draft-ietf-lpwan-schc-yang-data-model-04
featuring … ”feature”

Laurent Toutain (laurent.toutain@imt-atlantique.fr)

Ana Minaburo (ana@ackl.io)

```
feature fragmentation {
    description "Fragmentation is usually required only at the lowest level.";
}
```

```
container schc {
    leaf version{
        type uint64;
        description "used as an indication for versioning";
    }
    list rule {
        key "rule-id-value rule-id-length";
        uses rule-id-type;
        choice nature {
            case fragmentation {
                if-feature "fragmentation";
                uses fragmentation-content;
            }
            case compression {
                uses compression-content;
            }
            description "A rule is either for compression or fragmentation";
        }
        description "Set of rules compression or fragmentation rules identified by
        their rule-id ";
    }
description "a SCHC set of rule is composed of a list of rule which are either
compression or fragmentation";
}
```

```
module: ietf-schc
  +--rw schc
     +--rw version?   uint64
     +--rw rule* [rule-id-value rule-id-length]
        +--rw rule-id-value                uint32
        +--rw rule-id-length               uint8
        +--rw (nature)?
           +--:(fragmentation) {fragmentation}?
              |  +--rw direction              schc:direction-indicator-type
              |  +--rw dtagsize?              uint8
              |  +--rw wsize?                 uint8
              |  +--rw fcnsize                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 fragmentation-mode     schc:fragmentation-mode-type
              |  +--rw (mode)?
              |     +--:(no-ack)
              |     +--:(ack-always)
              |     +--:(ack-on-error)
              |        +--rw tile-size?       uint8
              |        +--rw tile-in-All1?    schc:all1-data-type
              |        +--rw ack-behavior?    schc:ack-behavior-type
              +--:(compression)
                 +--rw entry* [field-id field-position direction-indicator]
                    +--rw field-id                   schc:field-id-type
                    +--rw field-length               schc:field-length-type
                    +--rw field-position             uint8
                    +--rw direction-indicator        schc:direction-indicator-type
                    +--rw target-values* [position]
                    |  +--rw value?      union
                    |  +--rw position    uint16
                    +--rw matching-operator          schc:matching-operator-type
                    +--rw matching-operator-value* [position]
                    |  +--rw value?      union
                    |  +--rw position    uint16
                    +--rw comp-decomp-action         schc:comp-decomp-action-type
                    +--rw comp-decomp-action-value* [position]
                       +--rw value?      union
                       +--rw position    uint16
```