          **CoAP Communication with Alternative Transports**
          **draft-silverajan-core-coap-alternative-transports-08**

Abstract

   CoAP has been standardised as an application level REST-based
   protocol.  A single CoAP message is typically encapsulated and
   transmitted using UDP or DTLS as transports.  These transports are
   optimal solutions for CoAP use in IP-based constrained environments
   and nodes.  However compelling motivation exists for allowing CoAP to
   operate with other transports and protocols.  Examples are M2M
   communication in cellular networks using SMS, more suitable transport
   protocols for firewall/NAT traversal, end-to-end reliability and
   security such as TCP and TLS, or employing proxying and tunneling
   gateway techniques such as the WebSocket protocol.  This draft
   examines the requirements for conveying CoAP messages to end points
   over such alternative transports.  It also provides a new URI format
   for representing CoAP resources over alternative transports.

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 http://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 December 22, 2015.

Copyright Notice

Table of Contents

## 1.  Introduction

The Constrained Application Protocol (CoAP) [RFC7252] has been
standardised by the CoRE WG as a lightweight, HTTP-like protocol
providing a request/response model that constrained nodes can use to
communicate with other nodes, be those servers, proxies, gateways,
less constrained nodes, or other constrained nodes.  CoAP has been
definied to utilise UDP and DTLS as transports.

As the Internet evolves by integrating new kinds of networks,
services and devices, the need for a consistent, lightweight method
for resource representation, retrieval and manipulation becomes

evident.  Owing to its simplicity and low overhead, CoAP is a highly
suitable protocol for this purpose.  However, communicating CoAP
endpoints can reside in networks where end-to-end UDP-based
communication can be challenging.  These include networks separated
by NATs and firewalls, cellular networks in which the Short Messaging
Service (SMS) can be utilised as between nodes, or simply situations
where an endpoint has no possibility to communicate over UDP.
Consequently in addition to UDP and DTLS, alternative transport
channels for conveying CoAP messages should be considered.

Extending CoAP over alternative transports allows CoAP
implementations to have a significantly larger relevance in
constrained as well as non-constrained networked environments: it
leads to better code optimisation in constrained nodes and broader
implementation reuse across new transport channels.  As opposed to
implementing new resource retrieval mechanisms, an application in an
end-node can continue relying on using CoAP's REST-based resource
retrieval and manipulation for this purpose, while changes in end
point identification and the transport protocol can be addressed by a
transport-specific messaging sublayer.  This simplifies development
and memory requirements.  Resource representations are also visible
in an end-to-end manner for any CoAP client.  In certain conditions,
the processing and computational overhead for conveying CoAP Requests
and Responses from one underlying transport to another, would be less
than that of an application-level gateway performing protocol
translation of individual messages between CoAP and another resource
retrieval protocol such as HTTP.

This document first provides scenarios where usage of CoAP over
alternative transports is either currently underway, or may prove
advantageous in the future.  A simple transport type classification
for CoAP-capable nodes is provided next.  Then a new URI format is
described through which a CoAP resource representation can be
formulated that expresses transport identification in addition to
endpoint information and resource paths.  Following that, a
discussion of the various transport properties which influence how
CoAP Request and Response messages are mapped to transport level
payloads, is presented.

This document however, does not touch on application QoS
requirements, user policies or network adaptation, nor does it
advocate replacing the current practice of UDP-based CoAP
communication.

## 2.  Usage Cases

   Apart from UDP and DTLS, CoAP usage is being specified for the
   following environments as of this writing:

### 2.1.  Use of SMS

   CoAP messages can be sent via SMS between CoAP end-points in a
   cellular network [I-D.becker-core-coap-sms-gprs].  A CoAP Request
   message can also be sent via SMS from a CoAP client to a sleeping
   CoAP Server as a wake-up mechanism and trigger communication via IP.
   For this reason, the Open Mobile Alliance (OMA) specifies both UDP
   and SMS as transports for M2M communication in cellular networks.
   The OMA Lightweight M2M (LWM2M) protocol being drafted uses CoAP, and
   as transports, specifies both UDP as well as Short Message Service
   (SMS) bindings [OMALWM2M].  DTLS is being proposed for securing CoAP
   messages over SMS between Mobile Stations
   [I-D.fossati-dtls-over-gsm-sms].

