CoRE Working Group
Internet-Draft
Intended status: Informational
Expires: January 16, 2014

B. Silverajan
Tampere University of Technology
T. Savolainen
Nokia

July 15, 2013

CoAP Communication with Alternative Transports
draft-silverajan-core-coap-alternative-transports-02

Abstract

CoAP is being standardised as an application level REST-based protocol. A single CoAP message is typically encapsulated and transmitted using UDP. This draft examines the requirements and possible solutions for conveying CoAP packets to end points over alternative transports to UDP. UDP remains the optimal solution for CoAP use in IP-based constrained environments and nodes. However the need for M2M communication using non-IP networks, improved transport level end-to-end reliability and security, NAT and firewall traversal issues, and mechanisms possibly incurring a lower overhead to CoAP/HTTP translation gateways provide compelling motivation for understanding how CoAP can operate in various other environments.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

The Constrained Application Protocol (CoAP) [I-D.ietf-core-coap] is being 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.

As the Internet continues taking shape 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, the CoAP endpoint can reside in a non-IP network, be separated from its peer by NATs and firewalls or simply has no possibility to communicate over UDP. Consequently in addition to UDP, alternative transport channels for conveying CoAP packets could be considered.

This document looks at how CoAP can be used by nodes for resource retrieval, in an end-to-end manner regardless of the transport channel available. It looks at current usage of CoAP in this regard

today and provides other possible scenarios. Then the document looks at how resources using CoAP can contain resource information that provide endpoint as well as transport identifiers, without imposing incompatibilities with [I-D.ietf-core-coap] and maintaining conformance to [RFC3986].

This draft does not discuss on application QoS requirements, user policies or network adaptation, nor does it advocate replacing the current practice of UDP-based CoAP communication. The scope of this draft is limited towards a description and a requirements capture of how CoAP packets can be transmitted over alternative transports, especially how such protocols can be expressed at the CoAP layer, as well as how CoAP packets can be mapped at transport level payloads.

1.1. Using Alternative Transports

Extending CoAP's REST-based usage 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 implementation reuse across new transport channels. As opposed to implementing new resource retrieval schemes, an application in an end-node can continue relying on using CoAP for this purpose, but lets CoAP take into account the change in end point identification and transport protocol. This simplifies development and memory requirements. Resource representations are also visible in an end-to-end manner for any CoAP client.

Inevitably, if two CoAP endpoints reside in distinctly separate networks with orthogonal transports, a CoAP proxy node is needed between the 2 networks so that CoAP Requests and Responses can be fulfilled properly. The processing and computational overhead for conveying CoAP packets from one underlying transport to another, would be less than that of an application-level gateway performing individual packet-based, protocol translation between CoAP to another resource retrieval scheme.

1.2. Use Cases

CoAP has been designed to work on top of UDP, that is, on top of a transport that can lose, reorder, and duplicate packets. UDP has been chosen as the transport protocol over IP due its lightweight nature and connectionless characteristics. In addition to point-to-point communication, this allows multicast and group communications [I-D.ietf-core-groupcomm]. Moreover, DTLS can be employed to secure CoAP communication.

At this time of writing, the use of CoAP is also being specified for other environments as follows:

1. CoAP Request and Response messages can be sent via SMS or USSD 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 for subsequent communication via GPRS. The Open Mobile Alliance (OMA) specifies both UDP and SMS as transports for M2M communication in cellular networks. The OMA Lightweight M2M protocol being drafted uses CoAP, and as transports, specifies both UDP binding as well as Short Message Service (SMS) bindings [OMALWM2M] for the same reason.

2. The WebSocket protocol is being used as a transport channel between WebSocket enabled CoAP end-points on the Internet [I-D.savolainen-core-coap-websockets]. This is particularly useful as a means for web browsers, particularly in smart devices, to allow embedded client side scripts to upgrade an existing HTTP connection to a WebSocket connection through which CoAP Request and Response messages can be exchanged with a WebSocket-enabled server. This also allows a browser containing an embedded CoAP server to behave as a WebSocket client by opening a connection to a WebSocket enabled CoAP Mirror Server to register and update its resources.

3. [I-D.jimenez-p2psip-coap-reload] specifies 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.

We also envisage CoAP being extended atop other transport channels, such as:

1. Using TCP to facilitate the traversal of CoAP Request and Response messages. This 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-functionality [I-D.ietf-core-observe].

2. The transportation of CoAP messages in Delay-Tolerant Networks [RFC4838], using the Bundle Protocol [RFC5050] for reaching sensors in extremely challenging environments such as acoustic, underwater and deep space networks.

