ECN for RTP over UDP/IP

draft-westerlund-avt-ecn-for-rtp-00.txt
draft-carlberg-avt-rtp-ecn-02.txt
draft-carlberg-avt-rtcp-xr-ecn-01.txt

Magnus Westerlund
Ingemar Johansson
Colin Perkins
Ken Carlberg
Piers O’Hanlon
Motivation

• ECN provides for advanced warning of persistent congestion
  – RFC3168(§5.1): “the CE codepoint should not be set by a router based on the instantaneous queue size”
• ECN-CE warning is more useful to real-time flows (TCP can always ARQ)
  – Provides opportunity for adaption before loss occurs
• RTP/SDP provides a way forward
Dynamic adaptation RTP

• Many RTP flows do not do adaptation to loss
  – Using loss as a signal is a bit late
• There are now a number of variable bit rate codecs
• ECN allows
  – Early congestion response
    • Mechanisms are out of scope for this draft
  – Improved user experience
Background

- Explicit Congestion Notification (ECN)
  - Two Layer design (RFC-3168):
    - Network: hop-by-hop marking
    - Transport: negotiation and feedback
  - Active Queue Management (AQM)
    - E.g., Random Early Detection (RED), marks packets instead of dropping

- In-Band signaling
  - IP: two bits in diff-serv field
    - ECN Capable Transport (ECT) (01, 10)
    - Congestion Experience (CE) (11)
    - ECN not supported (00)
  - TCP: two bits
    - ECN Echo, Congestion Window Reduced
  - TCP ECN Nonce (RFC 3540)
ECN for RTP over UDP/IP

• Initially seems straight-forward:
  – Signal ECN support in SIP using SDP offer/answer
  – Set ECT on RTP data packets sent in UDP/IP
  – Send feedback piggybacked on RTCP reception reports
    • (No portable way to monitor received ECN marks on UDP)
  – Respond to ECN-CE by varying media encoding rate

• Yes, but...
Why is ECN for RTP Difficult? (1/3)

• Signalling
  – Signalling can negotiate end-point capability; says nothing about ability of media path ability to support ECN

• Feedback
  – RTCP feedback on congestion events is slow – seconds rather than RTT
    • AVPF helps, but may still limit amount of feedback that can be sent

• Congestion Response
  – Codecs adaptive, within some constraints; frequent variation destroys user experience; not TCP-friendly
Why is ECN for RTP Difficult? (2/3)

• Middle-boxes
  – RTP *translators* and *mixers* within the network
    • Translator is a middle-box; must interpose itself in the ECN negotiation, split the connection, respond to congestion
    • Mixer acts as end-point; terminates transport connections
  – Only *part* of an RTP session may support ECN
Why is ECN for RTP Difficult? (3/3)

• Multicast
  – RTP is inherently a *group* communication protocol
    • ASM with many-to-many groups and multicast feedback
    • SSM with unicast feedback, potentially very large groups
      – IPTV channels, potentially millions of receivers

  – ECN per sender tree? For the entire group? All receivers? Again, only *parts* of the session may support ECN
  – May require receiver driven congestion response (layered coding?)
ECN for RTP over UDP/IP: Proposal

• Four pieces to the proposed solution:
  – Negotiation of ECN capability
    • SIP with SDP offer/answer; ICE option
  – Initiation and verification of ECT
    • Using RTP and RTCP
    • Using STUN and ICE
  – Ongoing use of ECN with RTP session
  – Failure detection, verification, and fallback
Negotiation of ECN Capability

• SIP with SDP offer/answer
  – SDP offer include new attribute to indicate ECN capability of the offering entity
    • a=ecn-capable-rtp
    • a=rtp-ecn: <sendonly|sendrecv>
  – Answering entity replies; negotiates ECN capability

  – Portable APIs exist to set ECN bits on UDP packets, but not to read them from received packets
    • Should we support devices that can send ECN, but not receive it?
ECN Probing

• End-point ECN capability != path ECN capability

• Broken middle-boxes exist which can disrupt ECN
  – Drop packets with ECT marks
  – Zero out ECT marks in transit

• Need to probe path to determine if ECN supported
  – Using STUN as part of an ICE exchange
  – Using RTP and RTCP
ECN Probing using STUN/ICE (1/2)

• Additional signalling: capability to probe the path for ECN support using STUN as part of an ICE exchange
  – a=ice-options: rtp+ecn
  – Details to be resolved: a=ice-options poorly defined

• Possible for unicast flows where ICE is supported
  – Subset of possible use-cases
ECN Probing using STUN/ICE (2/2)

Figure 1: ECN Check Stun Attribute

V: Valid (1 bit) ECN Echo value field is valid when set to 1, and invalid when set 0.

