

A Concise Binary Object Representation (CBOR) of DNS Messages

draft-lenders-dns-cbor

(<https://datatracker.ietf.org/doc/draft-lenders-dns-cbor/>)

Martine S. Lenders (martine-lenders@tu-dresden.de),
Carsten Bormann, Thomas C. Schmidt, Matthias Wählisch
IETF 121, CBOR WG Session, 2024-11-05

Outline

Motivation

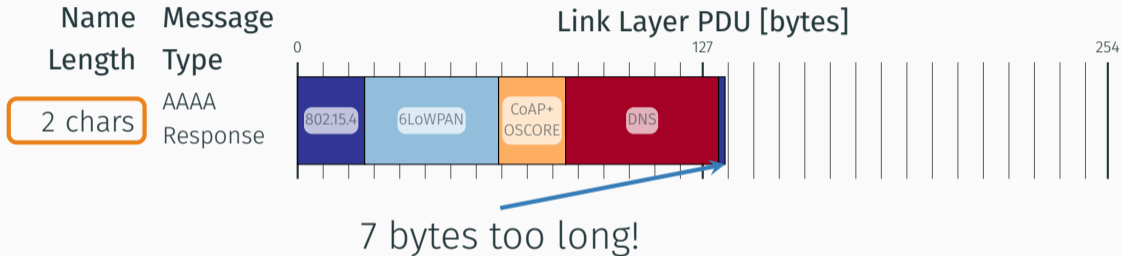
Objectives

Updates

Name Compression

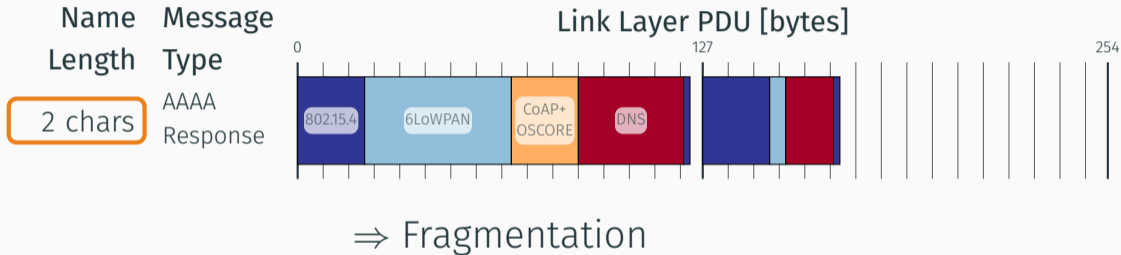
Motivation: DNS over CoAP in Constrained Networks

Constrained Networks, e.g., IEEE 802.15.4 with PDU of 127 bytes



Motivation: DNS over CoAP in Constrained Networks

Constrained Networks, e.g., IEEE 802.15.4 with PDU of 127 bytes



Motivation: DNS over CoAP in Constrained Networks

Constrained Networks, e.g., IEEE 802.15.4 with PDU of 127 bytes

Concise DNS messages are needed

`application/dns+cbor`

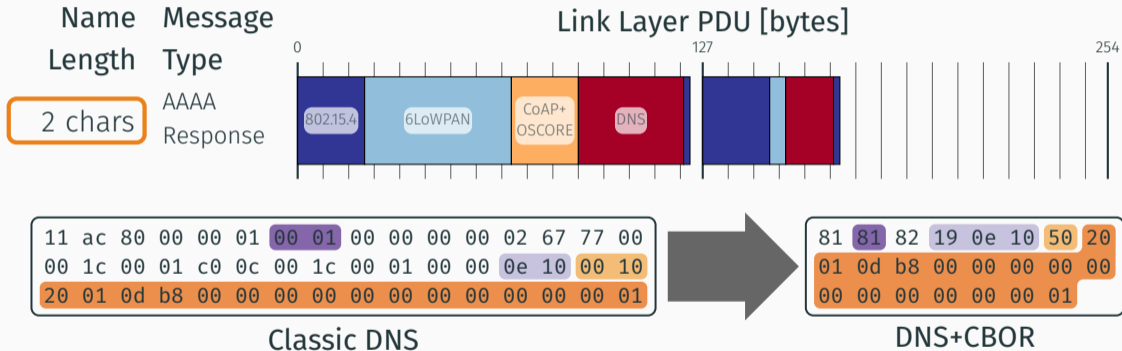
Media Type and Content-Format

(*i.e.*, usable with both DoC and DoH)

