

Martine S. Lenders (martine.lenders@tu-dresden.de)

From IoT Experiments to Internet Standards:

The road to DNS over CoAP

Dresden, 2024-12-18, TUD netd/HAW iNETRG Joint Christmas Lecture

Outline

Motivation

Designing DNS over CoAP

Going to IETF

A New Message Format is Needed

The Classic Internet Can Benefit

Conclusion

Outline

Motivation

Designing DNS over CoAP

Going to IETF

A New Message Format is Needed

The Classic Internet Can Benefit

Conclusion

Motivation

Attack Scenario: Name resolution by IoT devices



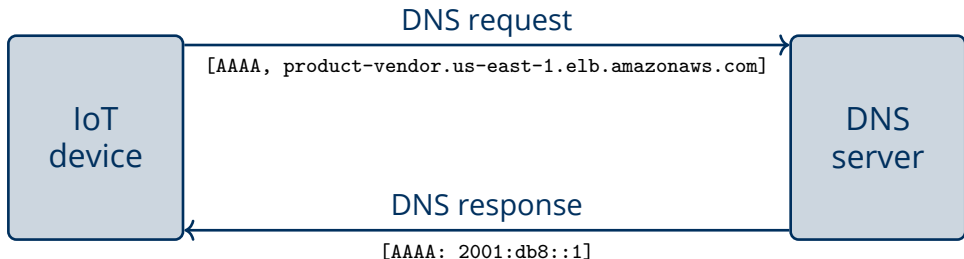
Motivation

Attack Scenario: Name resolution by IoT devices



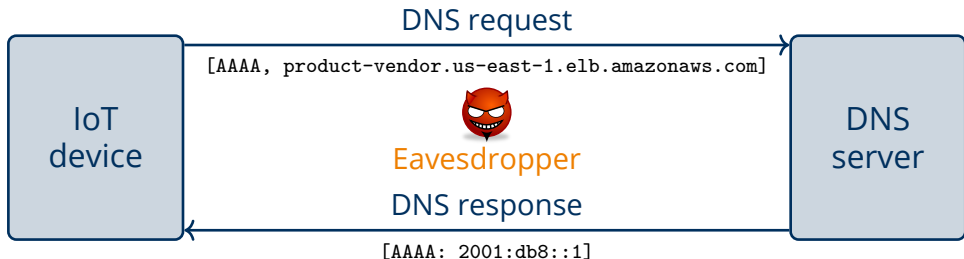
Motivation

Attack Scenario: Name resolution by IoT devices



Motivation

Attack Scenario: Name resolution by IoT devices

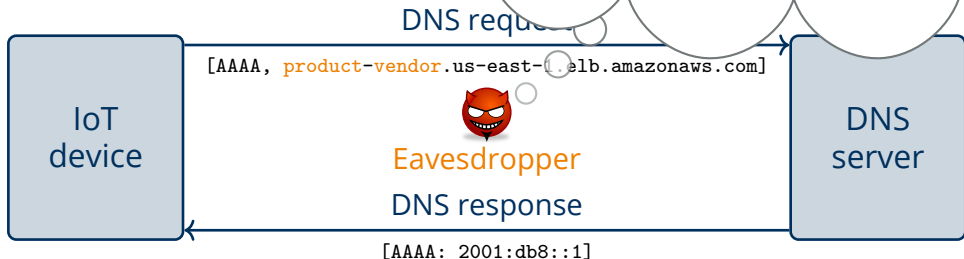


Motivation

Attack Scenario: Name resolution

<https://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=product+vendor>

Name	Description
CVE-2019-XXXX	Vendor Product are affected by a stack-based buffer overflow by an unauthenticated attacker which allows for remote code execution .
CVE-2023-YYYY	Vendor Product ...

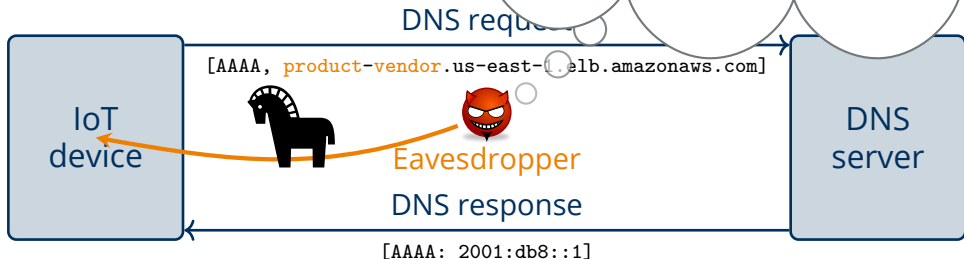


Motivation

Attack Scenario: Name resolution

<https://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=product+vendor>

Name	Description
CVE-2019-XXXXX	Vendor Product are affected by a stack-based buffer overflow by an unauthenticated attacker which allows for remote code execution .
CVE-2023-YYYYY	Vendor Product ...

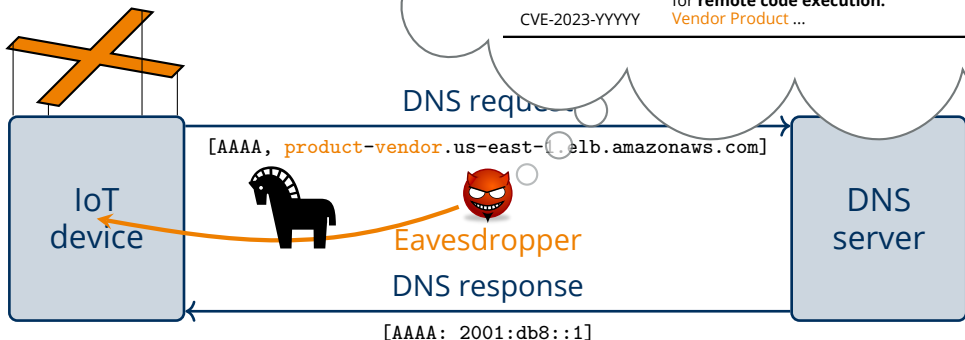


Motivation

Attack Scenario: Name resolution

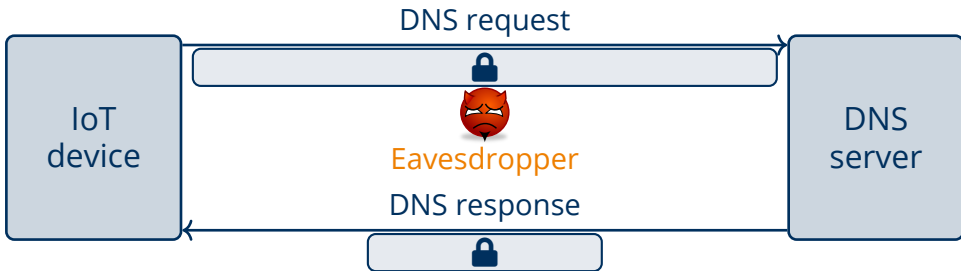
<https://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=product+vendor>

Name	Description
CVE-2019-XXXXX	Vendor Product are affected by a stack-based buffer overflow by an unauthenticated attacker which allows for remote code execution .
CVE-2023-YYYYY	Vendor Product ...



Motivation

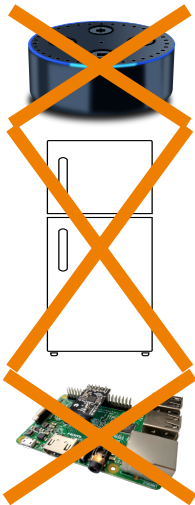
Attack Scenario: Name resolution by IoT devices



Countermeasure

Encrypt name resolution triggered by IoT devices against eavesdropping

Challenge: Constrained IoT



Constrained nodes (RFC 7228):

Characteristic	Class 0	Class 1	Class 2
Data size [KiB]	$\ll 10$	≈ 10	≈ 50
Code size [KiB]	$\ll 100$	≈ 100	≈ 250

Challenge: Constrained IoT



Constrained nodes (RFC 7228):

Characteristic	Class 0	Class 1	Class 2
Data size [KiB]	$\ll 10$	≈ 10	≈ 50
Code size [KiB]	$\ll 100$	≈ 100	≈ 250



Challenge: Constrained IoT



Constrained nodes (RFC 7228):

Characteristic	Class 0	Class 1	Class 2
Data size [KiB]	$\ll 10$	≈ 10	≈ 50
Code size [KiB]	$\ll 100$	≈ 100	≈ 250

Constrained networks:

- Low throughput, high packet loss, asymmetric link characteristics
- High penalties on large packets (link layer fragmentation)

Challenge: Constrained IoT

BLE



zigbee



LoRa[®]



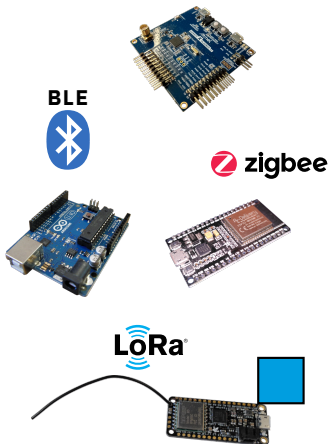
Constrained nodes (RFC 7228):

Characteristic	Class 0	Class 1	Class 2
Data size [KiB]	$\ll 10$	≈ 10	≈ 50
Code size [KiB]	$\ll 100$	≈ 100	≈ 250

Constrained networks:

- Low throughput, high packet loss, asymmetric link characteristics
- **High penalties on large packets** (link layer fragmentation)

Challenge: Constrained IoT



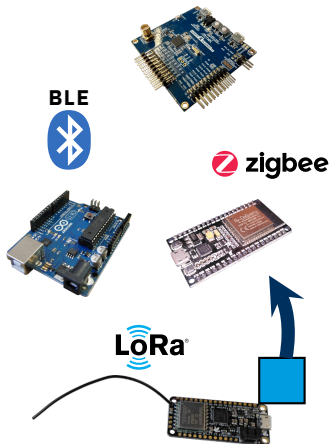
Constrained nodes (RFC 7228):

Characteristic	Class 0	Class 1	Class 2
Data size [KiB]	$\ll 10$	≈ 10	≈ 50
Code size [KiB]	$\ll 100$	≈ 100	≈ 250

Constrained networks:

- Low throughput, high packet loss, asymmetric link characteristics
- **High penalties on large packets** (link layer fragmentation)

Challenge: Constrained IoT



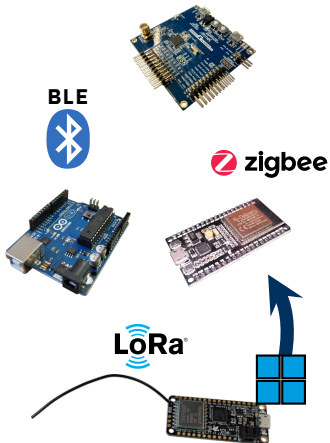
Constrained nodes (RFC 7228):

Characteristic	Class 0	Class 1	Class 2
Data size [KiB]	$\ll 10$	≈ 10	≈ 50
Code size [KiB]	$\ll 100$	≈ 100	≈ 250

