DNS over CoAP (DoC)

Securing Name Resolution in the Internet of Things

Martine S. Lenders (m.lenders@fu-berlin.de) [CC-BY 4.0]

Hacking in Parallel, 2022-12-29 (Day 3)
Which Internet of Things are we talking about?

<table>
<thead>
<tr>
<th>Device class</th>
<th>Data size [KiB]</th>
<th>Code size [KiB]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>≪10</td>
<td>≪100</td>
</tr>
<tr>
<td>1</td>
<td>≈10</td>
<td>≈100</td>
</tr>
<tr>
<td>2</td>
<td>≈50</td>
<td>≈250</td>
</tr>
</tbody>
</table>

Constrained networks (RFC 7228):

- Low throughput
- High packet loss
- Asymmetric link characteristics
- High penalties on large packets (link layer fragmentation)
- ...
Which Internet of Things are we talking about?

Constrained nodes (RFC 7228):

- Device class
- Data size [KiB]
- Code size [KiB]

- 0 ≪ 10 ≪ 100
- 1 ≈ 10 ≈ 100
- 2 ≈ 50 ≈ 250

Constrained networks (RFC 7228):

- Low throughput
- High packet loss
- Asymmetric link characteristics
- High penalties on large packets (link layer fragmentation)
- ...
Which Internet of Things are we talking about?

**Constrained nodes (RFC 7228):**

<table>
<thead>
<tr>
<th>Device class</th>
<th>Data size [KiB]</th>
<th>Code size [KiB]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>≪ 10</td>
<td>≪ 100</td>
</tr>
<tr>
<td>1</td>
<td>≈ 10</td>
<td>≈ 100</td>
</tr>
<tr>
<td>2</td>
<td>≈ 50</td>
<td>≈ 250</td>
</tr>
</tbody>
</table>

**Constrained networks (RFC 7228):**

- Low throughput
- High packet loss
- Asymmetric link characteristics
- High penalties on large packets (link layer fragmentation)
- ...

- BLE
Which Internet of Things are we talking about?

Constrained nodes (RFC 7228):

<table>
<thead>
<tr>
<th>Device class</th>
<th>Data size [KiB]</th>
<th>Code size [KiB]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>≪ 10</td>
<td>≪ 100</td>
</tr>
<tr>
<td>1</td>
<td>≈ 10</td>
<td>≈ 100</td>
</tr>
<tr>
<td>2</td>
<td>≈ 50</td>
<td>≈ 250</td>
</tr>
</tbody>
</table>

Constrained networks (RFC 7228):

- Low throughput
- High packet loss
- Asymmetric link characteristics
- High penalties on large packets (link layer fragmentation)
- ...
Which Internet of Things are we talking about?

Constrained nodes (RFC 7228):

- Device class
  - Data size [KiB]: \(0 \ll 10 \ll 100\)
  - Code size [KiB]: \(1 \approx 10 \approx 100\)
  - 2 \(\approx 50 \approx 250\)

Constrained networks (RFC 7228):

- Low throughput
- High packet loss
- Asymmetric link characteristics
- High penalties on large packets (link layer fragmentation)
- ...

BLE
Which Internet of Things are we talking about?

Constrained nodes (RFC 7228):

<table>
<thead>
<tr>
<th>Device class</th>
<th>Data size [KiB]</th>
<th>Code size [KiB]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>≪ 10</td>
<td>≪ 100</td>
</tr>
<tr>
<td>1</td>
<td>≈ 10</td>
<td>≈ 100</td>
</tr>
<tr>
<td>2</td>
<td>≈ 50</td>
<td>≈ 250</td>
</tr>
</tbody>
</table>

Constrained networks (RFC 7228):

- Low throughput
- High packet loss
- Asymmetric link characteristics
- High penalties on large packets (link layer fragmentation)
- ...

BLE
Which Internet of Things are we talking about?

