Outcome and outlook of the Friday T2TRG meeting

draft-core-transport-indication, draft-amsuess-coap-over-gatt, draft-core-resource-directory-extensions, draft-t2trg-onion-coap
… all about Naming Things

Christian Amsüss

IETF118 Prague, CoRE, 2023-11-09

Orientation

Origin: RFC 8323 introducing `coap+tcp://`

led to: draft silverajan-core-coap-protocol-negotiation (stalled)

used by: draft bormann-t2trg-slipmux (`coap+uart://`), draft amsuess-core-coap-over-gatt (`coap+gatt://`), draft becker-core-coap-sms-gprs (`coap+sms://`)

reboot: draft ietf-core-transport-indication – `rel=has-proxy`

more users: draft amsuess-t2trg-rdlink, draft amsuess-t2trg-onion-coap (both `coap://encodedpublickey.tbd.arpa`)

T2TRG meeting on Friday

Do we really need coap+foo?

Ambiguous: $\texttt{coap://}\underbrace{\texttt{[2001:db8::1]:1234}}_{\text{TCP port? UDP port?}}\texttt{/}$

Clarified: coap+tcp://[2001:db8::1]:1234/

Unambiguous:

- coap://0123456789ab.ble.arpa/ (was coap+ble://...)
- coap://7.6.5.4.3.2.1.5.5.5.3.4.e164.arpa/ (was coap+sms://+43555...)
- coap://ttyUSB0.uart.tbdlocal.arpa/[1]
- coap://001.004.usb.tbdlocal.arpa/

Criterion: coap+foo needed if the literal for of authority values are ambiguous with respect to transports. Or if the resolution process produces such values.

[1] Or ttyUSB0.alt?

Proposal to update `transport-indication`

- Recommend not adding `coap+x` schemes unless criterion is met.

Doesn't this create aliasing?

No. It's only aliasing when differert URIs point to the same resource.
Whether a server uses the same name for different transports or not is up to the operator.

...and CoAP-over-GATT (and slipmux?) will follow.[2]

[2] No development is known to be going on with the other transports.

Can't just rely on literals!

How do I know what to do with coap://c.example.com?

Naming systems provide metadata.

```
$ dig c.example.com
c.example.com. 32000 IN AAAA 2001:db8::42
```

insufficient literal[1]

but

```
$ dig BLE c.example.com
c.example.com. 32000 IN    BLE[2]  0123456789ab
```

Not much more, but has own discovery

[1] The literal is OK on its own, but insufficient when combined with just a port.

[2] With no desire to do this here or precisely that way, but it would suffice.

Other lookup systems can provide similar metadata

Extrapolating from draft core-resource-directory-extensions for forward proxy setup

```
GET coap://thedirectory.net/rd-lookup/
          ?anchor=coap://00publickey00.onion-ish.arpa

2.05 Content
Payload:
<coap+tcp://[2001:db8::5]/>;rel=has-proxy;
          anchor="coap://00publickey00.onion-ish.arpa"
```

Alternative history

If we had RFC 9460 SVCB back when RFC 8323 was done[3]...

```
$ dig SVCB _coap.c.example.com
c.example.com. 32000 IN SVCB 1 . transport="tcp"
c.example.com. 32000 IN SVCB 2 . transport="udp" port=61616
c.example.com. 32000 IN SVCB 3 . transport="udp"
```

Adopt for future IP based transports; solves no problems for us any more unless anyone wants to deprecate `+tcp` and `+ws`.

[3] CoAP over TCP was approved 2017-12-18, first draft schwartz-httpbis-dns-alt-svc was published 2018-01-16

Going forward

- Gather any objections here.
- Update GATT and transport-interfaces drafts to not do `coap+x` unless criterion is met.