Constrained networks:

- Low throughput, high packet loss, asymmetric link characteristics
- **High penalties on large packets** (link layer fragmentation)

Challenge: Constrained IoT



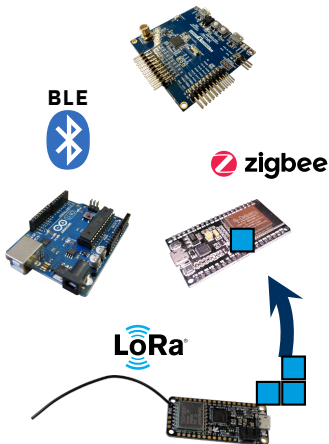
Constrained nodes (RFC 7228):

Characteristic	Class 0	Class 1	Class 2
Data size [KiB]	$\ll 10$	≈ 10	≈ 50
Code size [KiB]	$\ll 100$	≈ 100	≈ 250

Constrained networks:

- Low throughput, high packet loss, asymmetric link characteristics
- **High penalties on large packets** (link layer fragmentation)

Challenge: Constrained IoT



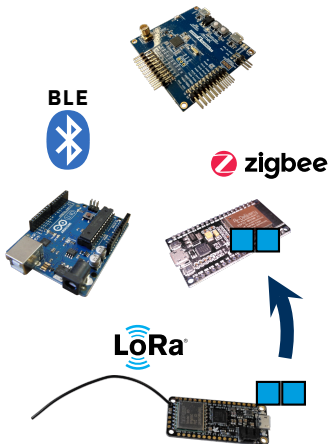
Constrained nodes (RFC 7228):

Characteristic	Class 0	Class 1	Class 2
Data size [KiB]	$\ll 10$	≈ 10	≈ 50
Code size [KiB]	$\ll 100$	≈ 100	≈ 250

Constrained networks:

- Low throughput, high packet loss, asymmetric link characteristics
- **High penalties on large packets** (link layer fragmentation)

Challenge: Constrained IoT



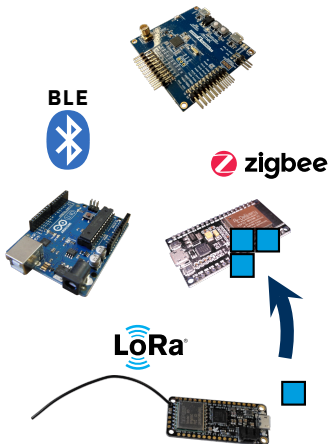
Constrained nodes (RFC 7228):

Characteristic	Class 0	Class 1	Class 2
Data size [KiB]	$\ll 10$	≈ 10	≈ 50
Code size [KiB]	$\ll 100$	≈ 100	≈ 250

Constrained networks:

- Low throughput, high packet loss, asymmetric link characteristics
- **High penalties on large packets** (link layer fragmentation)

Challenge: Constrained IoT



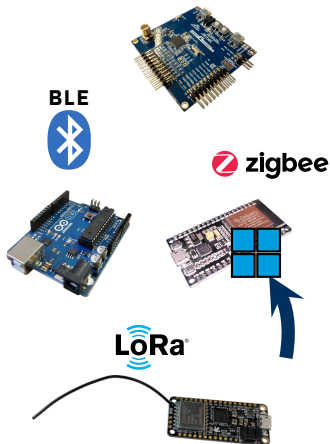
Constrained nodes (RFC 7228):

Characteristic	Class 0	Class 1	Class 2
Data size [KiB]	$\ll 10$	≈ 10	≈ 50
Code size [KiB]	$\ll 100$	≈ 100	≈ 250

Constrained networks:

- Low throughput, high packet loss, asymmetric link characteristics
- **High penalties on large packets** (link layer fragmentation)

Challenge: Constrained IoT



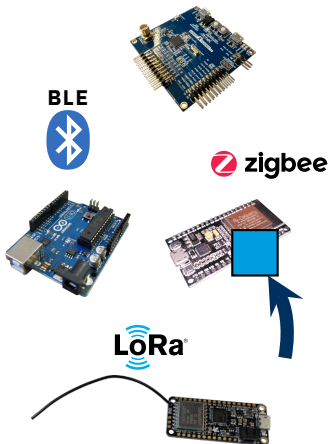
Constrained nodes (RFC 7228):

Characteristic	Class 0	Class 1	Class 2
Data size [KiB]	$\ll 10$	≈ 10	≈ 50
Code size [KiB]	$\ll 100$	≈ 100	≈ 250

Constrained networks:

- Low throughput, high packet loss, asymmetric link characteristics
- **High penalties on large packets** (link layer fragmentation)

Challenge: Constrained IoT



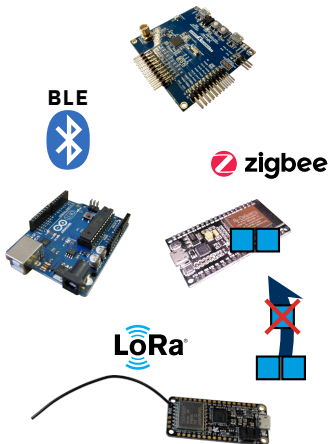
Constrained nodes (RFC 7228):

Characteristic	Class 0	Class 1	Class 2
Data size [KiB]	$\ll 10$	≈ 10	≈ 50
Code size [KiB]	$\ll 100$	≈ 100	≈ 250

Constrained networks:

- Low throughput, high packet loss, asymmetric link characteristics
- **High penalties on large packets** (link layer fragmentation)

Challenge: Constrained IoT



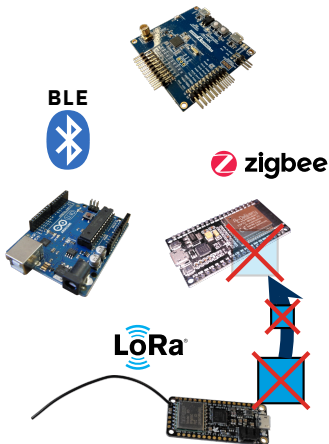
Constrained nodes (RFC 7228):

Characteristic	Class 0	Class 1	Class 2
Data size [KiB]	$\ll 10$	≈ 10	≈ 50
Code size [KiB]	$\ll 100$	≈ 100	≈ 250

Constrained networks:

- Low throughput, high packet loss, asymmetric link characteristics
- **High penalties on large packets** (link layer fragmentation)

Challenge: Constrained IoT



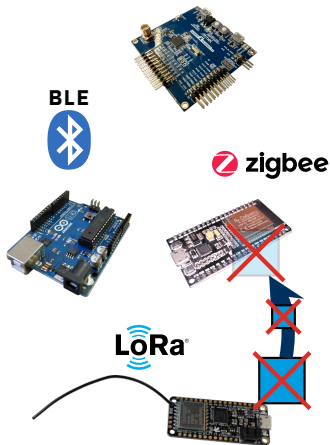
Constrained nodes (RFC 7228):

Characteristic	Class 0	Class 1	Class 2
Data size [KiB]	$\ll 10$	≈ 10	≈ 50
Code size [KiB]	$\ll 100$	≈ 100	≈ 250

Constrained networks:

- Low throughput, high packet loss, asymmetric link characteristics
- **High penalties on large packets** (link layer fragmentation)

Challenge: Constrained IoT



Constrained nodes (RFC 7228):

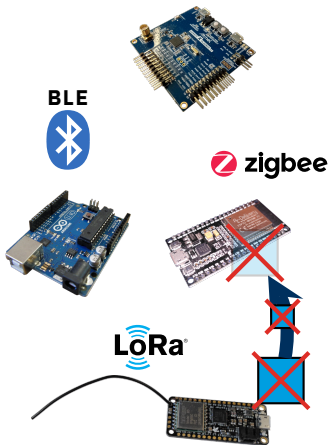
Characteristic	Class 0	Class 1	Class 2
Data size [KiB]	$\ll 10$	≈ 10	≈ 50
Code size [KiB]	$\ll 100$	≈ 100	≈ 250

Constrained networks:

- Low throughput, high packet loss, asymmetric link characteristics
- High penalties on large packets (link layer fragmentation)

Characteristic	IEEE 802.15.4	BLE	LoRaWAN
Data rate [kBit/s]	124-162	125-2000	0.3-5
Frame size [bytes]	127	≥ 1280	59-250

Challenge: Constrained IoT



Constrained nodes (RFC 7228):

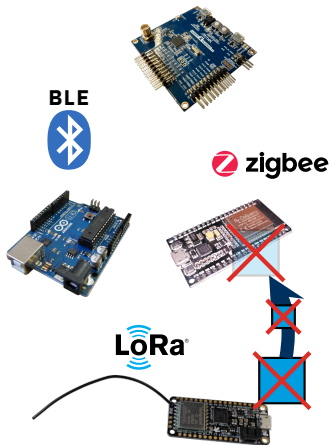
Characteristic	Class 0	Class 1	Class 2
Data size [KiB]	$\ll 10$	≈ 10	≈ 50
Code size [KiB]	$\ll 100$	≈ 100	≈ 250

Constrained networks:

- Low throughput, high packet loss, asymmetric link characteristics
- High penalties on large packets (link layer fragmentation)

Characteristic	IEEE 802.15.4	BLE	LoRaWAN
Data rate [kBit/s]	124-162	125-2000	0.3-5
Frame size [bytes]	127	≥ 1280	59-250

Challenge: Constrained IoT



Constrained nodes (RFC 7228):

Characteristic	Class 0	Class 1	Class 2
Data size [KiB]	$\ll 10$	≈ 10	≈ 50
Code size [KiB]	$\ll 100$	≈ 100	≈ 250

Constrained

- Low throughput
- High penalties on large packets (link layer fragmentation)

0.000003% – 0.0009%
slower than WiFi 6

Characteristic	IEEE 802.15.4	BLE	LoRaWAN
Data rate [kBit/s]	124-162	125-2000	0.3-5
Frame size [bytes]	127	≥ 1280	59-250

Possible Solutions for Encrypted DNS

DNS over HTTPS
(RFC 8484)

DNS over TLS
(RFC 7858)

Possible Solutions for Encrypted DNS

DNS over HTTPS
(RFC 8484)

DNS over TLS
(RFC 7858)

DNS over QUIC
(RFC 9250)

Possible Solutions for Encrypted DNS

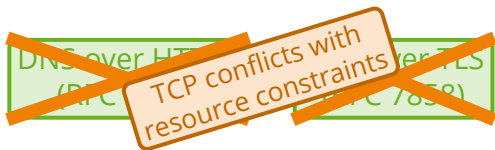
DNS over HTTPS
(RFC 8484)

DNS over TLS
(RFC 7858)

DNS over QUIC
(RFC 9250)

DNS over DTLS
(RFC 8094)

Possible Solutions for Encrypted DNS



DNS over QUIC
(RFC 9250)

DNS over DTLS
(RFC 8094)

Possible Solutions for Encrypted DNS



Possible Solutions for Encrypted DNS



Possible Solutions for Encrypted DNS

Our proposal: DNS over CoAP

(<https://datatracker.ietf.org/doc/draft-ietf-core-dns-over-coap/>)

