CoRE Working Group
Internet-Draft
Intended status: Informational
Expires: August 29, 2013

B. Silverajan
Tampere University of Technology
T. Savolainen
Nokia

February 25, 2013

CoAP Communication with Alternative Transports
draft-silverajan-core-coap-alternative-transports-01

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. While UDP remains an optimal solution for use in IP-based constrained environments and nodes, M2M communication using non-IP networks, NAT and firewall traversal issues and possibly mechanisms incurring a lower overhead to CoAP/ HTTP gateways provide compelling motivation for understanding how CoAP can operate in various other environments.

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 that provides 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's functionality and packet sizes have been specified in order to allow constrained nodes the ability to execute a simple application protocol to set and retrieve resources using a REST-based approach. To allow for very low communication overhead as well as the unreliability of constrained environments, CoAP is bound to UDP with optional reliability, to support unicast and multicast communication. Security is provided by means of the Datagram Transport Layer Security (DTLS). Interworking with web is being standardized by means of stateless HTTP mapping.

Owing to its simplicity, CoAP is an attractive option for all manner of uses. In addition to simple end-to-end communication between CoAP end-points as well as between CoAP and HTTP-based end-points, it is being used towards resource discovery and lookups, group-based communication, proxying and mirroring resources on behalf of sleeping nodes.

As the heterogeneity of interconnected networks and nodes continues increasing, alternative modes of transporting CoAP packets, in addition to UDP should be considered. This allows, for instance, retrieval of resource values and attributes of sensor nodes in non-IP networks and the ability of nodes to overcome firewall and NAT traversal issues. As the Internet of Things takes shape and begins integrating with new kinds of networks and services, it is important to understand the relevance of extending CoAP towards new transport protocols in order to have a uniform method of lightweight retrieval and modification of resources on constrained end-points and M2M communication. Not all constrained nodes might have the ability to take advantage of IP. At the same time, not all nodes with the ability to run CoAP over UDP will be confined to just one type of networking technology. As an example, the Lightweight M2M protocol being drafted by the Open Mobile Alliance uses CoAP, and as transports, specifies both UDP binding as well as Short Message Service (SMS) bindings [OMALWM2M]. The whys and hows of running CoAP over SMS, USSD and GPRS is an ongoing work, expanded upon in [I-D.becker-core-coap-sms-gprs]

This document generalizes the CoAP Unique Resource Identifier (URI), specified in [I-D.ietf-core-coap] and further expanded in

[I-D.becker-core-coap-sms-gprs]. These drafts describe CoAP using the "coap:", "coaps:" as well as "coap+tel:" URI schemes. In this draft we explore how the URI can be further extended towards specifying usable and alternative transports without imposing incompatibilities with current practices. The mechanisms introduced should remain in conformance to practices stipulated in [RFC4395].

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.

2. Rationale and Benefits

The variety of alternative transports is large. These include IETF specified transport protocols such as TCP and Websockets, Disruption Tolerant technologies such as the Bundle Protocol, non-IP transports based on Bluetooth Low Energy and Near-Field Communications (NFC). [I-D.ietf-core-coap] acknowledges that CoAP can be used in conjunction with XMPP and SIP and [I-D.becker-core-coap-sms-gprs] documents ongoing work on letting CoAP work with SMS. It is nevertheless important to understand the relevance of extending CoAP towards new transport protocols in order to have a uniform method of lightweight retrieval and modification of resources on constrained end-points by exploiting the underlying native characteristics of such networks and their transports without necessarily having to rely on an IP adaptation layer.

CoAP over alternative transports allows 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 networks, whereby a node can continue relying on the same REST-based API changing its end point identifier and transport protocol, when for example, its network technology migrates from a non-IP transport to an IP and UDP-based transport. This might be the case in a ZigBee or BLE node having CoAP over a proprietary network layer but subsequently supporting UDP/IP adaptation.

3. Use Cases

CoAP [I-D.ietf-core-coap] has been designed to work on top of UDP/IP, that is, on top of 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 allowing functions such as multicast and group communications [I-D.ietf-core-groupcomm]. As part of these design choices, CoAP uses the exponential backoff mechanism as a simple form of congestion control.

While the nature of UDP/IP transport for CoAP is well suited for constrained node communications [RFC6568], there are use cases where alternative transports would be better suited, or where UDP/IP is simply not available. In this section we discuss about a set of use cases where different transport channels could be useful.

A host with a CoAP client may reside behind a NAT or a firewall, and would like to talk to a CoAP server, possibly by using CoAP Observe-functionality [I-D.ietf-core-observe]. However, the host would wish to conserve resources, such as energy, and avoid NAT keepalives required to maintain NAT/firewall mappings. Furthermore, the application on the host may need to use HTTP for (initial) communications, but would preferably avoid use of HTTP/CoAP proxy, especially with "long polling" feature, required to be able to receive data from the CoAP server.