### 2.2.  Use of WebSockets

   The WebSocket protocol has been proposed as a transport channel
   between WebSocket enabled CoAP end-points on the Internet
   [I-D.savolainen-core-coap-websockets].  This is particularly useful
   to enable CoAP communication within HTML5 apps and web browsers,
   especially in smart devices, that do not have any means to use low-
   level socket interfaces.  Embedded client side scripts create new
   WebSocket connections to various WebSocket-enabled servers, through
   which CoAP messages can be exchanged.  This also allows a browser
   containing an embedded CoAP server to open a connection to a
   WebSocket enabled CoAP Mirror Server [I-D.vial-core-mirror-server] to
   register and update its resources.

### 2.3.  Use of P2P Overlays

   [I-D.jimenez-p2psip-coap-reload] specifices how CoAP nodes can use a
   peer-to-peer overlay network called RELOAD, as a resource caching
   facility for storing wireless sensor data.  When a CoAP node
   registers its resources with a RELOAD Proxy Node (PN), the node
   computes a hash value from the CoAP URI and stores it as a structure
   together with the PN's Node ID as well as the resources.  Resource
   retrieval by CoAP nodes is accomplished by computing the hash key
   over the Request URI, opening a connection to the overlay and using
   its message routing system to contact the CoAP server via its PN.

2.4.  Use of TCP and TLS

   Using TCP [I-D.tschofenig-core-coap-tcp-tls], allows easier
   communication between CoAP clients and servers separated by firewalls
   and NATs.  This also allows CoAP messages to be transported over push
   notification services from a notification server to a client app on a
   smartphone, that may previously have subscribed to receive change
   notifications of CoAP resource representations, possibly by using
   CoAP Observe [I-D.ietf-core-observe].
   [I-D.tschofenig-core-coap-tcp-tls] also discusses using TLS as a
   transport to securely convey CoAP messages over TCP.

3.  Node Types based on Transport Availability

   The term "alternative transport" in this document thus far has been
   used to refer to any non-UDP and non-DTLS transport that can convey
   CoAP messages in its payload.  A node however, may in fact possess
   the capability to utilise CoAP over multiple transport channels at
   its disposal, simultaneously or otherwise, at any point in time to
   communicate with a CoAP end-point.  Such communication can obviously
   take place over UDP and DTLS as well.  Inevitably, if two CoAP
   endpoints reside in distinctly separate networks with orthogonal
   transports, a CoAP proxy node is needed between the two networks so
   that CoAP Requests and Responses can be exchanged properly.

   In [RFC7228], Tables 1, 3 and 4 introduced classification schemes for
   devices, in terms of their resource constraints, energy limitations
   and communication power.  For this document, in addition to these
   capabilities, it seems useful to additionally identify devices based
   on their transport capabilities.

```
   +-------+----------------------------+
   | Name  | Transport Availability     |
   +-------+----------------------------+
   |  T0   | Single transport           |
   |       |                            |
   |  T1   | Multiple transports, with  |
   |       | one or more active at any  |
   |       | point in time              |
   |       |                            |
   |  T2   | Multiple active transports |
   |       | at all times               |
   +-------+----------------------------+
```

   Table 1: Classes of Available Transports

Type T0 nodespossess the capability of exactly 1 type of transport
channel for CoAP, at all times.  These include both active and sleepy
nodes, which may choose to perform duty cycling for power saving.

Type T1 nodes possess multiple different transports, and can retrieve
or expose CoAP resources over any or all of these transports.
However, not all transports are constantly active and certain
transport channels and interfaces could be kept in a mostly-off state
for energy-efficiency, such as when using CoAP over SMS (refer to
section 2.1)

Type T2 nodes possess more than 1 transport, and multiple transports
are simultaneously active at all times.  CoAP proxy nodes which allow
CoAP endpoints from disparate transports to communicate with each
other, are a good example of this.

