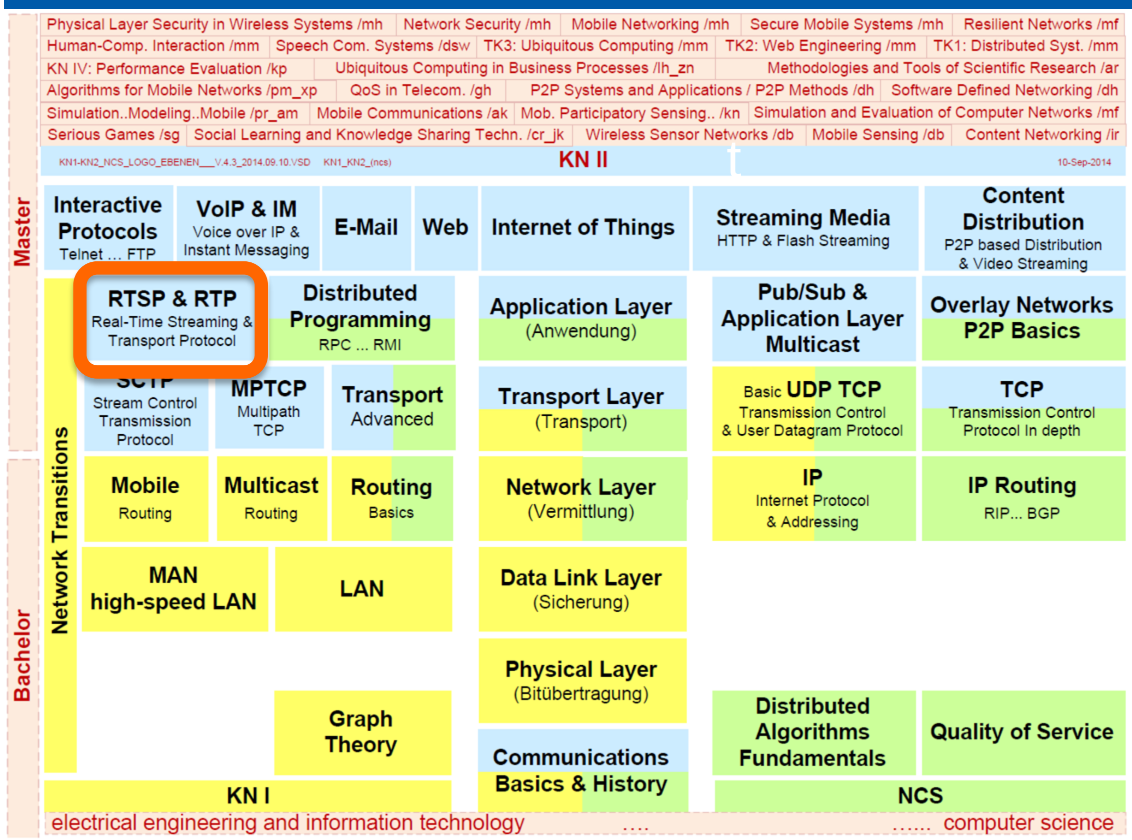
Communication Networks II

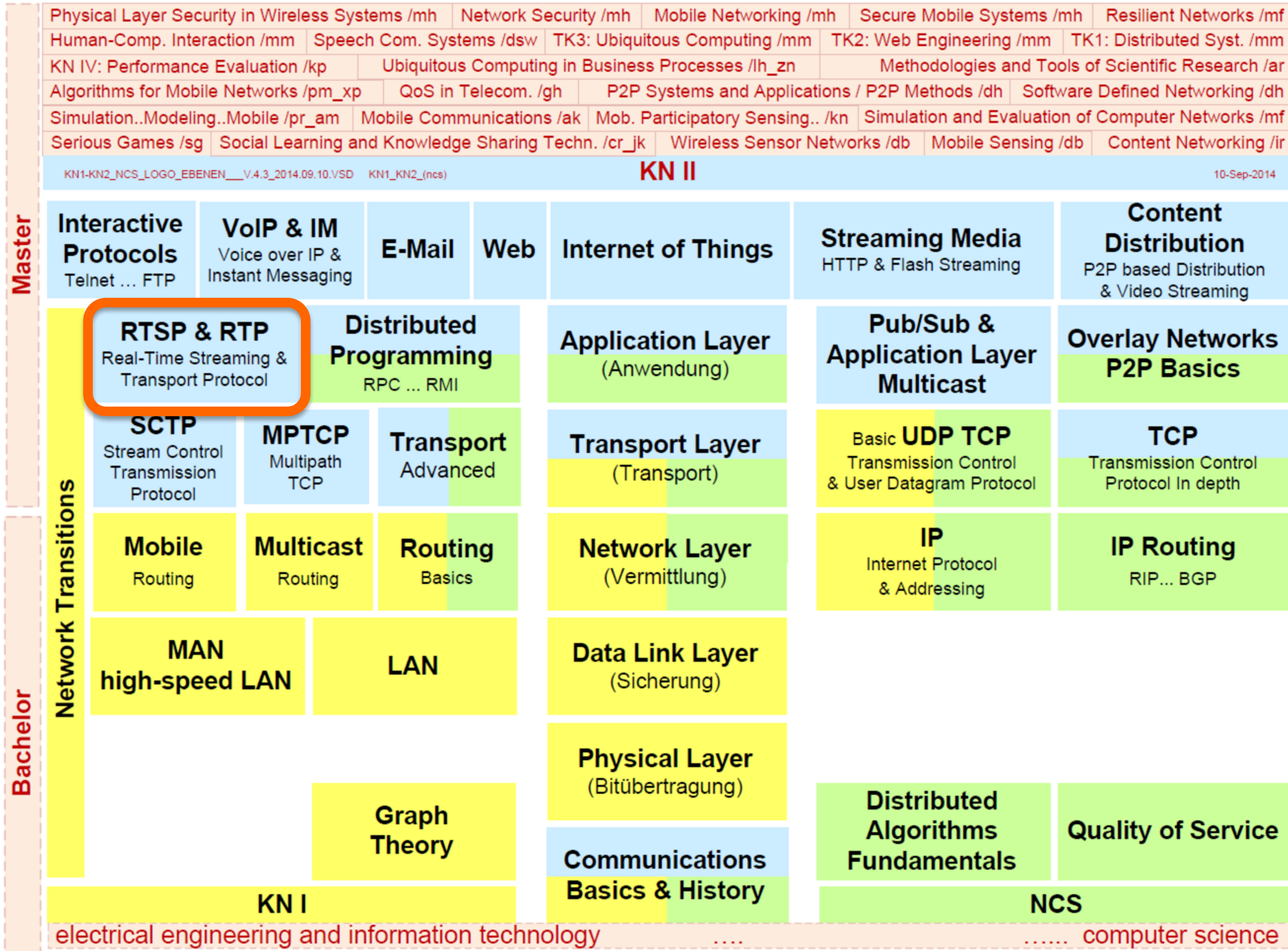
Real-Time and Multimedia Protocols of the Internet
Real-Time Streaming Protocol (RTSP)
Real-Time Transport Protocol (RTP)



