RTP Payloads

RTP Payload Types

- 7-bit payload type identifier
  - Some numbers statically assigned
  - Dynamic payload types identifiers for extensions – mapping to be defined outside of RTP (control protocol, e.g. SDP "a=rtpmap:"

Payload formats defined for many audio/video encodings

- Conferencing profile document RFC 3551
  - Audio: G.711, G.722, G.723.1, G.728, GSM, CD, DVI, ...

- In codec-specific RFCs
  - Audio: Redundant Audio, MP-3, ...
  - Video: JPEG, H.261, MPEG-1, MPEG-2, H.263, H.263+, BT.656
  - Others: DTMF, text, SONET, ...

- Generic formats
  - Generic FEC, (multiplexing)

IANA maintains a list
http://www.iana.org/assignments/rtp-parameters
Media Packetization Schemes (1)

General principle:
- Payload specific additional header (if needed)
- Followed by media data
  - Packetized and formatted in a well-defined way
  - Trivial ones specified in RFC 3551
  - RFC 2029, 2032, 2035, 2038, 2190, 2198, 2250, 2343, 2429, 2431, RFC 2435, 2658, 2733, 2793, 2833, 2862, and many further ones
  - Guidelines for writing packet formats: RFC 2736
- Functionality
  - Enable transmission across a packet network
  - Allow for semantics-based fragmentation
  - Provide additional information to simplify processing and decoding at the recipient
  - Maximize possibility of independent decoding of individual packets

Sample RTP Payload Types

Illustrate a variety of approaches to deal with packet loss in the Internet
Audio over RTP: PCM

<table>
<thead>
<tr>
<th>V</th>
<th>P</th>
<th>X</th>
<th>CC</th>
<th>M</th>
<th>PT = 0</th>
<th>Sequence Number</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Timestamp (8 KHz clock)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Sender SSRC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Audio Data</td>
</tr>
</tbody>
</table>

Audio Redundancy Coding (1)

- Audio Packets are small!
  - have to be because of interactivity
    - avoid large packetization delay
  - packet loss primarily depends on packet rate
    - rather than packet size

- Payloads for multiple time slots in one packet
  - send redundant information in packet n to reconstruct packets k, ..., n-1
  - redundant information typically sent at lower quality
  - details defined in RFC 2198
  - uses dynamic payload type

- Format specification, e.g. using SDP
  - m=audio 20002 RTP/AVP 96 0 0 0
  - a=rtpmap:96 red/8000/1
Audio Redundancy Coding (2)

Primary Encoding: PCM
Secondary Encoding: GSM

Packet #1
Packet #2
Packet #3
Packet #4

Primary
Redundant 2
Redundant 1

Data Block “Redundant 2” (160 Bytes)
Data Block “Redundant 1” (160 Bytes)
Data Block “Primary” (160 Bytes)

Audio Redundancy Coding (3)
Speex

- It is based on Code Excited Linear Prediction [CELP]
- Open source audio codec for VoIP applications
- When encoded at max bit-rate, a frame would be approx. 110 octets. Single frame is < path MTU.
- Speex frames can be combined and sent together, however no speex frame should be fragmented
- The padding begins with a 0 and is followed by a series of 1s to complete the octet

Example

```
0 8 16 24 31

RTP Header

Multiple Speex frame padding
```

DTMF over RTP (1)

- DTMF digits, telephony tones, and telephony signals
  - two payload formats
  - 8 kHz clock by default
  - audio redundancy coding for reliability
- Format 1: reference pre-defined events
  - 0 - 9 * # A - D (Hook)Flash [17]
  - modem and fax tones [18]
  - telephony signals and line events [43]
  - trunk events [44]
  - specified through identifier (8-bit value), volume, duration
- Format 2: specify tones by frequency
  - one, two, or three frequencies
  - addition, modulation
  - on/off periods, duration
  - specified through modulation, n x frequency, volume
**Video Redundancy Coding (1)**

- Video redundancy coding
  - For H.263+ video streams
  - Transmit several interleaved sequences of predicted frames (threads) instead of one
    - improves error resilience against packet loss

- Principle
  - create several (n) independently decodable streams
  - achieved by choosing different reference pictures
  - decode only streams with no packet losses
    - reduces temporal resolution by 1/n-th per affected stream
  - bit rate penalty due to larger deltas between frames
  - RFC 2429, revised version in progress
Video over RTP: H.261