4.  **CoAP Alternative Transport URI**

Based on the usage scenarios as well as the transport classes
presented in the preceding sections, this section discusses the
formulation of a new URI format for representing CoAP resources over
alternative transports.

CoAP is logically divided into 2 sublayers, whereby the upper layer
is responsible for the protocol functionality of exchanging request
and response messages, while the messaging layer is bound to UDP.
These 2 sublayers are tightly coupled, both being responsible for
properly encoding the header and body of the CoAP message.  The CoAP
URI is used by both logical sublayers.  For a URI that is expressed
generically as

URI = scheme ":" "//" authority path-abempty ["?" query ]

a simple example CoAP URI, "coap://server.example.com/sensors/
temperature" is interpreted as follows:

```
   coap :// server.example.com /sensors/temperature
   \___/    _____ _____/  _____ _____/
     |             \/                 \/
  protocol      endpoint         parameterised
 identifier    identifier           resource
                                    identifier
```
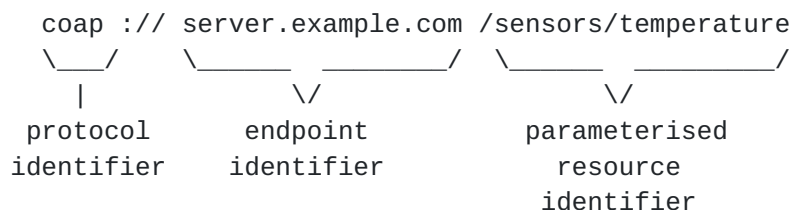
Figure 1: The CoAP URI format

The resource path is explicitly expressed, and the endpoint
identifier, which contains the host address at the network-level is
also directly bound to the scheme name containing the application-
level protocol identifier.  The choice of a specific transport for a
scheme, however, cannot be embedded with a URI, but is defined by
convention or standardisation of the protocol using the scheme.  As
examples, [RFC5092] defines the 'imap' scheme for the IMAP protocol
over TCP, while [RFC2818] requires that the 'https' protocol
identifier be used to differentiate using HTTP over TLS instead of
TCP.

## 4.1.  Design Considerations

Several ways of formulating a URI which express an alternative
transport binding to CoAP, can be envisioned.  When such a URI is
provided from an application to its CoAP implementation, the URI
component containing transport-specific information can be checked to
allow CoAP to use the appropriate transport for a target endpoint
identifier.

The following design considerations influence the formulation of a
new URI expressing CoAP resources over alternative transports:

1.  The CoAP Transport URI must conform to the generic syntax for a
    URI described in [RFC3986].  By ensuring conformance to RFC3986,
    the need for custom URI parsers as well as resolution algorithms
    can be obviated.  In particular, a URI format needs to be
    described in which each URI component clearly meets the syntax
    and percent-encoding rules described.

2.  A CoAP Transport URI can be supplied as a Proxy-Uri option by a
    CoAP end-point to a CoAP forward proxy.  This allows
    communication with a CoAP end-point residing in a network using a
    different transport.  Section 6.4 of [RFC7252] provides an
    algorithm for parsing a received URI to obtain the request's
    options.  Conformance to [RFC3986] is also necessary in order for
    the parsing algorithm to be successful.

3.  Request messages sent to a CoAP endpoint using a CoAP Transport
    URI may be responded to with a relative URI reference, for
    example, of the form "../../path/to/resource".  In such cases,
    the requesting endpoint needs to resolve the relative reference
    against the original CoAP Transport URI to then obtain a new
    target URI to which a request can be sent to, to obtain a
    resource representation.  [RFC3986] provides an algorithm to
    establish how relative references can be resolved against a base
    URI to obtain a target URI.  Given this algorithm, a URI format
    needs to be described in which relative reference resolution does

not result in a target URI that loses its transport-specific
information

4.  The host component of current CoAP URIs can either be an IPv4
    address, an IPv6 address or a resolvable hostname.  While the
    usage of DNS can sometimes be useful for distinguishing transport
    information (see section 4.3.1), accessing DNS over some
    alternative transport environments may be challenging.
    Therefore, a URI format needs to be described which is able to
    represent a resource without heavy reliance on a naming
    infrastructure, such as DNS.