3. Any type of non-IP networks supporting constrained nodes and low-energy sensors, such as Bluetooth and Bluetooth Low Energy (either through L2CAP or with GATT), ZigBee, Z-Wave, 1-Wire, DASH7 and so on.

4. Instant Messaging and Social Networking channels, such as Jabber and Twitter.

2. CoAP Transport URI

CoAP is logically divided into 2 sublayers, whereby a request/response 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" can be 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.

To express an alternative transport binding to CoAP, a scheme name can follow a convention of the form "coap+<transport-name>", where the name of the transport is clearly and unambiguously described. Each scheme name formed in this manner can be used to differentiate the use of CoAP over an alternative transport instead of the use of CoAP over UDP or DTLS. 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:

- coap://server.example.com/sensors/temperature for using CoAP over UDP

- coaps://server.example.com/sensors/temperature for using CoAP over DTLS

- coap+tel://+15105550101/sensors/temperature for using CoAP over SMS or USSD with the endpoint identifier being a telephone subscriber number

- coap+ws://www.example.com/WebSocket?/.well-known/core?rt=core.ms for using CoAP over WebSockets with the endpoint at ws://www.example.com/WebSocket

- coap+ble.l2cap://[12:34:56:78:90:AB]:4/pulse where the scheme name can possibly be used to describe the future use of CoAP over L2CAP using Bluetooth Low Energy, but not L2CAP using classic Bluetooth.

When such a URI is provided from an end-application to its CoAP implementation, the scheme name can be checked to allow the CoAP to use the appropriate transport for the specified endpoint identifier. The CoAP Transport URI can also be supplied as a Proxy-Uri option by a CoAP end-point to a CoAP forward proxy in order to communicate with a CoAP end-point residing in a network using a different transport. Section 6.4 of [I-D.ietf-core-coap] provides an algorithm for parsing the received URI to obtain the request's options.

3. Alternative Transport Analysis and Properties

In this section we consider the various characteristics of alternative transports for successfully supporting various kinds of functionality for CoAP. CoAP factors lossiness, unreliability, small packet sizes and connection statelessness into its protocol logic. We discuss general transport differences and their impact on carrying CoAP packets here. Note that Properties 1, 2, and 3 are related.

Property 1: Uniqueness of an end-point identifier.

Transport protocols providing non-unique end-point IDs for nodes may only convey a subset of the CoAP functionality. Such nodes may only serve as CoAP servers that announce data at specific intervals to a pre-specified end point, or to a shared medium.

Property 2: Unidirectional or bidirectional CoAP communication support.

This refers to the ability of the CoAP end-point to use a single transport channel for both request and response messages. Depending on the scenario, having a unidirectional transport layer would mean the CoAP end-point might utilise it only for outgoing data or incoming data. Should both functionalities be needed, 2 unidirectional transport channels would be necessary.

Property 3: 1:N communication support.

This refers to the ability of the transport protocol to support broadcast and multicast communication. CoAP's request/response behaviour depends on unicast messaging. Group communication in CoAP is bound to using multicasting. Therefore 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 4: Transport-level reliability.

This refers to the ability of the transport protocol to provide a guarantee of 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 5: 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, although techniques exist in allowing binary payloads to be transferred over text-based transport protocols such as base-64 encoding. A fuller discussion about performing CoAP message encoding for SMS can be found in Appendix A.5 of [I-D.bormann-coap-misc]

Property 6: 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 7: MTU correlation with CoAP PDU size.

Section 4.6 of [I-D.ietf-core-coap] 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-6lowpan-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 8: 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 packets is small enough not to interfere with these values for the proper operation of these functionalities.

3.1. Other Considerations

This section outlines miscellaneous considerations concerning transport bindings with the CoAP URI.

1. 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.

2. A CoAP server, such as a sensor, may make its data available over multiple types of transports concurrently. For example, this can be done to allow the value to be retrieved via UDP as well as TCP. However, this could be carried out only when neccesary to avoid a resource being identified by more than one URI. On the other hand, when using only a single underlying transport, URI aliasing should not be practised [WWWArchv1]. For some scenarios where there is an availability of DNS for lookups as well as updates, SRV records can be used. In these cases, the "_service" field can be "coap", and the "_proto" field carries the name of the transport to be used.

3. 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 [RFC4395], particularly where permanent URI scheme registration is concerned.

4. IANA Considerations

This memo includes no request to IANA.

5. Security Considerations

While we envisage no new security risks simply from the introduction of support for alternative transports, end-applications and CoAP implementations should take note if certain transports require privacy trade-offs that may arise if identifiers such as MAC addresses or phone numbers are made public in addition to FQDNs.