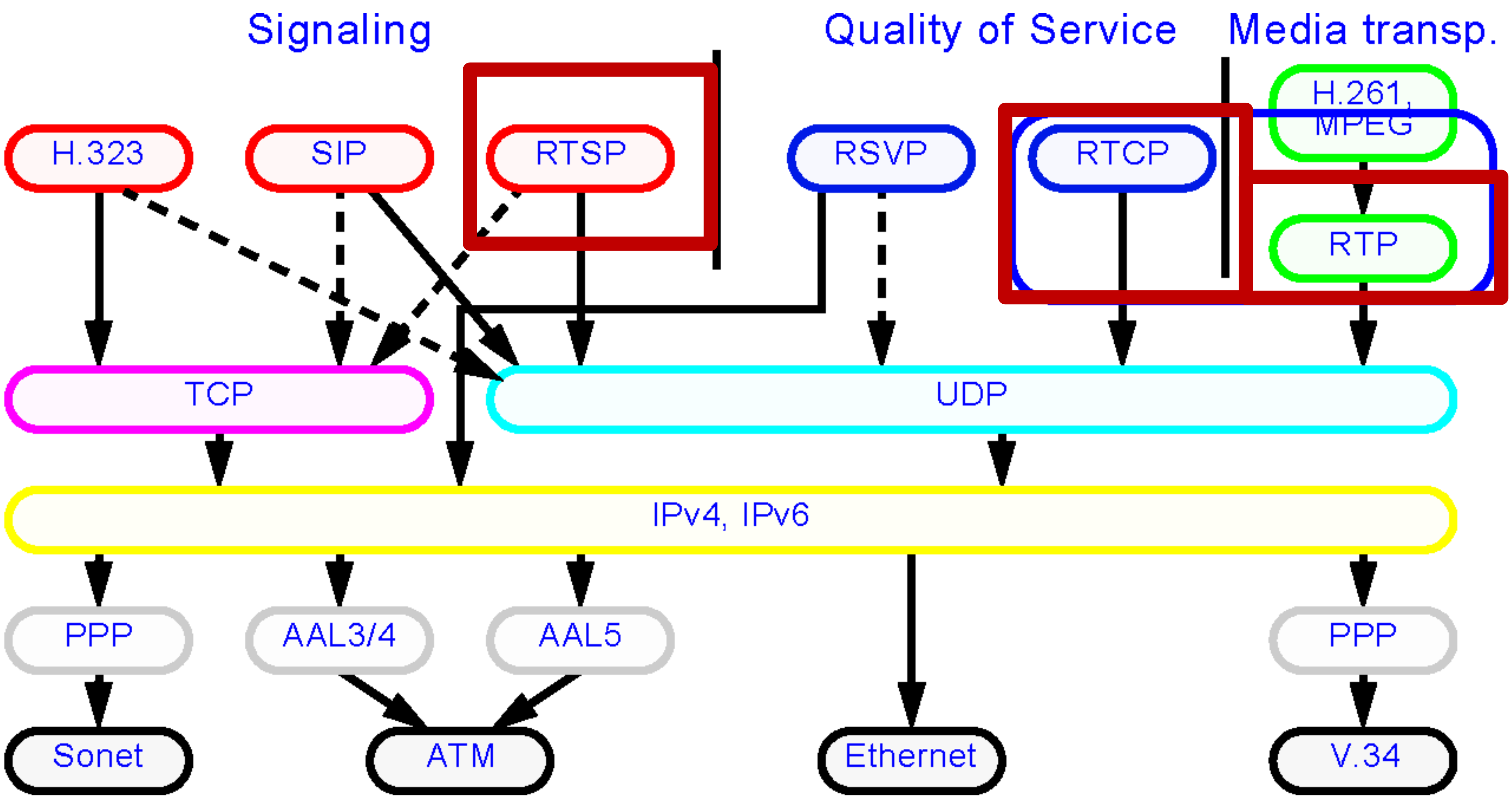
Prof. Dr.-Ing. **Ralf Steinmetz**
KOM - Multimedia Communications Lab

Overview

TECHNISCHE UNIVERSITÄT DARMSTADT



10. September 2014


Signaling
Quality of Service
Media transp.
H.323
SIP
RTSP
RSVP
RTCP
H.261, MPEG
RTP
TCP
UDP
IPv4, IPv6
PPP
AAL3/4
AAL5
PPP
Sonet
ATM
Ethernet
V.34

Real-Time

Real-time networked System:

“A networked system in which the correctness of a computation depends not only on obtaining the right result, but also upon providing the result on time.”

Real-time Process:

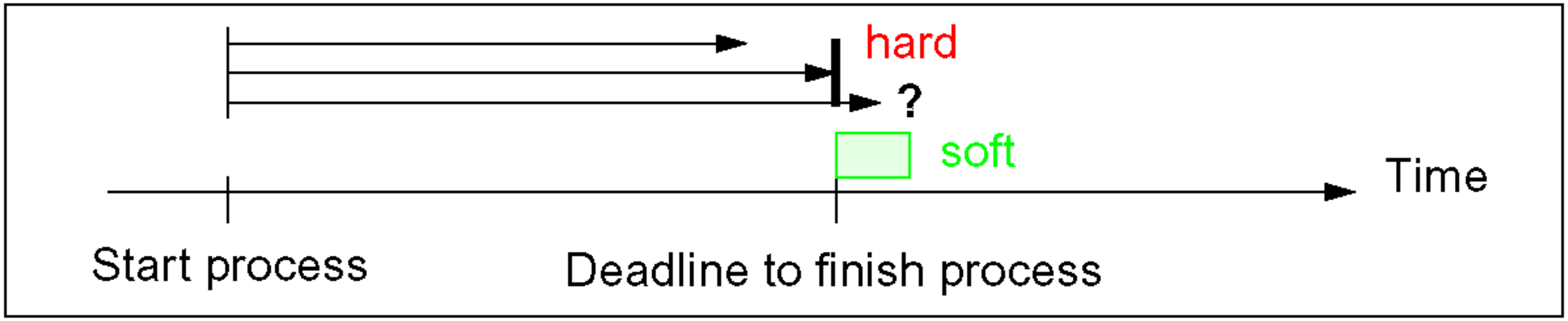
“A process which delivers the results of the processing in a given time-span.”

Real-time Application – examples

- Control of temperature in a chemical plant
  - driven by interrupts from external devices
  - these interrupts occur at irregular and unpredictable intervals
- Example: Control of a flight simulator
  - execution at periodic intervals
  - scheduled by timer-service which the application requests from the OS

- Common characteristics:
  - internal and external events that occur periodically or spontaneous
  - correctness also depends on meeting time constraints !