- **Encrypted communication** based on DTLS or OSCORE
- **Block-wise message transfer** provides message segmentation
- **En-route caching** mitigates high link layer packet loss
- **Share system resources** with CoAP applications
 - Same socket and buffers can be used
 - Re-use of the CoAP retransmission mechanism

vs.
r PDUS

Outline

Motivation

Designing DNS over CoAP

Going to IETF

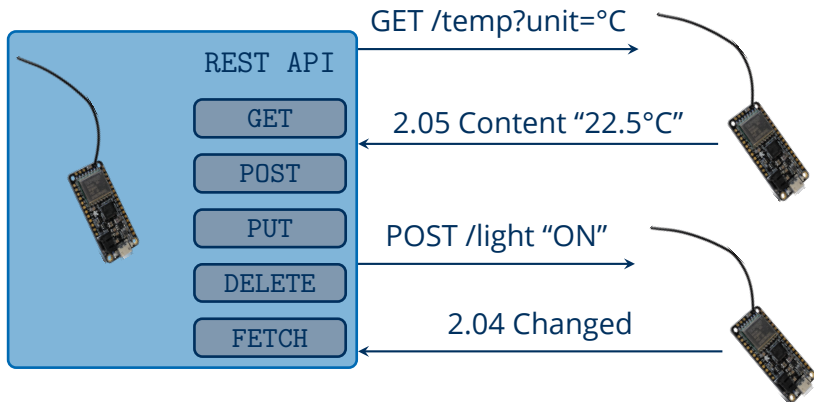
A New Message Format is Needed

The Classic Internet Can Benefit

Conclusion

CoAP: The **C**onstrained **A**pplication **P**rotocol

“REST over UDP” ~ The HTTP for IoT



CoAP Security Modes

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

Encrypted Transport



CoAP Security Modes

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

Encrypted Transport



CoAP Security Modes

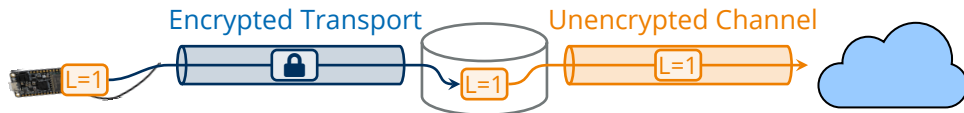
DTLS Datagram Transport Layer Security (\approx TLS over UDP)

Encrypted Transport



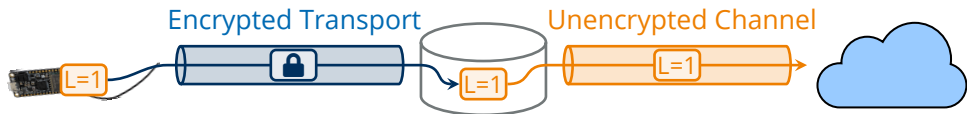
CoAP Security Modes

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



CoAP Security Modes

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

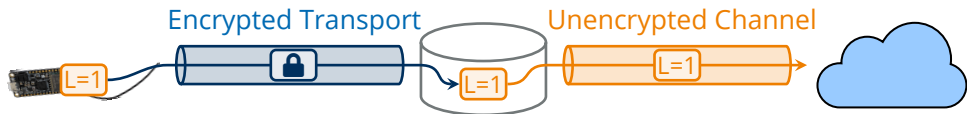


OSCORE Object Security for Constrained RESTful Environment



CoAP Security Modes

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

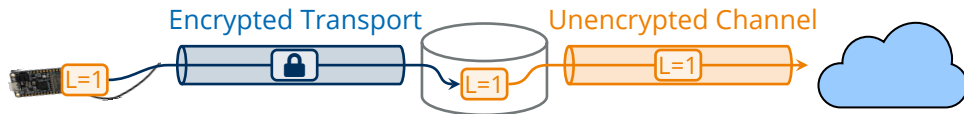


OSCORE Object Security for Constrained RESTful Environment



CoAP Security Modes

DTLS Datagram Transport Layer Security (\approx 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**?

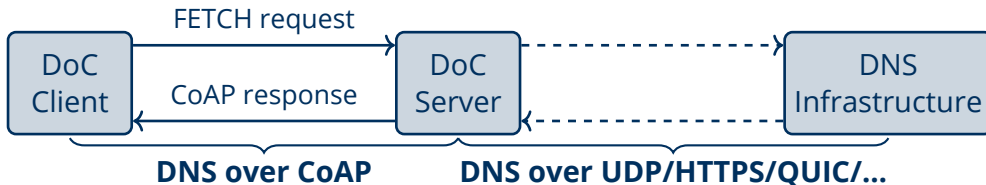
	HTTP	
	GET	POST
Responses cacheable	✓	✗
Application data carried in body	✗	✓
Block-wise transferable query	✗	✓

DNS over CoAP (DoC)

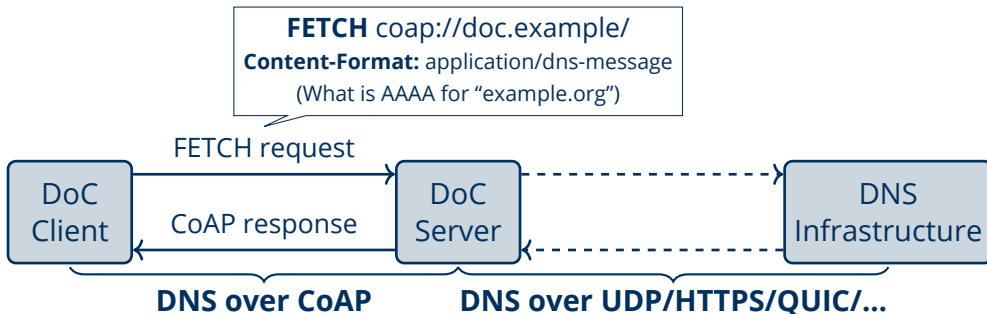
- Just map the DoH methods **GET** and **POST**?
- **FETCH** method in CoAP: best of both worlds (RFC 8132)

	CoAP		
	HTTP		
	GET	POST	FETCH
Responses cacheable	✓	✗	✓
Application data carried in body	✗	✓	✓
Block-wise transferable query	✗	✓	✓

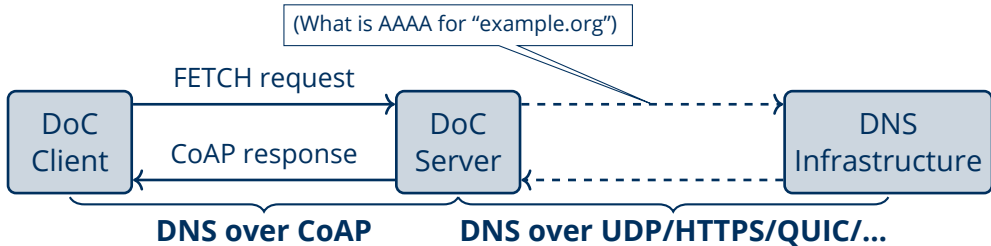
DNS over CoAP (DoC): Example Query



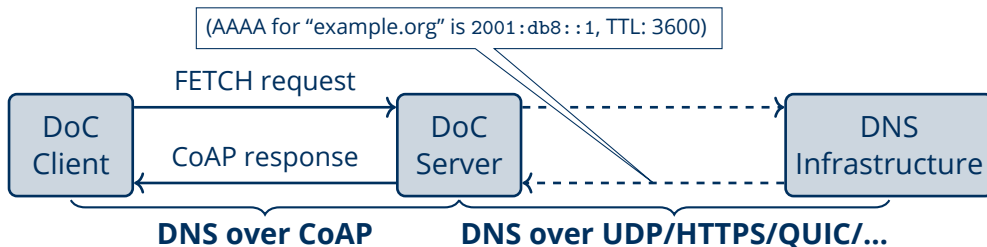
DNS over CoAP (DoC): Example Query



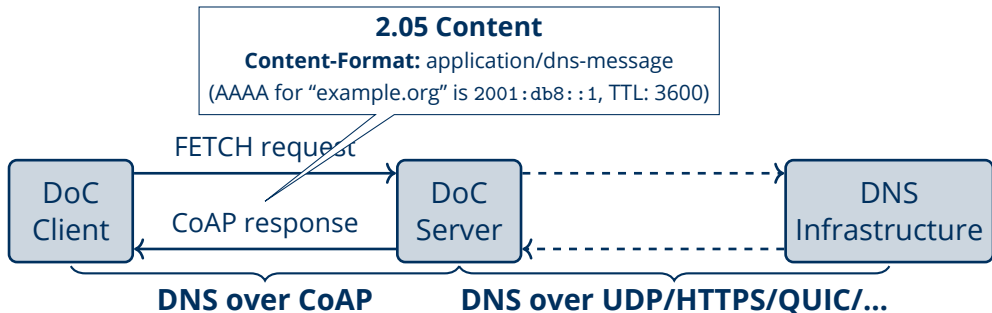
DNS over CoAP (DoC): Example Query



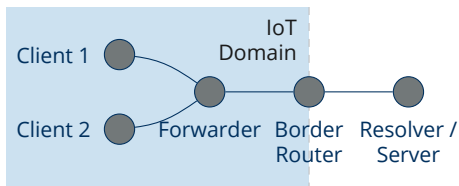
DNS over CoAP (DoC): Example Query



DNS over CoAP (DoC): Example Query

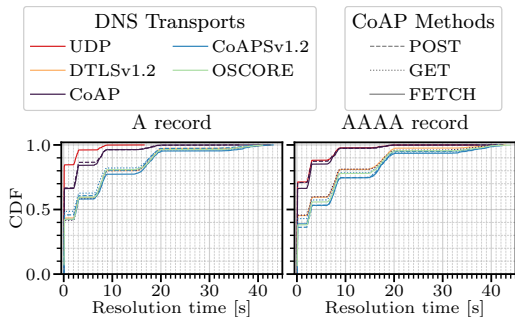


Evaluation Setup: DNS Transfer Protocol Comparison

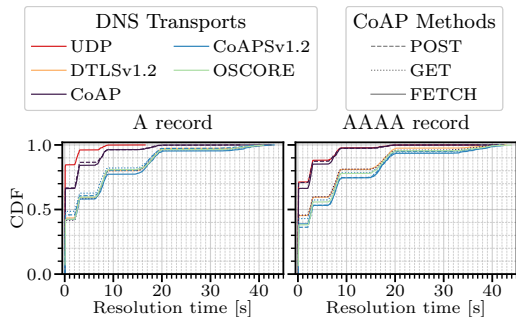


- Client 1 and 2 query records for names of length 24 chars (median of IoT data) via
 - UDP
 - DTLSv1.2
 - CoAP (unencrypted)
 - CoAPsv1.2 (CoAP over DTLSv1.2)
 - OSCORE
- 50 A and AAAA records each (most in IoT data)
- Poisson distribution: $\lambda = 5$ queries / sec
- 10 runs on IoT-nodes (incl. border router): Cortex-M3 with IEEE 802.15.4 radio