6. Acknowledgements

Discussions with Klaus Hartke, Graham Klyne, Markus Becker and Golnaz Karbaschi provided useful insights and ideas for this draft.

7. Informative References

[BTCorev4.0]
BLUETOOTH Special Interest Group, "BLUETOOTH Specification Version 4.0 ", June 2010.

[I-D.becker-core-coap-sms-gprs]
Becker, M., Li, K., Kuladinithi, K., and T. Poetsch, "Transport of CoAP over SMS, USSD and GPRS", draft-becker-core-coap-sms-gprs-03 (work in progress), February 2013.

[I-D.bormann-coap-misc]
Bormann, C. and K. Hartke, "Miscellaneous additions to CoAP", draft-bormann-coap-misc-25 (work in progress), May 2013.

[I-D.ietf-6lowpan-btle]
Nieminen, J., Savolainen, T., Isomaki, M., Patil, B., Shelby, Z., and C. Gomez, "Transmission of IPv6 Packets over BLUETOOTH Low Energy", draft-ietf-6lowpan-btle-12 (work in progress), February 2013.

[I-D.ietf-core-coap]
Shelby, Z., Hartke, K., and C. Bormann, "Constrained Application Protocol (CoAP)", draft-ietf-core-coap-18 (work in progress), June 2013.

[I-D.ietf-core-groupcomm]
Rahman, A. and E. Dijk, "Group Communication for CoAP", draft-ietf-core-groupcomm-10 (work in progress), July 2013.

[I-D.ietf-core-observe]
Hartke, K., "Observing Resources in CoAP", draft-ietf-core-observe-08 (work in progress), February 2013.

[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-03 (work in progress), February 2013.

[I-D.savolainen-core-coap-websockets]
Savolainen, T., Hartke, K., and B. Silverajan, "CoAP over WebSockets", draft-savolainen-core-coap-websockets-00 (work in progress), July 2013.

[IANA-paparazzi-uri]
https://www.iana.org/assignments/uri-schemes/prov/paparazzi, "Resource Identifier (RI) Scheme name: paparazzi ", September 2012.

[OMALWM2M]
Open Mobile Alliance (OMA), "Lightweight Machine to Machine Technical Specification ", 2013.

[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.

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

[RFC4395] Hansen, T., Hardie, T., and L. Masinter, "Guidelines and Registration Procedures for New URI Schemes", BCP 35, RFC 4395, February 2006.

[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.

[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 working group consensus or because they violate [RFC3986], and are incompatible with existing practices outlined in [I-D.ietf-core-coap]. They are however, retained in this section for historical documentation and completeness.

A.1. Transport in URI path or query component

For a URI that is expressed generically as

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

The transport protocol can alternatively be provided as a path or query component. The Diameter Base Protocol [RFC6733] is one example of a protocol that uses the "aaa" and "aaas" URI scheme names to reflect whether transport security is used, and at the same time provides the actual transport protocol to be used as a ";transport=" path component. Example valid Diameter URIs are aaa://host.example.com;transport=sctp and aaas://host.example.com:6666;transport=tcp

Adopting such a procedure for CoAP can be done in two ways. The first is to provide the transport as a path component, similar to the Diameter protocol. An example resulting URI could be coap://host.example.com;transport=tcp/.well-known/core?rt=core-rd specifying a CoAP endpoint discovering a Resource Directory and its base RD resource while using TCP as a transport instead of UDP. A URI-Path option would then be used to encode the transport used.

An alternative means of expressing the transport protocol used is to encode the transport as a query component instead. In this case, the resulting URI would then be coap://host.example.com/.well-known/core?rt=core-rd?tt=tcp where "tt" refers to the transport type. Such a scheme would mean that the CoAP implementation encodes two URI-query components.

It is also conceivable that an end point may wish to register its available transports and associated end point identifiers in a CoAP resource directory, and periodically update them. A "core-transport" resource type or a "tt" link target attribute may then need to be registered.

Encoding the transport as part of the URI path or query provides an advantage in that IANA registration is not required, as opposed to introducing new URI scheme names. New transports can be easily

introduced into the CoAP URI. As both the URI-Path and the URI-Query options fall into the "critical" class of options, caution must be exercised if an endpoint does not recognise them. In such cases, section 5.4.1 in [I-D.ietf-core-coap] provides handling guidelines.

A.2. Transport in the URI authority component

An application-specific, provisional resource identifier registered with IANA, has been done so by specifying the transport to be used as part of the authority [IANA-paparazzi-uri]:

```
paparazzi:[options] http:[//host>[:[port][transport]]/
```

While the URI is used by the application to obtain a screenshot of a non-secure webpage, usage of the transport parameter is unclear and if it is at all used.

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