## 4.2.  URI format

To meet the design considerations previously discussed, the transport
information is expressed as part of the URI scheme component.  This
is performed by minting new schemes for alternative transports using
the form "coap+<transport-name>" and/or "coaps+<transport-name>",
where the name of the transport is clearly and unambiguously
described.  Each scheme name formed in this manner is used to
differentiate the use of CoAP, or CoAP using DTLS, over an
alternative transport respectively.  The endpoint identifier, path
and query components together with each scheme name would be used to
uniquely identify each resource.

Examples of such URIs are:

o  coap+tcp://[2001:db8::1]:5683/sensors/temperature for using CoAP
   over TCP

o  coap+tls://[2001:db8::1]:5683/sensors/temperature for using CoAP
   over TLS

o  coaps+sctp://[2001:db8::1]:5683/sensors/temperature for using CoAP
   over DTLS over SCTP

o  coap+sms://0015105550101/sensors/temperature for using CoAP over
   SMS with the endpoint identifier being a telephone subscriber
   number

o  coaps+sms://0015105550101/sensors/temperature for using CoAP over
   DTLS over SMS with the endpoint identifier being a telephone
   subscriber number

o  coap+ws://www.example.com/sensors/temperature for using CoAP over
   WebSockets

o  coap+wss://www.example.com/sensors/temperature for using CoAP over
   secure WebSockets (WebSockets using TLS)

A URI of this format to distinguish transport types is simple to
understand and not dissimilar to the CoAP URI format.  As the usage
of each alternative transport results in an entirely new scheme, IANA
intervention is required for the registration of each scheme name.
The registration process follows the guidelines stipulated in
[I-D.ietf-appsawg-uri-scheme-reg], particularly where permanent URI
scheme registration is concerned.  CoAP resources transported over
UDP or DTLS must conform to Section 6 of [RFC7252] and utilise "coap"
or "coaps" for the URI scheme, instead of "coap+udp" or "coap+dtls".

It is also entirely possible for each new scheme to specify its own
rules for how resource and transport endpoint information can be
presented.  However, the URIs and resource representations arising
from their usage should meet the URI design considerations and
guidelines mentioned in Section 4.1.  In addition, each new transport
being defined should take into consideration the various transport-
level properties that can have an impact on how CoAP messages are
conveyed as payload.  This is elaborated on in the next section.

## 5. Alternative Transport Analysis and Properties

In this section the various characteristics of alternative transports
for successfully supporting various kinds of functionality for CoAP
are considered.  CoAP factors lossiness, unreliability, small packet
sizes and connection statelessness into its protocol logic.  General
transport differences and their impact on carrying CoAP messages here
are discussed.

Property 1: 1:N communication support.

This refers to the ability of the transport protocol to support
broadcast and multicast communication.  For example, group
communication for CoAP is based on multicasting Request messages and
receiving Response messages via unicast [RFC7390].  A protocol such
as TCP would be ill-suited for group communications using multicast.
Anycast support, where a message is sent to a well defined
destination address to which several nodes belong, on the other hand,
is supported by TCP.

Property 2: Transport-level reliability.

This refers to the ability of the transport protocol to support
properties such as guaranteeing reliability against packet loss,
ensuring ordered packet delivery and having error control.  When CoAP
Request and Response messages are delivered over such transports, the

CoAP implementations elide certain fields in the packet header.  As
an example, if the usage of a connection-oriented transport renders
it unnecessary to specify the various CoAP message types, the Type
field can be elided.  For some connection-oriented transports, such
as WebSockets, the version of CoAP being used can be negotiated
during the opening transfer.  Consequently, the Version field in CoAP
packets can also be elided.