Experiment: Resolution Time

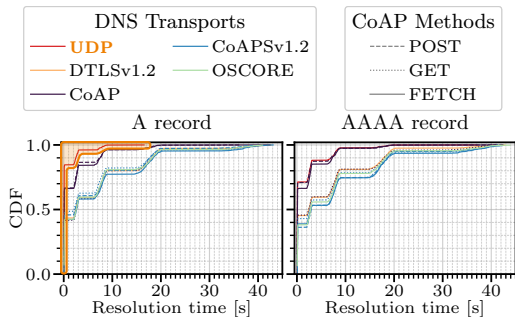


Experiment: Resolution Time



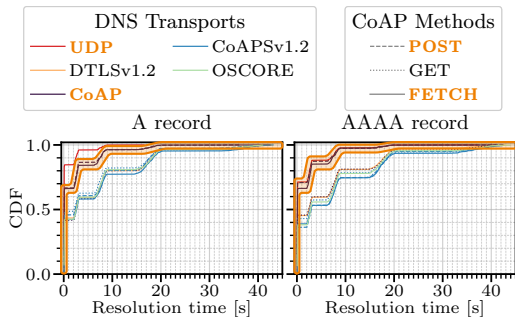
Clear performance groups visible

Experiment: Resolution Time



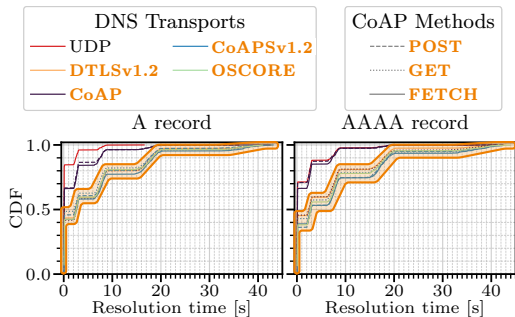
Group 1

Experiment: Resolution Time



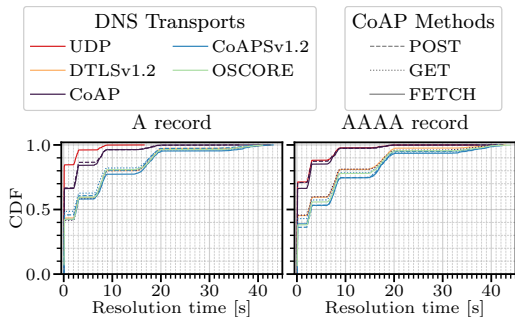
Group 2

Experiment: Resolution Time



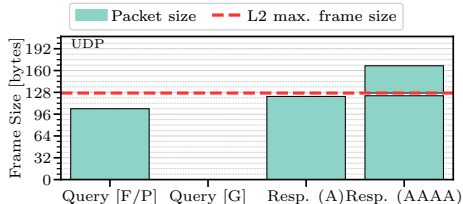
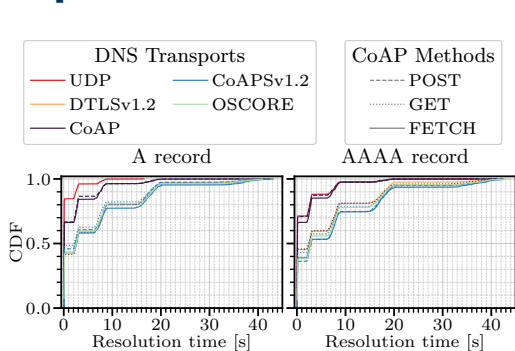
Group 3

Experiment: Resolution Time

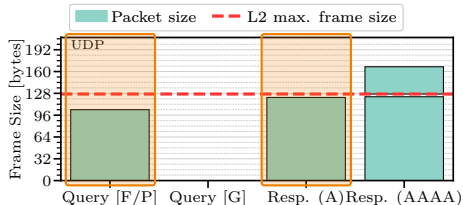
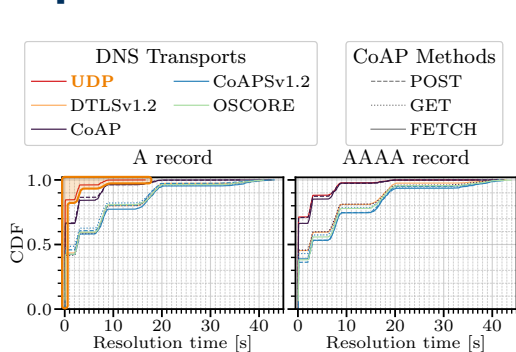


Where do performance groups come from?

Experiment: Resolution Time & Packet Sizes

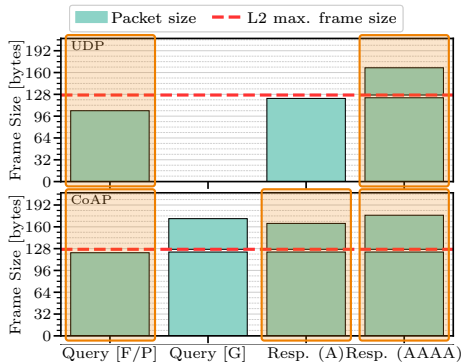
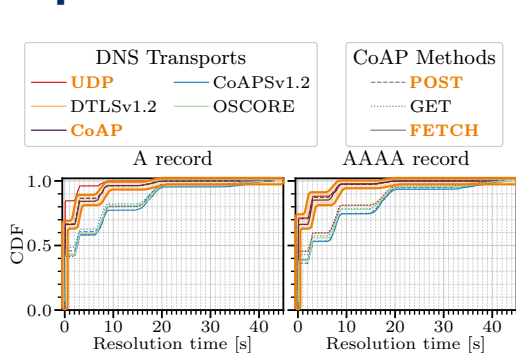


Experiment: Resolution Time & Packet Sizes



Group 1
No message fragmentation

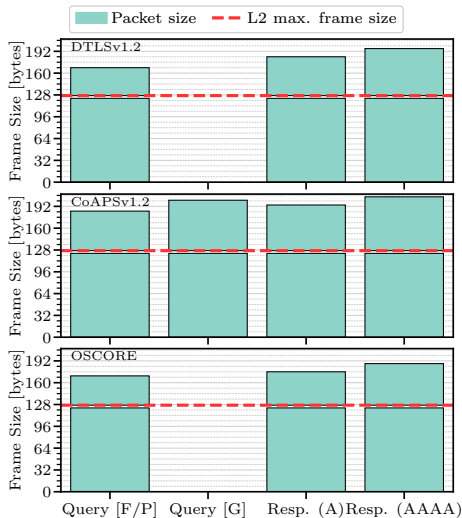
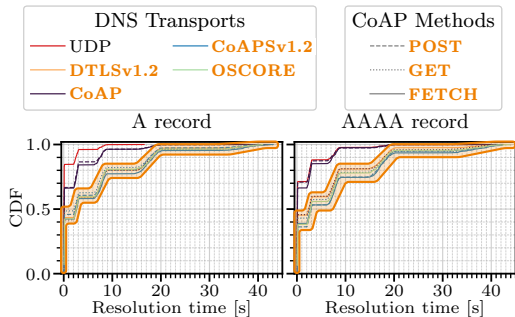
Experiment: Resolution Time & Packet Sizes



Group 2

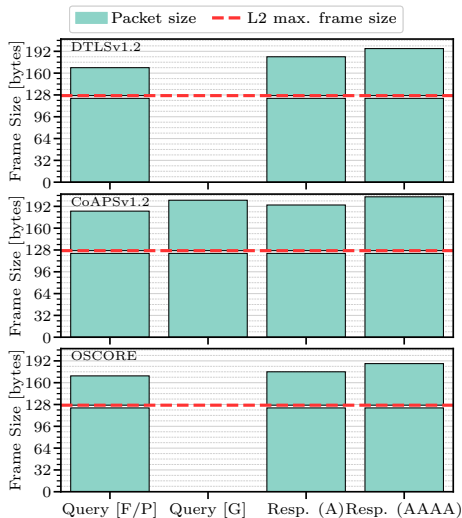
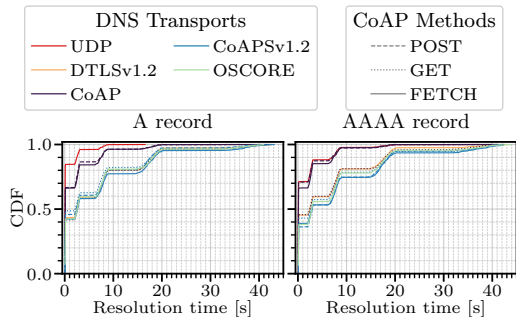
Query unfragmented
Response fragmented

Experiment: Resolution Time & Packet Sizes



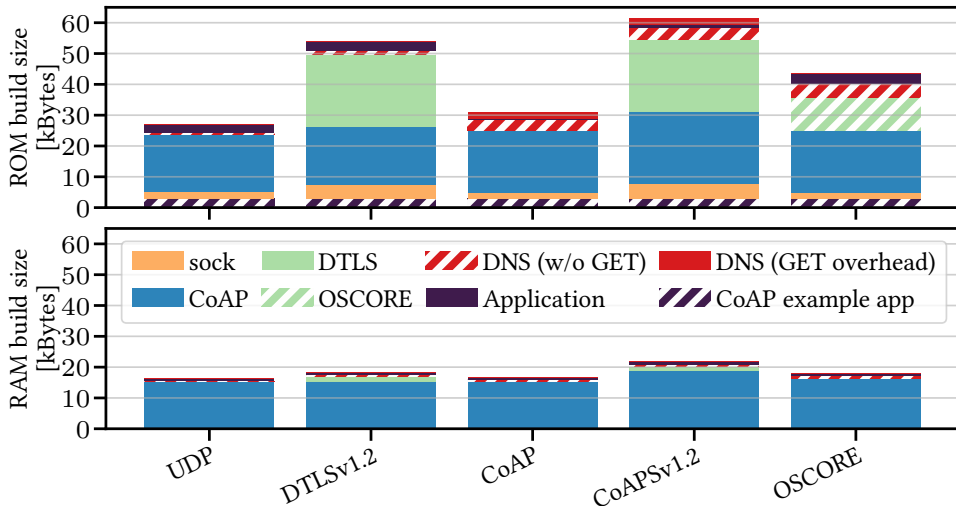
Group 3
Both messages fragmented

Experiment: Resolution Time & Packet Sizes

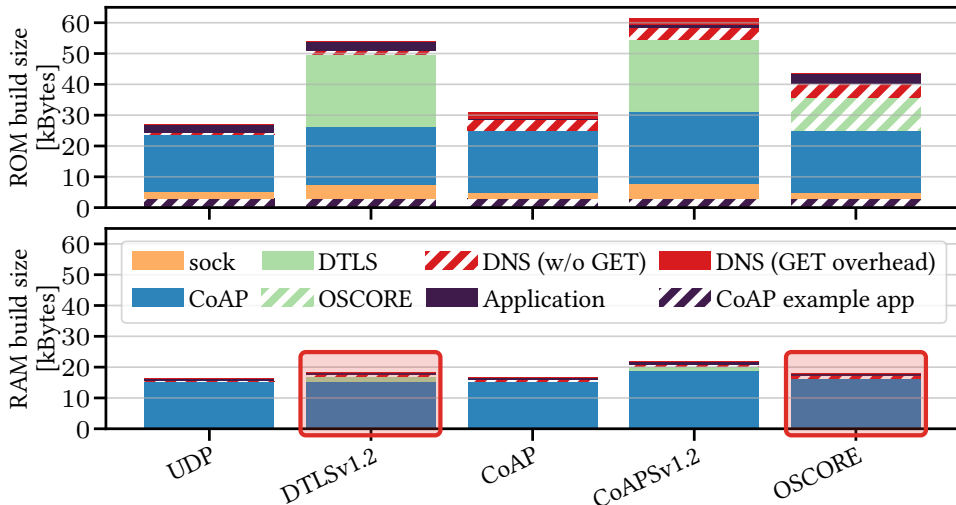


⇒ **Fragmentation has larger impact on performance** compared to transfer protocol or CoAP method

