RTP Frame Acknowledgement

https://github.com/sprangerik/frame-acknowledgement

Erik Språng

Gurtej Singh Chandok

Shridhar Majali

Draft Status

- Progress on GitHub
  - Open issues: 8
  - Closed issues: 8

- Current hot topics:
  - Increase Frame ID length #16
  - Explicit recovery request #19

Increase Frame ID length #16

- Initially defined with 8 bit length
  - Up to 16 reference buffers in the encoder - 4 bits * 16 wraparounds - should be enough?
- Could still lead to problems if:
  - High FPS application
  - Encoder decides to move on due to lack of feedback
  - Receiver had long latency spike *but no loss*
  - Then might accidentally ack new unreceived frames

=> Decision was made use 16 bits for simplicity

Explicit Recovery Request #19

Senders may not wait for feedback to encode new frames

How can the receiver signal a recovery request?

- Packet loss feedback (NACK, RFC8888, etc)
  - Complex, doesn't represent decoder state
- PLI
  - In practice a keyframe request, doesn't represent decoder state

Suggestion: Add a flag to frame acknowledgement RTCP feedback message, indicating "*I have observed a loss event, and I have last received and decoded frames {...}*"

Next Steps

- Close remaining github issues
- Address any new feedback
    - Contributions on github are welcome

Thanks!