CoRE: CoRECONF

— RFC 9254: YANG-CBOR

— In IESG: CORE-SID

— WGLC passed CORE-COMI

— WGLC passed CORE-YANG-LIBRARY

CoRE Topic Interim 2023-03-15: CORECONF/YANG
Meetecho • Notes

Agenda

issues in the area of CORECONF and other YANG-related efforts:
focus today's interim on:
bringing the YANG and CoRE ecosystems closer together.

- SID allocation (core-sid and LPWAN)
- COMI simplification/optimization
- More:
  - CoRE YANG Library
  - YANG models relevant to CoRE
  - Binary YANG Push

SID allocation

<draft-ietf-core-sid-20.txt>
<draft-toutain-lpwan-sid-allocation-02.txt>

CORE-SID: IESG processing + parallel WGLC (... 2023-03-16)

CORE-SID -20 is meant to deal with Rob Wilton's DISCUSS, adding status information to SID files:

- global file status
- per-SID status

SID completeness

Issue: no tool to validate examples/SID file against YANG
➔ manual fixes, but:

Does PYANG generate a complete set of SIDs?

— What would a complete set be?
— Evolve PYANG to emit this complete set,
or do we add the missing parts manually?
— (new SIDs can always be added, so missing one is not catastrophic.)

ietf-system (Appendix A of core-sid-20)

RPCs, actions; example:

```
1715,data,/ietf-system:set-current-datetime,
1716,data,/ietf-system:set-current-datetime/current-datetime,
1718,data,/ietf-system:system-restart,
1719,data,/ietf-system:system-shutdown,
```

what about input/output? more correct:?

```
171*,data,/ietf-system:set-current-datetime/input,
1716,data,/ietf-system:set-current-datetime/input/current-datetime,
```

SID process

<draft-toutain-lpwan-sid-allocation-02.txt>

… proposes allocations for certain LPWAN protocols.

— How do we (DE) check a document like this?

— Is it complete?

See also Appendix A of draft-bormann-cbor-cddl-csv-02 for an efficient way to discuss SID allocations.

LPWAN slides...

COMI: Access to YANG information bases via CoAP

<draft-ietf-core-comi-12.txt>: CORECONF ➔ CoAP
like RFC 8040 for RESTCONF ➔ HTTP

Already went through a Working Group Last-call
-11 discussed at IETF 115 hackathon

➔ simplifications (removing a fundamental bug as well)

concerns?

simplify/optimize URI form of instance identifiers (1)

Beyond -12:
further aligning FETCH... payload and GET... URI?

Using RFC 9254 (YANG-CBOR) notation:
example instance identifier: `[1533, "eth0"]`
COMI-12 uses the URI:

```
/c/X9?ZGV0aDA
1533^ ^"eth0"
```

Compare GET for `/c/X9` for entirety of SID 1533

simplify/optimize URI form of instance identifiers

compare FETCH: binary encoding of the CBOR data item `[1533, "eth0”]` in payload.
Do the same (as a base64-encoded CBOR sequence) in the URI?

```
comi-12: /c/X9?ZGV0aDA      ; separate 1533 and rest ["eth0"]
beyond:  /c/GQX9ZGV0aDA      ; combine [1533, "eth0"]
```

Or maybe inverse direction:
add special case "instance identifier with a single string key"

```
inverse: /c/X9?k=eth0        ; special-case looks good in examples
```

not sure it’s worth the additional code.

A YANG Library tailored to constrained environments:

<draft-ietf-core-yang-library-03.txt>
encodes a simplified form of the regular YANG library optimized for constrained devices

— yang-library next steps:
— Has the YANG environment changed, requiring us to make fundamental changes?
— Any small updates needed?

YANG models relevant or close to CoAP/CoRE:

<draft-marin-yang-edhoc-oscore-00.txt>
<draft-toutain-lpwan-access-control-01.txt>
<draft-tiloca-lpwan-8824-update-00.txt> (YANG model for that)

— “eat our own dog food”

— constrained vs. non-constrained use

What can we learn from these YANG models
to make use of for making CORECONF better?

Any best practices we could extract to tell model developers?

Binary YANG Push

<draft-ietf-netconf-udp-notif-09.txt>
<draft-ahuang-netconf-notif-yang-01.txt>

draft-ietf-netconf-udp-notif looks seriously weird

- text-based encaps for sequence of binary payloads?
- free get-out-of-jail card for congestion control?

draft-ahuang-netconf-notif-yang also interesting

anything else?
Bring your own issues!