254

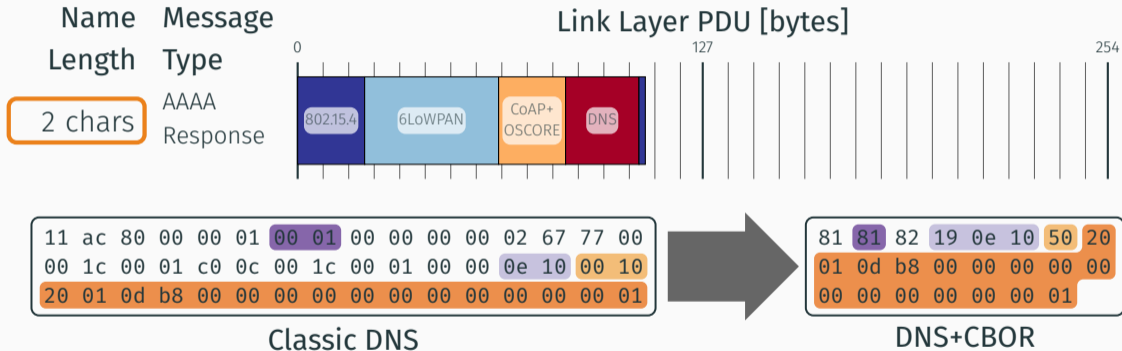
Motivation: DNS over CoAP in Constrained Networks

Constrained Networks, e.g., IEEE 802.15.4 with PDU of 127 bytes



Motivation: DNS over CoAP in Constrained Networks

Constrained Networks, e.g., IEEE 802.15.4 with PDU of 127 bytes



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

Provide concise packet format and compressed names and addresses in DNS queries and replies:

1. Using existing implementation: CBOR
2. Encoding of DNS messages in CBOR (conciseness)
3. Omit (redundant) DNS fields in DNS queries and responses (conciseness)
4. Easy to implement name compression, on-the-fly construction (compression)
5. Address and value compression using CBOR-packed (compression, optional)

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

Provide concise packet format and compressed names and addresses in DNS queries and replies:

1. Using existing implementation: CBOR
2. Encoding of DNS messages in CBOR (conciseness)
3. Omit (redundant) DNS fields in DNS queries and responses (conciseness)
4. Easy to implement name compression, on-the-fly construction (compression)
5. Address and value compression using CBOR-packed (compression, optional)

Updates since IETF 118

- + RFC1035-like name compression
- + Dedicated types for more structured RDATA
 - SOA, MX, SRV, SVCB & HTTPS
- + Answer section for queries (for Known Answers in mDNS)
- + Amend capability to carry more than one question
- + Switching boolean to queries to explicitly have question present in response
- + Comparison to classic DNS format in Appendix B
 - Make EDNS-Options a map

Updates since IETF 118 and hackathon results

- ✓ RFC1035-like name compression
- ✓ Dedicated types for more structured RDATA
 - ✓ SOA, ✓ MX, ✓ SRV, ✓ SVCB & HTTPS
- ✓ Answer section for queries (for Known Answers in mDNS)
- ✓ Amend capability to carry more than one question
- ✓ Switching boolean to queries to explicitly have question present in response
- + Comparison to classic DNS format in Appendix B
- ✓ Make EDNS-Options a map

Lessons learned during hackathon

- Name definition needs some mentioning of CBOR-packed
 - Currently says name parsing ends when element other than text string or tag TBDt is encountered
 - Needs mentioning that this is meant to be for *after* unpacking
- SVCB SvcParams: Provide extension point **\$\$svc-param-value**
 - Currently only = **bstr**
 - Allows for possible future structured types

Updates since IETF 118 and hackathon results

- + RFC1035-like name compression
- + **Dedicated types for more structured RDATA**
 - SOA, MX, SRV, SVCB & HTTPS
- + Answer section for queries (for Known Answers in mDNS)
- + Amend capability to carry more than one question
- + Switching boolean to queries to explicitly have question present in response
- + Comparison to classic DNS format in Appendix B
 - Make EDNS-Options a map