Deadlines in Real-Time (networked) Systems



Hard deadlines:

- should never be violated
- result presented too late after deadline has no value for the user
- violation means:
  severe (potentially catastrophic) system failure
- Example:
  - fly by wire,
  - nuclear power plant

Soft deadlines:

- deadlines are not missed by much
- in some cases the deadline may be missed, but not too many deadlines are missed
- violation:
  Result(s) has/have still some value for the user/application
- Example:
  - train/plain arrival-departure

Primary goal:

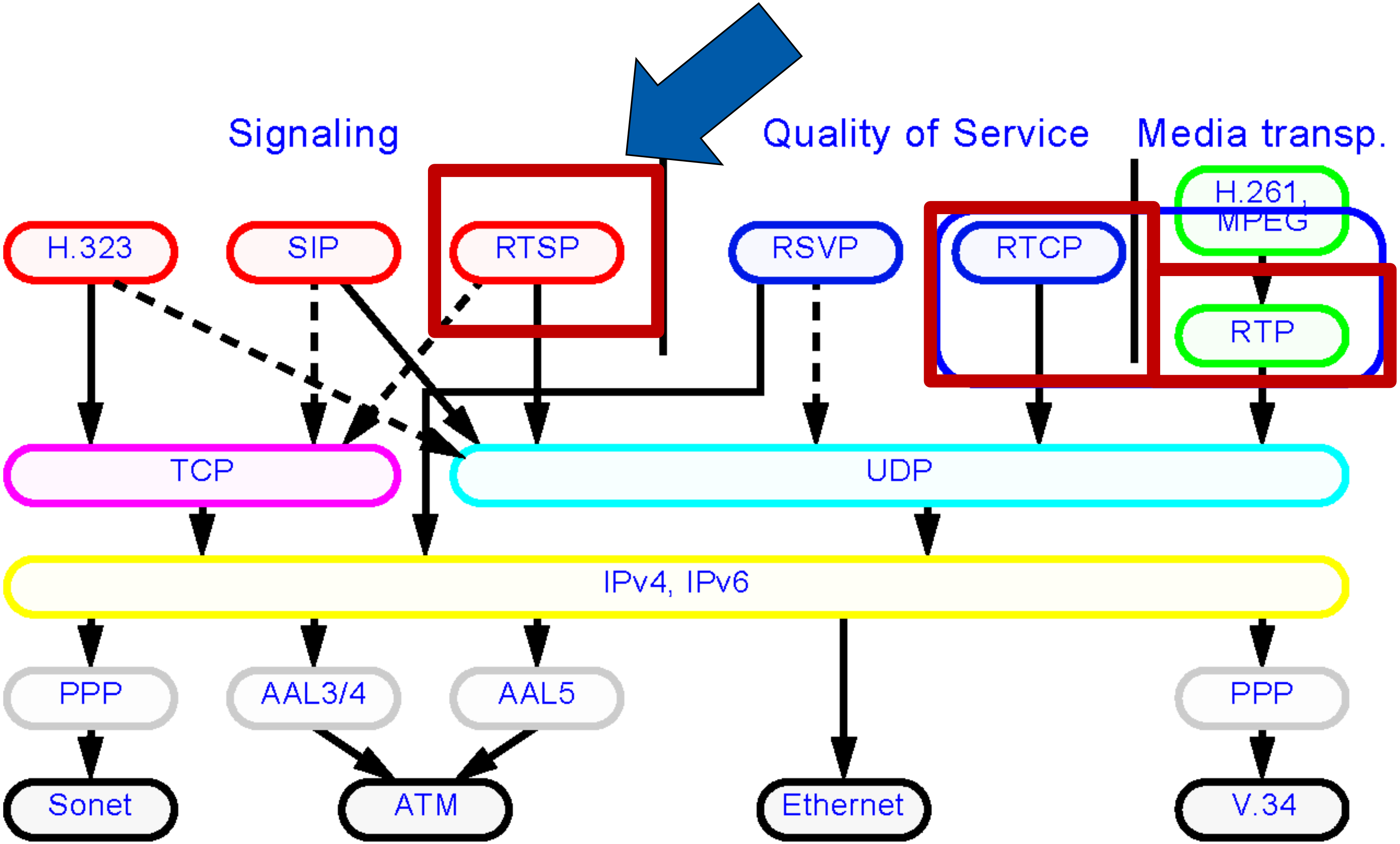
- deterministic behavior according to specification
- .. results in a variety of requirements

Mandatory requirements:

- predictable (and usually fast) handling of time-critical events
- adequate scheduling
- proper handling of overload conditions

Desirable and related requirements:

- Multi-tasking capabilities
- Short interrupt latency
- Fast context switching
- Control of memory management
- Proper scheduling
- Fine-granularity of timer services
- Rich set of interprocess communication and synchronization mechanisms


Signaling
Quality of Service
Media transp.
H.323
SIP
RTSP
RSVP
RTCP
H.261, MPEG
RTP
TCP
UDP
IPv4, IPv6
PPP
AAL3/4
AAL5
PPP
Sonet
ATM
Ethernet
V.34

History

- Specified by RFC 2326 in 1998

Goal of RTSP:

- Signaling and control of multimedia streams
- Being independent from transport of media content

Supported functions:

- To request streams from server (unicast, multicast)
- To invite a server to a conference
- To add media to a stream (record)

The Real Time Streaming Protocol, or RTSP, is an **application-level protocol** for **control** over the delivery of data with real-time properties. RTSP provides an extensible framework to enable controlled, on-demand delivery of real-time data, such as audio and video. Sources of data can include both **live data feeds** and **stored** clips. This protocol is intended to **control multiple data delivery sessions**, provide a means for choosing delivery channels such as UDP, multicast UDP and TCP, and provide a means for choosing delivery mechanisms based upon RTP.

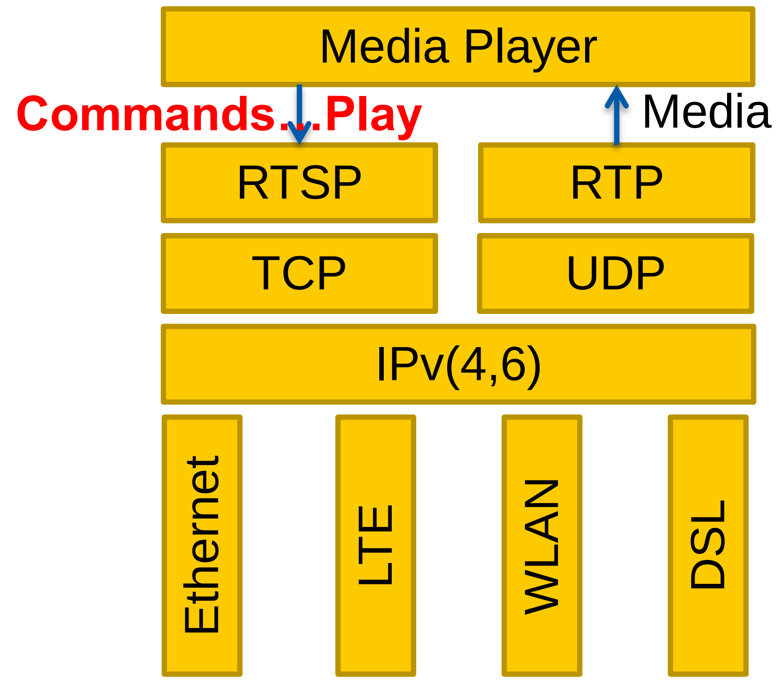
http://www.ietf.org/rfc/rfc2326.txt

Source: Slides Lecture Multimedia Kommunikation, Fakultät Informatik, TU Dresden

2.1 Properties of RTSP

Supported operations:

- Play, Pause, Resume
- Reset
- Fast forward, Fast backward
- Record



Text-based, in the style of HTTP 1.1

Differences to HTTP

- RTSP assumes a state-full server, HTTP is stateless

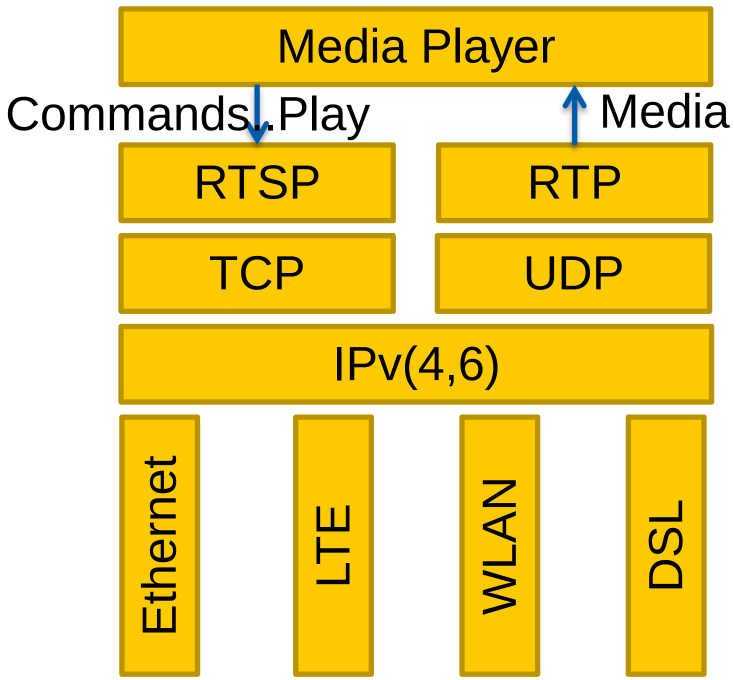
URIs

- ("rtsp:" | "rtspu:") "//" host [":" port] [abs_path]
- E.g. rtsp://media.example.com:554/twister/audiotrack

Properties of RTSP

What RTSP does NOT:

- To define compression methods for audio/video
- To define how media streams are split up in packets
- To make assumptions about the transport layer protocol being used
- To define how buffering is done by the media player



RTSP – Client-side Requests

As in traditional client/server protocol

- Client initiates session and sends requests to server
- Server executes commands and replies

Commands	Required / Optional / Recommended	Description
OPTIONS	Required	Request settings of server
SETUP	Recommended	Request description of RTSP-URI
SET_PARAMETER	Optional	Set specific parameter
PLAY	Required	Start transmission
PAUSE	Recommended	Pause transmission
RECORD	Optional	Record stream and send to server
TEARDOWN	Required	Stop transmission and tear down connection to server
…		

Source: Slides Lecture Multimedia Kommunikation, Fakultät Informatik, TU Dresden

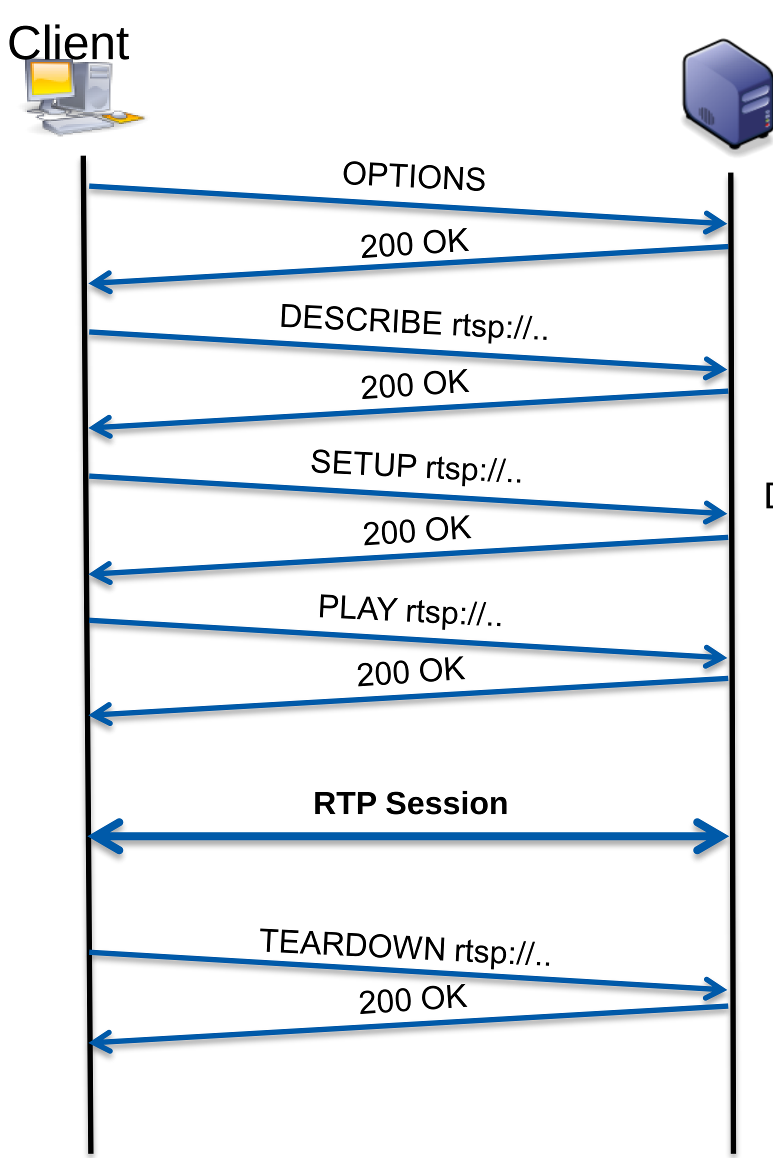
RTSP – Server Responses

Server replies with status code similar to HTTP

Satus Code	Short	Description
1xx	Information	Request has been received and will be processed
2xx	Success	Command has been executed successfully
3xx	Redirect	Resend request to the given new URL
4xx	Client-side error	Request contains error / not well formatted
5xx	Server-side error	Request correct, but cannot be executed

Source: Slides Lecture Multimedia Kommunikation, Fakultät Informatik, TU Dresden

2.2 RTSP - Session Example



Reply includes supported commands

Reply contains description of stream in SDP-format

Determine which Transport protocol to use for media stream

Reply contains start parameter for RTP-Session (stream-ID, seq no.)

Transmission of media stream within RTP Session

Teardown connection

Source: Slides Lecture Multimedia Kommunikation, Fakultät Informatik, TU Dresden

RTSP – OPTIONS Command

Client request

OPTIONS rtsp://62.201.170.10:554
RTSP/1.0\r\n
CSeq: 1\r\n
User-Agent: RealMedia Player
[…]\r\n
[…]

Server response

RTSP/1.0 200 OK\r\n
CSeq: 1\r\n
[…]
Server: Helix Server Version
9.0.7.1371\r\n **Public: OPTIONS, DESCRIBE, ANNOUNCE, PLAY, SETUP, GET_PARAMETER, SET_PARAMETER, TEARDOWN**\r\n

CSeq

- Sequence number
  for request / response pair
- Assigning responses to requests

Reply contains supported commands

Server does not change its state

RTSP – DESCRIBE Command

Client request

DESCRIBE
rtsp://62.201.170.10:554/events/[…]/20899581.rm?rmsrc=type.fd%7Cstation.rm RTSP/1.0\r\n

CSeq: 2\r\n
Accept: application/sdp\r\n
User-Agent: RealMedia Player HelixDNAClient/10.0.1.338 (win32)\r\n
Session: 809730198-1
Bandwidth: 28200400\r\n […] \r\n