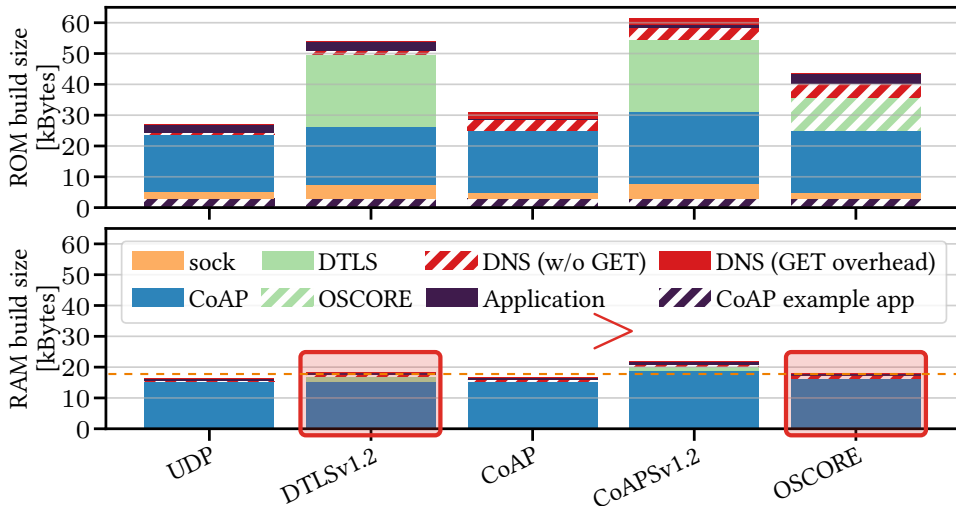
Memory Consumption



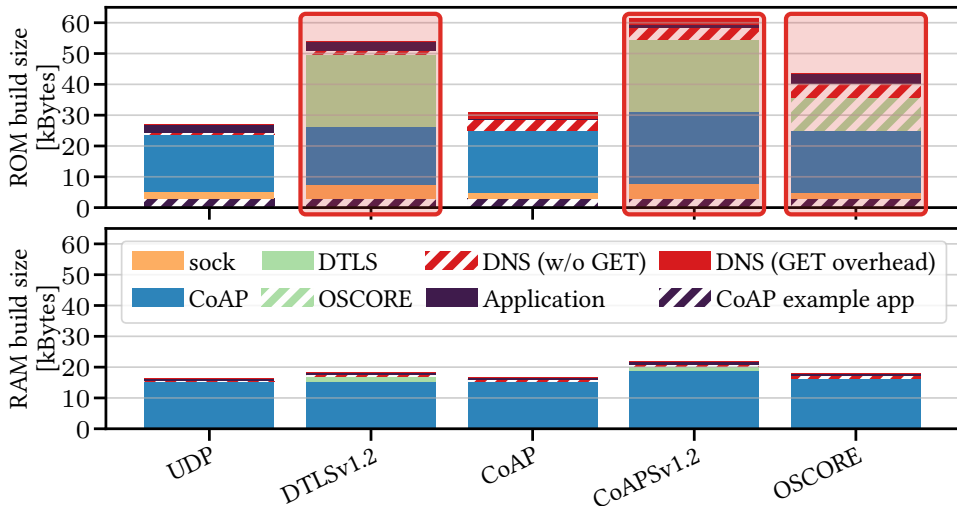
Memory Consumption



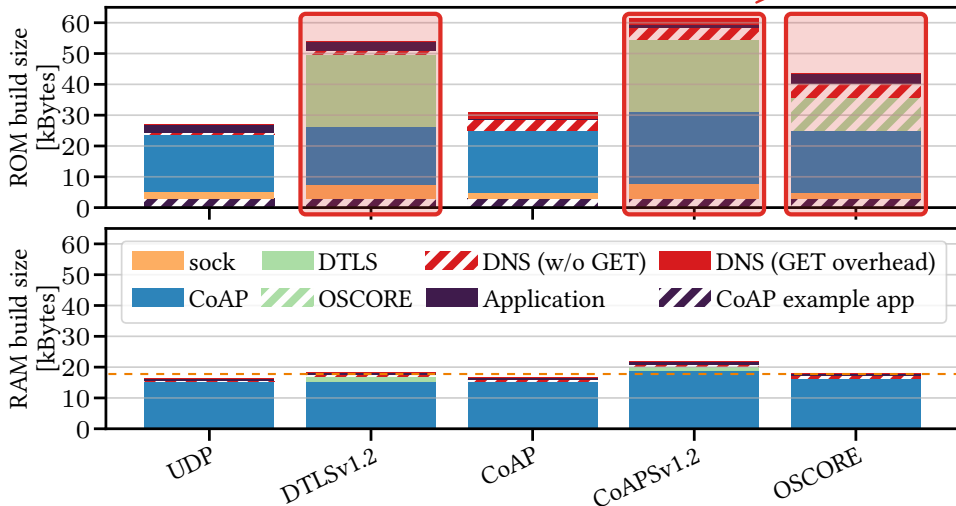
Memory Consumption



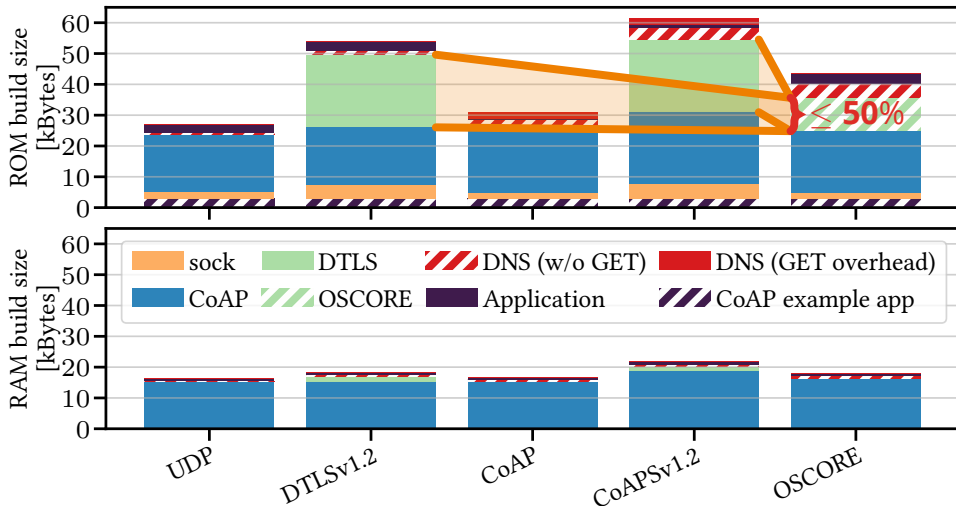
Memory Consumption



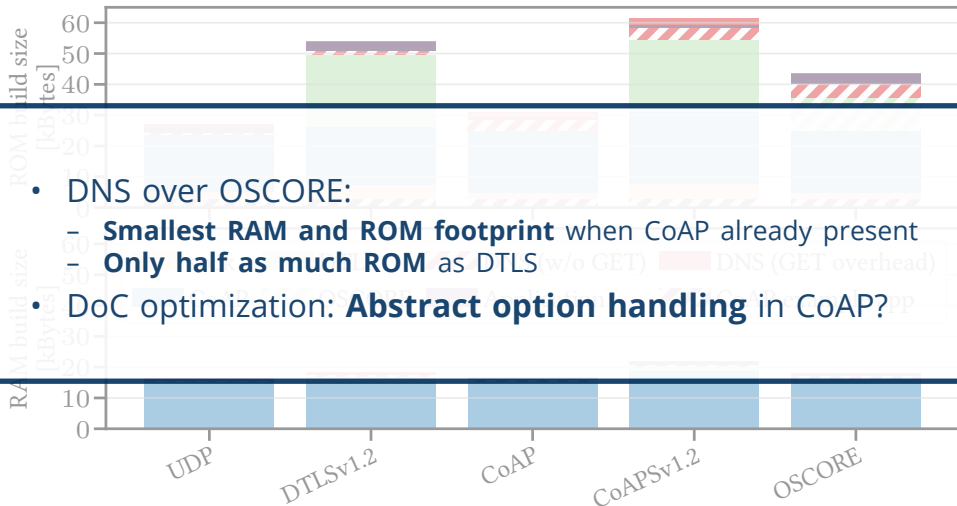
Memory Consumption



Memory Consumption



Memory Consumption



Outline

Motivation

Designing DNS over CoAP

Going to IETF

A New Message Format is Needed

The Classic Internet Can Benefit

Conclusion

So We Have a Protocol – What now?

- Anyone can design a protocol, how to make it a standard?
- Propose it to standardization bodies, *e.g.*, the



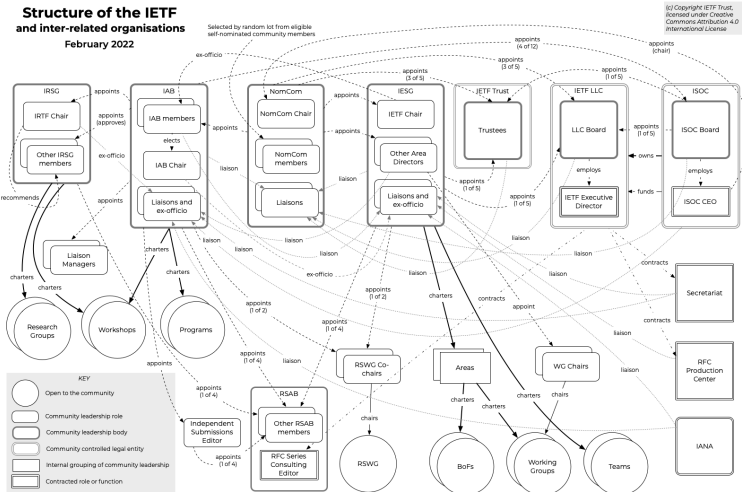
What is the IETF?

