

rdlink: Robust distributed links to constrained devices
draft-amsuess-t2trg-rdlink-00

Abstract

Thing to thing communication in Constrained RESTful Environments (CoRE) relies on URIs to link to servers. Next to hierarchical configuration and short-lived IP addresses, this document introduces a naming scheme for devices based on cryptographic identifiers. A special purpose domain is reserved for expressing those identifiers, and mechanisms for constrained devices to announce their names and to look them up are described.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 26, 2019.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in [Section 4](#).e of

the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

1. Introduction

Communication between constrained devices using the CoAP protocol largely happens in either of two scenarios at the moment: local networks with static IP addresses, and communication mediated by dedicated servers.

Direct communication between constrained devices across the Internet is currently an exceptional setup, and usually involves static configuration or bespoke mechanisms.

Direct communication with a CoAP servers is often guided by web links which point to the URI that both names the server (and a resource on it), and indicates how that server can be reached. Such links often indicate registered name which is typically looked up in the DNS hierarchy and thus relies on the operator to own and administer a domain, If they don't, they indicate an IP address; such links are of limited use for stable identifiers, e.g. due to mobile endpoints.

This document introduces a special purpose domain (rdlink.arpa) along the mechanisms with which it is used (employing a Distributed Hash Table (DHT)).

Constrained devices can announce and look up addresses without direct interaction with the DHT by interacting with a distributed resource directory ([[I-D.ietf-core-resource-directory](#)]).

Resolvable names are provided for compatibility with applications that are unaware of these provisions.

2. Terminology

2.1. Participants

This document uses several roles of devices:

named server A CoAP server owns a name in the special purpose domain.

name user A CoAP client that needs to resolve a name in the special purpose domain.

registration helper A server that assists constrained named servers in announcing their names.

lookup helper A server that assists constrained name users in finding the named server.

DHT participant Any device that is a node in the DHT overlay network.

Often, named servers act as name users towards other servers. The roles of registration helper, lookup helper and DHT participant are expected to be implemented together in typical use cases.

Note that a named server can act as a CoAP client towards a name user that has ongoing communication with it without being a name user on its own by just addressing the client on its own.

3. Auxiliary Mechanisms

This section describes mechanisms that are expected to be specified in different documents, which will then only be referenced.

3.1. The coap+at scheme

It is assumed that the CoRE working group at IETF will register a URI scheme "coap+at://" that can be used with DNS names, and that allows expressing CoAP URIs independent of the used transport.

The mechanisms outlined in [\[I-D.silverajan-core-coap-protocol-negotiation\]](#) are assumed to be one way of finding protocol URIs (e.g. "coap+tcp://...") that correspond to "coap+at" URIs when they are known to a Resource Directory.

3.2. The alternative-transport Link Relation Type

It is assumed that [\[I-D.silverajan-core-coap-protocol-negotiation\]](#) will have a parameter equivalent to the following:

This defines the "alternate-transport" Link Relation Type.

A link from a context resource to a target resource typed with the "alternative-transport" type declares that for any relative reference of the path-noscheme or path-empty form (see [\[RFC3986\] Section 4.2.](#)), the reference's resolution with the context as a Base URI can be substituted with the reference's resolution with the target as a Base URI.

The expression "can be substituted with" means that for every REST operation conducted on the original resource, the same operation on the new resource will give equivalent responses and have equivalent side effects.

Applications interpreting "alternative-transport" links need to carefully consider their trust model: They MUST either have obtained the statement from a source that is trusted to speak for the context authority, or make additional demands on the target when connecting to it (e.g. ask the target to identify as the context authority).

[If applications are defined for both CoAP and HTTP, and advertised the same way, hosts can only advertise alternatives if cross-proxying is possible; needs good generic phrasing.]

This link relation is roughly equivalent to the "at" RD parameter introduced in [[I-D.silverajan-core-coap-protocol-negotiation](#)], but suitable for multicast discovery.

4. The rdlink.arpa special purpose domain

The domain rdlink.arpa is reserved to represent devices by their cryptographic identifier (described using the "cryptident" ABNF of the next section).

