

Workgroup: CoRE
Internet-Draft:
draft-amsuess-core-coap-over-gatt-01
Published: 2 November 2020
Intended Status: Experimental
Expires: 6 May 2021
Authors: C. Amsüss

CoAP over GATT (Bluetooth Low Energy Generic Attributes)

Abstract

Interaction from computers and cell phones to constrained devices is limited by the different network technologies used, and by the available APIs. This document describes a transport for the Constrained Application Protocol (CoAP) that uses Bluetooth GATT (Generic Attribute Profile) and its use cases.

Note to Readers

Discussion of this document takes place on the CORE Working Group mailing list (core@ietf.org), which is archived at <https://mailarchive.ietf.org/arch/browse/core/>.

Source for this draft and an issue tracker can be found at <https://gitlab.com/chrysn/coap-over-gatt/>.

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 6 May 2021.

Copyright Notice

Copyright (c) 2020 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.

Table of Contents

- [1. Introduction](#)
 - [1.1. Procedural status](#)
 - [1.2. Application example](#)
- [2. Terminology](#)
- [3. Protocol description](#)
 - [3.1. Requests and responses](#)
 - [3.2. Addresses](#)
 - [3.2.1. Scheme-free alternative](#)
 - [3.3. Compression and reinterpretation of non-CoAP characteristics](#)
- [4. IANA considerations](#)
 - [4.1. Uniform Resource Identifier \(URI\) Schemes](#)
- [5. Security considerations](#)
- [6. References](#)
 - [6.1. Normative References](#)
 - [6.2. Informative References](#)
- [Appendix A. Change log](#)
- [Author's Address](#)

1. Introduction

The Constrained Application Protocol (CoAP) [[RFC7252](#)] can be used with different network and transport technologies, for example UDP on 6LoWPAN networks.

Not all those network technologies are available at end user devices in the vicinity of the constrained devices, which inhibits direct communication and necessitates the use of gateway devices or cloud services. In particular, 6LoWPAN is not available at all in typical end user devices, and while 6LoWPAN-over-BLE (IPSP, the Internet Protocol Support Profile of Bluetooth Low Energy (BLE), [[RFC7668](#)]) might be compatible from a radio point of view, many operating systems or platforms lack support for it, especially in a user-accessible way.

As a workaround to access constrained CoAP devices from end user devices, this document describes a way encapsulate generic CoAP exchanges in Bluetooth GATT (Generic Attribute Profile). This is

explicitly not designed as means of communication between two devices in full control of themselves - those should rather build an IP based network and transport CoAP as originally specified. It is intended as a means for an application to escape the limitations of its environment, with a special focus on web applications that use the Web Bluetooth [[webbluetooth](#)]. In that, it is similar to CoAP-over-WebSockets [[RFC8323](#)].

1.1. Procedural status

[This section will be removed before publication.]

The path of this document is currently not clear. It might attract interest in the CoRE working group, but might be easier to process as an independent submission.

1.2. Application example

Consider a network of home automation light bulbs and switches, which internally uses CoAP on a 6LoWPAN network and whose basic pairing configuration can be done without additional electronic devices.

Without CoAP-over-GATT, an application that offers advanced configuration requires the use of a dedicated gateway device or a router that is equipped and configured to forward between the 6LoWPAN and the local network. In practice, this is often delivered as a wired gateway device and a custom app.

With CoAP-over-GATT, the light bulbs can advertise themselves via BLE, and the configuration application can run as a web site. The user navigates to that web site, and it asks permission to contact the light bulbs using Web Bluetooth. The web application can then exchange CoAP messages directly with the light bulb, and have it proxy requests to other devices connected in the 6LoWPAN network.

For browsers that do not support Web Bluetooth, the same web application can be packaged into a native application consisting of a proxy process that forwards requests received via CoAP-over-WebSockets on the loopback interface to CoAP-over-GATT, and a browser view that runs the original web application in a configuration to use WebSockets rather than CoAP-over-GATT.

That connection is no replacement when remote control of the system is desired (in which case, again, a router is required that translates 6LoWPAN to the rest of the network), but suffices for many commissioning tasks.

2. Terminology

3. Protocol description

3.1. Requests and responses

[This section is not thought through or implemented yet, and could probably end up very different.]

CoAP-over-GATT uses individual GATT Characteristics to model a reliable request-response mechanism. Therefore, it has no message types or message IDs (in which it resembles CoAP-over-TCP [[RFC8323](#)]), and no tokens. In the place of tokens, different Bluetooth characteristics (comparable to open ports in IP based networks) can be used. All messages use GATT to ensure reliable transmission.

A GATT server announces service of UUID 8df804b7-3300-496d-9dfa-f8fb40a236bc (abbreviated US in this document), with one or more characteristics of UUID 2a58fc3f-3c62-4ecc-8167-d66d4d9410c2 (abbreviated UC).

[Right now, this only supports requests from the GATT client to the GATT server; role reversal might be added later.]

A client can start a CoAP request by writing to the UC characteristic a sequence composed of a single code byte, any options encoded in the option format of [[RFC7252](#)] Section 3.1, optionally followed by a payload marker and the request payload.

After the successful write, the client can read the response back from the server on the same characteristic. The client may need to attempt reading the characteristic several times until the response is ready, and may subscribe to indications to get notified when the response is ready.

The server does not need to keep the response readable after it has been read successfully.