Property 3: Message encoding.

While parts of the CoAP payload are human readable or are transmitted
in XML, JSON or SenML format, CoAP is essentially a low overhead
binary protocol.  Efficient transmission of such packets would
therefore be met with a transport offering binary encoding support.
Techniques exist in allowing binary payloads to be transferred over
text-based transport protocols such as base-64 encoding.  When using
SMS as a transport, for example, although binary encoding is
supported, Appendix A.5 of [I-D.bormann-coap-misc] indicates binary
encoding for SMS may not always be viable.  A fuller discussion about
performing CoAP message encoding for SMS can be found in Appendix A.5
of [I-D.bormann-coap-misc]

Property 4: Network byte order.

CoAP, as well as transports based on the IP stack use a Big Endian
byte order for transmitting packets over the air or wire, while
transports based on Bluetooth and Zigbee prefer Little Endian byte
ordering for packet fields and transmission.  Any CoAP implementation
that potentially uses multiple transports has to ensure correct byte
ordering for the transport used.

Property 5: MTU correlation with CoAP PDU size.

Section 4.6 of [RFC7252] discusses the avoidance of IP fragmentation
by ensuring CoAP message fit into a single UDP datagram.  End-points
on constrained networks using 6LoWPAN may use blockwise transfers to
accommodate even smaller packet sizes to avoid fragmentation.  The
MTU sizes for Bluetooth Low Energy as well as Classic Bluetooth are
provided in Section 2.4 of [I-D.ietf-6lo-btle].  Transport MTU
correlation with CoAP messages helps ensure minimal to no
fragmentation at the transport layer.  On the other hand, allowing a
CoAP message to be delivered using a delay-tolerant transport service
such as the Bundle Protocol [RFC5050] would imply that the CoAP
message may be fragmented (or reconstituted) along various nodes in
the DTN as various sized bundles and bundle fragments.

Property 6: Framing

When using CoAP over a streaming transport protocol such as TCP, as
opposed to datagram based protocols, care must be observed in
preserving message boundaries.  Commonly applied techniques at the
transport level include the use of delimiting characters for this
purpose as well as message framing and length prefixing.

Property 7: Transport latency.

A confirmable CoAP request would be retransmitted by a CoAP end-point
if a response is not obtained within a certain time.  A CoAP end-
point registering to a Resource Directory uses a POST message that
could include a lifetime value.  A sleepy end-point similarly uses a
lifetime value to indicate the freshness of the data to a CoAP Mirror
Server.  Care needs to be exercised to ensure the latency of the
transport being used to carry CoAP messages is small enough not to
interfere with these values for the proper operation of these
functionalities.

Property 8: Connection Management.

A CoAP endpoint using a connection-oriented transport should be
responsible for proper connection establishment prior to sending a
CoAP Request message.  Both communicating endpoints may monitor the
connection health during the Data Transfer phase.  Finally, once data
transfer is complete, at least one end point should perform
connection teardown gracefully.

## 6.  IANA Considerations

This memo includes no request to IANA.

## 7.  Security Considerations

New security risks are not envisaged to arise from the guidelines
given in this document, for describing a new URI format containing
transport identification within the URI scheme component.  However,
when specific alternative transports are selected for implementing
support for carrying CoAP messages, risk factors or vulnerabilities
can be present.  Examples include privacy trade-offs when MAC
addresses or phone numbers are supplied as URI authority components,
or if specific URI path components employed for security-specific
interpretations are accidentally encountered as false positives.
While this document does not make it mandatory to introduce a
security mode with each transport, it recommends ascribing meaning to
the use of "coap+" and "coaps+" prefixes in the scheme component,
with the "coaps+" prefix used for DTLS-based CoAP messages over the
alternative transport.

## 8. Acknowledgements

The draft has benefited greatly from reviews, comments and ideas from Thomas Fossati, Akbar Rahman, Klaus Hartke, Martin Thomson, Mark Nottingham, Dave Thaler, Graham Klyne, Carsten Bormann and Markus Becker.

