ECN and DSCP for Connect-UDP

Magnus Westerlund

Proposals

Two independent proposals of ECN for Connect-UDP was submitted:

- https://datatracker.ietf.org/doc/draft-seemann-masque-connect-udp-ecn/
  - Marten Seemann

- https://datatracker.ietf.org/doc/draft-westerlund-masque-connect-udp-ecn-dscp/
  - Magnus Westerlund
  - Mirja Kühlewind
  - Marcus Ihlar

Use Cases

ERICSSON

- Enable ECN to be negotiated end-to-end when tunneling:
  - WebRTC
  - Privacy Relay tunnel
  - 3GPP ATSSS to Private Networks
  - 3GPP Encapsulated XRM traffic (AS-> UPF)
- Traffic could be marked as ECT at any point in flow lifetime

- DSCP
  - Single DSCP per 5-tuple
  - Multiple DSCP per 5-tuple
    - WebRTC
    - UDP Tunnel Protocols

Two Alternative ECN Encoding

ERICSSON

- ECN codepoints bound to Context IDs
  - Context=0: (Not-ECT) UDP Payload
  - Context=2: ECT(0) + CID=0
  - Context=4: ECT(1) + CID=0
  - Context=6: CE + CID=0

- 1-Byte DSCP + ECN UDP Proxying Payload extension
  - Encodes DSCP + ECN in one byte prefixed before UDP Proxying Payload
  - CID=0: UDP Payload
  - CID=2: DSCP+ECN followed by CID=0

ECN on HTTP Transport Connection

ERICSSON

- Independent of ECN on End-to-End Traffic
- If ECN enabled on HTTP Transport Connection
  - If not Congestion Controlling E2E Traffic in datagrams
    - On each egress from transport connection mark up into encapsulated traffic or drop (RFC 6040 and RFC 9601)
  - If Congestion Controlled
    - Use AQM on Ingress to mark / drop end-to-end traffic
- HTTP Proxies
  - Handle each HTTP Transport connection independently

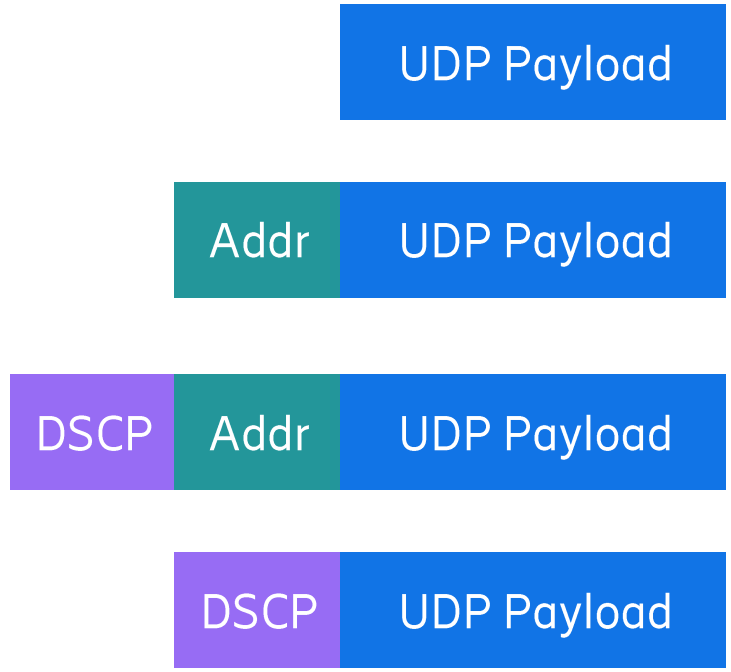
Setup Signaling

ERICSSON

- Capsules
  - Needed if dynamic allocation can occur in Connect-UDP request context
    - draft-ietf-masque-connect-udp-listen-07 **Proxying Bound UDP in HTTP**
      - Specifies dynamic binding
      - Intended for WebRTC
      - WebRTC has DSCP specified
    - ECN is defined for RTP (RFC 6679)
    - SCREAM v2 is an L4S enabled real-time media congestion controller
- Request and Response Header
  - Good for capability exchange
  - Possible for always established ECN using Context IDs
  - Does not work for dynamic payloads