**Constrained nodes** (RFC 7228):

<table>
<thead>
<tr>
<th>Device class</th>
<th>Data size [KiB]</th>
<th>Code size [KiB]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>\ll 10</td>
<td>\ll 100</td>
</tr>
<tr>
<td>1</td>
<td>\approx 10</td>
<td>\approx 100</td>
</tr>
<tr>
<td>2</td>
<td>\approx 50</td>
<td>\approx 250</td>
</tr>
</tbody>
</table>
Which Internet of Things are we talking about?

**Constrained nodes (RFC 7228):**

<table>
<thead>
<tr>
<th>Device class</th>
<th>Data size [KiB]</th>
<th>Code size [KiB]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>(\ll 10)</td>
<td>(\ll 100)</td>
</tr>
<tr>
<td>1</td>
<td>(\approx 10)</td>
<td>(\approx 100)</td>
</tr>
<tr>
<td>2</td>
<td>(\approx 50)</td>
<td>(\approx 250)</td>
</tr>
</tbody>
</table>
Which Internet of Things are we talking about?

**Constrained nodes** (RFC 7228):

<table>
<thead>
<tr>
<th>Device class</th>
<th>Data size [KiB]</th>
<th>Code size [KiB]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>≪ 10</td>
<td>≪ 100</td>
</tr>
<tr>
<td>1</td>
<td>≈ 10</td>
<td>≈ 100</td>
</tr>
<tr>
<td>2</td>
<td>≈ 50</td>
<td>≈ 250</td>
</tr>
</tbody>
</table>

Constrained networks (RFC 7228):
- Low throughput
- High packet loss
- Asymmetric link characteristics
- High penalties on large packets (link layer fragmentation)
- ...
Which Internet of Things are we talking about?

**Constrained nodes** (RFC 7228):

<table>
<thead>
<tr>
<th>Device class</th>
<th>Data size [KiB]</th>
<th>Code size [KiB]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>\ll 10</td>
<td>\ll 100</td>
</tr>
<tr>
<td>1</td>
<td>\approx 10</td>
<td>\approx 100</td>
</tr>
<tr>
<td>2</td>
<td>\approx 50</td>
<td>\approx 250</td>
</tr>
</tbody>
</table>

**Constrained networks** (RFC 7228):

- Low throughput
- High packet loss
- Asymmetric link characteristics
- High penalties on large packets (link layer fragmentation)
- ...

BLE
Which Internet of Things are we talking about?

**Constrained nodes** (RFC 7228):

<table>
<thead>
<tr>
<th>Device class</th>
<th>Data size [KiB]</th>
<th>Code size [KiB]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$\ll 10$</td>
<td>$\ll 100$</td>
</tr>
<tr>
<td>1</td>
<td>$\approx 10$</td>
<td>$\approx 100$</td>
</tr>
<tr>
<td>2</td>
<td>$\approx 50$</td>
<td>$\approx 250$</td>
</tr>
</tbody>
</table>

**Constrained networks** (RFC 7228):

- Low throughput
- High packet loss
- Asymmetric link characteristics
- High penalties on large packets (link layer fragmentation)
- ...

...
Why encrypted name resolution?
Why encrypted name resolution?

Who is example.org
Why encrypted name resolution?

Who is example.org

example.org is 2001:db8::1
Why encrypted name resolution?

Who is example.org

example.org is 2001:db8::1
Why encrypted name resolution?

example.org is 2001:db8::1

example.org is 2001:db8::1
Possible solutions

- DNS over HTTPS (RFC 8484)
- DNS over TLS (RFC 7858)
- DNS over QUIC (RFC 9250)
- DNS over DTLS (RFC 8094)

TCP conflicts with resource constraints
TLS over UDP conflicts with resource constraints
Path MTU problem vs. link layer fragmentation

Our proposal: DNS over CoAP

- Encrypted communication based on DTLS or OSCORE
- Block-wise message transfer to overcome Path MTU problem
- Shares system resources with CoAP applications
- Same socket and buffers can be used
- Re-use of the CoAP retransmission mechanism
Possible solutions

