CoRE Working Group
Internet-Draft
Intended status: Informational
Expires: October 21, 2017

C. Groves

W. Yang
Huawei

April 19, 2017

A WebRTC Data Channel Transport for the Constrained Application Protocol (CoAP)

draft-groves-coap-webrtcdc-02

Abstract

The WebRTC framework defines a generic transport service allowing WEB-browsers and other endpoints to exchange generic data from peer to peer utilizing a Stream Control Transmission Protocol (SCTP) transport. This service is known as Web Real Time Communication WebRTC data channels (WebRTC DC). The use of WebRTC DCs for the Constrained Application Protocol (CoAP) allows WebRTC enabled devices to exchange CoAP data between peers in a secure reliable manner.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Whilst the Constrained Application Protocol (CoAP) [RFC7252] was designed for Internet of Things (IoT) deployments in constrained network environments its ready adoption has seen the use of it in a multitude of different network environments. For example [I-D.silverajan-core-coap-alternative-transports] provides use cases for alternate CoAP transports.

[I-D.ietf-core-coap-tcp-tls] highlights a number of issues using the native User Datagram Transport (UDP) and envisages deployments more closely integrated with a Web environment. It also proposes the use of the WebSocket protocol [RFC6455]. The use of CoAP over WebRTC DCs has not yet been discussed.

WebRTC is a framework [I-D.ietf-rtcweb-overview] that defines real time protocols for browser-based applications. It allows communications between peer WebRTC endpoints (e.g. browsers) without the need to communicate through a web server.

In addition to protocols for the realtime transport of audio and video, the transport of generic peer-to-peer non-media data has been defined using WebRTC DCs. The non-media data is transported using the Stream Control Transmission Protocol (SCTP) [RFC4960] encapsulated in the Datagram Transport Layer Security (DTLS) [RFC6347]. It allows both reliable and partially reliable transport and provides confidentiality, source authenticated and integrity protected transfers. The use of Interactive Connectivity Establishment (ICE) [RFC5245] allows network address translator (NAT) traversal. The SCTP/DTLS association may be shared with existing audio and video streams enabling multiplexing of several data streams over a single port further facilitating NAT traversal.

Use cases for WebRTC DCs (section 3.1/[I-D.ietf-rtcweb-data-channel] envisage scenarios where the real-time gaming experience is enhanced by additional object state information. Additional scenarios are considered where information such as heart rate sensor or oxygen saturation sensors could augment audio and video in remote medicine scenarios. The transport of such sensor information is what CoAP has been designed for.

This is illustrated in Figure 1 showing the WebRTC Trapeziod with added sensor/CoAP information. The left hand side WebRTC endpoint acts as a CoAP to CoAP proxy.

```
                  +-----------+             +-----------+
                  |   Web     |             |   Web     |
                  |           |  Signaling  |           |
                  |           |-------------|           |
                  |  Server   |   path      |  Server   |
                  |           |             |           |
                  +-----------+             +-----------+
                       /                           \
                      /                             \ Application-
                     /                               \ defined over
                    /                                 \ HTTP/Websockets
                   /  Application-defined over         \
                  /   HTTP/Websockets                   \
                 /                                       \
          +-----------+                           +-----------+
          |JS/HTML/CSS|                           |JS/HTML/CSS|
          +-----------+                           +-----------+
          +-----------+                           +-----------+
   SensorA |           |                           |           |
   CoAP/UDP |           |                           |           |
     +------+  Browser  | ------------------------- |  Browser  +
          |           |          Media path       |           |
          |           |       (CoAP/WebRTC DC)    |           |
          +-----------+                           +-----------+
```

Figure 1: CoAP and WebRTC Trapeziod

By utilizing the WebRTC DC (SCTP over DTLS over ICE/UDP (or ICE/TCP)) transport for CoAP a number of important features are inherited including: congestion control, order and unordered messages delivery, large message transmission by providing segmentation and reassembly and multiple unidirectional streams. A more detailed analysis of the benefits of WebRTC DCs can be found in section 5/[I-D.ietf-rtcweb-data-channel]. [I-D.ietf-tsvwg-sctp-dtls-encaps] describes the usage of SCTP over DTLS.

WebRTC defines in-band and out-of-band methods for establishing a data channel and indicating its characteristics. The Data Channel Establishment Protocol (DCEP) [I-D.ietf-rtcweb-data-protocol] provides an in band means of establishing individual data channels. [I-D.ietf-mmusic-data-channel-sdpneg] uses the Session Description Protocol (SDP) [RFC4566] to provide an out-of-band means to establish data channels.