What is the IETF?

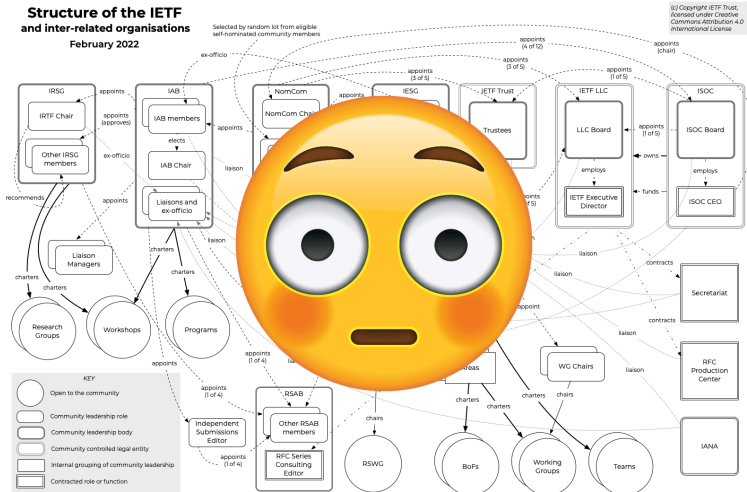
- Internet **E**ngineering **T**ask **F**orce
- Protocols standardization between and including Network and Application Layer: TCP/IP stack
- Open, grassroots-based standardization
 - Standardization freely available on the web in open formats as RFCs
 - No participation requirements¹
 - Decision-making based on “rough consensus and running code”

¹Except meeting registration and travel fees.

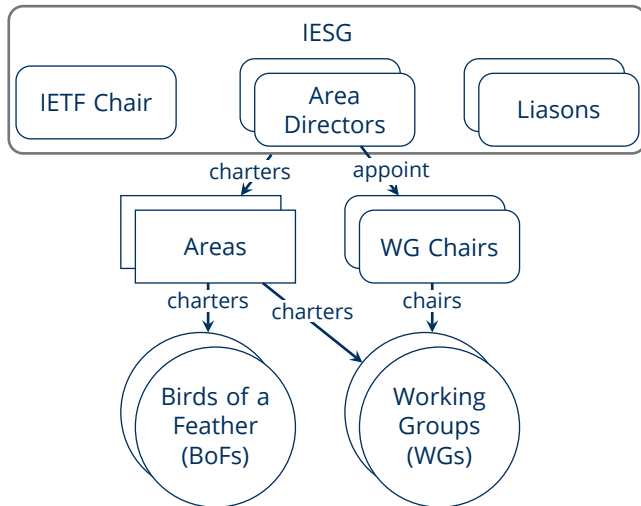
How does the IETF work?



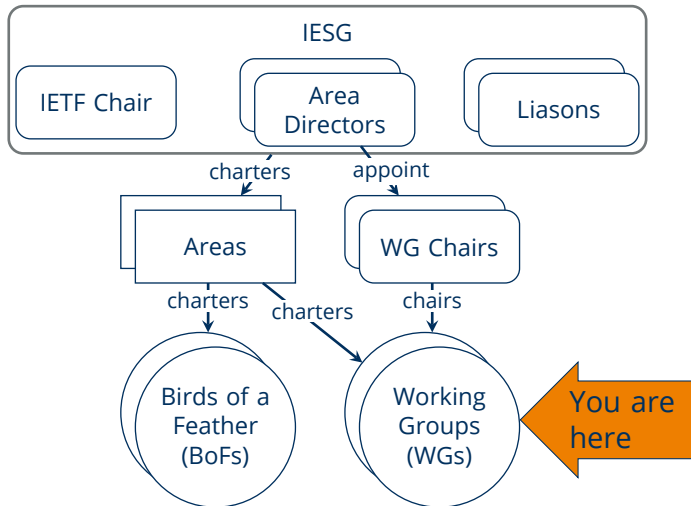
How does the IETF work?



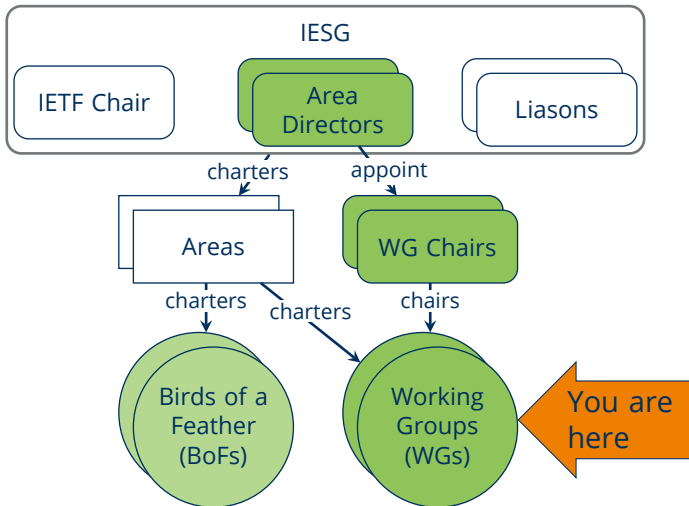
How does the IETF work? The Beginners View



How does the IETF work? The Beginners View



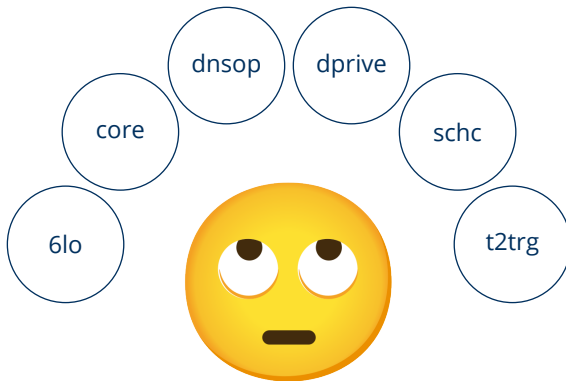
How does the IETF work? The Beginners View



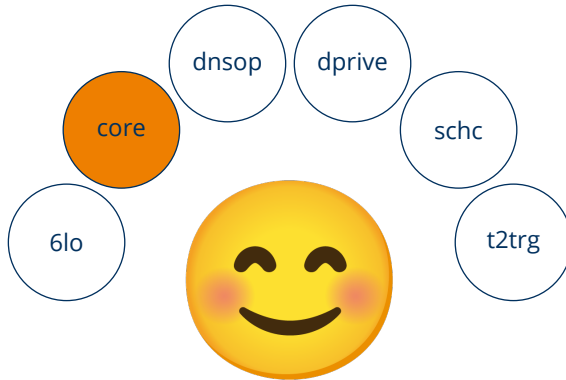
What is an RFC?

- **Request For Comments**
- Publications by the IETF, IRTF, or other related bodies:
 - Internet Standards
 - Informational
 - Experimental
- Start as Internet Draft (by working group or individual)
- Term RFC somewhat of an anachronistic misnomer:
 - Comment stage during draft stage
 - RFCs are not changed just updated (by new RFCs) or amended (by errata)

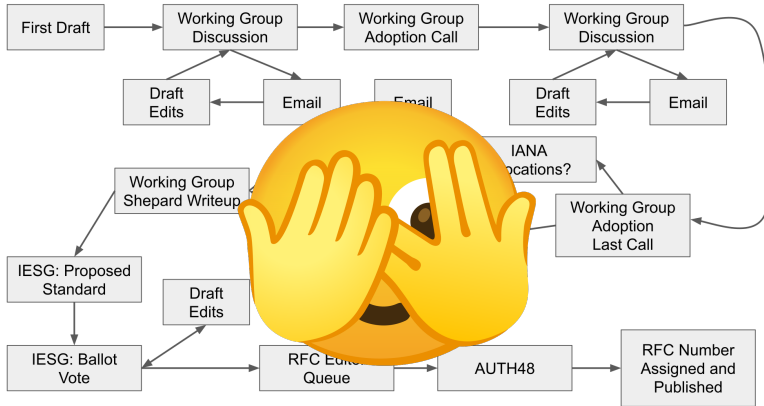
Where to go with DoC?



Where to go with DoC?

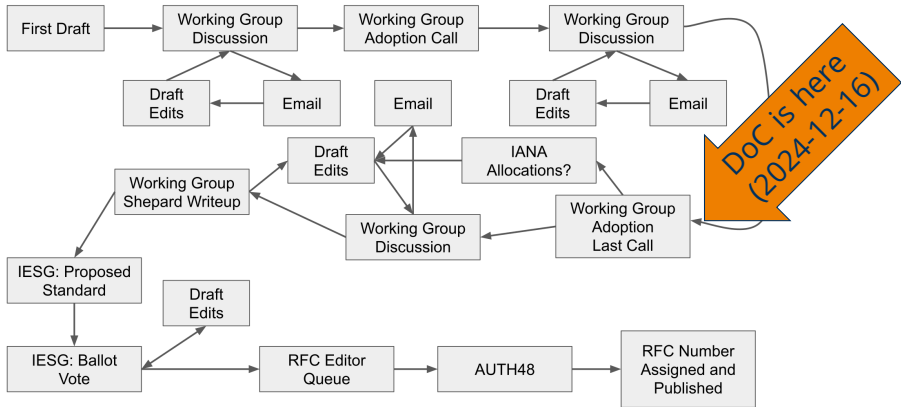


How to write an RFC?



Source: <https://blog.benjojo.co.uk/post/rfc-in-38-simple-steps>

How to write an RFC?