- DNS over HTTPS (RFC 8484)
- DNS over TLS (RFC 7858)
- DNS over QUIC (RFC 9250)
- DNS over DTLS (RFC 8094)

TCP conflicts with resource constraints
TLS over UDP conflicts with resource constraints
Path MTU problem vs. link layer fragmentation

Our proposal: DNS over CoAP
- Encrypted communication based on DTLS or OSCORE
- Block-wise message transfer to overcome Path MTU problem
- Shares system resources
- Same socket and buffers can be used
- Re-use of the CoAP retransmission mechanism
Possible solutions

DNS over HTTPS (RFC 8484)

DNS over TLS (RFC 7858)

DNS over QUIC (RFC 9250)

DNS over DTLS (RFC 8094)

TCP conflicts with resource constraints

Path MTU problem vs. link layer fragmentation

Our proposal: DNS over CoAP

• Encrypted communication based on DTLS or OSCORE
• Block-wise message transfer to overcome Path MTU problem
• Shares system resources
• Same socket and buffers can be used
• Re-use of the CoAP retransmission mechanism
Possible solutions

- DNS over HTTPS (RFC 8484)
- DNS over TLS (RFC 7858)
- DNS over QUIC (RFC 9250)
- DNS over DTLS (RFC 8094)

TCP conflicts with resource constraints
TLS over UDP conflicts with resource constraints

Our proposal: DNS over CoAP
- Encrypted communication based on DTLS or OSCORE
- Block-wise message transfer to overcome Path MTU problem
- Shares system resources
- Same socket and buffers can be used
- Re-use of the CoAP retransmission mechanism
Possible solutions

- DNS over HTTPS (RFC 8484)
- DNS over TLS (RFC 7858)
- DNS over QUIC (RFC 9250)
- DNS over DTLS (RFC 8094)

TCP conflicts with resource constraints
TLS over UDP conflicts with resource constraints
Path MTU problem vs. link layer fragmentation

Our proposal: DNS over CoAP
- Encrypted communication based on DTLS or OSCORE
- Block-wise message transfer to overcome Path MTU problem
- Shares system resources with CoAP applications
- Same socket and buffers can be used
- Re-use of the CoAP retransmission mechanism
Possible solutions

Our proposal: DNS over CoAP

- **Encrypted communication** based on DTLS or OSCORE
- **Block-wise message transfer** to overcome Path MTU problem
- **Share system resources** with CoAP applications
  - Same socket and buffers can be used
  - Re-use of the CoAP retransmission mechanism
What is CoAP? The Constrained Application Protocol = “REST over UDP”
What is CoAP? Provides optional reliability
What is CoAP? Provides optional reliability

NON (POST)

NON (2.01 Created)
What is CoAP? Provides optional reliability
What is CoAP? Provides optional reliability
What is CoAP? Provides optional reliability
What is CoAP? Provides optional reliability

CON (POST)

ACK (2.01 Created)
What is CoAP? Compatible with HTTP via proxy
What is CoAP? Compatible with HTTP via proxy

CoAP GET /dns

HTTP GET /dns

404 Not Found
What is CoAP? Compatible with HTTP via proxy

CoAP GET /dns

HTTP GET /dns

404 Not Found
What is CoAP? It comes with encryption!

**DTLS** Datagram **Transport Layer Security** (≈ TLS over UDP)
What is CoAP? It comes with encryption!

**DTLS** Datagram Transport Layer Security (≈ TLS over UDP)

Unencrypted Channel

OSCORE

Object Security for Constrained RESTful Environment

Encrypted Transport

Gateway
What is CoAP? It comes with encryption!

**DTLS** Datagram Transport Layer Security (≈ TLS over UDP)
What is CoAP? It comes with encryption!

**DTLS** Datagram Transport Layer Security (≈ TLS over UDP)

- **Encrypted Transport**
- **Unencrypted Channel**
What is CoAP? It comes with encryption!