Server response

RTSP/1.0 **200 OK**\r\n
CSeq: 2\r\n […]
Content-base:
rtsp://62.201.170.10:554/events/[…]/20899581.rm?rmsrc=type.fd%7Cstation.rm[…]/\r\n
ETag: 809730198-1\r\n
Session: 809730198-1
Content-type: **application/sdp**
Content-length: 2840
[…] - SDP

Accept

- Type of data that can be handled
- Session ID
- Bandwidth information

Reply contains

- Content type
- Content length

RTSP – SETUP Command

TECHNISCHE UNIVERSITÄT DARMSTADT

Client request

SETUP
rtsp://62.201.170.10:554/events/[…]/20899581.rm
RTSP/1.0\r\n
CSeq: 3\r\n
[…]
Transport: x-pn-tng/tcp;mode=play, x-real-rdt/tcp;mode=play, RTP/AVP/TCP;unicast;mode=play
[…]

Server response

RTSP/1.0 **200 OK**\r\n
CSeq: 3\r\n
Session: 809730200-1
[…]
Transport: **x-pn-tng/tcp;interleaved=0** \r\n

Provide several possibilities for transporting content to client

- TCP
- RDT: Real Data Transport (similar to RTP)

Server decides about which possibility to use

In this example:

- Transmit content via TCP over the same connection used by RTSP

RTSP – PLAY Command

TECHNISCHE UNIVERSITÄT DARMSTADT

Client request

PLAY
rtsp://62.201.170.10:554/events/[…]/20899581.rm
RTSP/1.0\r\n

CSeq: 6\r\n
User-Agent: RealMedia Player
HelixDNAClient/10.0.1.338 (win32)\r\n
Session: 809730200-1

Range: npt=0-202.106000\r\n
Bandwidth: 14869375\r\n \r\n

Server response

RTSP/1.0 **200 OK**\r\n
CSeq: 6\r\n
Session: 809730200-1 \r\n

Range to be played is requested

Allows to jump to an arbitrary position within the content

Pause playback with PAUSE

RTSP – Media Session

RTSP	Reply: RTSP/1.0 200 OK
RTSP	SET_PARAMETER rtsp://62.201.170.10:554/events/240826
RTSP	Reply: RTSP/1.0 200 OK
RDT	LATENCY-REPORT: t=0 DATA: stream-id=00 asm-rule=10
RDT	DATA: stream-id=00 asm-rule=11 seq=00001 ts=00116
RDT	DATA: stream-id=00 asm-rule=11 seq=00002 ts=00232
RDT	DATA: stream-id=00 asm-rule=11 seq=00004 ts=00464
RDT	DATA: stream-id=00 asm-rule=11 seq=00005 ts=00580
RDT	DATA: stream-id=00 asm-rule=11 seq=[illegible] ts=[illegible]
RDT	DATA: stream-id=00 asm-rule=11 seq=00007 ts=00813
RDT	DATA: stream-id=00 asm-rule=11 seq=00008 ts=00929
RDT	DATA: stream-id=00 asm-rule=11 seq=00009 ts=01045
RDT	DATA: stream-id=00 asm-rule=11 seq=00010 ts=01161
RDT	DATA: stream-id=00 asm-rule=11 seq=00012 ts=01393
RDT	DATA: stream-id=00 asm-rule=11 seq=00013 ts=[illegible]
RDT	DATA: stream-id=00 asm-rule=11 seq=00014 ts=01625
RDT	DATA: stream-id=00 asm-rule=11 seq=00015 ts=01741
RDT	DATA: stream-id=00 asm-rule=10 seq=00016 ts=01858
RDT	DATA: stream-id=00 asm-rule=11 seq=00017 ts=01974
RDT	DATA: stream-id=00 asm-rule=11 seq=00019 ts=02206
RDT	DATA: stream-id=00 asm-rule=11 seq=00020 ts=02322
RDT	DATA: stream-id=00 asm-rule=11 seq=00021 ts=02438
RDT	DATA: stream-id=00 asm-rule=11 seq=00022 ts=02554
RDT	DATA: stream-id=00 asm-rule=11 seq=00023 ts=02670
RDT	DATA: stream-id=00 asm-rule=11 seq=00024 ts=02786
RDT	DATA: stream-id=00 asm-rule=11 seq=00026 ts=03019
RDT	DATA: stream-id=00 asm-rule=11 seq=00027 ts=03135
RDT	DATA: stream-id=00 asm-rule=11 seq=00028 ts=03251
RDT	DATA: stream-id=00 asm-rule=11 seq=00029 ts=03367
RDT	DATA: stream-id=00 asm-rule=11 seq=00031 ts=03599
RDT	DATA: stream-id=00 asm-rule=10 seq=00032 ts=03715
RDT	DATA: stream-id=00 asm-rule=11 seq=00033 ts=03831
RDT	DATA: stream-id=00 asm-rule=11 seq=00034 ts=03947
RDT	LATENCY-REPORT: t=1030 DATA: stream-id=00 asm-rule=
RDT	DATA: stream-id=00 asm-rule=11 seq=00036 ts=04180

Stream ID

Sequence Number

Timestamp

Source: Slides Lecture Multimedia Kommunikation, Fakultät Informatik, TU Dresden

RTSP – TEARDOWN Command

TECHNISCHE UNIVERSITÄT DARMSTADT

Client request

TEARDOWN
rtsp://62.201.170.10:554/events/[…]/20899581.rm
RTSP/1.0\r\n
CSeq: 8\r\n
User-Agent: RealMedia Player HelixDNAClient/10.0.1.338 (win32)\r\n
Session: 809730200-1
\r\n

Playback is stopped

All resources are released

Restart requires:

SETUP and PLAY

Server response

RTSP/1.0 **200 OK**\r\n
CSeq: 8\r\n
Date: Sat, 24 Jan 2009 08:26:57 GMT\r\n
Session: 809730200-1 \r\n