By defining the use of CoAP over WebRTC DC it negates the need for the WebRTC endpoint to interwork between any CoAP messages received from local devices to a proprietary WebRTC DC format when signalling a remote WebRTC endpoint.

The SCTP Payload Protocol Identifier (PPID) allows the identification of whether a UTF-8 or Binary encoding is being used and thus facilitates the use of text or binary CoAP protocol serializations.

2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. Constrained Application Protocol

This section describes the use of CoAP over WebRTC DC as a delta to the information contained in section 2/[RFC7252].

Figure 2 shows the CoAP abstract layering as applied to the WebRTC framework.

```
                      +----------------------------+
                       |        Application        |
                       +------+------+-------------+ \
                       | DCEP | Requests/Responses | |
                       |      +--------------------| | CoAP
                       |      | Messages           | |
                       +------+--------------------+ /
                       |        SCTP               |
          +-----------------------------------------+
          | STUN | SRTP |        DTLS               |
          +-----------------------------------------+
          |                ICE                      |
          +-----------------------------------------+
          | UDP1 | UDP2 | UDP3 | ...                |
          +-----------------------------------------+
```

Figure 2: WebRTC protocol layers including CoAP

WebRTC DC mandates the use of SCTP over DTLS. Whilst the above diagram indicates the use of ICE over UDP the use of TCP is also possible in fall back scenarios.

3.1. Message Model

WebRTC DC allows application protocol messages to be exchanged by peers. WebRTC supports both a reliable and partially reliable methods of transmitting user messages.

CoAP [RFC7252] supports four message types "Confirmable, Non-Confirmable, Acknowledge and Reset". As SCTP provides the reliability mechanism the CoAP message types are not needed for CoAP over WebRTC DC.

WebRTC DC does not support multicast usage.

3.2. Request Response Model

WebRTC DCs are realized as a pair of one incoming and one outgoing SCTP stream (with the same identifier) allowing bi-directional communication. Each channel has properties (see section 6.4/[I-D.ietf-rtcweb-data-channel] as discussed below:

- o reliable or unreliable message transmission: WebRTC DCs support the per message indication whether user messages are reliable or partially reliable. Partial reliability indicates that message retransmission is limited to a certain number of retransmissions or lifetime. This loosely parallels to the CoAP usage of Confirmable (CON) or Non-confirmable (NON) messages.

- o in-order or out-of-order message delivery: WebRTC DCs support the per message indication whether user messages are delivered in or out of order. CoAP has been designed for unreliable transports and therefore assumes that messages may arrive out-of-order. CoAP implements a lightweight reliability mechanism to deal with this issue.

- o priority: WebRTC DCs allows a priority to specified for stream scheduling. The usage of this is application specific. Usage of CoAP has no impact on this parameter. It's up to the application using CoAP to set this indication.

- o an optional label: This is an application/implementation specific label. Uniqueness is not guaranteed. Usage of CoAP has no impact on this parameter.

- o an optional protocol: This is used to indicate the application protocol in use. A value is required to identify the usage of CoAP.

As discussed above WebRTC DC supports an unreliable / un-ordered delivery of messages. Implementations utilizing these data channel characteristics may use CoAP messages and request/response model largely unchanged. In this case the CoAP reliability mechanisms would be used. However as WebRTC DC's usage of SCTP is reliable or partially reliable there is some redundancy between the functionality that WebRTC DCs and CoAP provides.

The redundancies are identified and discussed in section 2/[I-D.ietf-core-coap-tcp-tls]. Namely:

1. There is no need to carry acknowledgement semantics at a CoAP level.

2. There is no need for duplicate delivery detection. This is part of the SCTP layer.

3.3. Intermediaries and Caching

As CoAP over WebRTC DC is peer to peer no intermediares or caching is expected.

3.4. Resource Discovery

The usage of CoAP over WebRTC DC has no foreseeable impacts on resource discovery.

3.5. Opening Handshake

Prior to the establishment of a CoAP over WebRTC DC the characteristics of the SCTP association and data channel may be negotiated by signalling. See Section 4 for further details. For example when using SDP [I-D.ietf-mmusic-sctp-sdp] the use of the "SDP max-message-size" attribute indicates the maximum received SCTP message size.

Further characteristics (such as those described in Section 3.2) are negotiated at the establishment of the WebRTC DC.