The rdlink.arpa domain does not provide DNS records for those names, but serves as a plain name for devices eligible to use their cryptident.

hostname = cryptident ".rdlink.arpa"

Names from this domain should probably only be used with the "coap+at" scheme, like this (assuming a host's cryptident is "nbswy3dpo5xxe3denbswy3dpo5xxe3de.ab"):

coap+at://nbswy3dpo5xxe3denbswy3dpo5xxe3de.ab.rdlink.arpa/.well-known/core

4.1. cryptident construction

cryptident = [ext-ident "."] cryptmain "." crypttype

The "cryptident" component describes a name under and describes a cryptographic identity the host can show, e.g. a public key, or the hash of a certificate.

The cryptmain component is base32 encoded binary data (as in [[RFC4648](#)], but lower case and without padding).

The crypttype is a registered designator for the meaning of cryptmain and ext-indent, which initially only has one value:

[Names are currently given with the same encoding as cryptmain to map to numbers, that may be a good or a bad idea.]

- o "ab": cryptmain is a 160bit number that represents an ed25519 public key.

A host proves being eligible to use this name by performing an EDHOC exchange in which the host demonstrates the ability to use that key as a Raw Public Key.

- o "ac": cryptmain is the 256bit hash value of a certificate using hash algorithm TBD. (Longer hashes would need changes in how this is put into DNS compatible names)

A host proves being eligible to use this name by performing an EDHOC exchange in which it demonstrates possession of the secret key indicated in the certificate.

- o "ad": Like "ac", but referring to any certificate in a chain.

For this crypttype, the common names in a certificate chain are concatenated [how exactly is TBD] into an ext-ident.

A host proves being eligible to use this name by performing an EDHOC exchange in which it demonstrates possession of any certificate together with a certificate chain to a certificate with the given hash where the certificate chain gives the claimed ext-ident name.

[Whether OSCORE's "ID_CRED_x" can be used in encoding this, or whether those can be substituted by a concept from HIP is up to further research; the rest of the document does not depend on the details of this construction.]

4.2. Equivalent resolvable names

For compatibility with devices that do not support the role of a constrained name user or even the coap+at scheme, resolvable names can be provided under a regular domain:

coap://nbswy3dpo5xxe3denbswy3dpo5xxe3de.ab.rd.link/.well-known/core

Note that a domain can only support a single non-coap+at scheme, as the addresses used by a named server for coap and coap+tcp may differ. The name servers for this domain would use the method described in [Section 6.1](#) to arrive at A/AAAA results.

Any equivalent URIs here create the issue of aliasing (see [\[RFC3986\]](#) [Section 6](#)). No more than two different names should be available for a device when this document has stabilized (and even that number would need to be justified, e.g. because one version leads to

enhanced backwards compatibility while the other has different benefits).

5. Announcing addresses

A named server has several ways of making itself available to clients:

5.1. Direct announcement

Protocol-qualified transport addresses for cryptidentifiers are announced by placing an entry in a global Distributed Hash Table (DHT).

The details of this are not yet laid out for this document, but [[I-D.jimenez-t2trg-drd](#)] already describes such a mechanism.

Entries in the DHT would contain:

- o Key: the cryptidentifier
- o Value:
 - * URLs that are alternative transports to the entry's coap+at://...rdlink.arpa URI
 - * A time stamp of the registration and its lifetime
 - * If the cryptidentifier alone is insufficient to verify signatories from it: additional information on the cryptidentifier, eg. a certificate (chain) for "ab" and "ac".
 - * If available in the crypttype: A signature on the rest of the value, signed by the owner of the cryptidentifier.

As a registration helper can not provide such a signature, instead of a signature on the entry there can be a signed datum that proves that the announcer was contacted by the identified device at a given time using the RD registration interface.

DHT participants and lookup helpers should verify the signatures on entries they propagate, but may do so only occasionally, or only when they detect duplicate entries.

For the signatures in which the registration helper creates a signed datum, it may make sense to use an unpredictable timestamping scheme (eg. the latest headlines from a widespread newspaper, or the head hash of a given block chain) to prevent malicious RD servers from

staying in control of the route to a given cryptident even after that device has picked a different RD server.

5.2. Announcement via a registration helper

Constrained named servers can enter their announcement by executing the RD registration operation ([\[I-D.ietf-core-resource-directory\] Section 5](#)) on a registration helper.