Successful stopping of playback acknowledged

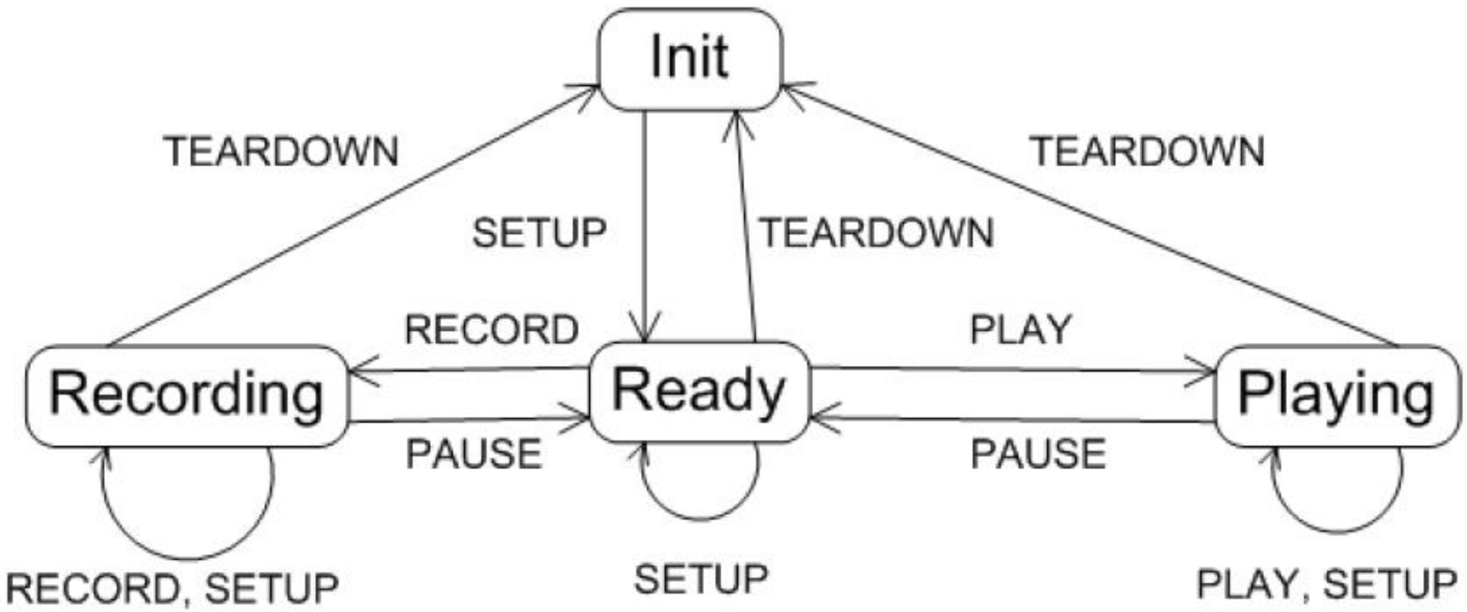
Source: Slides Lecture Multimedia Kommunikation, Fakultät Informatik, TU Dresden

2.3 RTSP State Diagram

State is maintained per Stream → Session-ID

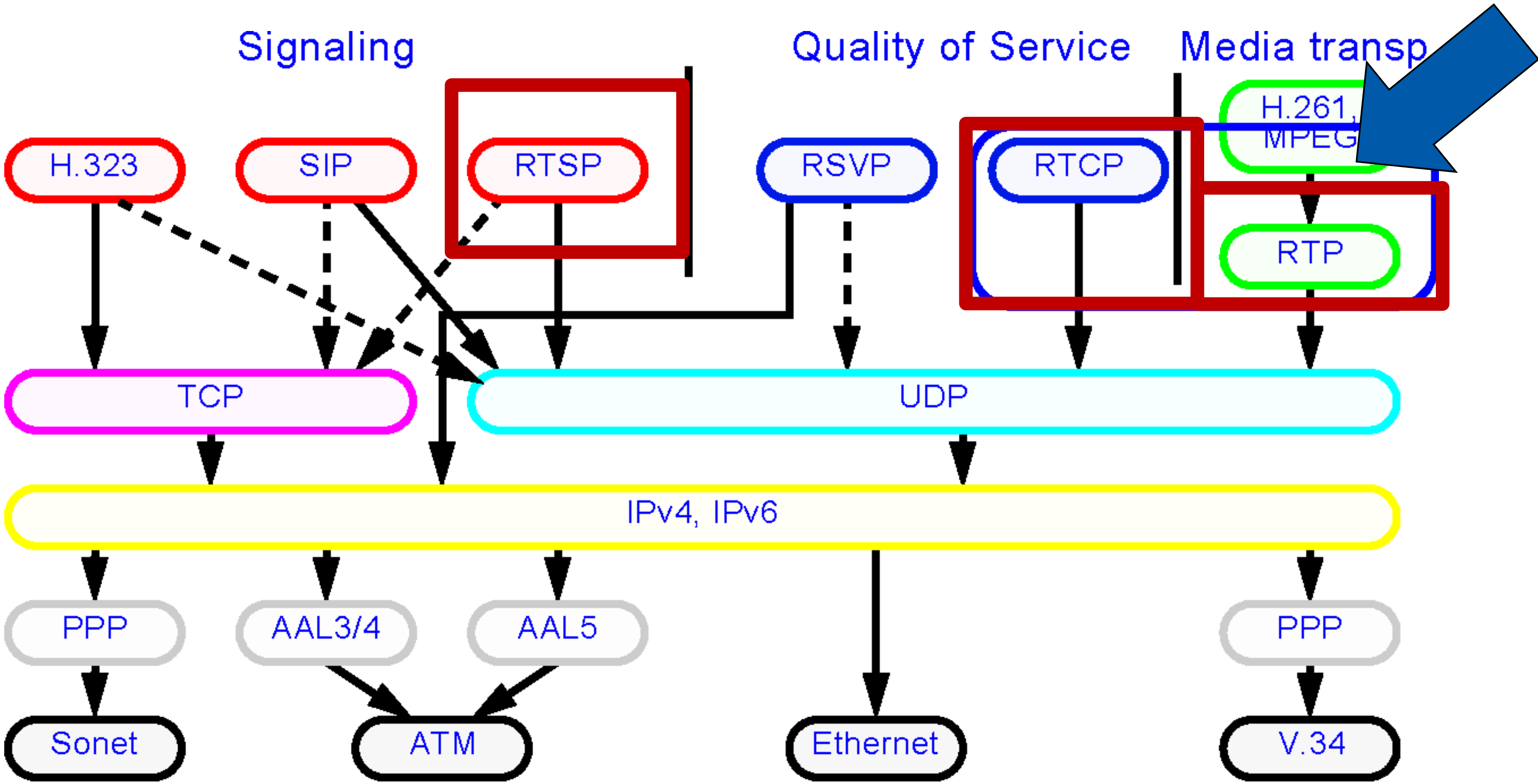
State Transition triggered at the reception of a request

- OPTIONS, DESCRIBE, SET_PARAMETER request do not trigger any state transitions



Source: Slides Lecture Multimedia Kommunikation, Fakultät Informatik, TU Dresden

TECHNISCHE UNIVERSITÄT DARMSTADT



Multimedia Transport Protocols

Need for

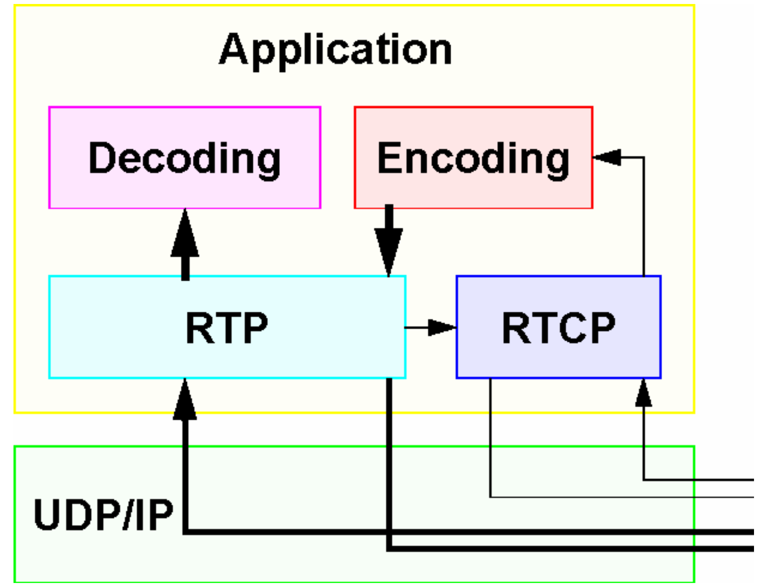
separate flows for each media stream
- simplifies applications
- allows for different QoS

receiver adaptation
- buffering to smooth out jitter
  - which always exists to some extend
- timestamp necessary

synchronization
- of various media streams
- adapting play-out buffers

framing service
- splitting media stream into adequate PDUs

➔ **Real-Time Transport Protocol RTP**



see
- RFC 1889, RTP: A Transport Protocol for Real-Time Applications
- RFC 1890, RTP Profile for Audio and Video Conferences with Minimal Control
- and around e.g. http://www.cs.columbia.edu/~hgs/rtp/ RFC 5574