Additional payload-specific header preceeds payload
- To avoid expensive bit shifting operations
  - Indicate # invalid bits in first (SBit) and last (EBit) octet of payload
- Indicate Intra encoding (I bit)
- Indicate the presence of motion vector data (V bit)
- Carry further H.261 header information to enable decoding in the presence of packet losses

Further mechanisms for video conferencing
- FIR: Full Intra Request
  - Ask sender to send a full intra encoded picture
- NACK: Negative Acknowledgement
  - Indicate specific packet loss to sender

Video over RTP: H.261 (2)
Video Redundancy Coding (2)

Video Redundancy Coding (3)
Video Redundancy Coding (4)

H.263+ coded data

RTP Header

rsrvd  Plen  Pebit TID  Trun  S

H.263+ Picture Header (optional)

Padding

H.263+ coded data

RTP Payload

H.264/AVC

- AVC stands for Advanced Video Coding
- RTP profile for H.264 AVC is standardized in RFC 3984
  - Updated draft 3984bis is being worked on

F is a flag for bit-error checking
NRI is 2 bits and indicates that usefulness of the NALU for reference frame reconstruction
Type is 5 bits and indicates NALU type. They are:
  - Single Unit (1-23), Single time aggregation packet (24, 25),
  - Multi-time aggregation packet (26, 27), Fragmentation unit (28-29).
SVC

- RTP profile for scalable Video coding is under final review will become an RFC soon
- SVC is an extension, Annex. G of H.264 standard
- Packet loss can cause artifacts in high quality video but by using more computational complexity an SVC compliant codec can achieve the same playback quality but at lower image quality.
- Extended NALU Type 14, 15 and 20 of H.264 AVC and defined as an 24 bit header.

Media Packetization Schemes (2)

Error-resilience for real-time media

- Input: Observation on packet loss characteristics
- Generic mechanisms (RFC 2354)
  - Retransmissions
    - in special cases only (e.g. with no interactivity!)
  - Interleaving
  - Forward Error Correction (FEC)
    - media-dependent vs. media-independent
    - Generic FEC: RFC 2733
- Feedback loops for senders
  - based upon generic and specific RTCP messages
  - adapt transmission rate, coding scheme, error control, ...
RTP Interleaving

- Distribute packets or packet contents for transmission
  - Avoid consecutive packet erasures in case of (burst) losses
  - Avoid loss of large consecutive data portions in case of single packet losses

Motivations

- Human perception tolerates individual losses better (with error concealment)
- Make simple FEC schemes work better with burst losses (e.g. XOR)

Drawback

- Re-ordering causes additional delay

RTP FEC (RFC 2733)

- Forward Error Correction scheme for RTP packets
  - Media-independent, flexible FEC (that can be enhanced)

Simple XOR-based (parity) FEC

- \( P_{fec} = P_1 \ XOR \ P_2 \ XOR \ P_3 \ XOR \ldots \ XOR \ P_n \)
  - Allows reconstruction of any single missing packets of \( P_1, \ldots, P_n, P_{fec} \)

RTP FEC stream transmitted independently of RTP stream

- Separate transport address (IP address, port number)
- Different SSRC

Recovery
### RTP Parity FEC Packet format

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>V = 2</td>
<td>Wait bit, 0 for normal packet, 1 for parity packet</td>
</tr>
<tr>
<td>P</td>
<td>Priority bit, 0 for normal packet, 1 for priority packet</td>
</tr>
<tr>
<td>FMT</td>
<td>Format type, 0 for audio, 1 for video</td>
</tr>
<tr>
<td>PT</td>
<td>Payload type, 0 for normal payload, 1 for priority payload</td>
</tr>
<tr>
<td>length</td>
<td>Length of the packet in bytes</td>
</tr>
<tr>
<td>SSRC of packet sender</td>
<td>Source identifier of the packet sender</td>
</tr>
<tr>
<td>SSRC of media source</td>
<td>Source identifier of the media source</td>
</tr>
<tr>
<td>SN base</td>
<td>Sequence number base for length recovery</td>
</tr>
<tr>
<td>E</td>
<td>Error bit</td>
</tr>
<tr>
<td>PT recovery</td>
<td>Payload type recovery mask</td>
</tr>
<tr>
<td>mask</td>
<td>Mask for payload recovery</td>
</tr>
<tr>
<td>TS recovery</td>
<td>Timestamp recovery</td>
</tr>
</tbody>
</table>

