Large Record Sizes for TLS and DTLS

draft-ietf-tls-super-jumbo-record-limit-01

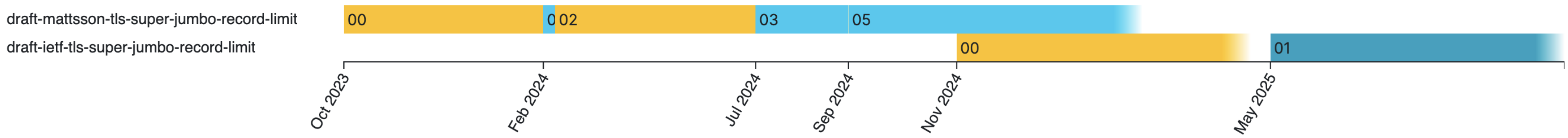
John Preuß Mattsson
Ericsson

Hannes Tschofenig
Univ. of Applied Sciences
Bonn-Rhein-Sieg

Michael Tüxen
Münster Univ. of Applied Sciences

Changes since IETF 119 “Brisbane”

- Major rewrite based on discussions at IETF 119
  - New independent extension ”large_record_size_limit” instead of flag extension used together with record_size_limit. LargeRecordSizeLimit denotes the maximum size, of inner plaintexts that the endpoint is willing to receive.
  - New record format TLSLargeCiphertext without opaque_type and legacy_record_version fields and with a 2–4 bytes length field. The size of the length field depends on the limit advertised by the receiver. This reduces overhead.
  - Support inner plaintext size up to 2^{32} - 256 bytes. This avoids fragmentation in almost all use cases. Not compatible with middleboxes expecting TLS 1.2 records.
  - Grammar and comprehension tweaks.
- **No changes since adoption (November 2024)**
- No known issues.



Large Record Sizes for TLS and DTLS

```
enum { u16(0), u24(1), u32(2) } Length;

struct {
    select (Length.type) {
        case u16: uint16;
        case u24: uint24;
        case u32: uint32;
    };
 } VarLength;

struct {
    VarLength length;
    opaque encrypted_record[TLSLargeCiphertext.length];
} TLSLargeCiphertext;
```

```
 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+
|0|0|1|C|S|L|E E|
+-+-+-+-+-+-+-+-+
| Connection ID |
| (if any,      |
/  length as    /
|  negotiated)  |
+-+-+-+-+-+-+-+-+
|  8 or 16 bit  |
|Sequence Number|
+-+-+-+-+-+-+-+-+
| 16, 24, or 32 |
|  bit Length   |
| (if present)  |
+-+-+-+-+-+-+-+-+
```

Next steps

- Draft is stable. No known issues.
- [Implementation in progress]
- More reviews?
- WGLC?

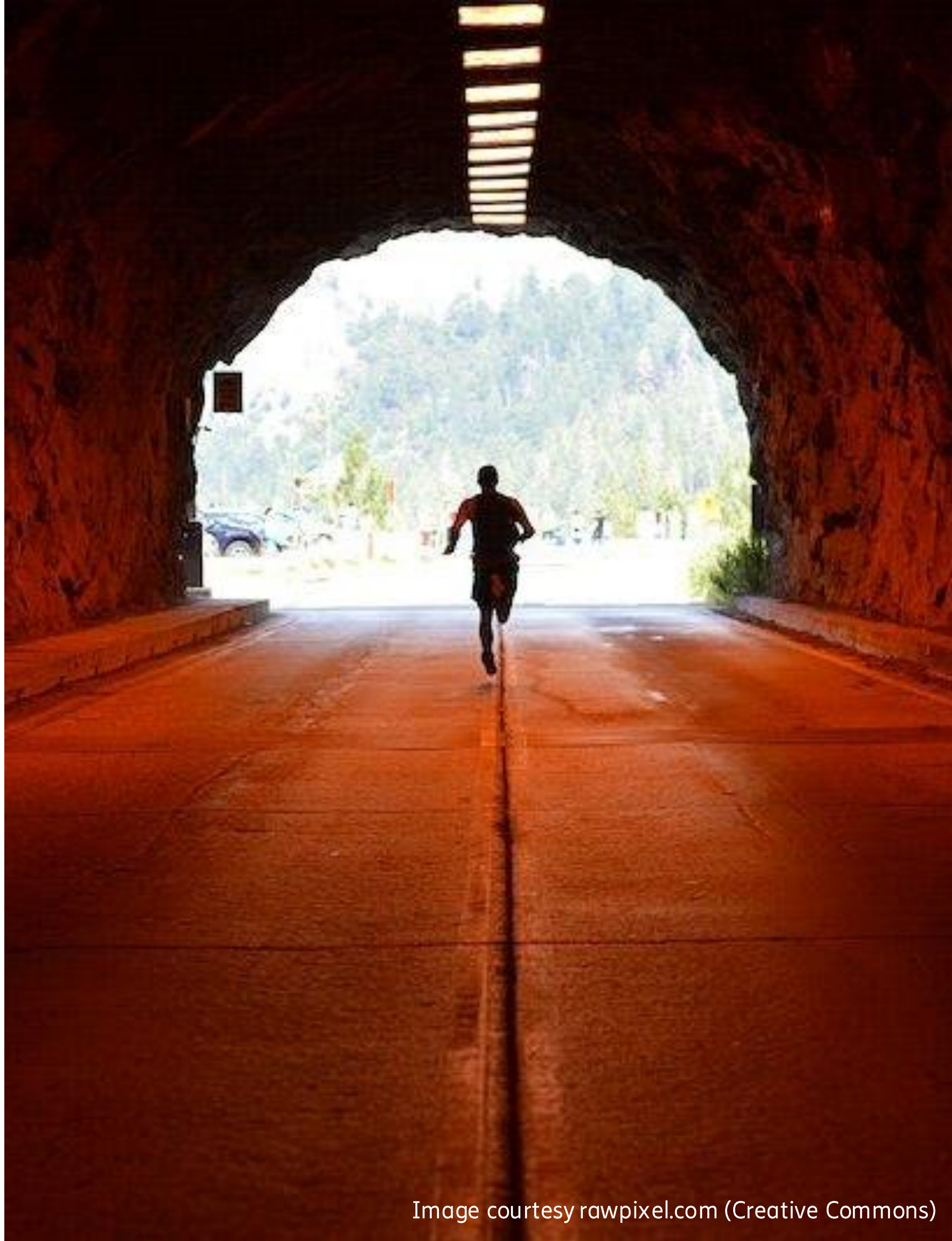


Image courtesy rawpixel.com (Creative Commons)