For the sake of simplicity, an application would like to communicate with constrained nodes using CoAP without using IP-based transport channels. For example, the application would like to use SMS [I-D.becker-core-coap-sms-gprs] or Bluetooth Low-Energy [BTCorev4.0] for communications. Furthermore, an application may be communicating via Delay-Tolerant Networks [RFC4838] using Bundle Protocol [RFC5050], and would like to transport CoAP formatted messages. In all of these cases it is not a given that UDP or IP are supported by a transport channel.

4. 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. When a local end-application supplies the URI to its own CoAP client implementation, it is parsed before the appropriate header values are encoded into the CoAP request. At the same time, the scheme specified in the URI is verified to determine whether plain UDP should be used (’coap’) or

whether the CoAP client should use DTLS instead ('coaps') to initiate communication with a CoAP origin server. Secondly, the CoAP client decomposes the URI into a set of options, namely Uri-Host, Uri-Port, Uri-Path and Uri-Query that are encoded into the CoAP packet sent to the CoAP origin server to specify the target resource. An origin server receiving such a packet can reconstruct the original CoAP URI from the option values.

A COAP URI used in this way can allow an end-application to notify its CoAP implementation of the transport mechanism to be used. The CoAP URI can also be used to distinguish the transport being used between communicating CoAP entities.

How the transport protocol is identified as a distinguishable component within the URI without violating [RFC3986], whilst ensuring little or no impact to [I-D.ietf-core-coap] is the challenge. We envision several alternatives for the CoAP URI based on available existing practices, each having its strengths and limitations.

4.1. Transport in URI scheme name

For a URI that is expressed generically as

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

The transport protocol can be expressed as part of the scheme name. According to [RFC3986] scheme names consist of a sequence of characters beginning with letter and followed by any combination of letters, digits, plus ("+"), period ("."), or hyphen ("-"). [I-D.ietf-core-coap] uses "coaps" instead of "coap" to specify DTLS as the transport, while the preferred form used by [I-D.becker-core-coap-sms-gprs] is "coap+tel". Using alternative transports would therefore invoke new scheme names, such as "coap+sip", "coap+tcp", "coap+l2cap" and so on. The authority component of the URI would invariably then be the end point identifier specific for that transport required, be it an IP-enabled endpoint or not.

Expressing the transport this way conforms to [RFC4395]. When such a URI is provided from an end-application to its CoAP implementation, URL parsing to retrieve the transport type and endpoint identifier is trivial. It is also expected that CoAP implementations not recognising new scheme names may simply discard the request or response procedure.

As the usage of each such scheme name results in an entirely new scheme, IANA intervention is required for the registration of each scheme name. Consequently such a registration process must conform to the guidelines stipulated in [RFC4395], particularly where

permanent URI scheme registration is concerned. Care must therefore be taken to ensure the scheme is well-defined and unambiguous in the transport description.

4.2. 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 [RFC3588] 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.

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.

4.3. Expressing transport in the URI in other ways

Other means of indicating the transport are also possible, and while these schemes might be incompatible with existing practices, they are presented for the sake of completeness.

4.3.1. 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.

4.3.2. 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.

4.4. Other Considerations

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

1. When CoAP communication over an alternative transport is desired, a clear, unambiguous name should be used. As an example, both Bluetooth Low Energy and Classic Bluetooth carry traffic over L2CAP. A "coap+l2cap" scheme name would therefore raise ambiguity.

2. 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 would then need to be registered.

5. Alternative Transport Analysis and Requirements

In this section we take a general look at alternative protocols for CoAP and the requirements CoAP imposes on underlying layer in order to successfully support various kinds of functionality. CoAP factors lossiness, unreliability, small packet sizes and connection statelessness into its protocol logic. We discuss general transport differences and requirements to carry CoAP packets here. Note that Reqs 1, 2, and 3 are related.

REQ1: Ability to provide unique 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.

REQ2: 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.

REQ3: 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.

REQ4: Binary encoding support.

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 to exist in allowing binary payloads to be transferred over text-based transport protocols

REQ5: 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.

REQ6: 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.

REQ7: 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 sleeping CoAP 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.

6. Acknowledgements

7. IANA Considerations

This memo includes no request to IANA.

8. 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.

9. 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.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., Bormann, C., and B. Frank, "Constrained Application Protocol (CoAP)", draft-ietf-core-coap-13 (work in progress), December 2012.

[I-D.ietf-core-groupcomm]
Rahman, A. and E. Dijk, "Group Communication for CoAP", draft-ietf-core-groupcomm-05 (work in progress), February 2013.

[I-D.ietf-core-observe]
Hartke, K., "Observing Resources in CoAP", draft-ietf-core-observe-07 (work in progress), October 2012.

[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.

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

[RFC3588] Calhoun, P., Loughney, J., Guttman, E., Zorn, G., and J. Arkko, "Diameter Base Protocol", RFC 3588, September 2003.

[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.

[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.

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