

[draft-ietf-lpwan-schc-yang-data-model-02](#)

Ana Minaburo

Laurent Toutain

Main changes

draft-ietf-lpwan-schc-yang-data-model-02

Available on github: <https://github.com/lp-wan/datamodel> (version 02/28)

- Just a YANG model, do not include CORECONF optimization
- Divided into 2 parts:
 - SCHC-ID : contains definition of types and identifier used in SCHC
 - Field-id id, MO id, CDA id
 - Type definitions for these fields
 - SCHC : defines the context model for compression and **fragmentation**
- Merged together when the model will be stable.

schc-id.yang

```

identity field-id-base-type {
    description "Field ID with SID";
}

identity fid-ipv6-version {
    base field-id-base-type;
    description "IPv6 version field from RFC8200";
}

identity fid-ipv6-trafficclass {
    base field-id-base-type;
    description "IPv6 Traffic Class field from
RFC8200";
}

identity fid-ipv6-trafficclass-ds {
    base field-id-base-type;
    description "IPv6 Traffic Class field from RFC8200,
DiffServ field from RFC3168";
}

identity fid-ipv6-trafficclass-ecn {
    base field-id-base-type;
    description "IPv6 Traffic Class field from RFC8200,
ECN field from RFC3168";
}

```

```

typedef field-id-type {
    description "Field ID generic type.";
    type identityref {
        base field-id-base-type;
    }
}

```

Subdivide some fields

Same for CoAP Code

- Class and details

Add types for fragmentation:

- Ack behavior
- RCS algorithm

```

identity /compression-decompression-action-base-type
identity /compression-decompression-action-base-type/cda-appiid
identity /compression-decompression-action-base-type/cda-compute-
n
identity /compression-decompression-action-base-type/cda-compute-
identity /compression-decompression-action-base-type/cda-deviid
identity /compression-decompression-action-base-type/cda-lsb
identity /compression-decompression-action-base-type/cda-mapping-sent
identity /compression-decompression-action-base-type/cda-not-sent
identity /compression-decompression-action-base-type/cda-value-sent
identity /direction-indicator-base-type
identity /direction-indicator-base-type/di-bidirectional
identity /direction-indicator-base-type/di-down
identity /direction-indicator-base-type/di-up
identity /field-id-base-type
identity /field-id-base-type/fid-coap-code
identity /field-id-base-type/fid-coap-code-class
identity /field-id-base-type/fid-coap-code-detail
identity /field-id-base-type/fid-coap-mid
identity /field-id-base-type/fid-coap-option-accept
identity /field-id-base-type/fid-coap-option-block1
identity /field-id-base-type/fid-coap-option-block2
identity /field-id-base-type/fid-coap-option-content-format
identity /field-id-base-type/fid-coap-option-end-option
identity /field-id-base-type/fid-coap-option-etag
identity /field-id-base-type/fid-coap-option-if-match
identity /field-id-base-type/fid-coap-option-if-none-match
identity /field-id-base-type/fid-coap-option-location-path
identity /field-id-base-type/fid-coap-option-location-query
identity /field-id-base-type/fid-coap-option-max-age
identity /field-id-base-type/fid-coap-option-no-response
identity /field-id-base-type/fid-coap-option-observe
identity /field-id-base-type/fid-coap-option-proxy-scheme
identity /field-id-base-type/fid-coap-option-proxy-uri
identity /field-id-base-type/fid-coap-option-size1
identity /field-id-base-type/fid-coap-option-size2
identity /field-id-base-type/fid-coap-option-uri-host

```

```

identity /field-id-base-type/fid-coap-option-uri-port
identity /field-id-base-type/fid-coap-option-uri-query
identity /field-id-base-type/fid-coap-tkl
identity /field-id-base-type/fid-coap-token
identity /field-id-base-type/fid-coap-type
identity /field-id-base-type/fid-coap-version
identity /field-id-base-type/fid-ipv6-appiid
identity /field-id-base-type/fid-ipv6-appprefix
identity /field-id-base-type/fid-ipv6-deviid
identity /field-id-base-type/fid-ipv6-devprefix
identity /field-id-base-type/fid-ipv6-flowlabel
identity /field-id-base-type/fid-ipv6-hoplimit
identity /field-id-base-type/fid-ipv6-nextheader
identity /field-id-base-type/fid-ipv6-payloadlength
identity /field-id-base-type/fid-ipv6-trafficclass
identity /field-id-base-type/fid-ipv6-trafficclass-ds
identity /field-id-base-type/fid-ipv6-trafficclass-ecri
identity /field-id-base-type/fid-ipv6-version
identity /field-id-base-type/fid-udp-app-port
identity /field-id-base-type/fid-udp-checksum
identity /field-id-base-type/fid-udp-dev-port
identity /field-id-base-type/fid-udp-length
identity /field-length-base-type
identity /field-length-base-type/fl-token-length
identity /field-length-base-type/fl-variable
identity /matching-operator-base-type
identity /matching-operator-base-type/mo-equal
identity /matching-operator-base-type/mo-ignore
identity /matching-operator-base-type/mo-matching
identity /matching-operator-base-type/mo-msb

```

CoAP identityref

- divide fields into sub-fields (coap-code-class, coap-code-detail,...)
- CoAP option naming space:
 - Carsten proposed to reserve the whole space to link the option repository to the id
 - How can we do that in Yang ?
 - What size we reserve ?
 - Largest one in IANA : 2053 OCF-Content-Format-Version [[Michael Koster](#)]

0-255 IETF Review or IESG Approval

256-2047 Specification Required

2048-64999 Expert Review

65000-65535 Experimental use (no operational use)

- may be a waste of space, what procedure when new option created ?
- CoAP End Option (0xFF) is treated as an option
 - Conflict if Core uses this value for a specific option.

SCHC model

```

module: schc
++-rw schc
  +-rw version? uint64
  +-rw rule* [rule-id rule-length]
    +-rw rule-id          uint32
    +-rw rule-length      uint8
  +-rw (nature)?
    +-:(fragmentation)
      +-rw direction?     schc-id:direction-indicator-type
      +-rw dtagsize?      uint8
      +-rw wsize?          uint8
      +-rw fcsize?         uint8
      +-rw RCS-algorithm? RCS-algorithm-type
      +-rw maximum-window-size? uint16
      +-rw retransmission-timer? uint64
      +-rw inactivity-timer? uint64
      +-rw max-ack-requests? uint8
      +-rw maximum-packet-size uint16
    +-rw (mode)
      +-:(no-ack)
      +-:(ack-always)
      +-:(ack-on-error)
        +-rw tile-size?     uint8
        +-rw tile-in-All1?   boolean
        +-rw ack-behavior   schc-id:ack-behavior-type

```

after-All0, after-All1, always

```

+--:(compression)
  +-rw entry* [field-id field-position direction-indicator]
    +-rw field-id          schc-id:field-id-type
    +-rw field-length       schc-id:field-length-type
    +-rw field-position     uint8
    +-rw direction-indicator schc-id:direction-indicator-type
    +-rw target-values* [position]
      | +-rw value?         union
      | +-rw position       uint16
    +-rw mo
      +-rw mo-value* [position]
        | +-rw value?         union
        | +-rw position       uint16
    +-rw cda
      +-rw cda-value* [position]
        +-rw value?         union
        +-rw position       uint16

```

True, false
Yes, No, Optional

Summary

- Added version
 - TV generic
 - New identities
 - Fragmentation
-
- Open question: CoAP options
 - Add more description
 - Model is stable, YANG doctor review ?