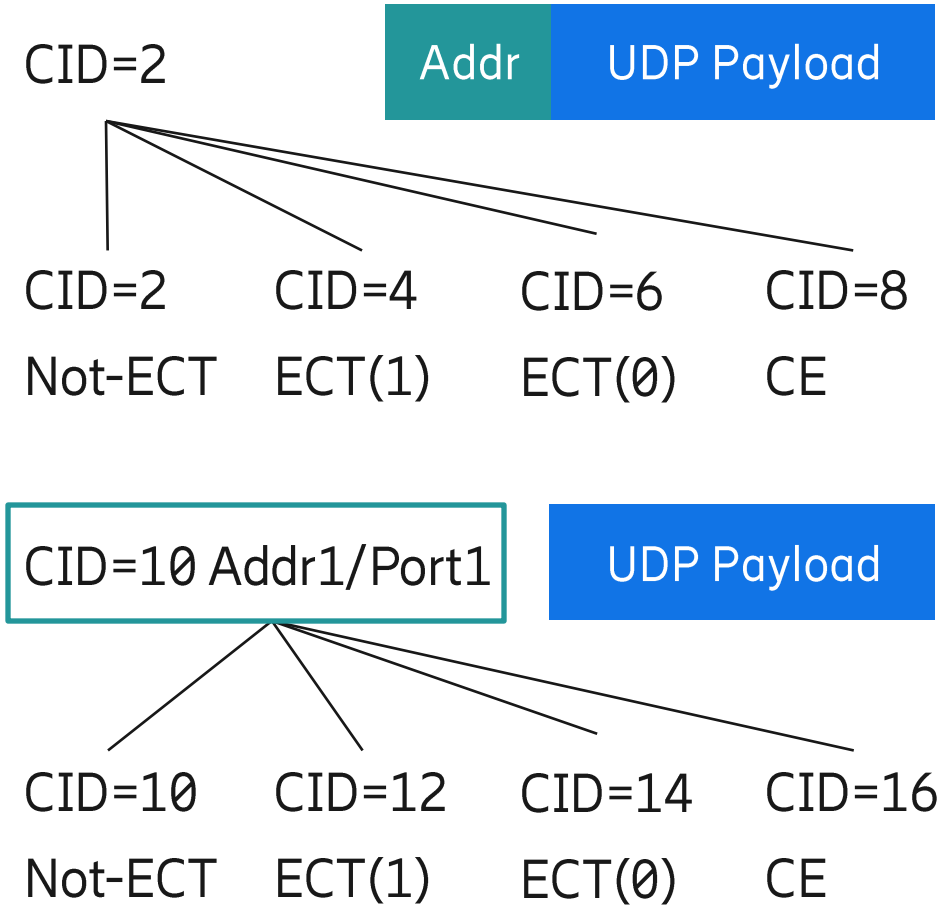
Chaining of Context IDs

- draft-ietf-masque-connect-udp-listen-07
  Proxying Bound UDP in HTTP
  - Creates a new UDP proxying Payload
- 1 Byte DSCP+ECN UDP Proxying Payload extension
  - Desirable to just specify a chain rather than having to define new payloads for all relevant combinations
  - Defines UDP Payload Extensions as DSCP+ECN followed by next fields per CID definition:
    - CID 4: DSCP + CID=2
    - CID 6: DSCP + CID=0