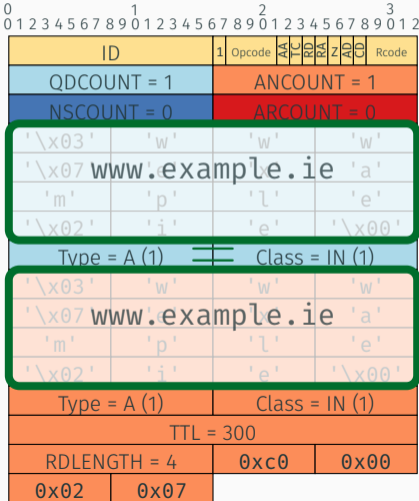
Excursion: Name Compression in RFC 1035 (DNS)

A DNS Response (62 bytes)

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	
ID										1	Opcode		AA	TC	RD	RA	Z	AD	C	Rcode			
QDCOUNT = 1										ANCOUNT = 1													
NSCOUNT = 0										ARCOUNT = 0													
' \x03 '		' w '		' w '		' w '																	
' \x07 '		' e '		' x '		' a '																	
' m '		' p '		' l '		' e '																	
' \x02 '		' i '		' e '		' \x00 '																	
Type = A (1)										Class = IN (1)													
' \x03 '		' w '		' w '		' w '																	
' \x07 '		' e '		' x '		' a '																	
' m '		' p '		' l '		' e '																	
' \x02 '		' i '		' e '		' \x00 '																	
Type = A (1)										Class = IN (1)													
TTL = 300																							
RDLENGTH = 4										0xc0				0x00									
0x02		0x07																					

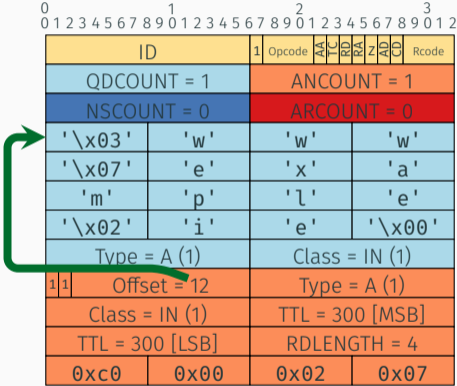
Excursion: Name Compression in RFC 1035 (DNS)

A DNS Response (62 bytes)



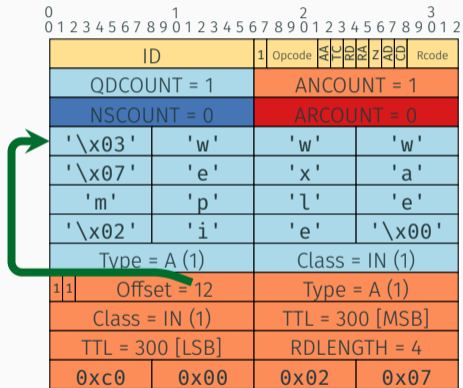
Excursion: Name Compression in RFC 1035 (DNS)

A DNS Response (48 bytes)



Excursion: Name Compression in RFC 1035 (DNS)

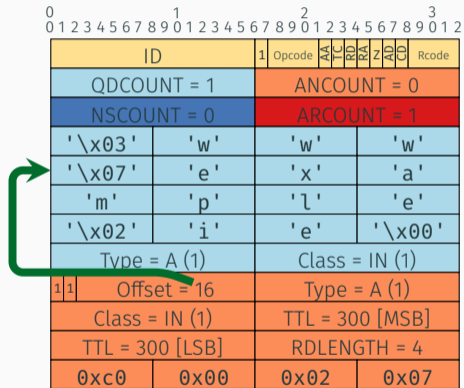
A DNS Response (48 bytes) ⇒ Do not re-invent the wheel!



How to translate byte pointer to CBOR?

Excursion: Name Compression in RFC 1035 (DNS)

A DNS Response (48 bytes) ⇒ Do not re-invent the wheel!



How to translate byte pointer to CBOR?
How to include name suffixes?

Name Compression in DNS+CBOR

How to include name suffixes?

EDN

```
[  
  ["www.example.ie",  
   1],  
  [  
    ["www.example.ie",  
     5, "example.ie"],  
    ["example.ie", 1,  
     h'c0000207']  
  ]  
]
```

Binary (hex, 65 bytes)

```
82  
82 6e 7777772e6578616d706c652e6965  
  01  
82  
83 6e 7777772e6578616d706c652e6965  
  05 6a 6578616d706c652e6965  
83 6a 6578616d706c652e6965 01  
  44 c0000207
```

Name Compression in DNS+CBOR

How to include name suffixes?

