DNS over CoAP (DoC) & DNS messages in CBOR
[draft-ietf-core-dns-over-coap, draft-lenders-dns-cbor]

Martine S. Lenders, Christian Amsüss, Cenk Gündoğan, Thomas C. Schmidt, Matthias Wählisch
IETF 115 CoRE Meeting, 2022-11-07
Outline

Draft 1: DNS over CoAP (DoC)

Next Steps for draft-ietf-core-dns-over-coap

Draft 2: A Concise Binary Object Representation (CBOR) of DNS Messages

Next Steps for draft-lenders-dns-over-coap
Motivation for DNS over CoAP

Attack Scenario

Countermeasure: Encrypt name resolution triggered by IoT devices against eavesdropping
Our Proposal: DNS over CoAP (DoC), draft-ietf-core-dns-over-coap

- Encrypted communication based on DTLS or OSCORE
- Block-wise message transfer to overcome Path MTU problem (DNS over DTLS)
- Share system resources with CoAP applications
  - Same socket and buffers can be used
  - Re-use of the CoAP retransmission mechanism
Add security considerations on ID=0 in unencrypted use
  • Replace layer violating statement for CON with statement of fact
  – Remove “DoC Server Considerations” (moved to draft-lenders-dns-cns)
Feedback from DNSOP (thanks!):

- Why isn’t DoH via CoAP gateway sufficient? The draft should explain.
- Explain why TTL rewriting proposed is notably different from DoH.
- Does DoC live at a URI path? If so, consider defining a default path, if this is a common practice in CoAP.
- Recommendation to add a section describing how to bootstrap DoC in a SVCB-DNS record. May require to allocate a new ALPN ID for CoAP/DTLS.
Next Steps

DoC

• Address feedback where possible
• Pick ID for application/dns-message Content-Format

Guidance on DNS in constrained networks

• Details see draft-lenders-dns-cns
• Do we see value in such guidance?
Draft 2: A Concise Binary Object Representation (CBOR) of DNS Messages
Packet size exceeds 802.15.4 PDU depending on queried name length

⇒ Fragmentation

<table>
<thead>
<tr>
<th>Frame Size [bytes]</th>
<th>Query</th>
<th>Response (AAAA)</th>
<th>Query</th>
<th>Response (AAAA)</th>
<th>Query</th>
<th>Response (AAAA)</th>
<th>Query</th>
<th>Response (AAAA)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
<td>64</td>
<td></td>
<td>128</td>
<td></td>
<td>192</td>
<td></td>
</tr>
<tr>
<td>256</td>
<td></td>
<td></td>
<td>256</td>
<td></td>
<td>256</td>
<td></td>
<td>256</td>
<td></td>
</tr>
<tr>
<td>320</td>
<td></td>
<td></td>
<td>320</td>
<td></td>
<td>320</td>
<td></td>
<td>320</td>
<td></td>
</tr>
</tbody>
</table>

IEEE 802.15.4+6LoWPAN RIOT-most (w/o L2 security)
CoAP with OSCORE, Content-Format and URI-Path “/dns”

IEEE 802.15.4+6LoWPAN RIOT-most (w/ L2 security)
CoAP with OSCORE, Content-Format and URI-Path “/dns”
Drawback of DNS in Constrained Networks

Packet size exceeds 802.15.4 PDU depending on queried name length

⇒ Fragmentation

Compression of DNS messages is needed!

application/dns+cbor
Objectives of draft-lenders-dns-cbor (application/dns+cbor)

Reduce packet sizes of DNS queries and replies:

1. Encoding of DNS messages in CBOR
2. Omit (redundant) DNS fields in DNS queries and responses
3. Address and name compression using packed CBOR (optional)
DNS Query

Using CDDL (RFC8610)

```
domain-name = tstr

type-spec = (  
    record-type: uint,
    ? record-class: uint,
  )

dns-question = (  
    ? id: uint,
    name: domain-name,
    ? type-spec,
  )

dns-query = [dns-question]
```

CBOR array:

- At minimum containing text string domain name (IDNA encoded)
- Optional ID and record type specification
  (ID defaults to 0, record-type to AAAA, record-class to IN)
DNS Resource Record

```plaintext
rr = (
    ? name: domain-name,
    ttl: uint,
    ? type-spec,
    rdata: bstr / domain-name,
)
dns-rr = [rr]
```