Transport Layer

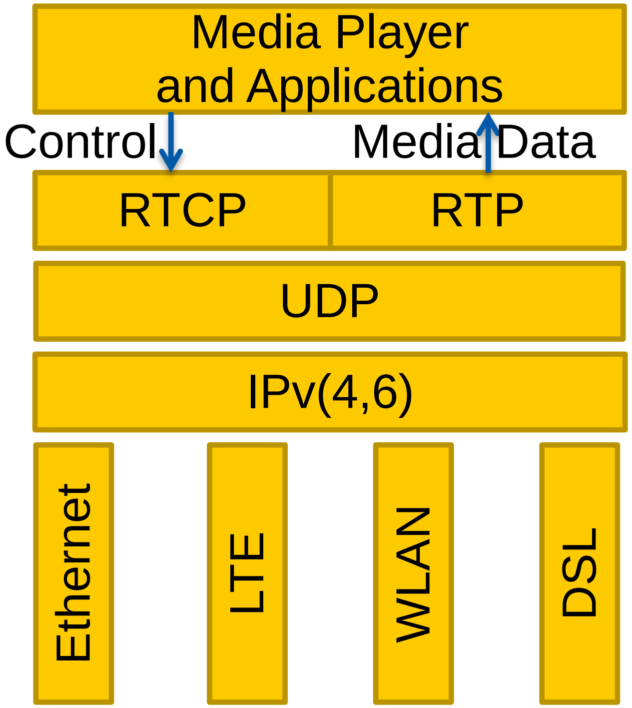
- Real-Time Transport Protocol RTP
- Real-Time Transport Control Protocol RTCP

End-to-end transport functions but is NOT

- a real transport protocol
  - no checksums
  - no multiplexing
- a real-time protocol
  - no reservations
  - no guarantees

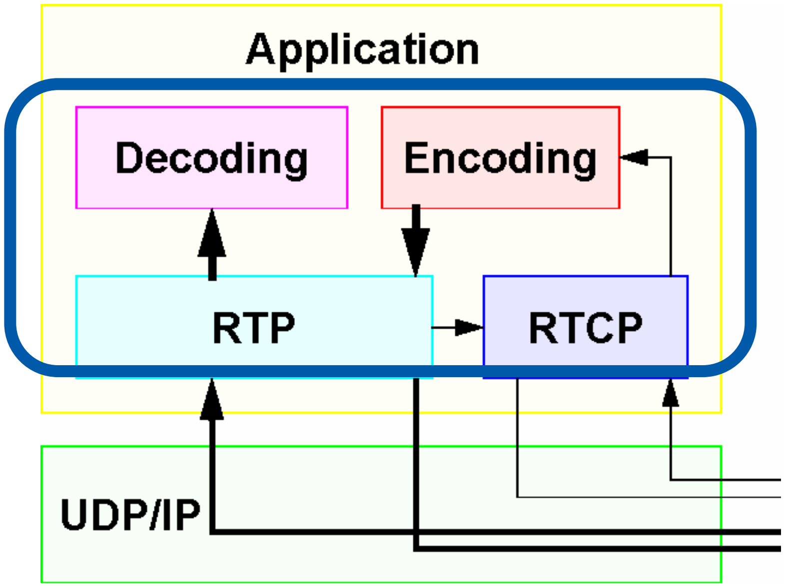
Adds functionality to existing transport protocols

- designed to work with UDP
  - works also with TCP
- functions like
  - session layer (in OSI terminology)
  - integrated with applications



Real-Time Transport Control Protocol (RTCP)

companion protocol to RTP



functions:

- to monitor QoS
- to convey information about
  - participants
  - session relationships

I.e.

- To monitor application performance
  - feedback to sender about delivery quality, loss, etc.
- automatic adjustment to overhead
  - report frequency based on participant count

Typically,

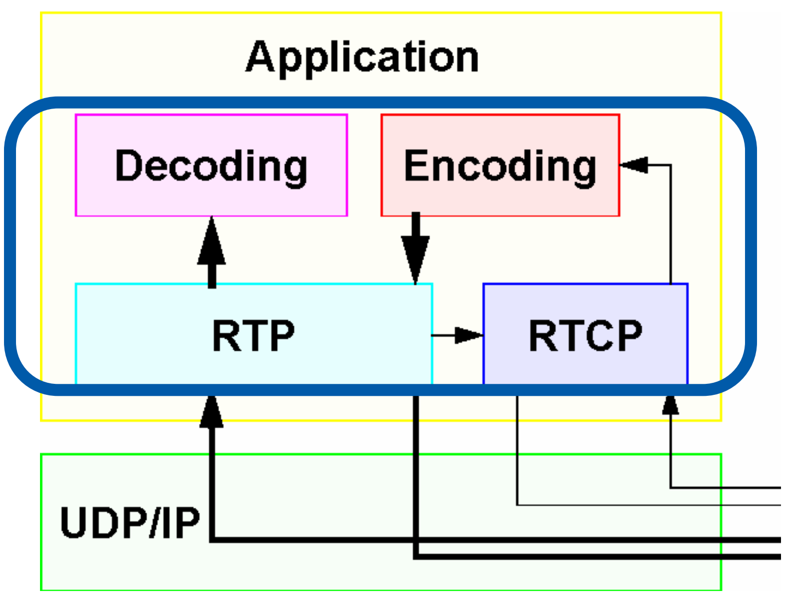
- “RTP does ...” means “RTP with RTCP does ...”

RTP with RTCP Functions

TECHNISCHE UNIVERSITÄT DARMSTADT

RTP with RTCP provides:

