University of Glasgow | School of Computing Science

RTP Requirements for RTC-Web

Colin Perkins – University of Glasgow
Magnus Westerlund – Ericsson
Jörg Ott – Aalto University

Outline

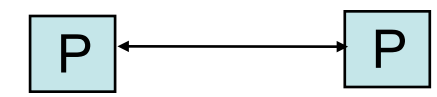
- Purpose of the draft
- Assumptions and topologies
- Requirements RTP places on RTC-Web
- Recommendations
  - Choice of RTP profile
  - RTP optimisations
  - RTP extensions
  - Improving transport robustness
  - Rate adaptation

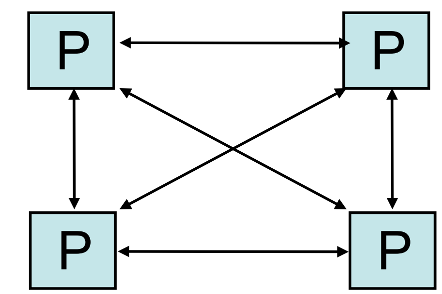
Purpose of the Draft

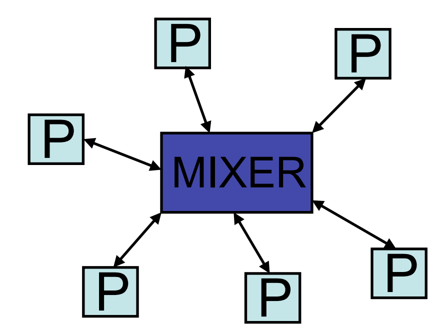
- General agreement that RTP appropriate for media transfer in RTC-Web
- How do RTP and RTC-Web interact?
  - What requirements does RTP impose on the RTC-Web solution?
  - How can the RTC-Web use of RTP be compatible with other uses of RTP, and with the basic operation of the RTP protocol?
- What is an appropriate set of RTP features and extensions for the RTC-Web solution?
  - For compatibility with other uses of RTP
  - To provide a solid base on which to build high-performance, flexible, and robust RTC-Web implementations

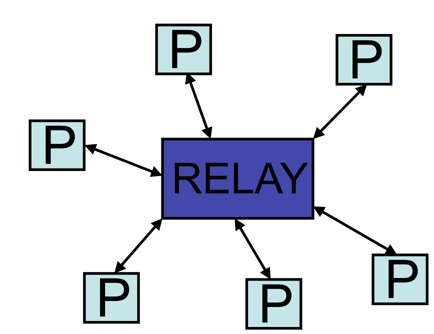
Assumptions and Topologies

- Transport flows are established and kept alive by other parts of the RTC-Web solution
- Some signalling solution exists to negotiate and configure media transport
- Media flows are configured to a subset of the topologies supported by RTP
  - Peer-to-peer
  - Multi-unicast or full mesh
  - Centralised mixer
  - Relay or transport translator
  - These are explicitly group communication scenarios, even though they use unicast connection
    - An application cannot determine the number of participants from the nature of the transport connections (it may be able to do so from the signalling)







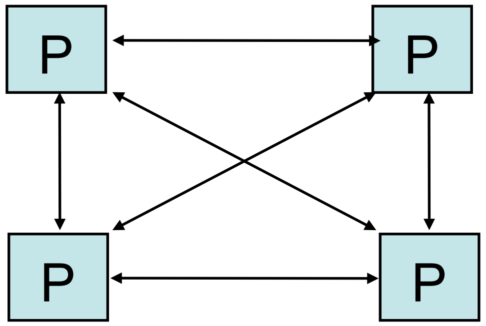


Requirements from RTP

- The RTP protocol imposes some requirements on the RTC-Web solution:
  - RTP sessions and multiplexing points
  - RTCP
  - Signalling Requirements for RTP Sessions
  - (Lack of) Signalling for Payload Format Changes

Sessions and Multiplexing Points

- RTP defines three multiplexing points:
  - RTP session – used for a specific purpose and type of media
    - E.g., audio for all game participants; video of anyone speaking in a lecture; video of a product being demonstrated – audio and video run on different RTP sessions
    - Defined by a common SSRC space, and can be distributed over several transport layer flows/connections
      - Transport addresses/ports separate RTP sessions, but *don't* define the session
      - E.g., a four participant full mesh teleconference, that comprises six peer-to-peer transport connections, will comprise *one* RTP session