On establishment of the CoAP over WebRTC DC the client and server MAY send a CoAP Capability and Settings message (CSM see Section 4.3/[I-D.ietf-core-coap-tcp-tls]) as its first message on the connection to establish CoAP specific capabilities. Any capabilities signalled SHALL not contradict previously negotiated chracteristics. Consideration for the individual options are below:

- Server-Name Setting: CoAP over WebRTC DC clients MAY use the server-name setting option. The initial value is derived based on the signalling method used to establish the WebRTC peer to peer communications. WebRTC does not mandate a signalling method. For example if Websockets is used then the value may be taken from the HTTP host header field.

- Max-message size Capability: The CoAP Max-Message-Size shall not exceed the SCTP message size.

- o Block-wise Transfer Capability: CoAP over WebRTC DC client and server MAY support the use of BERT (Section 5/[I-D.ietf-core-coap-tcp-tls]). See Section 4.6 for message size considerations.

- o Ping and Pong Messages: Ping and Pong messages MAY be sent by CoAP over WebRTC DC clients and servers. However its use as a basic keepalive is not required as WebRTC defines a method to determine liveness (see Section 4.1).

- o Release Messages: CoAP over WebRTC DC clients and servers may support the CoAP Release message. On receipt of a release message the CoAP over WebRTC DC SHALL be closed as per Section 4.

- o Abort Messages: CoAP over WebRTC DC clients and servers may support the CoAP Abort message. Senders SHALL then close the CoAP over WebRTC DC as per Section 4.

3.6. Message Format

As discussed in [I-D.ietf-core-coap-tcp-tls] the use of a reliable underlying transport allows the use of a modified CoAP header format. The modified format removes the "Type (T)" and "Message ID" fields and introduces a "length" as illustrated below in Figure 3.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Len=13 |  TKL  | Length (8-bit)|      Code     | TKL bytes ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Options (if any) ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|1 1 1 1 1 1 1 1|    Payload (if any) ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3: CoAP Header with TCP with 8-bit Length in Header

CoAP over WebRTC DC implementations shall also use the message format in Figure 3 with the following consideration:

- o The length field was added for message delimitation to keep messages separate in TCP. WebRTC DC uses the message orientation of SCTP to preserve message boundaries thus the use of single application message per SCTP user message is mandated by the WebRTC framework. The length field shall be set to 0.

CoAP [RFC7252] supports the use of different content-formats. WebRTC DC defines the use of PPIDs per SCTP user message as follows:

- o WebRTC String: to identify a non-empty JavaScript string encoded in UTF-8.

- o WebRTC Binary: to identify a non-empty JavaScript binary data (ArrayBuffer, ArrayBufferView or Blob).

Depending on the content-format (see section 12.3/[RFC7252]) an appropriate PPID to the encoding type SHOULD be used to minimise the need for translating between encodings. For example content type of "text/plain" would result in the use of PPID "WebRTC String".

Author's note: Specific mappings for each content-format could be provided however given that the formats may change in the future it may be sufficient to offer broad guidance instead.

3.7. Option Format and Value

There are no impacts to option formats or values due to the use of CoAP over WebRTC DCs.

Author's note: Given that the host is determined by the usage of WebRTC are the Uri-Host and Uri-Port relevant? It would seem that this may be valuable to establish a resource tree independent of WebRTC.

4. Message Transmission

In order to use a WebRTC DC, a SCTP over DTLS over ICE/UDP (or ICE/TCP) association must be established. A DTLS connection is established followed by an SCTP association. The out-of-band establishment method through the use of SDP-based Data Channel Negotiation [I-D.ietf-mmusic-data-channel-sdpneg] allows the negotiation of SCTP over DTLS over ICE/UDP as well as the negotiation and establishment of the characteristics of an individual WebRTC DC.

The in-band establishment method through the use of the Data Channel Establishment Protocol (DCEP) [I-D.ietf-rtcweb-data-protocol] only allows for the establishment of a WebRTC DC once the SCTP over DTLS is established. It relies on DATA_CHANNEL_OPEN and DATA_CHANNEL_ACK messages on the relevant SCTP stream to negotiate the properties of the channel. A separate SCTP PPID (50) indicates that the SCTP user message is a WebRTC DCEP message to allow de-multiplexing by the endpoint.

WebRTC DCs are realized as a pair of one incoming and one outgoing SCTP stream (with the same identifier). Requests are sent on an outgoing SCTP stream and received on the peer incoming stream. The SCTP stream identifier is bound to the WebRTC DC instance at the

establishment of the data channel. The establishment protocol provides rules for determining the SCTP stream IDs.

WebRTC DC closure (Stream Reset) is supported through the use of the SCTP stream reconfiguration extension defined in [RFC6525]. The SCTP Stream Reconfiguration reset has the effect of setting the numbering sequence of the SCTP stream back to zero. This is separate function to the CoAP "Reset" message. There is no mapping between the SCTP Stream Reset and the CoAP "Reset" message.