The registrant (= constrained named server) does not need to send a cryptident or other endpoint identifier; the helper will construct the cryptident from the chosen authentication method and construct an endpoint name from it.

The registrant may send a base URI (but may just as well rely on the RD (= the registration helper) to announce its network address). An alternative transport option (at=; [\[I-D.silverajan-core-coap-protocol-negotiation\] Section 4.1](#)) indicating the coap+at rdlink.arpa URI constructed from the cryptident is implicitly configured by the RD.

While performing the authentication step, the RD ensures that the registrant signs a timestamp and its IP address by embedding them in the "OSCORE C_V". [Or something similar, this part is still very experimental.]

The registrant may submit discoverable resources with its registration, but it is expected that most clients will only reveal them later to authenticated clients.

5.2.1. Finding a registration helper

The registrant can find a registration helper at the anycast address TBDv4 or TBDv6. The helpers work in "distinct registration point" mode (cf. [\[I-D.amsuess-core-rd-replication\] Section 6.2](#)), but do not implement the anycast variation suggested there in [Section 6.2.2](#), but rather give their explicit unicast addresses in a full URI during path discovery to ensure that updates wind up with them. [That should be added there in an updated rd-replication document].

5.3. Local announcement

To enable the use of coap+at rdlink.arpa URIs even in absence of an announcement server (eg. on ad-hoc networks), endpoints should join the link- and site-local All CoAP Nodes groups, provide an alternative-transport link to their own address, and answer to filtered multicast requests as described in [\[RFC6690\]](#):


```
Req: GET coap://ff02::fd/.well-known/core?href=coap+arpa://  
nbswy3dpo5xxe3denbswy3dpo5xxe3de.ab.rdlink.arpa&rev=alternative-transport
```

```
Res: 2.05 Content
```

```
<coap+at://nbswy3dpo5xxe3denbswy3dpo5xxe3de.ab.rdlink.arpa>;rev="alternative-  
transport"
```

[5.4.](#) Not announcing addresses

A named server is under no obligation to make its name publicly visible, especially when it is not expecting to host services.

The generated name can still be of use: It can be used in direct communication that the device has initiated in the role of a CoAP client with a different server. When that server accesses the named server under role reversal, it can address it by a rdlink.arpa name.

[6.](#) Lookup of rdlink.arpa URIs

A name user has several ways of finding transports of an rdlink.arpa URI:

[6.1.](#) Direct lookup

Alternative transport URLs for a given coap+at rdlink.arpa URI can be looked up in the DHT described in [Section 5.1](#); this mechanism is only conveniently usable by unconstrained devices.

[6.2.](#) RD lookup based

Analogous to [Section 5.2](#), clients can perform endpoint lookup to find alternative transport URLs for a given coap+at rdlink.arpa URI.

Clients look up actual transport addresses based on a filter on the alternative transport attribute (eg. by requesting "coap://[2001:db8::1]/rd-lookup/ep?at=coap+at://nbswy3dpo5xxe3denbswy3dpo5xxe3de.ab.rdlink.arpa"), and can specify the transport they are looking for using the transport type query parameter (tt=; [\[I-D.silverajan-core-coap-protocol-negotiation\]](#) [Section 4.2](#)).

Note that due to the distributed nature of this directory, lookups that do not specify an cryptident based URI can not be performed (as that would mean iterating through all published entries in the DHT); such requests are probably best answered with 4.00 "Bad Request".

6.3. Local lookup

Alternative transports to a coap+at URI can be discovered using multicast; see [Section 5.3](#) for an example.

7. Operation considerations

While the DHT can be run with very little management (probably just managing bootstrap servers), running the helpers at the anycast addresses will need some degree of management.

Steps to involve multiple parties in hosting such RD servers and policies that guide which of these servers are announced on the anycast addresses are to be developed in parallel to this document.

Device vendors may operate their servers under additional addresses, but are encouraged to join in the server pool. Devices may be configured to query such vendor servers by default, but need to use the public ones at least as a fallback.

Note that in private networks, operators may run their own helpers at the anycast addresses. If communication with other DHT nodes is not possible or administratively prohibited, discovery across such border is blocked, but the addresses used are still persistent, and discovery between services on the local network is unaffected.