## 9. References

### 9.1. Normative References

[I-D.ietf-appsawg-uri-scheme-reg]
          Thaler, D., Hansen, T., Hardie, T., and L. Masinter,
          "Guidelines and Registration Procedures for URI Schemes",
          draft-ietf-appsawg-uri-scheme-reg-06 (work in progress),
          April 2015.

[RFC3986]  Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform
          Resource Identifier (URI): Generic Syntax", STD 66, RFC
          3986, January 2005.

[RFC7228]  Bormann, C., Ersue, M., and A. Keranen, "Terminology for
          Constrained-Node Networks", RFC 7228, May 2014.

[RFC7252]  Shelby, Z., Hartke, K., and C. Bormann, "The Constrained
          Application Protocol (CoAP)", RFC 7252, June 2014.

### 9.2. Informative References

[BTCorev4.1]
          BLUETOOTH Special Interest Group, "BLUETOOTH Specification
          Version 4.1", December 2013.

[I-D.becker-core-coap-sms-gprs]
          Becker, M., Li, K., Kuladinithi, K., and T. Poetsch,
          "Transport of CoAP over SMS", draft-becker-core-coap-sms-
          gprs-05 (work in progress), August 2014.

[I-D.bormann-coap-misc]
          Bormann, C. and K. Hartke, "Miscellaneous additions to
          CoAP", draft-bormann-coap-misc-27 (work in progress),
          November 2014.

   [I-D.fossati-dtls-over-gsm-sms]
              Fossati, T. and H. Tschofenig, "Datagram Transport Layer
              Security (DTLS) over Global System for Mobile
              Communications (GSM) Short Message Service (SMS)", draft-
              fossati-dtls-over-gsm-sms-01 (work in progress), October
              2014.

   [I-D.ietf-6lo-btle]
              Nieminen, J., Savolainen, T., Isomaki, M., Patil, B.,
              Shelby, Z., and C. Gomez, "IPv6 over BLUETOOTH(R) Low
              Energy", draft-ietf-6lo-btle-13 (work in progress), May
              2015.

   [I-D.ietf-core-observe]
              Hartke, K., "Observing Resources in CoAP", draft-ietf-
              core-observe-16 (work in progress), December 2014.

   [I-D.jimenez-p2psip-coap-reload]
              Jimenez, J., Lopez-Vega, J., Maenpaa, J., and G.
              Camarillo, "A Constrained Application Protocol (CoAP)
              Usage for REsource LOcation And Discovery (RELOAD)",
              draft-jimenez-p2psip-coap-reload-09 (work in progress),
              June 2015.

   [I-D.savolainen-core-coap-websockets]
              Savolainen, T., Hartke, K., and B. Silverajan, "CoAP over
              WebSockets", draft-savolainen-core-coap-websockets-04
              (work in progress), March 2015.

   [I-D.tschofenig-core-coap-tcp-tls]
              Bormann, C., Lemay, S., Technologies, Z., and H.
              Tschofenig, "A TCP and TLS Transport for the Constrained
              Application Protocol (CoAP)", draft-tschofenig-core-coap-
              tcp-tls-04 (work in progress), June 2015.

   [I-D.vial-core-mirror-server]
              Vial, M., "CoRE Mirror Server", draft-vial-core-mirror-
              server-01 (work in progress), April 2013.

   [OMALWM2M]
              Open Mobile Alliance (OMA), "Lightweight Machine to
              Machine Technical Specification Version 1.0", 2015.

   [RFC2119]  Bradner, S., "Key words for use in RFCs to Indicate
              Requirement Levels", BCP 14, RFC 2119, March 1997.

   [RFC2609]  Guttman, E., Perkins, C., and J. Kempf, "Service Templates
              and Service: Schemes", RFC 2609, June 1999.

[RFC2818]  Rescorla, E., "HTTP Over TLS", RFC 2818, May 2000.