EDN

```
[  
  ["www", "example", "ie",  
   1],  
  [  
    ["www", "example", "ie",  
     5, "example", "ie"],  
    ["example", "ie", 1,  
     h'c0000207']  
  ]  
]
```

Split names into labels

Binary (hex, 65 bytes)

```
82  
84 63 777777 67 6578616d706c65 62 6965  
01  
82  
86 63 777777 67 6578616d706c65 62 6965  
05 67 6578616d706c65 62 6965  
84 67 6578616d706c65 62 6965 01  
44 c0000207
```

Name Compression in DNS+CBOR

How to include name suffixes?

EDN

```
[  
  ["www", "example", "ie",  
   1],  
  [  
    ["www", "example", "ie",  
     5, "example", "ie"],  
    ["example", "ie", 1,  
     h'c0000207']  
  ]  
]
```

Split names into labels

Binary (hex, 65 bytes)

82

84 63 777777 67 6578616d706c65 62 6965

No additional overhead!
(When labels are
short enough)

86 63 777777 67 6578616d706c65 62 6965

05 67 6578616d706c65 6965

84 67 6578616d706c65 62 6965 01

44 c0000207

Name Compression in DNS+CBOR

How to translate byte pointer to CBOR?

EDN

```
[  
  ["www", "example", "ie",  
   1],  
  [  
    ["www", "example", "ie",  
     5, "example", "ie"],  
    ["example", "ie", 1,  
     h'c0000207']  
  ]  
]
```

Binary (hex, 65 bytes)

```
82  
84 63 777777 67 6578616d706c65 62 6965  
01  
82  
86 63 777777 67 6578616d706c65 62 6965  
05 67 6578616d706c65 62 6965  
84 67 6578616d706c65 62 6965 01  
44 c0000207
```

Name Compression in DNS+CBOR

How to translate byte pointer to CBOR?

EDN

```
[  
  ["www", "example", "ie",  
    1],
```

Names are the
only text strings...

```
  ["example", "ie", 1,  
    h'c0000207']  
]
```

```
]
```

Binary (hex, 65 bytes)

82

```
84 63 777777 67 6578616d706c65 62 6965  
01
```

82

```
86 63 777777 67 6578616d706c65 62 6965  
05 67 6578616d706c65 62 6965  
84 67 6578616d706c65 62 6965 01  
44 c0000207
```

Name Compression in DNS+CBOR

How to translate byte pointer to CBOR?

EDN

```
[  
  ["www", "example", "ie",  
  1],  
  ["www", "example", "ie",  
  5, "example", "ie"],  
  ["example", "ie", 1,  
  h'c0000207']  
]
```

Names are the
only text strings...
Let's count them
for reference!

Binary (hex, 65 bytes)

```
82  
84 63 777777 67 6578616d706c65 62 6965  
01  
82  
86 63 777777 67 6578616d706c65 62 6965  
05 67 6578616d706c65 62 6965  
84 67 6578616d706c65 62 6965 01  
44 c0000207
```


Name Compression in DNS+CBOR

How to translate byte pointer to CBOR?

EDN

```
[  
  [ 0"www", 1"example", 2"ie",  
    1 ],  
  [  
    ["www", "example", "ie",  
     5, "example", "ie"],  
    ["example", "ie", 1,  
     h'c0000207']  
  ]  
]
```

Binary (hex, 65 bytes)

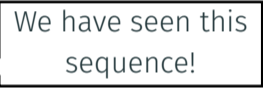
```
82  
84 63 777777 67 6578616d706c65 62 6965  
01  
82  
86 63 777777 67 6578616d706c65 62 6965  
05 67 6578616d706c65 62 6965  
84 67 6578616d706c65 62 6965 01  
44 c0000207
```

Name Compression in DNS+CBOR

How to translate byte pointer to CBOR?