While helpers may offer the proxy extension ([\[I-D.amsuess-core-resource-directory-extensions\]](#)), it is not expected that the public RD servers will offer that feature.

8. IANA considerations

[TBD: alternative-transport]

9. Security considerations

Alternative transports: "trusted to speak for" is usually not any resource on the device

[...]

...

10. References

10.1. Normative References

- [RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, [RFC 3986](#), DOI 10.17487/RFC3986, January 2005, <<https://www.rfc-editor.org/info/rfc3986>>.
- [RFC4648] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", [RFC 4648](#), DOI 10.17487/RFC4648, October 2006, <<https://www.rfc-editor.org/info/rfc4648>>.

10.2. Informative References

- [I-D.amsuess-core-rd-replication]
Amsuess, C., "Resource Directory Replication", [draft-amsuess-core-rd-replication-02](#) (work in progress), March 2019.
- [I-D.amsuess-core-resource-directory-extensions]
Amsuess, C., "CoRE Resource Directory Extensions", [draft-amsuess-core-resource-directory-extensions-00](#) (work in progress), January 2019.
- [I-D.ietf-core-resource-directory]
Shelby, Z., Koster, M., Bormann, C., Stok, P., and C. Amsuess, "CoRE Resource Directory", [draft-ietf-core-resource-directory-20](#) (work in progress), March 2019.
- [I-D.ietf-hip-rfc4423-bis]
Moskowitz, R. and M. Komu, "Host Identity Protocol Architecture", [draft-ietf-hip-rfc4423-bis-20](#) (work in progress), February 2019.
- [I-D.jimenez-t2trg-drd]
Jimenez, J., Liu, M., and E. Harjula, "A Distributed Resource Directory (DRD)", [draft-jimenez-t2trg-drd-00](#) (work in progress), March 2018.
- [I-D.silverajan-core-coap-protocol-negotiation]
Silverajan, B. and M. Ocak, "CoAP Protocol Negotiation", [draft-silverajan-core-coap-protocol-negotiation-09](#) (work in progress), July 2018.
- [RFC6690] Shelby, Z., "Constrained RESTful Environments (CoRE) Link Format", [RFC 6690](#), DOI 10.17487/RFC6690, August 2012, <<https://www.rfc-editor.org/info/rfc6690>>.

Appendix A. Ideas for the future of this document

- o Add a note on how to deal with PSK cases
 - * Often the PSKs will stem from a previous RPK or certificate process, in which case that base URI persists.
 - * If not, how did HIP deal with them? We could sure look them up in the DHT, but anyone may spam them.
 - * If it gets established in EDHOC that the implicit Uri-Host values in an EDHOC-derived PSK pair are the original key material's (which'd make sense IMO), does this affect the usability of implicit base in [Section 5.2](#)? (After all, the implicit role inversion address would be coap+hash:// then.)

- o Add notes on TLS/DTLS

Conceptually all of this should work on TLS/DTLS as well.

Appendix B. Existing approaches that don't solve the complete problem

- o IPv6 stable mobile addresses

Didn't take off.

- o HIP / ORCHIDv2

Feasible alternative, nice CIRI-compact addresses. Stack support unclear. Not extensible to sub-names (dev123.HASHHASHHASH - or can it, with notarized identities?). Hard limit on hash lengths (eg. Tor went from 80bit to 160bit, can't do that).

Using them at application layer only might be an option (cf. [\[I-D.ietf-hip-rfc4423-bis\]](#) last paragraph of introduction) See whether HIP's DoS protection can be applied in EDHOC.

Keep reading.

- o RD-DNS-SD

Hierarchical or limited to link-local.

- o TOR

basically got it right, just that we don't do onion routing here, and pull protocol negotiation in.

(might even consider 1:1 using their addresses, or at least take much from the discussion at <https://gitweb.torproject.org/torspec.git/tree/rend-spec-v3.txt> -> ONIONADDRESS-REFS); given we're likely never to write addresses by hand, checksumming might be left out, and version move to "domain".

- o IPFS / IPNS

Doesn't integrate well into URIs. Uses a concept of multihashes that has inspired the current cryptident design.

Author's Address

Christian Amsuess
Hollandstr. 12/4
1020
Austria

Phone: +43-664-9790639
Email: christian@amsuess.com

