ECN and DSCP support for HTTPS's Connect-UDP

draft-westerlund-masque-connect-udp-ecn-dscp-01

Magnus Westerlund
Marten Seemann
Mirja Kühlewind
Marcus Ihlar

Updates since last meeting

- Marten Seemann joint as author to work on a common approach

- Clarifications
  - Renamed “ECN/DSCP” to “ECN plus DSCP”
  - documented that optimistic sending of UDP packets is a design goal
  - Rewrote the paragraph discussing congestion controlled tunnel transport
  - Fix for client initiated context IDs

Two Extensions

ECN-zero-byte Extension

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

```
UDP Proxying HTTP Datagram Payload {
  Context ID (i),
  UDP Proxying Payload (..)
}
```

ECN plus DSCP Extension

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

```
ECN/DSCP UDP Proxying Payload {
  DSCP(6),
  ECN(2),
  UDP Proxying Payload (..) # Payload per another Context ID
}
```

Two Options to negotiate use of each Extension

Request and Response Header

– ECN-zero-byte

  • `ECN-Context-ID: (10,12,14,8), (2,4,6,0)`

– ECN plus DSCP

  • `DSCP-ECN-Conext-ID: (10,8), (6,4), (2,0)`

Capsules (for dynamic allocation)

– ECN-zero-byte

```
ECN_CONTEXT_ASSIGN Capsule {
  Type (i) = TBA_2
  Length (i),
  ECN_CONTEXT_ASSIGNMENT (..) ...,
}
```

– ECN plus DSCP

```
DSCP_ECN_CONTEXT_ASSIGN Capsule {
  Type (i) = TBA_2
  Length (i),
  CONTEXT ASSIGNMENT (..) ...,
}
```

```
ECN_CONTEXT_ASSIGNMENT {
  ECT_1_CONTEXT (i),
  ECT_0_CONTEXT (i),
  CE_CONTEXT (i),
  PAYLOAD_CONTEXT (i)
}
```

Issue #14: Acknowledgement capsules

- Add an Ack capsule?
  - Enables the sender of the assignment capsule to start using the Context ID without risking loss
  - Is that a generally recommended approach or just overhead?

Issue #3 Indication of capsule support

- Is it the right mechanism to use an HTTP header with no configurations to indicate the capability to support later configuration using capsules?

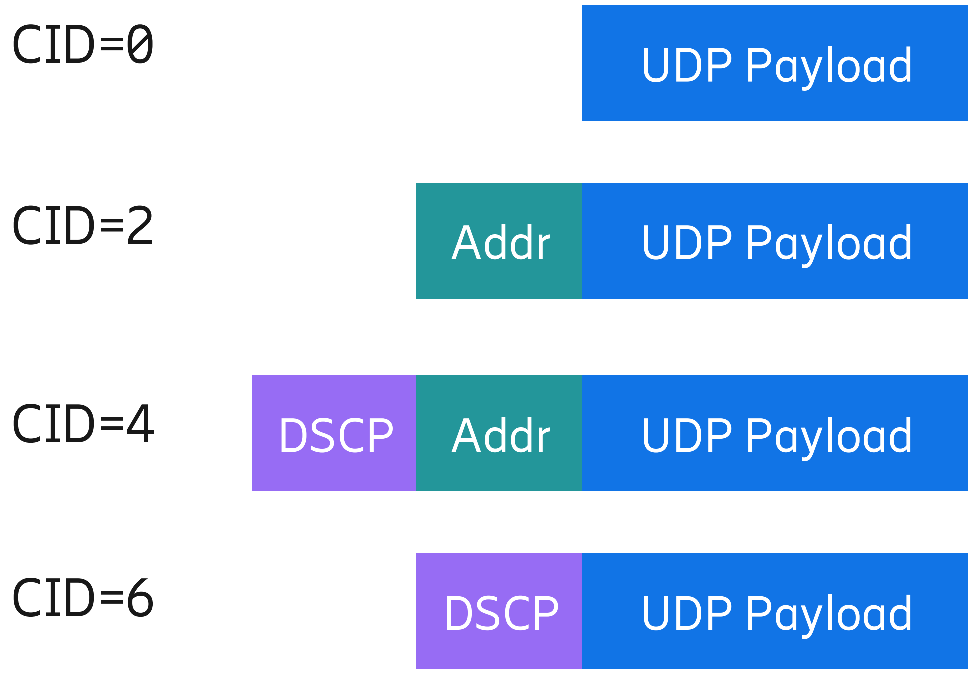
Issue [#17 Which extensibility solutions to support]