**DTLS** Datagram Transport Layer Security ($\approx$ TLS over UDP)

**OSCORE** Object Security for Constrained RESTful Environment
What is CoAP? It comes with encryption!

**DTLS** Datagram Transport Layer Security (≈ TLS over UDP)

**OSCORE** Object Security for Constrained RESTful Environment
What is CoAP? It comes with encryption!

**DTLS** Datagram Transport Layer Security (≈ TLS over UDP)

**OSCORE** Object Security for Constrained RESTful Environment
DNS over CoAP (DoC)

- Just map the DoH methods **GET** and **POST**?
DNS over CoAP (DoC)

- Just map the DoH methods **GET** and **POST**?

<table>
<thead>
<tr>
<th></th>
<th>HTTP</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>GET</td>
<td>POST</td>
<td></td>
</tr>
<tr>
<td>Cacheable</td>
<td>✔</td>
<td>✗</td>
<td></td>
</tr>
<tr>
<td>Application data carried in body</td>
<td>✗</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>Block-wise transferable query</td>
<td>✗</td>
<td>✔</td>
<td></td>
</tr>
</tbody>
</table>
### DNS over CoAP (DoC)

- Just map the DoH methods **GET** and **POST**?
- **FETCH** method in CoAP: best of both worlds (RFC 8132)

<table>
<thead>
<tr>
<th>HTTP</th>
<th>GET</th>
<th>POST</th>
<th>FETCH</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cacheable</td>
<td>✔</td>
<td>✗</td>
<td>✔</td>
</tr>
<tr>
<td>Application data carried in body</td>
<td>✗</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Block-wise transferable query</td>
<td>✗</td>
<td>✔</td>
<td>✔</td>
</tr>
</tbody>
</table>
DNS over CoAP (DoC)

- Just map the DoH methods **GET** and **POST**?
- **FETCH** method in CoAP: best of both worlds (RFC 8132)

<table>
<thead>
<tr>
<th>CoAP</th>
<th>HTTP</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td>GET</td>
</tr>
<tr>
<td>POST</td>
<td>POST</td>
</tr>
<tr>
<td>FETCH</td>
<td>POST</td>
</tr>
</tbody>
</table>

- Cacheable
- Application data carried in body
- Block-wise transferable query

---

**CoAP request** `coap://[2001:db8::1]/`

- **DoC Client**
- **DoC Server**
- **DNS Server**
What questions do we want to evaluate?

- How do name resolutions perform in a constrained network?
What questions do we want to evaluate?

- How do name resolutions perform in a constrained network?
- Which transport is best suited for encrypted name resolution?
What questions do we want to evaluate?

• How do name resolutions perform in a constrained network?
• Which transport is best suited for encrypted name resolution?
• What side effects may come from using encrypted name resolution?
Evaluation: Setup

**Name properties:** Based on empirical query data from IoT devices

**Testbed experiments:**

- Clients query 50 A or AAAA records for names of length 24 chars via DNS over UDP / DTLSv1.2 / CoAP (unencrypted) / CoAPSv1.2 / OSCORE
- Poisson distribution: $\lambda = 5$ queries / sec
- 10 runs on physical IoT-nodes (incl. BR): Cortex-M3 with IEEE 802.15.4 radio
Experiment: Resolution time

![Resolution time CDF for DNS transports and CoAP methods]

- **DNS Transports**
  - UDP
  - DTLSv1.2
  - CoAP
  - CoAPSv1.2
  - OSCORE

- **CoAP Methods**
  - FETCH
  - POST
  - GET

---

**Can we get the packets smaller to prevent fragmentation?**
Experiment: Resolution time

Clear performance groupings visible
Experiment: Resolution time

A record

AAAA record

Group 1
Group 2
## Experiment: Resolution time

### Group 3

<table>
<thead>
<tr>
<th>DNS Transports</th>
<th>CoAP Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>UDP</td>
<td>FETCH</td>
</tr>
<tr>
<td>CoAPSv1.2</td>
<td>POST</td>
</tr>
<tr>
<td>DTLSv1.2</td>
<td>GET</td>
</tr>
<tr>
<td>OSCORE</td>
<td></td>
</tr>
</tbody>
</table>

