CoAP Transport Indication\(^1\)

ietf-core-transport-indication-05

Christian Amsüss, Martine Lenders

2024-03-20

CoRE at IETF 119 in Brisbane

\(^1\)Used to be Protocol Indication, document name was transport-indication already
Goals

1. Enablement of transport discovery
2. No Aliasing
3. Optimization (no cost per request)
4. Proxy usability
5. Proxy announcement

- Give way forward after `coap://` and `coap+tcp` diverged
Changes since IETF118

Why did coap:// and coap+tcp diverge? – rehash of IETF118

Ambiguous: coap://[2001:db8::1]:1234/

TCP port? UDP port?

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

Unambiguous:

- coap://0123456789ab.ble.arpa/ (was coap+ble://...)
- coap://ttyUSB0.dev.alt/ (was coap+uart://ttyUSB0/)

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.
Guidance to new transports: `coap://`
... and if is IP based, do SVCB as described in the appendix E.
... and if you need that with IP literals, finish appendix F.

Security guidance simplified

Consistently talk of “transports”
IETF118 changes are applied.

Let’s get this done...
Open question: Scope of has-proxy

“only through link relations”
- URIs regarded as opaque
- Relations are explicit
- Allows to exclude individual resources from transport indication
- Uses RFC 6690 rel=hosts which is not very clear
- Hard to keep track of what works where

vs.

“Applies per Origin”
- HTTP’s mechanism
- Simple
- Way shorter wording in terminology section

\(^2\)No support adding transport indication for individual resources is planned
Open question: Self description style guide
e.g. after multicast discovery

“Explicitly using canonical names”

<coap://myhostname/foo>,...,  
<coap://[2001:db8::1]>;rel=has-unique-proxy;anchor="coap://myhostname"

vs.

“Relying on client to re-interpret the document after having parsed parts of it”

</foo>,...,  
<coap://[2001:db8::1]>;rel=has-unique-proxy;anchor="coap://myhostname"

... Or do we not care so much about RFC 6690 anyway, because other formats would allow setting a base anyway?
Open question: Appendices

- Is alternative history fiction a thing here?
- (Where) should literals for service parameters (and other data from DNS) go?
Summary

Open questions:
- Decide “only through link relation” vs “always when Origin matches”
- Advertising self-proxies / importance of RFC 6690
- Fate of appendices

What else before this is done?

Interop?