[RFC4838]  Cerf, V., Burleigh, S., Hooke, A., Torgerson, L., Durst,
           R., Scott, K., Fall, K., and H. Weiss, "Delay-Tolerant
           Networking Architecture", RFC 4838, April 2007.

[RFC5050]  Scott, K. and S. Burleigh, "Bundle Protocol
           Specification", RFC 5050, November 2007.

[RFC5092]  Melnikov, A. and C. Newman, "IMAP URL Scheme", RFC 5092,
           November 2007.

[RFC6455]  Fette, I. and A. Melnikov, "The WebSocket Protocol", RFC
           6455, December 2011.

[RFC6568]  Kim, E., Kaspar, D., and JP. Vasseur, "Design and
           Application Spaces for IPv6 over Low-Power Wireless
           Personal Area Networks (6LoWPANs)", RFC 6568, April 2012.

[RFC6733]  Fajardo, V., Arkko, J., Loughney, J., and G. Zorn,
           "Diameter Base Protocol", RFC 6733, October 2012.

[RFC7390]  Rahman, A. and E. Dijk, "Group Communication for the
           Constrained Application Protocol (CoAP)", RFC 7390,
           October 2014.

[WWWArchv1]
           http://www.w3.org/TR/webarch/#uri-aliases, "Architecture
           of the World Wide Web, Volume One", December 2004.

## Appendix A.  Expressing transport in the URI in other ways

Other means of indicating the transport as a distinguishable
component within the CoAP URI are possible, but have been deemed
unsuitable by not meeting the design considerations listed, or are
incompatible with existing practices outlined in [RFC7252].  They are
however, retained in this section for historical documentation and
completeness.

## A.1.  Transport information as part of the URI authority

A single URI scheme, "coap-at" can be introduced, as part of an
absolute URI which expresses the transport information within the
authority component.  One approach is to structure the component with
a transport prefix to the endpoint identifier and a delimiter, such
as "<transport-name>-endpoint_identifier".

Examples of resulting URIs are:

o  coap-at://tcp-server.example.com/sensors/temperature

o  coap-at://sms-0015105550101/sensors/temperature

An implementation note here is that some generic URI parsers will
fail when encountering a URI such as "coap-at://tcp-
[2001:db8::1]/sensors/temperature".  Consequently, an equivalent, but
parseable URI from the ip6.arpa domain needs to be formulated
instead.  For [2001:db8::1] using TCP, this would result in the
following URL:

coap-at://tcp-1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.b.d.0
.1.0.0.2.ip6.arpa:5683/sensors/temperature

Usage of an IPv4-mapped IPv6 address such as [::ffff.192.100.0.1] can
similarly be expressed with a URI from the ip6.arpa domain.

This URI format allows the usage of a single scheme to represent
multiple types of transport end-points.  Consequently, it requires
consistency in ensuring how various transport-specific endpoints are
identified, as a single URI format is used.  Attention must be paid
towards the syntax rules and encoding for the URI host component.
Additionally, against a base URI of the form "coap-at://tcp-
server.example.com/sensors/temperature", resolving a relative
reference, such as "//example.net/sensors/temperature" would result
in the target URI "coap-at://example.net/sensors/temperature", in
which transport information is lost.

### A.1.1.  Usage of DNS records

DNS names can be used instead of IPv6 address literals to mitigate
lengthy URLs referring to the ip6.arpa domain, if usage of DNS is
possible.

DNS SRV records can also be employed to formulate a URL such as:

coap-at://srv-_coap._tcp.example.com/sensors/temperature

in which the "srv" prefix is used to indicate that a DNS SRV lookup
should be used for _coap._tcp.example.com, where usage of CoAP over
TCP is specified for example.com, and is eventually resolved to a
numerical IPv4 or IPv6 address.