### Resolution time [s]

<table>
<thead>
<tr>
<th>0.0</th>
<th>0.5</th>
<th>1.0</th>
</tr>
</thead>
</table>

### Frame Size [bytes]

<table>
<thead>
<tr>
<th>UDP</th>
<th>0</th>
<th>32</th>
<th>64</th>
<th>96</th>
<th>128</th>
<th>160</th>
<th>192</th>
</tr>
</thead>
<tbody>
<tr>
<td>DTLSv1.2</td>
<td>0</td>
<td>32</td>
<td>64</td>
<td>96</td>
<td>128</td>
<td>160</td>
<td>192</td>
</tr>
<tr>
<td>CoAP</td>
<td>0</td>
<td>32</td>
<td>64</td>
<td>96</td>
<td>128</td>
<td>160</td>
<td>192</td>
</tr>
<tr>
<td>CoAPSv1.2</td>
<td>0</td>
<td>32</td>
<td>64</td>
<td>96</td>
<td>128</td>
<td>160</td>
<td>192</td>
</tr>
<tr>
<td>OSCORE</td>
<td>0</td>
<td>32</td>
<td>64</td>
<td>96</td>
<td>128</td>
<td>160</td>
<td>192</td>
</tr>
</tbody>
</table>

### Packet size

<table>
<thead>
<tr>
<th>L2 max. frame size</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
</tbody>
</table>

Can we get the packets smaller to prevent fragmentation?
Where do performance groups come from?

Experiment: Resolution time

A record

AAAA record

DNS Transports
- UDP
- DTLSv1.2
- CoAP
- CoAPSv1.2
- OSCORE

CoAP Methods
- FETCH
- POST
- GET

Frame Size [bytes]
- UDP
- DTLSv1.2
- CoAP
- CoAPSv1.2
- OSCORE

Packet size
- L2 max. frame size

Query
- [F/P]
- [G]
- Resp. (A)
- Resp. (AAAA)
Experiment: Resolution time & packet sizes

A record

AAAA record

DNS Transports
UDP
CoAP
DTLSv1.2
OSCORE
CoAPSv1.2

CoAP Methods
FETCH
GET
POST

Frame Size [bytes]
UDP
0
32
64
96
128
160
192

Packet size
L2 max. frame size

Query [F/P]
Query [G]
Resp. (A)
Resp. (AAAA)

Can we get the packets smaller to prevent fragmentation?
**Experiment: Resolution time & packet sizes**

**A record**

**AAAA record**

**Group 1**

No message fragmentation
**Experiment: Resolution time & packet sizes**

**A record**
- CDF of Resolution time [s]
- DNS Transports: UDP, DTLSv1.2, CoAP, CoAPSv1.2, OSCORE

**AAAA record**
- CDF of Resolution time [s]
- DNS Transports: UDP, DTLSv1.2, CoAP, CoAPSv1.2, OSCORE

**Group 2**
- Query unfragmented
- Response fragmented

**Frame Size [bytes]**
- UDP
- DTLSv1.2
- CoAP
- CoAPSv1.2

**Packet size L2 max. frame size**

Can we get the packets smaller to prevent fragmentation?
Experiment: Resolution time & packet sizes

A record

AAAA record

Group 3
Both messages fragmented
Experiment: Resolution time & packet sizes

 ⇒ Fragmentation has larger impact on performance compared to transport or CoAP method
Experiment: Resolution time & packet sizes

Can we get the packets smaller to prevent fragmentation?
A solution: CBOR-based message format

The **Concise Binary Object Representation (CBOR, RFC 8948)**:

- Binary and lightweight alternative to JSON

  \[ \{ "d": 1, "v": [false, true] \} \rightarrow A2 61 64 01 61 76 82 F4 F5 \]
  (9 bytes instead of 24 characters)

Content format option in CoAP header can indicate type of content:

**Use CBOR to get smaller DNS messages!**
Compressing DNS with CBOR: The query

**Wire format**

c7 0c 01 20 00 01 00 00 00 00 07 65 78 61 6d 70 6c 65 03 6f 72 67 00 00 1c 00 01

(29 bytes)
Compressing DNS with CBOR: The query

Wire format

\[
\text{c 7 0 c 0 1 2 0 0 0 0 0 0 0 0 0 7 6 5 7 8 6 1 6 d 7 0 6 c 6 5 0 3 6 f 7 2 6 7 0 0 0 0 1 c 0 0 0 1 (29 bytes)}
\]
Compressing DNS with CBOR: The query

Wire format

```
c 1 0 0 0 0 0 0
0 0 0 0 0 0 0 7 6 5 7 8 6 1
6 d 7 0 6 c 6 5 0 3 6 f 7 2 6 7
0 0 0 0 1 c 0 0 0 1
```

(29 bytes)
Compressing DNS with CBOR: The query

Wire format

<table>
<thead>
<tr>
<th>ID</th>
<th>Flags</th>
<th>#QD</th>
<th>#AN</th>
<th>#NS</th>
<th>#AR</th>
</tr>
</thead>
<tbody>
<tr>
<td>6d</td>
<td>70</td>
<td>65</td>
<td>78</td>
<td>61</td>
<td></td>
</tr>
<tr>
<td>6c</td>
<td>65</td>
<td>03</td>
<td>6f</td>
<td>72</td>
<td>67</td>
</tr>
<tr>
<td>00</td>
<td>00</td>
<td>1c</td>
<td>00</td>
<td>01</td>
<td></td>
</tr>
</tbody>
</table>

(29 bytes)
Compressing DNS with CBOR: The query

Wire format

```
00 00 1c 00 01
```

(29 bytes)
Compressing DNS with CBOR: The query

**Wire format**

```
<table>
<thead>
<tr>
<th>ID</th>
<th>Flags</th>
<th>#QD</th>
<th>#AN</th>
<th>#NS</th>
<th>#AR</th>
<th>Name &amp; (example.org)</th>
<th>Type</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td></td>
<td>01</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>07 65 78 61</td>
<td>AAAA</td>
<td>IN</td>
</tr>
</tbody>
</table>
```

(29 bytes)
Compressing DNS with CBOR: The query

Wire format

```plaintext
05 0c 01 20 00 01 00 00 00 00 07 65 78 61 6d 70 6c 65 03 6f 72 67 00 00 1c 00 01
```

(29 bytes)
Compressing DNS with CBOR: The query

Wire format

```
0c 01 20 00 01 00 00 00 07 65 78 61 6d 70 6c 65 03 6f 72 67 00 01
```
(29 bytes)
Compressing DNS with CBOR: The query

Wire format

```
03 01 20 00 01 00 00 00 07 65 78 61 6d 70 6c 65 03 6f 72 67 00 00 1c 00 01
```

29 bytes

ID Flags #QD #AN #NS #AR

Name (\7example\3org\0)

AAAA IN

14
Compressing DNS with CBOR: The query

Wire format

```
C7 0C 01 20 00 01 00 00 00 00 07 65 78 61 6d 70 6c 65 03 6f 72 67 00 00 1c 00 01
```

(29 bytes)
Compressing DNS with CBOR: The query

Wire format

C 7 c 0 c 0 1 2 0 0 0 1 0 0 0 0 7 6 5 7 8 6 1 0 3 6 0 7 6 5 7 8 6 1 (29 bytes)

Name (\7example\3org\0)

AAAA

IN

14
Compressing DNS with CBOR: The query

**Wire format**

```
c7 0c 01 20 00 01 00 00 00 00 07 65 78 61 6d 70 6c 65 03 6f 72 67 00 00 1c 00 01
```

(29 bytes)
Compressing DNS with CBOR: The query

Wire format

```
c7 0c 01 20 00 01 00 00 00 00 07 65 78 61 6d 70 6c 65 03 6f 72 67 00 00 1c 00 01
```
(29 bytes)