4.1. Messages and Endpoints

As per section 2.5/[I-D.ietf-core-coap-tcp-tls] requests can be sent from both the connecting host and the endpoint that accepted the connection. Who initiated the SCTP/DTLS connection has no bearing on the meaning of the CoAP terms client and server.

WebRTC DC mandates the use of DTLS thus the endpoint is identified depending on the security mode.

WebRTC DCs allows the indication of whether a SCTP user message is empty through the use of PPIDs (WebRTC String Empty and WebRTC Binary Empty). CoAP defines the use of empty messages. However from the perspective of SCTP these CoAP messages would still contain header information thus PPIDs for empty data MUST not be used.

CoAP uses an Empty Confirmable message to provoke a Reset message to check the liveness of an endpoint (so called "CoAP" ping). In WebRTC liveness and the ability to send data is determined through the usage of Session Traversal Utilities for NAT (STUN) Usage for Consent Freshness [RFC7675]. Therefore endpoints utilising CoAP over WebRTC DC MUST not use CoAP "reset" messages.

CoAP also uses Empty messages to acknowledge a request. This is not required due to the SCTP level acknowledgement. Therefore Empty messages MUST not be used with CoAP over WebRTC.

4.2. Messages Transmitted Reliably

For CoAP messages marked as confirmable the sender SHALL use a reliable SCTP user message.

A CoAP endpoint MUST use the ordered delivery SCTP service, as described in [RFC4960], for the CoAP protocol.

CoAP receivers MUST NOT generate CoAP "ACK" or "reset" messages. SCTP level acknowledgement mechanisms are used.

4.3. Messages Transmitted without Reliability

WebRTC DC makes use of the SCTP Partial Reliability (SCTP-PR) Extension [RFC3758]. This extension allows a user to indicate on a per message basis how persistent the transport service should be in attempting to send the message to the receiver. One of the benefits of using this extension identified by [RFC3758] is:

1. Some application layer protocols may benefit from being able to use a single SCTP association to carry both reliable content, - such as text pages, billing and accounting information, setup signaling - and unreliable content, e.g., state that is highly sensitive to timeliness, where generating a new packet is more advantageous than transmitting an old one.

This benefit is also one of the reasons the CoAP "Non-Confirmable" message was introduced. However the SCTP-PR and the CoAP "Non-Confirmable" message mechanisms differs in their approach. The SCTP-PR mechanism focuses on sender side behaviour (e.g. when to abandon retransmission). The CoAP "Non-Confirmable" message focuses on receiver side behaviour (e.g. must not send a CoAP ACK). Even with the use of SCTP-PR an SCTP receiver will send an SCTP level ACK for a successfully received SCTP CHUNK. The CoAP "Non-Confirmable" message has no effect on the SCTP level function.

Therefore the use of a CoAP "Non-Confirmable" message type is redundant as the CoAP receiver will never send a CoAP ACK message in response.

SCTP-PR provides a complimentary function and thus CoAP senders who send Non-confirmable messages SHALL also use SCTP-PR for that message.

4.4. Message Correlation

Due to reliability being handled at the SCTP layers the CoAP "Message ID" is not required.

4.5. Message Duplication

The SCTP layer provides message duplication protection. The CoAP application level procedure is not required.

4.6. Message Size

The considerations in section 4.1/[I-D.ietf-core-coap-tcp-tls] regarding message size limitations also apply to the use of WebRTC DCs. However [I-D.ietf-rtcweb-data-channel] indicates that senders

SHOULD limit the maximum message size to 16KB to avoid monopolization of the SCTP association. Section 5/[I-D.ietf-tsvwg-sctp-dtls-encaps] provides further details regarding segmentation and reassembly and path maximum transmission unit (MTU) discovery.

Interleaving of large user messages is supported by an SCTP protocol extension defined in [I-D.ietf-tsvwg-sctp-ndata].

4.7. Congestion Control

SCTP provides congestion control on a per-association basis (see section 5/[I-D.ietf-rtcweb-data-channel].

4.8. Transmission Parameters

The application level parameters defined in section 4.8/[RFC7252] are not relevant to SCTP.

5. Request/Response Semantics

Request and response semantics for CoAP over WebRTC DC is as per section 5/[RFC7252] with the following exceptions:

- o section 5.2/[RFC7252]: separate responses MUST be used. Given that WebRTC DC provides an SCTP level acknowledgement it is not possible to piggy back CoAP responses.

- o section 5.3.1/[RFC7252]: due to the use of DTLS the advice regarding token use without using TLS is invalid.