Source: <https://blog.benjojo.co.uk/post/rfc-in-38-simple-steps>

Outline

Motivation

Designing DNS over CoAP

Going to IETF

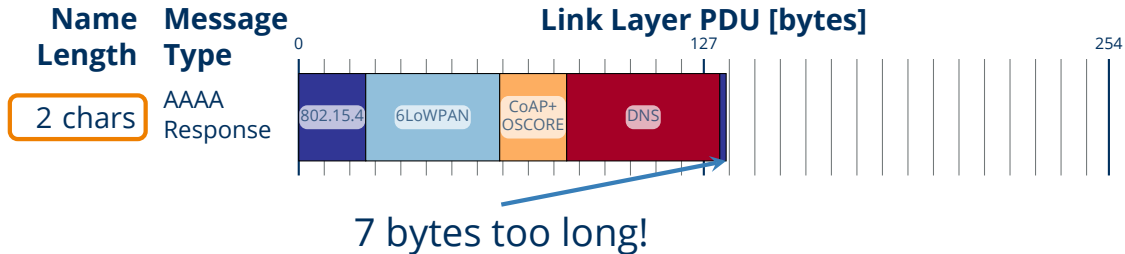
A New Message Format is Needed

The Classic Internet Can Benefit

Conclusion

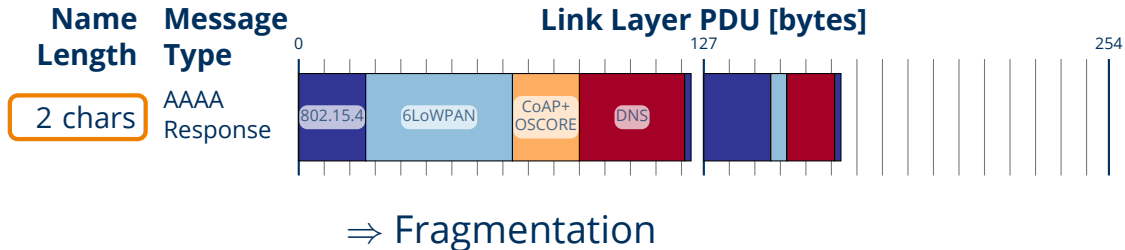
Concise DNS Message Representation

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



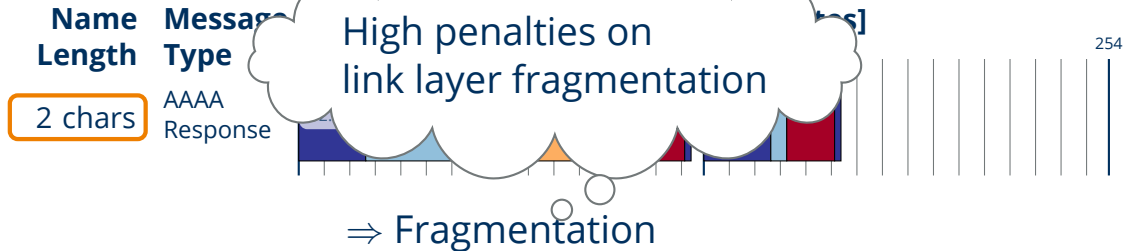
Concise DNS Message Representation

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



Concise DNS Message Representation

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



Concise DNS Message Representation

Concise DNS messages are needed

`application/dns+cbor`

Media Type and Content-Format
(*i.e.*, usable with both DoC and DoH)

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

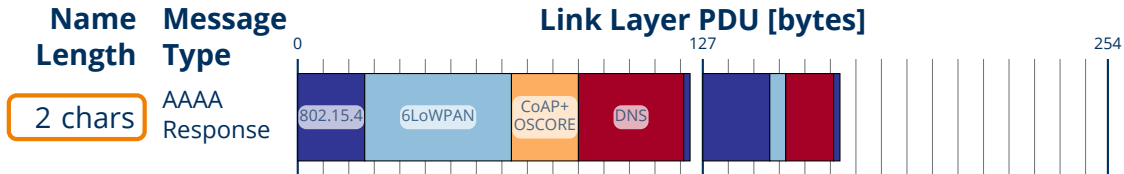
Objectives:

- Concise encoding of DNS messages using existing implementation (CBOR)
- Omits (redundant) DNS fields in DNS queries and responses
- Provides (optional) address and name compression using Packed CBOR

254

Concise DNS Message Representation

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



```
11 ac 80 00 00 01 00 01 00 00 00 02 67 77 00
00 1c 00 01 c0 0c 00 1c 00 01 00 00 0e 10 00 10
20 01 0d b8 00 00 00 00 00 00 00 00 00 00 01
```

Classic DNS

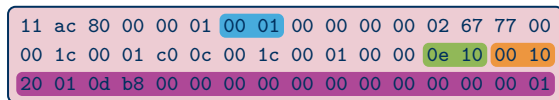
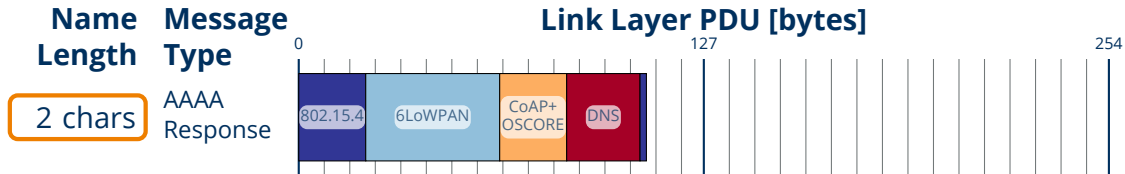


```
81 81 82 19 0e 10 50 20
01 0d b8 00 00 00 00 00
00 00 00 00 00 00 01
```

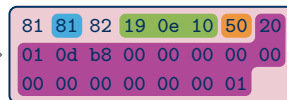
DNS+CBOR

Concise DNS Message Representation

Constrained Networks, e.g., IEEE 82.15.4 with PDU of 127 bytes



Classic DNS



DNS+CBOR

CBOR is smaller than JSON

A simple example, integers:

JSON

12

CBOR is smaller than JSON

A simple example, integers:

JSON

0x3132


12

(2 bytes)

CBOR is smaller than JSON

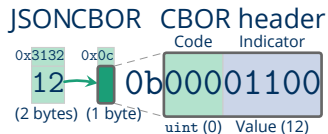
A simple example, integers:

JSON CBOR

0x3132 0x0c
12 
(2 bytes) (1 byte)

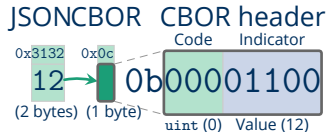
CBOR is smaller than JSON

A simple example, integers:

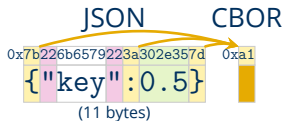


CBOR is smaller than JSON

A simple example, integers:

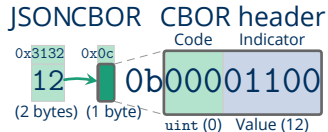


A more advanced example, maps:



CBOR is smaller than JSON

A simple example, integers:

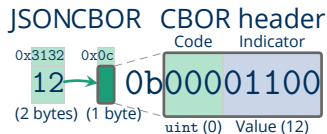


A more advanced example, maps:

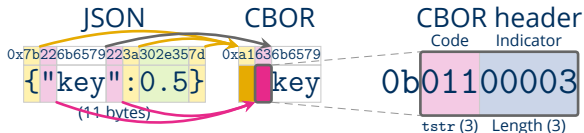


CBOR is smaller than JSON

A simple example, integers:

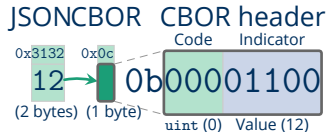


A more advanced example, maps:

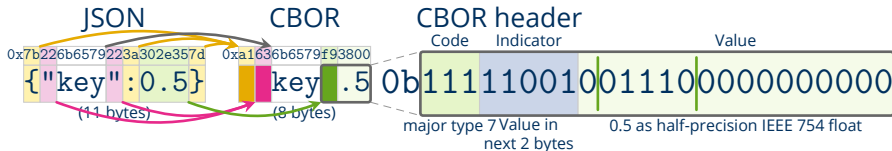


CBOR is smaller than JSON

A simple example, integers:

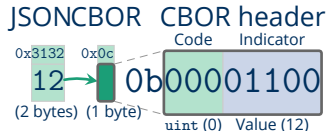


A more advanced example, maps:

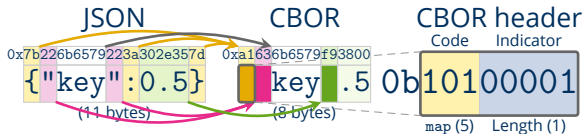


CBOR is smaller than JSON

A simple example, integers:



A more advanced example, maps:



CBOR structure allows for elision

application/dns-message

0 1 2 3
012345678901234567890123456789012

ID		0	Opcode	AA	TC	RD	RA	Z	RC	Rcode
QDCOUNT = 1		ANCOUNT = 0								
NSCOUNT = 0		ARCOUNT = 0								
'\x03'	'w'	'w'	'w'							
'\x07'	'e'	'x'	'a'							
'm'	'p'	'l'	'e'							
'\x02'	'd'	'e'	'\x00'							
Type = A (1)		Class = IN (1)								

application/dns+cbor query

```
[ 0x0000, 0x0000,
  1,      0,
  0,      0,
  [ ["www.example.de",
    1,  1]],
  [],
  [],
  []
]
```

CBOR structure allows for elision

application/dns-message

0 1 2 3
012345678901234567890123456789012

ID		0	Opcode	AA	TC	RD	RA	Z	AR	GC	Rcode
QDCOUNT = 1		ANCOUNT = 0									
NSCOUNT = 0		ARCOUNT = 0									
'\x03'	'w'	'w'	'w'								
'\x07'	'e'	'x'	'a'								
'm'	'p'	'l'	'e'								
'\x02'	'd'	'e'	'\x00'								
Type = A (1)		Class = IN (1)									

(32 bytes)

application/dns+cbor query

(in CBOR diagnostic notation)

```
8a 0x000000, 0x000000,
01, 00,
00, 00,
81836e7777772e6578616d706c652e6465",
01, 01]],
80],
80],
80]
```

(29 bytes)

CBOR structure allows for elision

application/dns-message

0 1 2 3
012345678901234567890123456789012

ID		0	Opcode	AA	TC	RD	RA	Z	AR	RC	Rcode
QDCOUNT = 1		ANCOUNT = 0									
NSCOUNT = 0		ARCOUNT = 0									
'\x03'	'w'	'w'	'w'								
'\x07'	'e'	'x'	'a'								
'm'	'p'	'l'	'e'								
'\x02'	'd'	'e'	'\x00'								
Type = A (1)		Class = IN (1)									

(32 bytes)

application/dns+cbor query

(as hex dump)

```
[ 0x0000, 0x0000, 8a000001
1, 0, 00000081
0, 0, 836e7777
["www.example772e6578
616d706c
652e6465
01018080
1, 1]], 80
],
],
]
```

(29 bytes)

CBOR structure allows for elision

application/dns-message

0 1 2 3
012345678901234567890123456789012

ID		0	Opcode	AA	TC	RD	RA	Z	AR	RC	Rcode
QDCOUNT = 1		ANCOUNT = 0									
NSCOUNT = 0		ARCOUNT = 0									
'\x03'	'w'	'w'	'w'								
'\x07'	'e'	'x'	'a'								
'm'	'p'	'l'	'e'								
'\x02'	'd'	'e'	'\x00'								
Type = A (1)		Class = IN (1)									

(32 bytes)

application/dns+cbor query

```
[ 0x0000, 0x0000,
  1,      0,
  0,      0,
  [ ["www.example.de",
    1,  1]],
  [],
  [],
  []
]
```

(29 bytes)

CBOR structure allows for elision

application/dns-message

0 1 2 3
012345678901234567890123456789012

ID		0	Opcode	AA	TC	RD	RA	Z	AR	GC	Rcode
QDCOUNT = 1		ANCOUNT = 0									
NSCOUNT = 0		ARCOUNT = 0									
'\x03'	'w'	'w'	'w'								
'\x07'	'e'	'x'	'a'								
'm'	'p'	'l'	'e'								
'\x02'	'd'	'e'	'\x00'								
Type = A (1)		Class = IN (1)									

(32 bytes)

application/dns+cbor query

(in CBOR diagnostic notation)

```
[ 0x0000, 0x0000,  
  1, 0,  
  0, 0,  
  8["www.example.de",  
    1, 1]],  
  80],  
  80],  
  80]  
]
```

