

CoRE: CORECONF & SIDs

YANG: NETCONF, RESTCONF, CORECONF

- NETCONF is based on YANG/XML -- text-based
 - appropriate for rack-mounted hardware
- RESTCONF adds YANG/JSON -- still text-based
 - still appropriate for rack-mounted hardware
- CORECONF adds **YANG/CBOR** -- binary, concise
 - can replace SNMP even on low-resource hardware

The YANG Name Problem

```
identifier          = (ALPHA / "_")
                    *(ALPHA / "_" / DIGIT / "-" / ".")
node-identifier     = [prefix ":" ] identifier
absolute-schema-nodeid = 1*("/") node-identifier
```

...

YANG Names often take up bulk of NETCONF/RESTCONF

- transmission
- processing

Going to binary names

- Prior art: ASN.1 Object Identifiers [SNMP!]
 - still variable length byte string
 - mirrors registry structure, can get quite long
- Radical idea: use a single **unsigned integer**
 - but how to handle extensibility?

YANG SID

YANG SID: unsigned 63-bit integer

- allocated in little **SID ranges** (~ 100 SIDs at a time)
- most frequently interchanged as a delta
 - locality leads to mostly single-byte deltas
- wasting some number space is not a big problem!

SID evolution/lifecycle

YANG modules use up a complete SID range (~ 100+)

- keep 1/3 to 1/2 free

Updates to YANG modules use up remaining space

- deltas stay small

Allocate a second SID range if needed

- deltas do get larger; sorry about that

SID and YANG module development

Unstable SIDs are assigned during development

- can change around randomly between draft revisions

Made stable on \$EVENT (approval?)

Updates keep the stable SIDs, add unstable SIDs

- `pyang` support for adding SIDs to a stable baseline (SID files)

How to get SID ranges

SID ranges are allocated out of **mega-ranges** (~1 000 000)

- IANA gets 0 to 999 999 for its own range registry
- SDOs can get one or more mega-ranges
- Organizations (vendors, open source projects), too

Minimum requirements on **capacity** to manage and operate a YANG SID Range Registry out of the mega-range

SID space management Objectives

- **MUST**: Keep stable mapping from SID to YANG name
- **SHOULD**: Keep number of SIDs allocated per name == 1

[draft-ietf-core-sid](#) contains processes that are intended to maintain the MUST and help the SHOULD

(In IESG processing, fine tuning remains)

SIDs: jobs for the community

- Get a mega-range from IANA (if **capacity** is there:)
 - Allocate ranges to YANG module developers
- Provide infrastructure for lookup (SID \Leftrightarrow YANG Name)
 - optional! — can avoid disclosure if not desired
- Make sure YANG tools handle SIDs well