Options

1. Chaining of Context IDs: Field in context ID negoatiation that indicates the context ID following the ECN code points field (ECN Plus DSCP) or payload extension (ECN-zero-overhead)
   - Avoids overhead; requires use of more context IDs
   - Desirable to just specify a chain rather than having to define new payloads for all relevant combinations
2. Encoding next Context ID in payload
   - Simple but additional per-packet overhead
3. Define the combination of connect-UDP payloads explicitly in the extension
   - Risk combinatorial extension explosion?

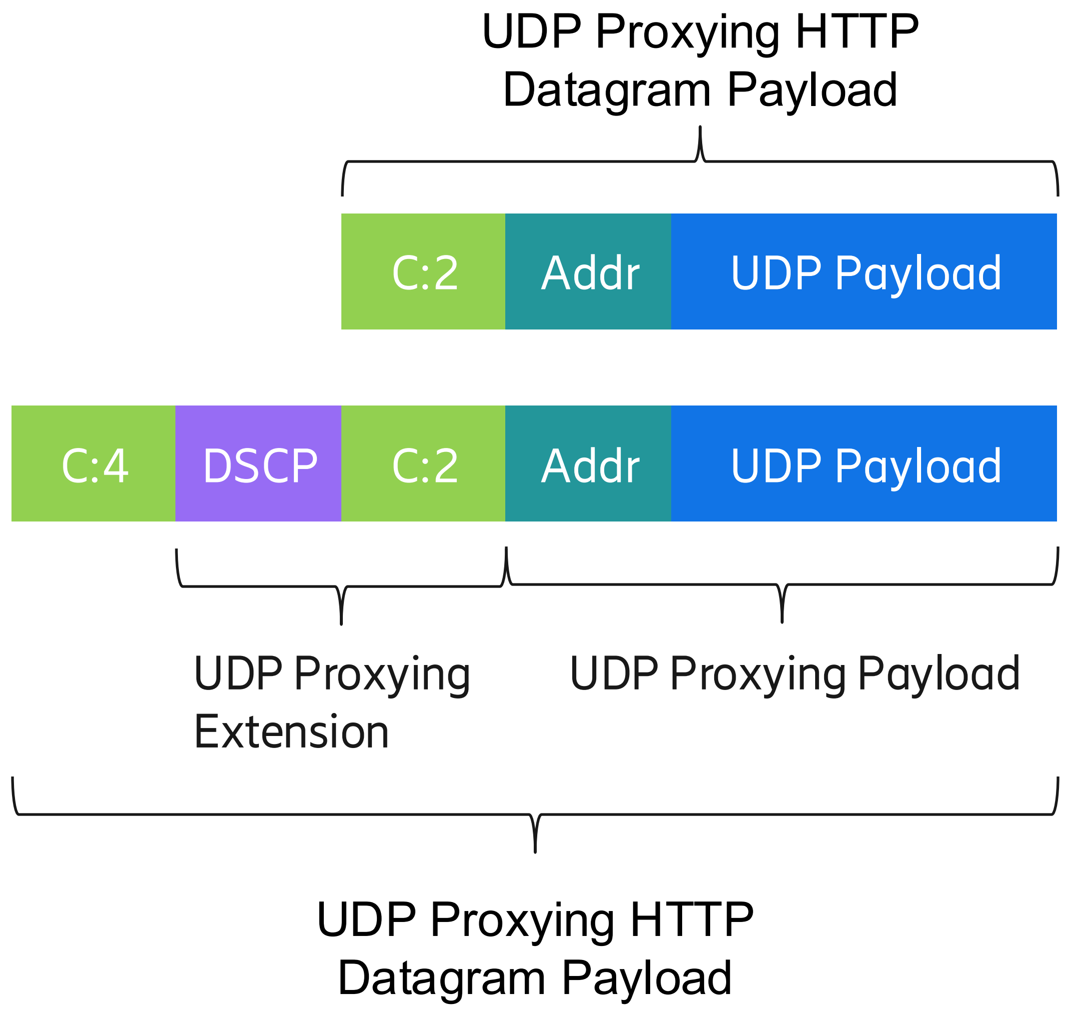
Option 1: Chaining of Context IDs

- Defines UDP Payload Extensions as DSCP+ECN followed by next fields per CID definition