- o section 5.3.2/[RFC7252]: In addition CoAP request/response matching is unique to a particular WebRTC DC (SCTP StreamID pair).

- o section 5.8/[RFC7252]: It is not possible to use a 4.05 piggybacked response.

6. CoAP URI

CoAP [RFC7252] defines the "coap" and "coaps" URI schemes for identifying CoAP resources and providing a means of locating the resource. [RFC7252] defines these resources for use with CoAP over UDP.

Section 8/[RFC7252] (Multicast CoAP), does not apply to the URI schemes defined in the present specification.

Resources made available via the "coaps+wr" schemes have no shared identity with the other scheme or with the "coap" or "coaps" scheme,

even if their resource identifiers indicate the same authority (the same host listening to the same port). The schemes constitute distinct namespaces and, in combination with the authority, are considered to be distinct origin servers.

6.1. coaps+wr URI scheme

```
coaps-wr-URI = "coaps+wr:" "//" host [ ":" port ] path-abempty
                  [ "?" query ]
```

The semantics defined in section 6.3/[RFC7252], apply to this URI scheme, with the following changes:

- o The port SHALL be omitted. The underlying UDP or TCP port and SCTP port is negotiated prior to the establishment of the CoAP over WebRTC DC.

7. Discovery

7.1. Service Discovery

WebRTC does not define peer discovery mechanisms. Peers discover each other through the use of the ICE protocol. ICE candidates need to be sent from peer to peer via signalling. The Javascript Session Establishment Protocol (JSEP) [I-D.ietf-rtcweb-jsep] details the generic SDP media descriptions for peer endpoints to determine the characteristics of a session. The actual signalling protocol between application servers is unspecified. WebRTC endpoints MUST implement the network functions detailed by JSEP including ICE functionality.

Whilst the inter-application server signalling protocol is unspecified, the Session Initiation Protocol (SIP) is able to carry SDP for the purposes of establishing a CoAP over WebRTC DC session. SIP allows the use of media feature tags to indicate user agent capabilities [RFC3840]. In order to indicate that a SIP user agent supports the use of CoAP a new "sip.coap" media feature tag is proposed. A CoAP-capable endpoint SHOULD include this media feature tag in its REGISTER requests and OPTION responses. It SHOULD also include the media feature tag in INVITE and UPDATE [RFC3311] requests and responses. Presence of the media feature tag in the contact field of a requestor response can be used to determine that the far end supports CLUE.

The exchange of SDP results in: the underlying transport address (e.g. IPv4 or IPv6), the underlying transport port (e.g. UDP port) the SCTP port and the SCTP StreamID used for the CoAP WebRTC DC being exchanged between the peer endpoints.

7.2. Resource Discovery

On establishment of a CoAP WebRTC DC endpoints are able to use the resource discovery mechanism defined in [RFC6690] for CoAP resources.

8. Multicast CoAP

WebRTC DCs do not support multicast.

9. Securing CoAP

This document defines how to convey CoAP over WebRTC DCs. The WebRTC security architecture [I-D.ietf-rtcweb-security-arch] mandates the use of DTLS for data channels. The use of DTLS 1.2 is compatible with CoAP [RFC7252] which allows makes use of DTLS 1.2.

The use of DTLS for WebRTC is detailed in [I-D.ietf-rtcweb-security-arch].

10. Interworking

An WebRTC endpoint supporting CoAP may in affect act as a gateway between local sensor devices and a remote peer endpoint. The local sensors may utilise CoAP over an alternate signalling transport such as UDP to the local WebRTC endpoint. The WebRTC endpoint may then utilise CoAP over WebRTC to signal to the remote peer.

A CoAP gateway when converting to and from a WebRTC transport will in general perform the following functions:

- Map received Empty CoAP message to SCTP level operations and discard the empty message.

- Map received ACK message to SCTP level operations and discard the ACK message.

- Separate piggy-backed messages.

- Provide a mapping between received and sent Tokens in order to match requests and responses.

Other behaviour depends on the type of proxy behaviour the gateway is performing. See section 5.7/[RFC7252] for more details.

11. Security Considerations

Security considerations for WebRTC are discussed in [I-D.ietf-rtcweb-security].

The use of CoAP over WebRTC can potentially negate the risks mentioned in:

- section 11.3/[RFC7252] on insecure UDP and multicast being used to aid an amplification attack.

- section 11.4/[RFC7252] on IP address spoofing and section 11.5/[RFC7252] on Cross-Protocol attacks.

- section 11.6/[RFC7252] may also not be relevant as WebRTC endpoints are not expected to be severely constrained.