ECF: ECN Echo value field (2 bits) contains the ECN filed value of the STUN packet it echoes back when field is valid. If invalid the content is arbitrary.

Reserved: Reserved bits (29 bits) SHALL be set to 0 and SHALL be ignored on reception.
ECN Probing using RTP/RTCP

• Basic RTP/RTCP probing mechanism:
  – Sender starts by ECT marking small fraction of RTP packets
    • Comfort noise, no-op, or similar
  – Receivers report reception of ECT marked packets
    • New RTCP report blocks sent using AVPF, described later
  – Sender waits for receiver population to stabilise
  – If all receivers reported reception of ECT marked packets, sender may switch to ECT marking all packets

• Per-sender; gracefully supports groups; conservative
ECN Usage with RTP

• Sender ECT-marks all packets
• Receivers send ECN feedback
  – Regular RTCP: indicate continued receipt of ECT-marks
  – AVPF feedback: receipt of ECN-CE packets
• Respond to ECN-CE as-if packet loss occurred; reduce path data rate

• Need to continually monitor, since path may fail
  – Discussion later
RTCP Feedback: Regular

- Use new RTCP XR report
- Initial straw man for the data it should report:
  - Start + end sequence numbers, bitmaps of lost and marked packets, ECN nonce value
  - Considering alternative that avoid ECN nonce

```
+--------+--------+--------+--------+
| 0 1 2 3| 4 5 6 7| 8 9 0 1| 2 3 4 5|
+--------+--------+--------+--------+
 INV | RNV | Z | C | P | Reserved | Chunk 1 |  
+--------+--------+--------+--------+
```
RTCP Feedback: Congestion/Probe

• Need rapid feedback during probing period, or if ECN-CE marked packet received
• Use new AVPF feedback packet
  – Should be small enough to use immediate mode
  – Aim for similar format to regular reports
Congestion Response

• Receipt of ECN-CE indicates congestion
  – Path data rate must be reduced, or packet loss will occur
  – Two options:
    • Sender-based rate reduction: change media encoding options
    • Receiver-driven rate reduction: layered media coding
  – Lots of options for how to adapt; probably not TCP-friendly

• Incentive to react to ECN-CE:
  – If you react, you control how media quality is reduced
  – If you don’t react, network will drop packets – worse quality
Ongoing Verification of ECN

• Why might ECN support change?
  – New receivers join a multicast group
  – Mobility changes the path, putting a new broken middle box on path

• How to detect and fallback?
  – Regular RTCP feedback will show (some) receivers not getting ECT-marked packets
  – Fall-back to occasional ECT-probes for safety
    • This is deliberately conservative for multicast groups
ECN Usage with RTP: Translators

- Translator that doesn’t modify media
  - Multicast ↔ unicast; IPv4 ↔ IPv6
  - Pass ECN and RTCP unchanged

- Translator that combines or splits packets
  - Split → copy ECN marks; combine → pick worst ECN mark
  - Rewrite RTCP ECN feedback to match

- Translator that is a media transcoder
  - Must interpose translator into ECN negotiation
  - Must generate and respond to ECN feedback on each segment → non-trivial
ECN Usage with RTP: Mixers

• Mixer acts as an RTP endpoint for ECN purposes
  – Treats all paths independently
  – For each path:
    • Negotiate capability and check path support
    • Generate RTCP ECN feedback
    • Respond to ECN feedback
  – Possible that some paths support ECN, others don’t
Implementation Experiences

• Host capability to get/set ECN (TOS) bits
  – Set ECN/TOS on most platforms (setsockopt())
  – Get ECN per packet is only possible on Linux
    • setsockopt(IP_RECVTOS,), recvmsg() cmsghdr
  – Design to cope with differing hosts

• Network paths (tunnels, middleboxes, routers etc)
  – Currently most paths reset DSCP bits
  – Currently some paths reset ECN bits
  – Design to cope with differing paths

• Current implementation using UCL PhD’s (Soo Hyun)
  TFWC congestion control
Input and Future Directions

• Any questions or comments?

• Authors working on a combined Internet-Draft

• Desire that this becomes a working group draft
  – Suggest AVT as the formal home for the work, with regular review by TSVWG
  – Target: standards track