**Sample Packet:**
```
E = 1
mask = 10110000 00000000 00000000
```

**Payload XOR:**
```
XOR of Payloads indicated by SN Base and mask
```

### Unequal Error Protection

- **Observation:** not all parts of a packet are equally important
  - Beginning of packet contains headers/parameters, more relevant contents
  - Holds for both audio and video

- **Uneven Level Protection (ULP):**
  - Create independent parity packets for different parts of packets
  - Allows for selectively more overhead for the more important parts

- **Level 0:**
  - Packet A: 50% FEC
  - Packet B: 50% FEC
  - Packet C: 25% FEC

- **Level 1:**
  - Packet D: 25% FEC

- **Related thoughts:**
  - Partial checksums
    - Live with bit errors in the less important parts (rather than dropping a packet)
ULP FEC payload format

-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          RTP Header (12 octets or more)                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          FEC Header (10 octets)                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       FEC Level 0 Header                                    |
| +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                              FEC Level 0 Payload                              |
| |                                                                               |
| +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     FEC Level 1 Header                                    |
| |                                                                               |
| +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                              FEC Level 1 Payload                              |
| |                                                                               |
| +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Cont.                                               |
| |                                                                               |
| +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

- FEC header is constructed with one or more level of FEC encoding

New FEC packet

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|E|L|P|X| CC |M| PT recovery | SN base |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| TS recovery |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| length recovery |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

- Standardized so that it is agnostic to Audio/Video data as much as possible
- E and L are flags for extension bits (E), long mask (L)
- P, X, CC, M and PT are the recovery fields for the corresponding fields in the FEC’ed packets.

defined in RFC 5104
FEC level header

<table>
<thead>
<tr>
<th>Protection Length</th>
<th>mask</th>
</tr>
</thead>
<tbody>
<tr>
<td>mask cont. (present only when L = 1)</td>
<td></td>
</tr>
</tbody>
</table>

- FEC level header indicates which packets are associated with the FEC packet
- The L flag specifies if the mask is 16 bits or 48 bits.
- The rules for masking are based on the levels of protection.
- So a media packet protected at LEVEL “P” should also be protected at LEVEL “P-1” in any FEC packet

RTP Payload Type Overview (1)

- RFC 3551 Collection of simple packetization formats (formerly RFC 1890)
- RFC 2029 Sun CellB Video encoding
- RFC 2032,4587 H.261 video
- RFC 2435 JPEG video (was RFC 2035)
- RFC 2250 MPEG-1/MPEG-2 video (was RFC 2038)
- RFC 2190 H.263 video (historic)
- RFC 2343 Bundled MPEG
- RFC 2429 H.263+ video & video redundancy support
- RFC 2431 BT.656 video
- RFC 2658 PureVoice audio
- RFC 2793,4103 Text conversation
- RFC 2833 DTMF, telephony tones, and telephony signals
- RFC 2862 Real-time Pointers
- RFC 3016 MPEG-4 Audio/visual streams
- RFC 3047 G.722.1 audio
- RFC 3119 Loss-tolerant format for MP3
- RFC 3189 DV video
- RFC 3190 12-bit DAT and 20-/24-bit linear audio
RTP Payload Type Overview (2)

- RFC 3267, 4352 Adaptive Multirate (AMR, AMR-WB+) audio
- RFC 3389 Comfort noise
- RFC 3497 SMPTE 292M video
- RFC 3557 ETSI Distributed speech recognition (ES 201 108)
- RFC 3558 Enhanced variable rate codecs and selectable mode vocoders
- RFC 3640 MPEG-4 elementary streams
- RFC 3952 Low Bit Rate Codec (ILBC) Speech
- RFC 3984 H.264 Video
- RFC 4040 64 kbit/s Transparent Call
- RFC 4060 Distributed speech recognition encoding (ES 202 050/211/212)
- RFC 4175, 4421 Uncompressed Video
- RFC 4184, 4598 AC-3 Audio, Enhanced AC-3
- RFC 4298 BroadVoice Speech codec
- RFC 4348, 4424 Variable Rate Multimodal Wideband Audio (VMR-WB)
- RFC 4351 Text conversation interleaved with audio stream
- RFC 4396 3GPP Timed Text
- RFC 4425 Video Codec 1 (VC-1)
- RFC 4588 Retransmission payload format

Many more to come…