Of particular relevance to the support of CoAP over WebRTC DC is access to local devices. Devices generating CoAP data are essentially the same as cameras and microphones in that they may expose sensitive data about the user or the location of the device. Thus the guidance of section 4.1/[I-D.ietf-rtcweb-security] applies to devices generating CoAP data. Whilst CoAP has been designed for constrained devices where there is no user interface to inform/ request consent, it is assumed that device utilising WebRTC DC for CoAP is more likely at minimum a Class 2 [RFC7228] device that could facilitate consent.

The CoAP media feature tag defined by this document tag may be present in sessions not utilising CoAP, which increases the metadata available about the sending device, which can help an attacker differentiate between multiple devices and help them identify otherwise anonymised users via the fingerprint of features their device supports. To prevent this, SIP signalling SHOULD always be encrypted using TLS [RFC5630].

12. IANA Considerations

12.1. New WebRTC DC Protocol Value

NOTE: This registration is exactly the same as the registration in [I-D.savolainen-core-coap-websockets].

This document requests the registration of the subprotocol name "coap.v1" in the WebSocket Subprotocol Name Registry.

- Subprotocol Identifier: coap.v1

- o Subprotocol Common Name: Constrained Application Protocol (CoAP)

- o Subprotocol Definition: This document

12.2. Secure Service Name and Port Number Registration

No need has been identified to register a new service name and port number for CoAP over WebRTC. Port number allocation is dynamic. The use of the SCTP over DTLS over UDP/TCP results in a layering of services.

12.3. ALPN Protocol ID

[I-D.ietf-core-coap-tcp-tls] defines a new "coap" application protocol negotiation protocol identity. However as the DTLS connection is used to establish a WebRTC application the protocol identifiers defined in [I-D.ietf-rtcweb-alpn] MUST be used. Note: that confidentiality protection does not extend to WebRTC DCs.

12.4. URI Schemes

This document registers a new URI scheme "coaps+wr" for the use of CoAP over WebRTC DCs. The "coaps+wr" URI schemes can be compared to the "https" URI scheme.

The IANA is requested to add this new URI schemes to the registry established with [RFC7595].

12.5. New SIP Media Feature Tag

This specification registers a new media feature tag in the SIP [RFC3264] tree per the procedures defined in [RFC2506] and [RFC3840].

Media feature tag name: sip.coap

ASN.1 Identifier: 1.3.6.1.8.4.30

Summary of the media feature indicated by this tag: This feature tag indicates that the device supports the Constrained Application Protocol (CoAP).

Values appropriate for use with this feature tag: Boolean.

The feature tag is intended primarily for use in the following applications, protocols, services, or negotiation mechanisms: This feature tag is useful to indicate the support of CoAP.

Related standards or documents: This document

Security Considerations: Security considerations for this media feature tag are discussed in Section 11.

Name(s) and email address(es) of person(s) to contact for further information:

- CORE workgroup: core@ietf.org

- CORE chairs: core-chairs@ietf.org

Intended usage: COMMON

13. Examples

The example SDP Offer shows a CoAP over WebRTC DC utilising out-of-band negotiation [I-D.ietf-mmusic-data-channel-sdpneg]. It is based on the example in section 7.2/[I-D.ietf-rtcweb-jsep]. Modified lines are indicated with ">>>" at the start of the line. These indicators are NOT part of the SDP syntax. Note: some lines have been broken into two lines for formatting reasons.

```
   v=0
      o=- 4962303333179871723 1 IN IP4 0.0.0.0
      s=-
      t=0 0
      a=group:BUNDLE a1 d1
      a=ice-options:trickle
      m=audio 9 UDP/TLS/RTP/SAVPF 96 0 8 97 98
      c=IN IP4 0.0.0.0
      a=rtcp:9 IN IP4 0.0.0.0
      a=mid:a1
      a=msid:57017fee-b6c1-4162-929c-a25110252400
             e83006c5-a0ff-4e0a-9ed9-d3e6747be7d9
      a=sendrecv
      a=rtpmap:96 opus/48000/2
      a=rtpmap:0 PCMU/8000
      a=rtpmap:8 PCMA/8000
      a=rtpmap:97 telephone-event/8000
      a=rtpmap:98 telephone-event/48000
      a=maxptime:120
      a=ice-ufrag:ATEn1v9DoTMB9J4r
      a=ice-pwd:AtSK0WpNtpUjkY4+86js7ZQl
      a=fingerprint:sha-256
                    19:E2:1C:3B:4B:9F:81:E6:B8:5C:F4:A5:A8:D8:73:04
                   :BB:05:2F:70:9F:04:A9:0E:05:E9:26:33:E8:70:88:A2
      a=setup:actpass
      a=rtcp-mux
      a=rtcp-rsize
      a=extmap:1 urn:ietf:params:rtp-hdrext:ssrc-audio-level
      a=extmap:2 urn:ietf:params:rtp-hdrext:sdes:mid
      a=ssrc:1732846380 cname:FocUG1f0fcg/yvY7

      m=application 0 UDP/DTLS/SCTP webrtc-datachannel
      c=IN IP4 0.0.0.0
      a=bundle-only
      a=mid:d1
      a=fmtp:webrtc-datachannel max-message-size=65536
      a=sctp-port 5000
      a=fingerprint:sha-256
                    19:E2:1C:3B:4B:9F:81:E6:B8:5C:F4:A5:A8:D8:73:04
                    :BB:05:2F:70:9F:04:A9:0E:05:E9:26:33:E8:70:88:A2
      a=setup:actpass
   >>>a=dcmap:0 subprotocol="coap.v1";"label="coap"
```