If the request and initial response establish an observation, the client may keep reading; the server may keep the latest notification available indefinitely (especially if it turns out that "has been read successfully" is hard to determine) or make it readable only once for each new state.

Once the client writes a new request to a UC characteristic, any later reads pertain to that request, and any observation previously established is cancelled implicitly.

Attribute values are limited to 512 Bytes ([[bluetooth52](#)] Part F Section 3.2.9), practically limiting blockwise operation ([[RFC7959](#)]) to size exponents to 4 (resulting in a block size of 256 byte). Even smaller messages might enhance the transfer efficiency when they avoid fragmentation at the L2CAP level.

If a server provides multiple OC typed characteristics, parallel requests or observations are possible; otherwise, this transport is limited to a single pending request.

3.2. Addresses

```
[ ... coap+bluetooth://00-11-22-33-44-55-66-77-88-99/.well-known/
core ... ]
```

Note that when using Web Bluetooth [[webbluetooth](#)], neither the own nor the peer's address are known to the application. They may come up with an application-internal authority component (e. g. coap+bluetooth://id-SomeInternalIdentifier/.well-known/core), but must be aware that those can not be expressed towards anything outside the local stack.

3.2.1. Scheme-free alternative

As an alternative to the abovementioned scheme, a zone in .arpa could be registered to use addresses like

```
coap://001122334455.ble.arpa/.well-known/core
```

where the .ble.arpa address do not resolve to any IP addresses.

```
[ Accepting this will require a .arpa registering IANA consideration
to replace the URI one. ]
```

3.3. Compression and reinterpretation of non-CoAP characteristics

The use of SCHC is being evaluated in combination with CoAP-over-GATT; the device can use the characteristic UUID to announce the static context used.

Together with non-traditional response forms ([[I-D.bormann-core-responses](#)]) and contexts that expand, say, a numeric value 0x1234 to a message like

```
2.05 Content Response-For: GET /temperature Content-Format:
application/senml+cbor Payload (in JSON-ish equivalent): [ {1 /*
unit */: "K", 2 /* value */: 0x1234} ]
```

This enables a different use case than dealing with limited environments: Accessing BLE devices via CoAP without application

specific gateways. Any required information about the application can be expressed in the SCHC context.

4. IANA considerations

4.1. Uniform Resource Identifier (URI) Schemes

IANA is asked to enter a new scheme into the "Uniform Resource Identifier (URI) Schemes" registry set up in [[RFC7595](#)]:

*URI Scheme: "coap+gatt"

*Description: CoAP over Bluetooth GATT (sharing the footnote of coap+tcp)

*Well-Known URI Support: yes, analogous to [[RFC7252](#)]

5. Security considerations

All data received over GATT is considered untrusted; secure communication can be achieved using OSCORE [[RFC8613](#)].

Physical proximity can not be inferred from this means of communication.

6. References

6.1. Normative References

[[RFC7252](#)] Shelby, Z., Hartke, K., and C. Bormann, "The Constrained Application Protocol (CoAP)", RFC 7252, DOI 10.17487/RFC7252, June 2014, <<https://www.rfc-editor.org/info/rfc7252>>.

[[RFC7595](#)] Thaler, D., Ed., Hansen, T., and T. Hardie, "Guidelines and Registration Procedures for URI Schemes", BCP 35, RFC 7595, DOI 10.17487/RFC7595, June 2015, <<https://www.rfc-editor.org/info/rfc7595>>.

6.2. Informative References

[[RFC7668](#)] Nieminen, J., Savolainen, T., Isomaki, M., Patil, B., Shelby, Z., and C. Gomez, "IPv6 over BLUETOOTH(R) Low Energy", RFC 7668, DOI 10.17487/RFC7668, October 2015, <<https://www.rfc-editor.org/info/rfc7668>>.

[[webbluetooth](#)] Grant, R. and O. Ruiz-Henríquez, "Web Bluetooth", 24 February 2020, <<https://webbluetoothcg.github.io/web-bluetooth/>>.

[RFC8323]

Bormann, C., Lemay, S., Tschofenig, H., Hartke, K., Silverajan, B., and B. Raymor, Ed., "CoAP (Constrained Application Protocol) over TCP, TLS, and WebSockets", RFC 8323, DOI 10.17487/RFC8323, February 2018, <<https://www.rfc-editor.org/info/rfc8323>>.

[RFC8613]

Selander, G., Mattsson, J., Palombini, F., and L. Seitz, "Object Security for Constrained RESTful Environments (OSCORE)", RFC 8613, DOI 10.17487/RFC8613, July 2019, <<https://www.rfc-editor.org/info/rfc8613>>.

[RFC7959]

Bormann, C. and Z. Shelby, Ed., "Block-Wise Transfers in the Constrained Application Protocol (CoAP)", RFC 7959, DOI 10.17487/RFC7959, August 2016, <<https://www.rfc-editor.org/info/rfc7959>>.

[bluetooth52] "Bluetooth Core Specification v5.2", 31 December 2019,

<https://www.bluetooth.org/docman/handlers/downloaddoc.ashx?doc_id=478726>.

[I-D.bormann-core-responses]

Bormann, C., "CoAP: Non-traditional response forms", Work in Progress, Internet-Draft, draft-bormann-core-responses-00, 12 November 2017, <<http://www.ietf.org/internet-drafts/draft-bormann-core-responses-00.txt>>.

Appendix A. Change log

Since -00:

*Add note on SCHC possibilities.

Author's Address

Christian Amsüss
Austria

Email: christian@amsuess.com