# CoAP Protocol Indication

draft-amsuess-core-transport-indication-02

Christian Amsüss

2021-10-27, CoRE interim

# Recap: What this is about

Different schemes for CoAP exist, URI aliasing happens.

Proxying can be used to send over different transports without aliasing.

New link relations explain existing aliasing, and allow protocol selection...

...all while not really changing anything.

# Proposed mechanism

```
</cfe>;rt="tag:...:coffemachine";rel=hosts;anchor="/",
<coap+tcp://[2001:db8::1]/>;rel=has-proxy;anchor="/"
```

Request can go to through TCP but still request canonical URI `coap://`.

## Goals (1-2/5)

**Enablement** Inform clients of the availability of other transports of servers.

**No Aliasing** Any URI aliasing must be opt-in by the server. Any defined mechanisms must allow applications to keep working on the canonical URIs given by the server.

# Optimized mechanism with trade-offs

```
<coap+tcp://[2001:db8::1]/>;rel=has-unique-proxy;anchor="/"
```

Request can go to `coap+tcp://` on the wire; application still think in terms of `coap://`.

## Goals (3/5)

**Optimization** Do not incur per-request overhead from switching protocls. This may depend on the server's willingness to create aliased URIs.

By the way, this goes for the host name as well as for the scheme. ("Do I have to send the Uri-Host option?" can now be answered.)

# Proxy interaction

## Goals (4/5)

**Proxy usability** All information provided must be usable by aware proxies to reduce the need for duplicate cache entries.

With only `has-proxy`, acutal proxies always see one URI. Proxies that see the `has-proxy` statement may forward requests through the other transport.

With `has-unique-proxy` (or as-things-are-now), actual proxies may see any aliased URI. Proxies that see the `has-unique-proxy` statement may contract caches at canonical URI.

# Proxy interaction

## Goals (5/5)

**Proxy announcement** Allow third parties to announce that they provide alternative transports to a host.

...with security considerations: Security can not be downgraded, access to ciphertext is a per-application trade-off.

Bycatch: Also define how forward proxies can announce themselves:
`<>;rt=TBDcore.proxy;proxy-schemes="coap coap+tcp coap+ws http"`
Overlap in applications and security considerations.

# Other news in -02

- Considerations for MPTCP
- Concrete RD extension proposal to query suitable proxies along the way
- Security considerations simplified by...

# Security Considerations

## Just As With Any Proxy.

OK, there's more in the text, but that's the gist.

~~Are all-valid certificates common when 3rd-party proxies are used with DTLS? Do we want to endorse them?~~
With input from IETF111: No, we don't do that.

# Take-home message

- It can probably be just this simple.
- No URI aliasing introduced in applications.

Questions? Comments? Way forward?

# Backup slide / FAQ

*Didn't we want to do this with DNS?*

We[1] still can, just need to phrase the equivalent statements in DNS.

Straw man for "`coap://device.example.com` has CoAP-over-TCP running on port 1234":

```
_has-coap-proxy._tcp.device.example.com SRV 0 0 device.example.com 1234
device.example.com AAAA 2001:db8::1
```

*How does this relate to HTTP's `Alt-Svc`?*

Generally similar; links instead of headers (as common in CoAP), and no need for protocol-id because we have schemes already.

[1] Whoever wants to use it will need to volunteer as coauthor.