Figure 4: Example SDP Offer

14. Acknowledgements

We would like to thank the authors of [I-D.ietf-core-coap-tcp-tls] and [I-D.savolainen-core-coap-websockets] for providing a framework for this document. In addition we would like to thank Carsten Bormann for his feedback on message format.

15. Changelog

draft-groves-coap-webrtcdc-02:

- o Keep alive update. No changes.

draft-groves-coap-webrtcdc-01:

- o Updated message format to align with draft-core-coap-tcp-tls-04
- o Updates to align with draft-core-coap-tcp-tls-04 as a result of the merger with websockets. Added section on opening handshake. Added support of CoAP capability messages and BERT.

16. References

16.1. Normative References

[I-D.ietf-mmusic-data-channel-sdpneg]
Drage, K., Makaraju, M., Stoetzer-Bradler, J., Ejzak, R., and J. Marcon, "SDP-based Data Channel Negotiation", draft-ietf-mmusic-data-channel-sdpneg-12 (work in progress), March 2017.

[I-D.ietf-mmusic-sctp-sdp]
Holmberg, C., Shpount, R., Loreto, S., and G. Camarillo, "Session Description Protocol (SDP) Offer/Answer Procedures For Stream Control Transmission Protocol (SCTP) over Datagram Transport Layer Security (DTLS) Transport.", draft-ietf-mmusic-sctp-sdp-25 (work in progress), March 2017.

[I-D.ietf-rtcweb-alpn]
Thomson, M., "Application Layer Protocol Negotiation for Web Real-Time Communications (WebRTC)", draft-ietf-rtcweb-alpn-04 (work in progress), May 2016.

[I-D.ietf-rtcweb-data-channel]
Jesup, R., Loreto, S., and M. Tuexen, "WebRTC Data Channels", draft-ietf-rtcweb-data-channel-13 (work in progress), January 2015.

[I-D.ietf-rtcweb-data-protocol]
Jesup, R., Loreto, S., and M. Tuexen, "WebRTC Data Channel Establishment Protocol", draft-ietf-rtcweb-data-protocol-09 (work in progress), January 2015.

[I-D.ietf-rtcweb-jsep]
Uberti, J., Jennings, C., and E. Rescorla, "Javascript Session Establishment Protocol", draft-ietf-rtcweb-jsep-20 (work in progress), March 2017.

[I-D.ietf-rtcweb-overview]
Alvestrand, H., "Overview: Real Time Protocols for Browser-based Applications", draft-ietf-rtcweb-overview-18 (work in progress), March 2017.

[I-D.ietf-rtcweb-security]
Rescorla, E., "Security Considerations for WebRTC", draft-ietf-rtcweb-security-08 (work in progress), February 2015.

[I-D.ietf-rtcweb-security-arch]
Rescorla, E., "WebRTC Security Architecture", draft-ietf-rtcweb-security-arch-12 (work in progress), June 2016.

[I-D.ietf-tsvwg-sctp-dtls-encaps]
Tuexen, M., Stewart, R., Jesup, R., and S. Loreto, "DTLS Encapsulation of SCTP Packets", draft-ietf-tsvwg-sctp-dtls-encaps-09 (work in progress), January 2015.

[I-D.ietf-tsvwg-sctp-ndata]
Stewart, R., Tuexen, M., Loreto, S., and R. Seggelmann, "Stream Schedulers and User Message Interleaving for the Stream Control Transmission Protocol", draft-ietf-tsvwg-sctp-ndata-09 (work in progress), March 2017.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC2506] Holtman, K., Mutz, A., and T. Hardie, "Media Feature Tag Registration Procedure", BCP 31, RFC 2506, DOI 10.17487/RFC2506, March 1999, <http://www.rfc-editor.org/info/rfc2506>.

[RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", RFC 3264, DOI 10.17487/RFC3264, June 2002, <http://www.rfc-editor.org/info/rfc3264>.

[RFC3311] Rosenberg, J., "The Session Initiation Protocol (SIP) UPDATE Method", RFC 3311, DOI 10.17487/RFC3311, October 2002, <http://www.rfc-editor.org/info/rfc3311>.

[RFC3758] Stewart, R., Ramalho, M., Xie, Q., Tuexen, M., and P. Conrad, "Stream Control Transmission Protocol (SCTP) Partial Reliability Extension", RFC 3758, DOI 10.17487/RFC3758, May 2004, <http://www.rfc-editor.org/info/rfc3758>.

[RFC3840] Rosenberg, J., Schulzrinne, H., and P. Kyzivat, "Indicating User Agent Capabilities in the Session Initiation Protocol (SIP)", RFC 3840, DOI 10.17487/RFC3840, August 2004, <http://www.rfc-editor.org/info/rfc3840>.

[RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, DOI 10.17487/RFC4566, July 2006, <http://www.rfc-editor.org/info/rfc4566>.

[RFC4960] Stewart, R., Ed., "Stream Control Transmission Protocol", RFC 4960, DOI 10.17487/RFC4960, September 2007, <http://www.rfc-editor.org/info/rfc4960>.

[RFC5245] Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", RFC 5245, DOI 10.17487/RFC5245, April 2010, <http://www.rfc-editor.org/info/rfc5245>.

[RFC5630] Audet, F., "The Use of the SIPS URI Scheme in the Session Initiation Protocol (SIP)", RFC 5630, DOI 10.17487/RFC5630, October 2009, <http://www.rfc-editor.org/info/rfc5630>.

[RFC6347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, DOI 10.17487/RFC6347, January 2012, <http://www.rfc-editor.org/info/rfc6347>.

[RFC6525] Stewart, R., Tuexen, M., and P. Lei, "Stream Control Transmission Protocol (SCTP) Stream Reconfiguration", RFC 6525, DOI 10.17487/RFC6525, February 2012, <http://www.rfc-editor.org/info/rfc6525>.

[RFC6690] Shelby, Z., "Constrained RESTful Environments (CoRE) Link Format", RFC 6690, DOI 10.17487/RFC6690, August 2012, <http://www.rfc-editor.org/info/rfc6690>.

[RFC7228] Bormann, C., Ersue, M., and A. Keranen, "Terminology for Constrained-Node Networks", RFC 7228, DOI 10.17487/RFC7228, May 2014, <http://www.rfc-editor.org/info/rfc7228>.

[RFC7252] Shelby, Z., Hartke, K., and C. Bormann, "The Constrained Application Protocol (CoAP)", RFC 7252, DOI 10.17487/RFC7252, June 2014, <http://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, <http://www.rfc-editor.org/info/rfc7595>.

[RFC7675] Perumal, M., Wing, D., Ravindranath, R., Reddy, T., and M. Thomson, "Session Traversal Utilities for NAT (STUN) Usage for Consent Freshness", RFC 7675, DOI 10.17487/RFC7675, October 2015, <http://www.rfc-editor.org/info/rfc7675>.

16.2. Informative References

[I-D.ietf-core-coap-tcp-tls]
Bormann, C., Lemay, S., Tschofenig, H., Hartke, K., Silverajan, B., and B. Raymor, "CoAP (Constrained Application Protocol) over TCP, TLS, and WebSockets", draft-ietf-core-coap-tcp-tls-07 (work in progress), March 2017.

[I-D.savolainen-core-coap-websockets]
Savolainen, T., Hartke, K., and B. Silverajan, "CoAP over WebSockets", draft-savolainen-core-coap-websockets-07 (work in progress), June 2016.

[I-D.silverajan-core-coap-alternative-transports]
Silverajan, B. and T. Savolainen, "CoAP Communication with Alternative Transports", draft-silverajan-core-coap-alternative-transports-09 (work in progress), December 2015.

[RFC6455] Fette, I. and A. Melnikov, "The WebSocket Protocol", RFC 6455, DOI 10.17487/RFC6455, December 2011, <http://www.rfc-editor.org/info/rfc6455>.

Authors' Addresses

Christian Groves

Email: cngroves.std@gmail.com

Weiwei Yang
Huawei

Email: tommy@huawei.com