CBOR diagnostic format

```
[ "example.org"
]
```
(29 bytes)

CBOR binary

```
81 6b 65 78 61 6d 70 6c 65 2e 6f 72 67
65 2e 6f 72 67
```
(13 bytes)
Compressing DNS with CBOR: The query

Wire format
```
C 0C 01 20 00 01 00 00 00 07 65 78 61 6d 70 6c 65 03 6f 72 67 00 00 1c 00 01
```
(29 bytes)

CBOR diagnostic format
```
[ "example.org"
]
```
(13 bytes)

CBOR binary
```
81 6b 65 78 61 6d 70 6c 65 2e 6f 72 67
65 2e 65 78 61 6d 70 6c 65 03 6f 72 67
```
(13 bytes)
Compressing DNS with CBOR: The query

**Wire format**
```
c7 0c 01 20 00 01 00 00 00 00 00 00 07 65 78 61 6d 70 6c 65 03 6f 72 67 00 00 1c 00 01 (29 bytes)
```

**CBOR diagnostic format**
```
[
  "example.org"
]
```

**CBOR binary**
```
81 6b 65 78 61 6d 70 6c 65 2e 6f 72 67
```

(13 bytes)
## Compressing DNS with CBOR: The query

<table>
<thead>
<tr>
<th>Wire format</th>
<th>CBOR diagnostic format</th>
<th>CBOR binary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>c7 0c 01 20 00 01 00 00 00 00 07 65 78 61 6d 70 6c 65 03 6f 72 67 00 00 1c 00 01</code></td>
<td><code>[ &quot;example.org&quot; ]</code></td>
<td><code>81 6b 65 78 61 6d 70 6c 65 03 6f 72 67 65 2e 6f 72 67 65 2e 6f 72 67</code></td>
</tr>
<tr>
<td>(29 bytes)</td>
<td>(13 bytes)</td>
<td>(13 bytes)</td>
</tr>
</tbody>
</table>
Compressing DNS with CBOR: The response

Wire format

c7 0c 81 80 00 01 00 01
00 00 00 00 07 65 78 61
6d 70 6c 65 03 6f 72 67
00 00 1c 00 01 c0 0c 00
1c 00 01 00 00 39 c0 00
10 20 01 0d b8 00 00 00
00 00 00 00 00 00 00 00
01

(57 bytes)
Compressing DNS with CBOR: The response

Wire format

```
c7 0c 81 80 00 01 00 01 00 00 00 07 65 78 61 6d 70 6c 65 03 6f 72 67 00 00 1c 00 01 c0 0c 00 1c 00 01 00 00 39 c0 00 10 20 01 0d b8 00 00 00 00 00 00 00 00 01
```

(57 bytes)
Compressing DNS with CBOR: The response

Wire format

Assumes that query can be mapped to response by transport!
Compressing DNS with CBOR: The response

Wire format

```
wireformat
```

```
\x00 \x07 \x65 \x78 \x61 \x6d \x70 \x6c \x65 \x03 \x6f \x72 \x67
```

Assumes that query can be mapped to response by transport!
Compressing DNS with CBOR: The response

Wire format

<table>
<thead>
<tr>
<th>ID</th>
<th>Flags</th>
<th>#QD</th>
<th>#AN</th>
<th>#NS</th>
<th>#AR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>00</td>
<td>01</td>
<td>00</td>
<td>00</td>
</tr>
<tr>
<td>10</td>
<td>20</td>
<td>01</td>
<td>0d</td>
<td>b8</td>
<td>00</td>
</tr>
<tr>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
</tr>
<tr>
<td>01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(57 bytes)

Assumes that query can be mapped to response by transport!
Compressing DNS with CBOR: The response

Wire format

```
1c 00 01 00 00 1c 00 01 00 00 39 c0 00 10 20 01 0d b8 00 00 00 00 00 00 00 00 00 00 01
```

(57 bytes)

Assumes that query can be mapped to response by transport!
Compressing DNS with CBOR: The response