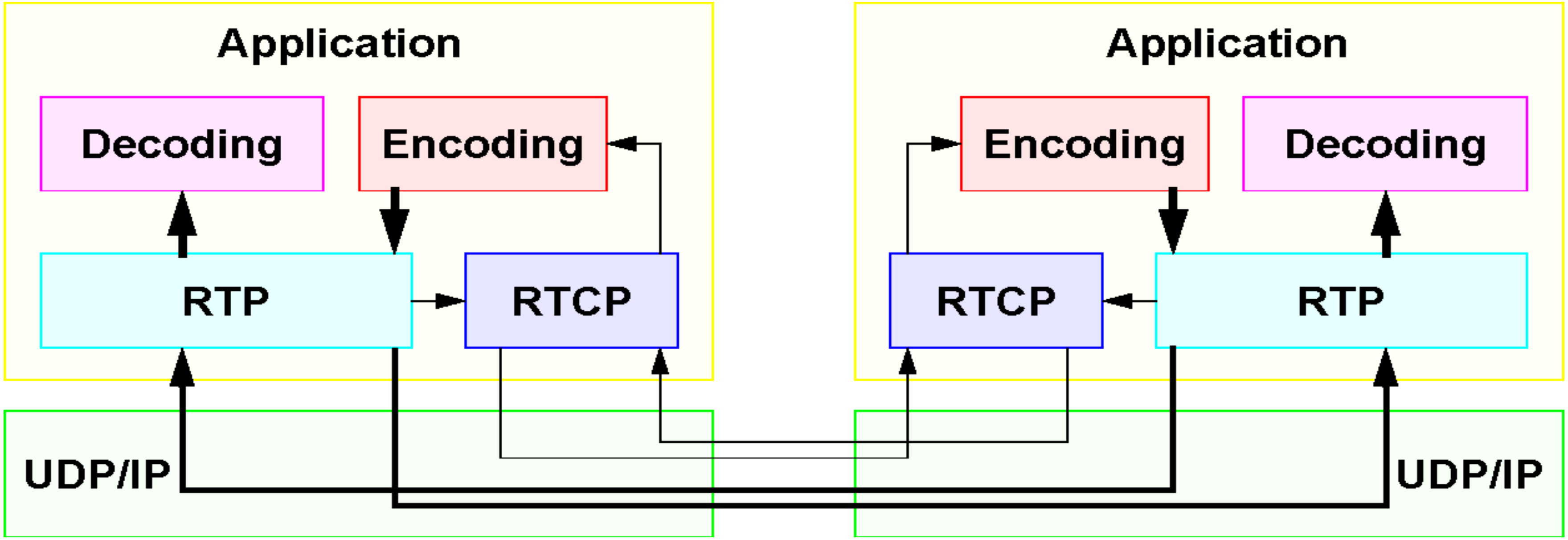
- support for transmission of real-time data
- over multicast or unicast network services



functional basis

- sequence numbering
- determination of media encoding
  - i.e. payload type identification
- source identification (process)
- synchronization
- framing i.e. follows principle of
  - application level framing and
  - integrated layer processing
- error detection
  - i.e. delivery monitoring
- encryption
- Timing
  - i.e. time stamping
- unicast and multicast support
- support for stream “translation” and “mixing”

RTP+RTCP: Quality Control



Component interoperations for control of quality

- evaluation of sender and receiver reports
- modification or encoding schemes and parameters
- adaptation of transmission rates

TECHNISCHE UNIVERSITÄT DARMSTADT

Header structure

bytes	20	8	12	
	IP HDR	UDP	RTP HDR	Payload

RTP and RTCP uses (most commonly) UDP

- simple
- unreliable
- connectionless
- multicast

RTCP to control the stream

- media encoding with profiles

RTP Profile	Media Enc.	Bits/Sample	Sampl. Rate	Packet Rate
0	PCM µLaw	8	variable	
6	DVI14	4	16 kHz	20 ms
9	G.722	8	16 kHz	20 ms

RTP Header

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|V=2|P|X|  CC   |M|     PT      |       sequence number         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           timestamp                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           synchronization source (SSRC) identifier            |
+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
|            contributing source (CSRC) identifiers             |
|                             ....                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

3.3 RTCP Packets

Characteristics

- periodic control packets
- quality control
- participants list
- occupies only a fraction of the used bandwidth

Header

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|V=2|P| subtype |   PT= SR =200 |             length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           SSRC/CSRC                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          name (ASCII)                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   application-dependent data                ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

with

- PT (packet type) non application specific, e.g.
  - SR Sender Report (statistics from active senders: bytes sent -> estimate rate)
  - RR Receiver Report (statistics from receivers)
  - SDES Source Descriptions (Canonical Name = user@host; name, email, location, …)
  - BYE explicit leave
- application/media specific
  - APP extensions, application specific

RTCP to control the stream

- media encoding with profiles

Payload type identification

RTP Profile	Media Enc.	Bits/Sample	Sampling Rate	Packet Rate
0	**PCM µLaw**	**8**	**variable**	
6	**DVI14**	**4**	**16 kHz**	**20 ms**
9	**G.722**	**8**	**16 kHz**	**20 ms**

each content encoding needs its coding specification

- defines a set of payload type codes and their mapping to payload formats like
  - H.261, H.263, H.263+ (ITU-T), Real, ..
  - Motion JPEG, MPEG1 & MPEG2, Bundled MPEG, CellB video encoding
  - BT.656-3 encoding
  - HTTP encoding
  - ASF (Advanced Streaming Format)
  - DTMF (dial tone multiple frequency) Digits
  - Layered Multimedia Streams
  - Redundant Encodings Audio Data

3.5 Further Details: Mixer & Translator

Mixer functions:

reconstructs
- constant spacing generated by sender

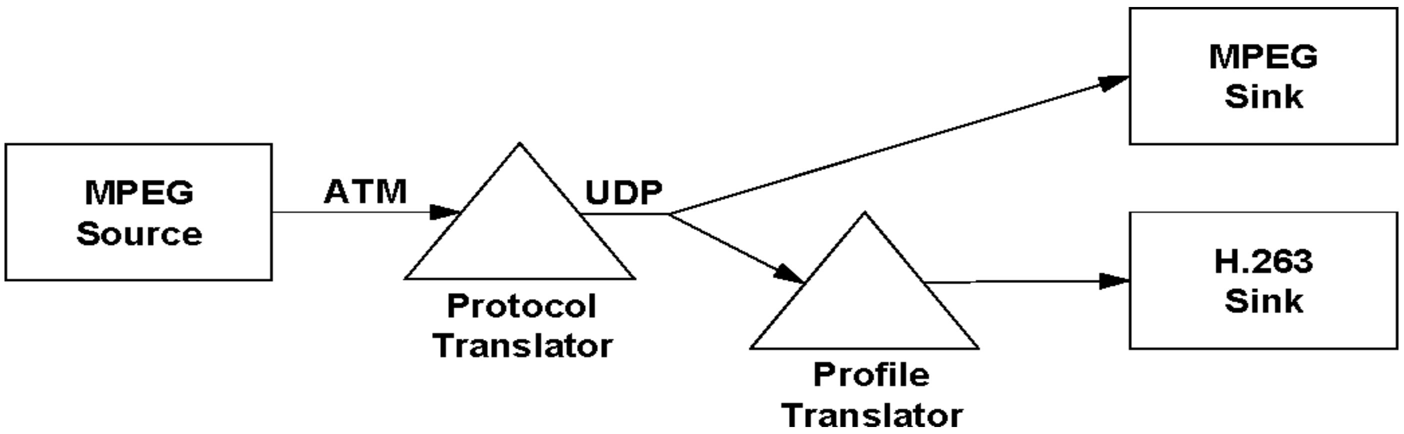
translates
- e.g. audio encoding to a lower-bandwidth

mixes
- reconstructed audio streams into a single stream

resynchronizes
- incoming audio packets
  - new synchronization source value (SSRC) stored in packet
  - incoming SSRCs are copied into the contributing sync. source list (CSRC)

forwards
- mixed packet stream

Mixer & Translator



translation between

- IP and other protocols or protocol families
- e.g., between
  - IP and
  - e.g. Stream Type Protocol ST-2

two translators are installed

- may change the encoding of data
- no resynchronization in translators

SSRC and CSRC remain unchanged

SSRC

- is synchronization source (random number), identification of sender, whose timestamp is master time stamp

CSRC

- list of identifiers of those contributing to (mixed) packet

RTP Identifiers