  - Media flows – identified by SSRC within session
    - The RTCP CNAME indicates if from the same or different participants
  - Payload type – allows switching between different codecs of the same sort (e.g., between two audio codecs) at different times
- Important not to conflate these multiplexing points
  - In particular, trying to use the SSRC to separate different RTP sessions running on the same transport address is problematic

RTCP

- RTCP is an integral part of RTP – it is *not* optional
  - To identify participants within an RTP session
  - To ensure that other participants know of receiver-only SSRCs
  - To learn the media quality performance to other participants
  - To synchronise media streams (for lip-sync or other reasons)
  - To enable transport related media adaptation and control
  - To improve performance of media switching middleboxes for group conferencing scenarios

Signalling Requirements

- RTP requires a number of parameters to be signalled before a session can be created:
  - Addresses and port for transport-layer flows to be used
  - RTP profile to be used
  - RTP payload types and their media encoding parameters
  - RTP extensions to be used

- Note that RTP does not require signalling before change in payload type during a session
  - Can switch between any of the negotiated payload types at any time
  - Important to allow adaptation to network conditions

Recommendation for RTP Profile

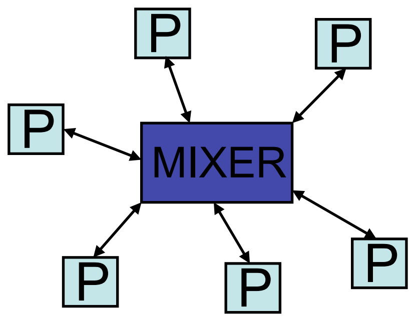
- Require support for RTP/SAVPF profile only:
  - Secured media is going to be required → the SAVP part
    - Using null crypto may be simpler than negotiating profile!
  - Feedback is essential → AVPF part
    - Greatly improved timeliness in RTCP behaviour
    - Reduces bit-rate consumption when there are no RTCP events to be sent
    - Conferencing extensions greatly improve performance of centralised conferencing
    - Rapid feedback provides improved robustness to packet loss
    - Transparent compatibility with RTP/AVP
- Negotiation of RTP profile is problematic: better to mandate a full-featured baseline, than to deal with signalling complexity

RTP Optimisations

- The draft proposes the following optimisations:
  - RTP and RTCP multiplexing – both flows on the same underlying transport flows
  - Reduced Size RTCP – allow feedback messages to be sent without redundant reception reports and SDES information
  - Symmetric RTP – allow bi-directional media on the same transport flow
  - RTCP CNAME generation – generate canonical name identifiers that work through NAT devices
- These generally simplify implementation compared to RFC 3550, although they increase specification complexity

RTP Extensions

- RTP conferencing extensions
  - Mixer selects or combines media streams
  - Needs RTCP feedback and control messages to improve performance and scalability
    - Full Intra Request
    - Picture Loss Indicator
    - TMMBR – also for rate adaptation



- RTP Header Extensions
  - RFC 5285 format required to support stacking and future extensibility
  - RFC 6051 rapid media synchronisation extensions recommended
  - If you want another header extension, please argue for it!
    - Client↔mixer audio level indications?

Improving RTP Transport Robustness

- Support for RTP/AVPF NACK and retransmission is recommended

- Support for FEC might be desirable, but unclear what to recommend
  - Many FEC schemes are encumbered
  - Range of options in use – no clear winner?

Rate Control and Media Adaptation

- RTC-Web will mainly be used on best effort Internet
  - Highly variable bottlenecks around the world
  - Must be able to adapt to available resources – else can starve yourself!
  - We *do not* mandate TCP-Friendly behaviour
    - Difficult to define, and inappropriate for many media flows
    - But, require to be self fair: two instances of an RTC-Web running over the same bottleneck will share fairly between each other
- No well established solution exist:
  - Fairly well developed draft for TFRC within RTP/UDP/IP
  - RTP running over DCCP with TFRC or TFRC-SP support?
    - DCCP over UDP as a baseline transport protocol? Complex but flexible
- Needs work!

Next Steps?

- Are these appropriate recommendations?
- Adopt this draft as a working group item?