Wire format

```
C 7 0c 81 80 00 01 00 01 00 00 00 00 07 65 78 61 6d 70 6c 65 03 6f 72 67 00 00 1c 00 01 c0 0c 00 1c 00 01 00 00 39 c0 00 10 20 01 0d b8 00 00 00 00 00 00 00 00 01
```

(57 bytes)

Assumes that query can be mapped to response by transport!
Compressing DNS with CBOR: The response

Wire format

```
00 1c 00 01
00 01 00 00 39 c0 00
10 20 01 0d b8 00 00 00
00 00 00 00 00 00 00 00
01
```

(57 bytes)

Assumes that query can be mapped to response by transport!
Compressing DNS with CBOR: The response

Wire format

```
07 65 78 61 6d 70 6c 65 03 6f 72 67 00 00 1c 00 01 c0 0c 00 1c 00 01 00 00 39 c0 00 10 20 01 0d b8 00 00 00 00 00 00 00 00 01
```

(57 bytes)

Assumes that query can be mapped to response by transport!
Compressing DNS with CBOR: The response

Wire format

Assumes that query can be mapped to response by transport!
Compressing DNS with CBOR: The response

Wire format

(57 bytes)

Assumes that query can be mapped to response by transport!
Compressing DNS with CBOR: The response

Wire format

(57 bytes)

Assumes that query can be mapped to response by transport!
Compressing DNS with CBOR: The response

Wire format

CBOR diagnostic format

CBOR binary

(57 bytes)

(21 bytes)

Assumes that query can be mapped to response by transport!
Compressing DNS with CBOR: The response

Wire format

CBOR diagnostic format

[ 14784, h'20010db80000000000000000000000001' ]

CBOR binary

82 19 147840 50 20 01 0d b8 00 2001:db8:0::1 00 00
00 00 00 00 01

(57 bytes)

(21 bytes)

Assumes that query can be mapped to response by transport!
Compressing DNS with CBOR: The response

Wire format

```
c7 0c 81 80 00 01 00 00 00 00 07 65 78 61 6d 70 6c 65 03 6f 72 67 00 00 1c 00 01 c0 0c 00 1c 00 01 00 00 39 c0 00 10 20 01 0d b8 00 00 00 00 00 00 00 00 01
```

(57 bytes)

CBOR diagnostic format

```
[14784, h'2001:db8::1']
```

(21 bytes)

CBOR binary

```
82 19 147840 50 20 01 0d b8 00 2001:db8::1 00 00 00 00 01
```

Assumes that query can be mapped to response by transport!
Compressing DNS with CBOR: The response

<table>
<thead>
<tr>
<th>Wire format</th>
<th>CBOR diagnostic format</th>
<th>CBOR binary</th>
</tr>
</thead>
<tbody>
<tr>
<td>c7 0c 81 80 00 01 00 01 00 00 00 07 65 78 61 6d 70 6c 65 03 6f 72 00 00 1c 00 01 c0 0c 00 1c 00 01 00 00 39 c0 00 10 20 01 0d b8 00 00 00 00 00 00 00 00 00 00 00 00 01</td>
<td>[14784, h'20010db800000000000000000000000001']</td>
<td>82 19 39 c0 50 20 01 0d b8 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 01</td>
</tr>
</tbody>
</table>

(57 bytes)                                                                                              (21 bytes)

Assumes that query can be mapped to response by transport!
Conclusion

• DNS over CoAP (DoC) provides encrypted DNS for the constrained IoT
• With FETCH and OSCORE en par with established DNS solutions
• Fragmentation has a high impact on performance
  ⇒ Compressed message format needed
Thank you!

https://datatracker.ietf.org/doc/draft-ietf-core-dns-over-coap
https://datatracker.ietf.org/doc/draft-lenders-dns-cbor
https://arxiv.org/abs/2207.07486

📧 m.lenders@fu-berlin.de  🗞️ 0xD355B9E03C098C7
🔗 https://blog.martine-lenders.eu
 münchen@ohai.social  🫶️ @miri64  🐦 @miri_64