CBOR array:
- At minimum containing TTL and resource data
- Optional name and record type specification (both default to question values)
extra-sections = (  
  ? authority: [+ dns-rr],  
  additional: [+ dns-rr]  
)
sections = (()  
  ? id: uint,  
  answer: [+ dns-rr]  
)// (  
  ? id: uint,  
  question: dns-query,  
  answer: [+ dns-rr],  
  ? extra-sections,  
))
dns-response = [sections]

CBOR array of arrays:

- At minimum containing answer section (array of DNS resource records)
- **Generally assumes that transport can map query to response!**
  (original question and ID may be amended optionally)
Query IPv6 address for example.org
(13 bytes vs. 52 bytes wire-format: compression 400%)
["example.org"]

Corresponding response (24 bytes vs. 68 bytes wire-format: compression 283.3%):
[[3600, h'20010db8000000000000000000000001']]
Query ANY record for `example.org` (cf. DNS-SD)
(17 bytes vs. 52 bytes wire-format: compression 305,9%)

[['example.org', 255, 255]]

Corresponding response (200 bytes vs. 195 bytes wire-format: compression 97.5%):

[
  ['example.org', 12, 1],
  [[3600, '_coap._udp.local']],
  [[3600, 2, 'ns1.example.org'], [3600, 2, 'ns2.example.org']],
  [
    ['_coap._udp.local', 3600, 28, '20010DB8000000000000000000000001'],
    ['ns1.example.org', 3600, 28, '20010DB8000000000000000000000035'],
    ['ns2.example.org', 3600, 28, '20010DB8000000000000000000000035']
  ]
]
A More Complex Example

Query ANY record for \texttt{example.org} (cf. DNS-SD)
(17 bytes vs. 52 bytes wire-format: compression 305,9%)

["example.org", 255, 255]

Corresponding response (200 bytes vs. 195 bytes wire-format: compression 97.5%):

\[
\begin{array}{l}
\[
["example.org", 12, 1],
[[3600, "_coap._udp.local"],
[[3600, 2, "ns1.example.org"], [3600, 2, "ns2.example.org"],
["_coap._udp.local", 3600, 28, h'20010db800000000000000000000000001'],
["ns1.example.org", 3600, 28, h'20010db80000000000000000000000035'],
["ns2.example.org", 3600, 28, h'20010db80000000000000000000003535']
\]
\end{array}
\]

⇒ \textbf{Larger than wire-format!} Address and name compression needed
A More Complex Example

Query ANY record for example.org (cf. DNS-SD)
(17 bytes vs. 52 bytes wire-format: compression 305,9%)
["example.org", 255, 255]

Corresponding response (200 bytes vs. 195 bytes wire-format: compression 97.5%):

```
["example.org", 12, 1],
[[3600, "_coap._udp.local"],
[[3600, 2, "ns1.example.org"],
["_coap._udp.local", 3600, 28, h'20010db8000000000000000000000035'],
["ns1.example.org", 3600, 28, h'20010db80000000000000000000003535'],
["ns2.example.org", 3600, 28, h'20010db800000000000000000000353535']
]
```

⇒ Larger than wire-format! Address and name compression needed

Packed CBOR (draft-ietf-cbor-packed) comes with prefix- and suffix-based compression.
Our Proposal: Name and Address Compression Using Packed CBOR

- Optional packed CBOR support *for responses* negotiated using parameter `application/dns+cbor;packed=1` (own media type in draft -01)
- Make shared value and argument tables one list for that media type

```plaintext
compr-dns-response = any  # TBD; how to express packed CBOR in CDDL?
packed-dns-response = [[pack-table], compr-dns-response]
pack-table = any
```

Response becomes another CBOR array of two arrays:

1. Packing table (combined shared value and argument table)
2. Compressed `dns-response`
   (structure as defined before: CBOR array of sections)
Why Not SCHC?

Device

- CoAP
- App 2
- App 3
- UDP
- IPv6
- SCHC C/D & F/R

Apps

- CoAP
- App 2
- App 3
- UDP
- IPv6

LPWAN

There is an established connection between Device and NGW & SCHC C/D!
Why Not SCHC?

Device

- CoAP
- App 2
- App 3
- UDP
- IPv6
- SCHC C/D & F/R

Apps

- CoAP
- UDP
- IPv6
- App 2
- App 3
- IPv6

RGW

NGW

SCHC C/D

SCHC F/R

Internet

LPWAN
Why Not SCHC?