EDN

```
[  
  [ "www", "example", "ie",  
    1 ],  
  [ "www", "example", "ie",  
    5, "example", "ie" ],  
  [ "example", "ie", 1,  
    h'c0000207' ]  
]
```



Binary (hex, 65 bytes)

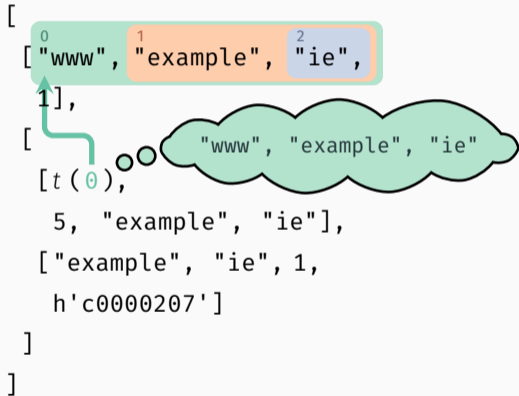
```
82  
84 63 777777 67 6578616d706c65 62 6965  
01  
82  
86 63 777777 67 6578616d706c65 62 6965  
05 67 6578616d706c65 62 6965  
84 67 6578616d706c65 62 6965 01  
44 c0000207
```

Name Compression in DNS+CBOR

How to translate byte pointer to CBOR?

EDN

```
[  
  [0  
  ["www", "example", "ie",  
  ],  
  [1  
  [t(0),  
    5, "example", "ie"],  
  ["example", "ie", 1,  
    h'c0000207']  
  ]  
]
```

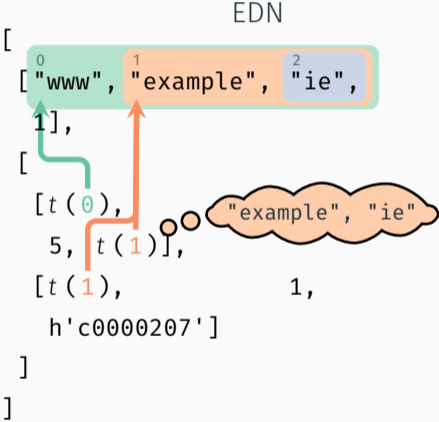


Binary (hex, 65 bytes)

```
82  
84 63 777777 67 6578616d706c65 62 6965  
01  
82  
84 ct 00  
05 67 6578616d706c65 62 6965  
84 67 6578616d706c65 62 6965 01  
44 c0000207
```

Name Compression in DNS+CBOR

How to translate byte pointer to CBOR?



Binary (hex)

```
82  
84 63 777777 67 6578616d706c65 62 6965  
01  
82  
83 ct 00  
05 ct 01  
83 ct 01  
44 c0000207
```

Name Compression in DNS+CBOR

How to translate byte pointer to CBOR?

EDN	Binary (hex)						
<pre>[[t(0), 5, t(1)], t(1), h'c0000207']</pre>	<p>How should <i>t</i> be defined?</p> <hr/> <table border="0"> <tr> <td>1+0 tag</td> <td>1+1 tag</td> </tr> <tr> <td>≥2 bytes</td> <td>≥3 bytes</td> </tr> <tr> <td>ct XX</td> <td>d8 tt XX</td> </tr> </table> <hr/> <p>1, Similar in size to DNS pointers</p> <p>Less crowded real-estate</p>	1+0 tag	1+1 tag	≥2 bytes	≥3 bytes	ct XX	d8 tt XX
1+0 tag	1+1 tag						
≥2 bytes	≥3 bytes						
ct XX	d8 tt XX						

6965

How to Compress More Hidden Names?

Make RRs with names structured

- Definitions in `draft-09`: SOA, MX, SRV, SVCB, HTTPS
- CDDL extension point `$$structured-ts-rd` for future record types

Example: MX record to example.org with TTL 3600

- `draft-08`: `[3600, 15, h'0001076578616d706c65036f726700']`
83 190e10 0f 4f 0001076578616d706c65036f726700
- `draft-09`: `[3600, 15, [1, "example", "org"]]`
83 190e10 0f 83 01 67 6578616d706c65 63 6f7267

Further Optimizations?

Resource Record Sets instead of Resource Records

- Summarize Record Sets instead of sending every record each
- `[300, 1, [h'c0000207', h'c0000208']]` instead of `[300, 1, h'c0000207']`, `[300, 1, h'c0000208']`
- + More concise
- + Easier to sign with, e.g., DNSSEC (or COSE)
- Introduces 1 byte overhead per RR for RRset size 1

Punycode labels when shorter than UTF-8?

- Measurements show this can be a thing.

What do you think?

CoAP Content-Format for new Media Types

Content Type	Content Coding	ID
application/dns+cbor		TBD
application/dns+cbor;packed=1		TBD

CoAP Content-Format for new Media Types

Content Type	Content Coding	ID
application/dns+cbor		53?
application/dns+cbor;packed=1		54?

Should DNS+CBOR be adopted by
CBOR WG or should we go to
DNSOP?