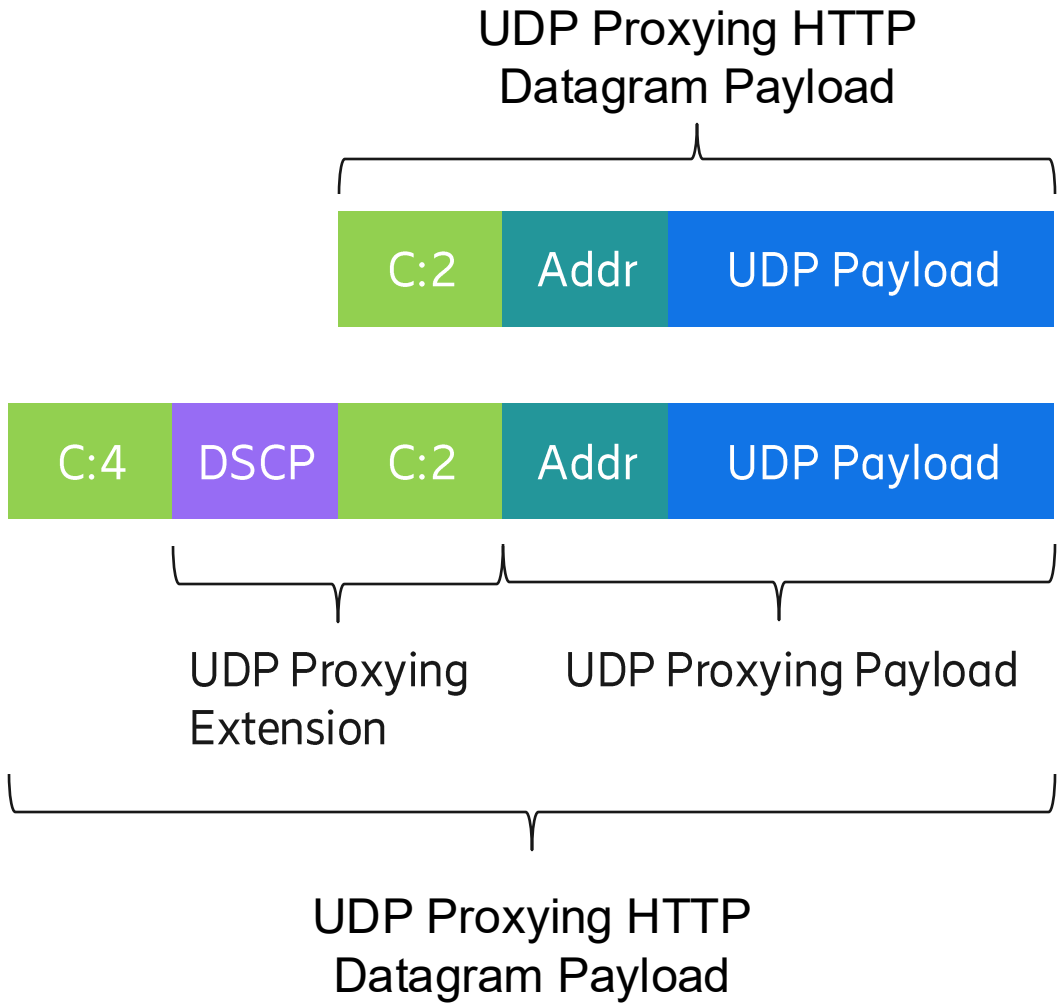
Combinatorial explosion when chaining ECN encoded in Context ID

- At a minimal a signaling time chain costs one additional Context ID per extension:
  - DSCP+ECN and Uncompress Bound
    - Requires Three context IDs (0, 2, 4)
- If encoding ECN bit values in Context ID
  - Three additional: i.e. four Context IDs
- Each address registration will consume four Context IDs
- If we would encode DSCP value also
  - Multiply with number of DSCP values
  - 3 DSCP values -> 12 Context IDs per compressed address



Encoding next CID in payload

- Alternative realization is to have each Extension include a CID Field to define what is next in payload
  - Single Context ID per extension in varint
  - Likely One byte per extension per packet
- Context ID Chaining package the combination value in a single varint



Way Forward

ERICSSON

- Appear to exist interest in the ECN Extension
- Would like to work in the WG to develop this extension
- Believe it is important to discuss acceptable solutions to the extension combination issue


ERICSSON