Device

CoAP
UDP
IPv6
SCHC C/D & F/R

Apps

CoAP
UDP
IPv6

Internet

Rule ID
Compression Residue
Payload

Rule n
Rule ...
Rule 2
Rule 1

<table>
<thead>
<tr>
<th>FID</th>
<th>FL</th>
<th>FP</th>
<th>DI</th>
<th>TV</th>
<th>MO</th>
<th>CDA</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP6.V</td>
<td>4</td>
<td>1</td>
<td>Bi</td>
<td>6</td>
<td>equal</td>
<td>not-sent</td>
</tr>
<tr>
<td>IP6.TC</td>
<td>8</td>
<td>1</td>
<td>Bi</td>
<td>0</td>
<td>ignore</td>
<td>not-sent</td>
</tr>
<tr>
<td>IP6.FL</td>
<td>20</td>
<td>1</td>
<td>Bi</td>
<td>0</td>
<td>ignore</td>
<td>not-sent</td>
</tr>
<tr>
<td>IP6.LEN</td>
<td>16</td>
<td>1</td>
<td>Bi</td>
<td>0</td>
<td>ignore</td>
<td>comp-length</td>
</tr>
<tr>
<td>UDP.DEV UP</td>
<td>16</td>
<td>1</td>
<td>UP</td>
<td>5683</td>
<td>equal</td>
<td>not-sent</td>
</tr>
<tr>
<td>UDP.DEV DW</td>
<td>16</td>
<td>1</td>
<td>DW</td>
<td>0x1f40</td>
<td>equal</td>
<td>LSB</td>
</tr>
<tr>
<td>CoAPV</td>
<td>2</td>
<td>1</td>
<td>Bi</td>
<td>1</td>
<td>equal</td>
<td>not-sent</td>
</tr>
</tbody>
</table>
Why Not SCHC?

There is an established connection between Device and NGW & SCHC C/D!
Why Not SCHC?

- Pre-defined rule sets needed between client and server
- Not generally provided in DNS client/server relationship
- Only few global compression contexts in DNS, e.g., TLDs

Device and NGW & SCHC C/D!
Example: ANY Record Response in application/dns+cbor;packed=1

Original CBOR response (200 bytes)

[["example.org", 12, 1],
 [[3600, "_coap._udp.local"]],
 [[3600, 2, "ns1.example.org"],
  [3600, 2, "ns2.example.org"],
 [["_coap._udp.local", 3600, 28,
  h'20010db8000000000000000000000001' ],
 [[ns1.example.org", 3600, 28,
  h'20010db80000000000000000000000035' ],
 [["ns2.example.org", 3600, 28,
  h'20010db80000000000000000000003535' ]]]

Packed CBOR response (119 bytes)

[[h'20010db8000000000000000000000000',
  "_coap._udp.local", "example.org", 3600, 28, 2
 ],
 [[simple(2), 12, 1],
  [[simple(3), simple(1)]],
  [[simple(2), simple(5), 219("ns1."),
    simple(2), simple(5), 219("ns2.")]]],
  [[simple(1), simple(3), simple(4), 6(h'0001')],
  [219("ns1."), simple(3), simple(4), 6(h'0035')],
  [219("ns2."), simple(3), simple(4), 6(h'3535')]]]

(cmp. 195 bytes wire-format: compression 163.9%)
Example: ANY Record Response in application/dns+cbor;packed=1

Original CBOR response (200 bytes)

[["example.org", 12, 1],
 [[3600, "_coap._udp.local"]],
 [[3600, 2, "ns1.example.org"],
  [3600, 2, "ns2.example.org"]],
 [["_coap._udp.local", 3600, 28,
   h'20010db800000000000000000000001'],
 ["ns1.example.org", 3600, 28,
  h'20010db8000000000000000000000035'],
 ["ns2.example.org", 3600, 28,
  h'20010db8000000000000000000003535']]

Packed CBOR response (119 bytes)

[[h'20010db800000000000000000000000000000000000',
  "_coap._udp.local", "example.org",
  3600, 28, 2
 ],
 [[simple(2), 12, 1],
  [[simple(3), simple(1)]],
  [[simple(2), simple(5), 219("ns1."),
    simple(2), simple(5), 219("ns2.")]]],
  [[simple(1), simple(3), simple(4), 6(h'0001')],
   [219("ns1."), simple(3), simple(4), 6(h'0035')],
   [219("ns2."), simple(3), simple(4), 6(h'3535')]]]

(cmp. 195 bytes wire-format: compression 163.9%)

Implied DNS-specific table entries for global compression contexts (e.g. TLDs) enable potential for more elision
Next Steps

Define more details on using packed CBOR:

- How to construct packing table?
- Global compression contexts, DNS-specific implied table entries
- ⟨Your thoughts.⟩