RTP packet header

Offsets	Octet	0								1								2								3							
Octet	Bit [a]	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Version				P	X	CC		M	PT							Sequence number															
4	32	Timestamp																															
8	64	SSRC identifier																															
12	96	CSRC identifiers ...																															
12+4×CC	96+32×CC	Profile-specific extension header ID																Extension header length															
16+4×CC	128+32×CC	Extension header ...																															

```go
type Header struct {
    Version          uint8
    Padding          bool
    Extension        bool
    Marker           bool
    PayloadType      uint8
    SequenceNumber   uint16
    Timestamp        uint32
    SSRC             uint32
    CSRC             []uint32
    ExtensionProfile uint16
    Extensions       []Extension
}

// Packet represents an RTP Packet
type Packet struct {
    Header
    Payload     []byte
    PaddingSize byte
}
```

The bit string is formed for each source packet by concatenating the following fields together in the order specified:

o The first 16 bits of the RTP header (16 bits), though the first two (version) bits will be ignored by the recovery procedure.

o Unsigned network-ordered 16-bit representation of the source packet length in bytes minus 12 (for the fixed RTP header), i.e., the sum of the lengths of all the following if present: the CSRC list, extension header, RTP payload and RTP padding (16 bits).

o The timestamp of the RTP header (32 bits).

o All octets after the fixed 12-byte RTP header. (Note the SSRC field is skipped.)

The FEC bit string is generated by applying the parity operation on the bit strings produced from the source packets. The FEC header is generated from the FEC bit string as follows:

The FEC bit string is generated by applying the parity operation on the bit strings produced from the source packets. The FEC header is generated from the FEC bit string as follows:

o The first (most significant) 2 bits in the FEC bit string, which contain the RTP version field, are skipped. The R and F bits in the FEC header are set to the appropriate value, i.e., it depends on the chosen format variant. As a consequence of overwriting the RTP version field with the R and F bits, this payload format only supports RTP version 2.

o The next bit in the FEC bit string is written into the P recovery bit in the FEC header.

o The next bit in the FEC bit string is written into the X recovery bit in the FEC header.

o The next 4 bits of the FEC bit string are written into the CC recovery field in the FEC header.

o The next bit is written into the M recovery bit in the FEC header.

o The next 7 bits of the FEC bit string are written into the PT recovery field in the FEC header.

o The next 16 bits are written into the length recovery field in the FEC header.

o The next 32 bits of the FEC bit string are written into the TS recovery field in the FEC header.

o The lowest Sequence Number of the source packets protected by this repair packet is written into the Sequence Number Base field in the FEC header. This needs to be repeated for each SSRC that has packets included in the source block.

o Depending on the chosen FEC header variant, the mask(s) are set when F=0, or the L and D values are set when F=1. This needs to

be repeated for each SSRC that has packets included in the source block.

o The rest of the FEC bit string, which contains everything after the fixed 12-byte RTP header of the source packet, is written into the Repair "Payload" following the FEC header, where "Payload" refers to everything after the fixed 12-byte RTP header, including extensions, CSRC list, true payloads, and padding.

If the lengths of the source packets are not equal, each shorter packet MUST be padded to the length of the longest packet by adding octet 0's at the end.

```
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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|0|1|P|X|  CC   |M| PT recovery |         length recovery       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          TS recovery                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          SN base_i            |  L (columns)  |    D (rows)   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    ... next SN base and L/D for CSRC_i in CSRC list ...       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
:               Repair "Payload" follows FEC Header             :
:                                                               :
```

```go
type FecHeaderLD struct {
    R                 bool
    F                 bool
    P                 bool
    X                 bool
    CC                uint8
    M                 bool
    PTRecovery        uint8
    LengthRecovery    uint16
    TimestampRecovery uint32
    SN_base           uint16
    L                 uint8
    D                 uint8
}
```

6.3.2. Recovering the RTP Header

For a given set T, the procedure for the recovery of the RTP header of the missing packet, whose sequence number is denoted by SEQNUM, is as follows:

1. For each of the source packets that are successfully received in T, compute the 80-bit string by concatenating the first 64 bits of their RTP header and the unsigned network-ordered 16-bit representation of their length in bytes minus 12.

2. For the repair packet in T, extract the FEC bit string as the first 80 bits of the FEC header.

3. Calculate the recovered bit string as the XOR of the bit strings generated from all source packets in T and the FEC bit string generated from the repair packet in T.

4. Create a new packet with the standard 12-byte RTP header and no payload.

5. Set the version of the new packet to 2. Skip the first 2 bits in the recovered bit string.

6. Set the Padding bit in the new packet to the next bit in the recovered bit string.

7. Set the Extension bit in the new packet to the next bit in the recovered bit string.

8. Set the CC field to the next 4 bits in the recovered bit string.

9. Set the Marker bit in the new packet to the next bit in the recovered bit string.

10. Set the Payload type in the new packet to the next 7 bits in the recovered bit string.

11. Set the SN field in the new packet to SEQNUM.

12. Take the next 16 bits of the recovered bit string and set the new variable Y to whatever unsigned integer this represents (assuming network order). Convert Y to host order. Y represents the length of the new packet in bytes minus 12 (for the fixed RTP header), i.e., the sum of the lengths of all the following if present: the CSRC list, header extension, RTP payload and RTP padding.

13. Set the TS field in the new packet to the next 32 bits in the recovered bit string.

14. Set the SSRC of the new packet to the SSRC of the missing source RTP stream.

This procedure recovers the header of an RTP packet up to (and including) the SSRC field.

6.3.3. Recovering the RTP Payload

Following the recovery of the RTP header, the procedure for the recovery of the RTP "payload" is as follows, where "payload" refers to everything following the fixed 12-byte RTP header, including extensions, CSRC list, true payload and padding.

1. Allocate Y additional bytes for the new packet generated in Section 6.3.2.

2. For each of the source packets that are successfully received in T, compute the bit string from the Y octets of data starting with the 13th octet of the packet. If any of the bit strings generated from the source packets has a length shorter than Y, pad them to that length. The zero-padding octets MUST be added at the end of the bit string. Note that the information of the first 8 octets are protected by the FEC header.

3. For the repair packet in T, compute the FEC bit string from the repair packet payload, i.e., the Y octets of data following the FEC header. Note that the FEC header may be different sizes depending on the variant and bitmask size.

4. Calculate the recovered bit string as the XOR of the bit strings generated from all source packets in T and the FEC bit string generated from the repair packet in T.

5. Set the last Y octets in the new packet to the recovered bit string.