### A.2.  Making CoAP Resources Available over Multiple Transports

The CoAP URI used thus far is as follows:

```
URI          = scheme ":" hier-part [ "?" query ]
hier-part    = "//" authority path-abempty
```

A new URI format could be introduced, that does not possess an
"authority" component, and instead defining "hier-part" to instead
use another component, "path-rootless", as specified by RFC3986
[RFC3986].  The partial ABNF format of this URI would then be:

```
URI           = scheme ":" hier-part [ "?" query ]
hier-part   = path-rootless
path-rootless = segment-nz *( "/" segment )
```

The full syntax of "path-rootless" is described in [RFC3986].  A
generic URI defined this way would conform to the syntax of
[RFC3986], while the path component can be treated as an opaque
string to indicate transport types, endpoints as well as paths to
CoAP resources.  A single scheme can similarly be used.

A constrained node that is capable of communicating over several
types of transports (such as UDP, TCP and SMS) would be able to
convey a single CoAP resource over multiple transports.  This is also
beneficial for nodes performing caching and proxying from one type of
transport to another.

Requesting and retrieving the same CoAP resource representation over
multiple transports could be rendered possible by prefixing the
transport type and endpoint identifier information to the CoAP URI.
This would result in the following example representation:

```
coap-at:tcp://example.com?coap://example.com/sensors/temperature
        _____  _____/ _____  _____/
               \/                          \/
         Transport-specific          CoAP Resource
             Prefix
```

Figure 2: Prefixing a CoAP URI with TCP transport

Such a representation would result in the URI being decomposed into
its constituent components, with the CoAP resource residing within
the query component as follows:

Scheme: coap-at

Path: tcp://example.com

Query: coap://example.com/sensors/temperature

The same CoAP resource, if requested over a WebSocket transport,
would result the following URI:


```
coap-at:ws://example.com/endpoint?coap://example.com/sensors/temperature
         _____  _____/ _____  _____/
                    \/                         \/
             Transport-specific            CoAP Resource
                  Prefix
```


        Figure 3: Prefixing a CoAP URI with WebSocket transport

While the transport prefix changes, the CoAP resource representation
remains the same in the query component:

Scheme: coap-at

Path: ws://example.com/endpoint

Query: coap://example.com/sensors/temperature

The URI format described here overcomes URI aliasing [WWWArchv1] when
multiple transports are used, by ensuring each CoAP resource
representation remains the same, but is prefixed with different
transports.  However, against a base URI of this format, resolving
relative references of the form "//example.net/sensors/temperature"
and "/sensor2/temperature" would again result in target URIs which
lose transport-specific information.

Implementation note: While square brackets are disallowed within the
path component, the '[' and ']' characters needed to enclose a
literal IPv6 address can be percent-encoded into their respective
equivalents.  The ':' character does not need to be percent-encoded.
This results in a significantly simpler URI string compared to
section 2.2, particularly for compressed IPv6 addresses.
Additionally, the URI format can be used to specify other similar
address families and formats, such as Bluetooth addresses
[BTCorev4.1].

## A.3.  Transport as part of a 'service:' URL scheme

The "service:" URL scheme name was introduced in [RFC2609] and forms
the basis of service description used primarily by the Service
Location Protocol.  An abstract service type URI would have the form

"service:<abstract-type>:<concrete-type>"

where <abstract-type> refers to a service type name that can be
associated with a variety of protocols, while the <concrete-type>
then providing the specific details of the protocol used, authority
and other URI components.

Adopting the "service:" URL scheme to describe CoAP usage over
alternative transports would be rather trivial.  To use a previous
example, a CoAP service to discover a Resource Directory and its base
RD resource using TCP would take the form

service:coap:tcp://host.example.com/.well-known/core?rt=core-rd

The syntax of the "service:" URL scheme differs from the generic URI
syntax and therefore such a representation should be treated as an
opaque URI as Section 2.1 of [RFC2609] recommends.

Authors' Addresses

Bilhanan Silverajan
Tampere University of Technology
Korkeakoulunkatu 10
FI-33720 Tampere
Finland

Email: bilhanan.silverajan@tut.fi


Teemu Savolainen
Nokia
Hermiankatu 12 D
FI-33720 Tampere
Finland

Email: teemu.savolainen@nokia.com