- E.g. ECN Plus DSCP extension could be combined with **Proxying Bound UDP in HTTP** (draft-ietf-masque-connect-udp-listen-07)
  - CID 0: default UDP payload only
  - CID 2: connect-UDP listen with addr field
  - CID 4: DSCP + CID=2
  - CID 6: DSCP + CID=0



Option 2: Encoding next CID in payload

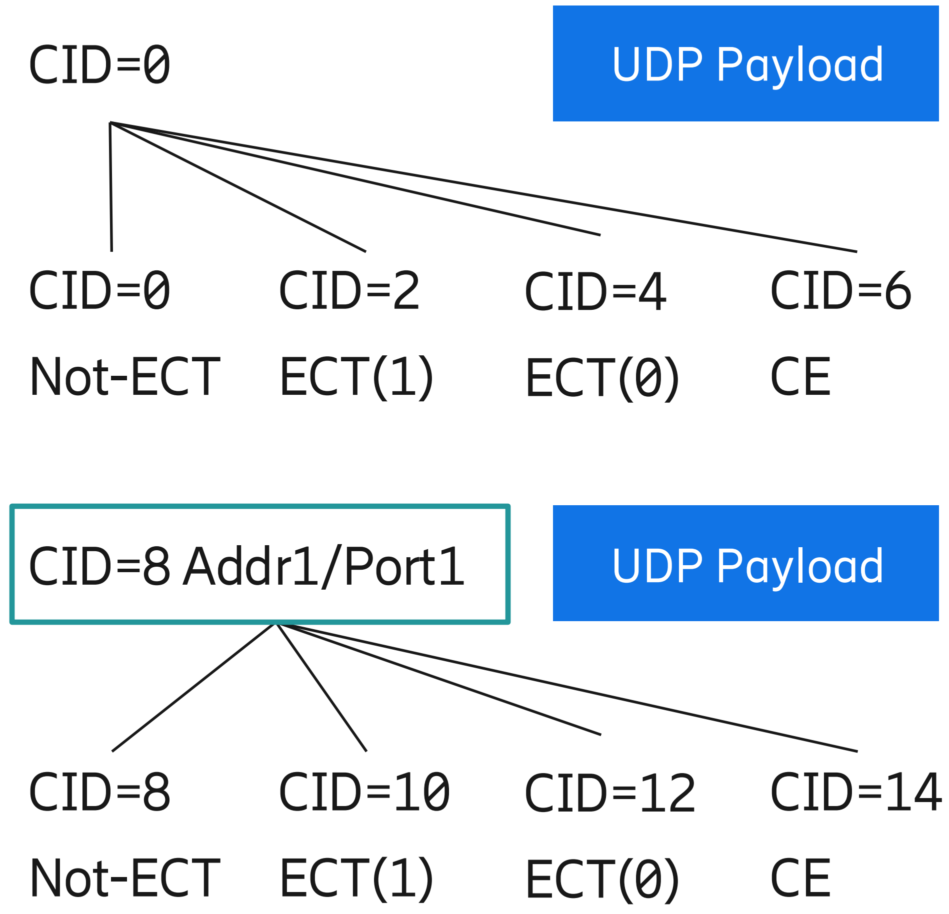
- Each Extension include a CID Field to define what is next in payload
  - Overhead accumulates with one byte per extension



Option 3: Chaining ECN encoded in one combined Context ID

Combinational Explosion

- ECN bit values in Context ID need three additional: i.e. in total four Context IDs
- Each address registration will consume another 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



ERICSSON