(29 bytes)

CBOR structure allows for elision

application/dns-message

0 1 2 3
012345678901234567890123456789012

ID		0	Opcode	AA	TC	RD	RA	Z	AR	CG	Rcode
QDCOUNT = 1		ANCOUNT = 0									
NSCOUNT = 0		ARCOUNT = 0									
'\x03'	'w'	'w'	'w'								
'\x07'	'e'	'x'	'a'								
'm'	'p'	'l'	'e'								
'\x02'	'd'	'e'	'\x00'								
Type = A (1)		Class = IN (1)									

(32 bytes)

application/dns+cbor query

(in CBOR diagnostic notation)

[0x0000, 0x0000,

[["www.example.de",

1, 1]],

[],

[],

[]

]

(25 bytes)

CBOR structure allows for elision

application/dns-message

0 1 2 3
012345678901234567890123456789012

ID		0	Opcode	AA	TC	RD	RA	Z	AR	GC	Rcode
QDCOUNT = 1		ANCOUNT = 0									
NSCOUNT = 0		ARCOUNT = 0									
'\x03'	'w'	'w'	'w'								
'\x07'	'e'	'x'	'a'								
'm'	'p'	'l'	'e'								
'\x02'	'd'	'e'	'\x00'								
Type = A (1)		Class = IN (1)									

(32 bytes)

application/dns+cbor query

(in CBOR diagnostic notation)

[0x0000, 0x0000,

[["www.example.de",

1, 1]],
~~1, 1]~~

]

(25 bytes)

CBOR structure allows for elision

application/dns-message

0 1 2 3
012345678901234567890123456789012

ID		0	Opcode	AA	TC	RD	RA	Z	AR	CG	Rcode
QDCOUNT = 1		ANCOUNT = 0									
NSCOUNT = 0		ARCOUNT = 0									
'\x03'	'w'	'w'	'w'								
'\x07'	'e'	'x'	'a'								
'm'	'p'	'l'	'e'								
'\x02'	'd'	'e'	'\x00'								
Type = A (1)		Class = IN (1)									

(32 bytes)

application/dns+cbor query

(in CBOR diagnostic notation)

[0x0000, 0x0000,

[["www.example.de",

1, 1]]

]

(22 bytes)

CBOR structure allows for elision

application/dns-message

0 1 2 3
012345678901234567890123456789012

ID		0	Opcode	AA	TC	RD	RA	Z	AR	CG	Rcode
QDCOUNT = 1		ANCOUNT = 0									
NSCOUNT = 0		ARCOUNT = 0									
'\x03'	'w'	'w'	'w'								
'\x07'	'e'	'x'	'a'								
'm'	'p'	'l'	'e'								
'\x02'	'd'	'e'	'\x00'								
Type = A (1)		Class = IN (1)									

(32 bytes)

application/dns+cbor query

(in CBOR diagnostic notation)

[0x0000, 0x0000,

~~X~~"www.example.de",

1, ~~1X~~]

(22 bytes)

CBOR structure allows for elision

application/dns-message

0 1 2 3
012345678901234567890123456789012

ID		0	Opcode	AA	TC	RD	RA	Z	AR	CG	Rcode
QDCOUNT = 1		ANCOUNT = 0									
NSCOUNT = 0		ARCOUNT = 0									
'\x03'	'w'	'w'	'w'								
'\x07'	'e'	'x'	'a'								
'm'	'p'	'l'	'e'								
'\x02'	'd'	'e'	'\x00'								
Type = A (1)		Class = IN (1)									

(32 bytes)

application/dns+cbor query

(in CBOR diagnostic notation)

[0x0000, 0x0000,

["www.example.de",

1, 1]

]

(21 bytes)

CBOR structure allows for elision

application/dns-message

0										1										2										3																																																											
ID										Opcode										AA										TC										RD										RA										Z										AD										Rcode									
QDCOUNT = 1										ANCOUNT = 0																																																																															
NSCOUNT = 0										ARCOUNT = 0																																																																															
' \x03 '					' w '					' w '					' w '																																																																										
' \x07 '					' e '					' x '					' a '																																																																										
' m '					' p '					' l '					' e '																																																																										
' \x02 '					' d '					' e '					' \x00 '																																																																										
Type = A (1)										Class = IN (1)																																																																															

(32 bytes)

application/dns+cbor query

(in CBOR diagnostic notation)

[~~0x0000~~, 0x0000,

["www.example.de",

1, 1]

]

(21 bytes)

CBOR structure allows for elision

application/dns-message

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	2	3						
ID										0	Opcode	AA	TC	RD	RA	Z	AD	CG	Rcode																				
QDCOUNT = 1										ANCOUNT = 0																													
NSCOUNT = 0										ARCOUNT = 0																													
' \x03 '					' w '					' w '					' w '																								
' \x07 '					' e '					' x '					' a '																								
' m '					' p '					' l '					' e '																								
' \x02 '					' d '					' e '					' \x00 '																								
Type = A (1)										Class = IN (1)																													

(32 bytes)

application/dns+cbor query

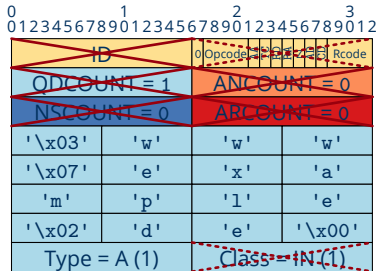
(in CBOR diagnostic notation)

```
[ 0x0000,  
  
  "www.example.de",  
  
  1, 1 ]
```

(20 bytes)

CBOR structure allows for elision

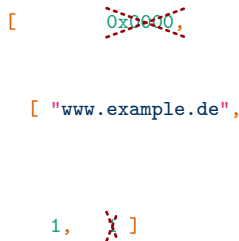
application/dns-message



(32 bytes)

application/dns+cbor query

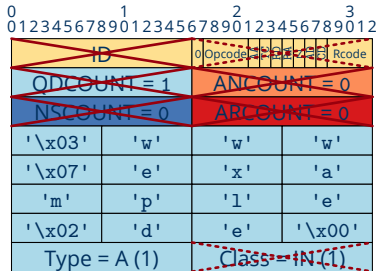
(in CBOR diagnostic notation)



(20 bytes)

CBOR structure allows for elision

application/dns-message



(32 bytes)

application/dns+cbor query

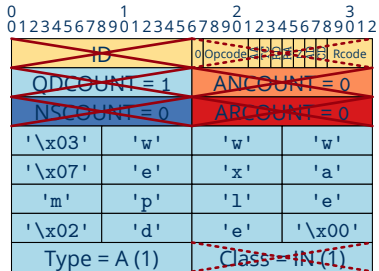
(in CBOR diagnostic notation)

```
[ 0x0000 ,  
  "www.example.de" ,  
  1 , 1 ]
```

(18 bytes)

CBOR structure allows for elision

application/dns-message



(32 bytes)

application/dns+cbor query

(in CBOR diagnostic notation)

```
[
  0x0000,
  "www.example.de",
  1, 1 ]
```

(18 bytes)

CBOR structure allows for elision

application/dns-message

0	1	2	3
012345678901	2345678901	2345678901	2345678901
ID		1 Opcode	
QDCOUNT = 1		ANCOUNT = 1	
NSCOUNT = 0		ARCOUNT = 0	
'\x03'	'w'	'w'	'w'
'\x07'	'e'	'x'	'a'
'm'	'p'	'l'	'e'
'\x02'	'd'	'e'	'\x00'
Type = A (1)	Class = IN (1)		
Offset = 0xc00c	Type = A (1)		
Class = IN (1)	TTL = 300 (MSB)		
TTL = 300 (LSB)	RDLENGTH = 4		
RD = 192.0.2.7			

(48 bytes)

application/dns+cbor response

(in CBOR diagnostic notation)

```
[ 0x8000 ,  
  
  [ "www.example.de" ,  
  
    1, 1 ],  
  [ [ "www.example.de" , 300 , 1 ,  
    1 ,  
    4 ,  
    ip'192.0.2.7' ] ] ]
```

(46 bytes)

CBOR structure allows for elision

application/dns-message

0 1 2 3		0 1 2 3	
ID		1 Opcode 2 Rcode	
QDCOUNT = 1		ANCOUNT = 1	
NSCOUNT = 0		ARCOUNT = 0	
'\x03'	'w'	'w'	'w'
'\x07'	'e'	'x'	'a'
'm'	'p'	'l'	'e'
'\x02'	'd'	'e'	'\x00'
Type = A (1)		Class = IN (1)	
Offset = 0xc00c		Type = A (1)	
Class = IN (1)		TTL = 300 (MSB)	
TTL = 300 (LSB)		RDLENGTH = 4	
RD = 192.0.2.7			

(48 bytes)

application/dns+cbor response

(in CBOR diagnostic notation)

```
[
  0x8000,
  "www.example.de",
  1, 1],
["www.example.de", 300, 1,
1,
44c0000207.7']]
```

(46 bytes)

CBOR structure allows for elision

application/dns-message

0 1 2 3			
012345678901234567890123456789012			
ID		1 Opcode	
QDCOUNT = 1		ANCOUNT = 1	
NSCOUNT = 0		ARCOUNT = 0	
'\x03'	'w'	'w'	'w'
'\x07'	'e'	'x'	'a'
'm'	'p'	'l'	'e'
'\x02'	'd'	'e'	'\x00'
Type = A (1)		Class = IN (1)	
Offset = 0xc00c		Type = A (1)	
Class = IN (1)		TTL = 300 (MSB)	
TTL = 300 (LSB)		RDLENGTH = 4	
RD = 192.0.2.7			

(48 bytes)

application/dns+cbor response

(in CBOR diagnostic notation)

```
[
  0x8000,
  "www.example.de",
  1, 1 ],
["www.example.de", 300, 1,
1,
44c0000207.7']]
```

(46 bytes)

CBOR structure allows for elision

application/dns-message

0 1 2 3			
012345678901234567890123456789012			
ID		1 Opcode	
QDCOUNT = 1		ANCOUNT = 1	
NSCOUNT = 0		ARCOUNT = 0	
'\x03'	'w'	'w'	'w'
'\x07'	'e'	'x'	'a'
'm'	'p'	'l'	'e'
'\x02'	'd'	'e'	'\x00'
Type = A (1)		Class = IN (1)	
Offset = 0xc00c		Type = A (1)	
Class = IN (1)		TTL = 300 (MSB)	
TTL = 300 (LSB)		RDLLENGTH = 4	
RD = 192.0.2.7			

(48 bytes)

application/dns+cbor response

(in CBOR diagnostic notation)

```
[ 0x8000,  
  
  "www.example.de",  
  
  1, 1 ],  
[ "www.example.de", 300, 1,  
  1,  
  ip'192.0.2.7' ] ]
```

(45 bytes)

CBOR structure allows for elision

application/dns-message

0	1	2	3
012345678901	2345678901	2345678901	2345678901
ID		1 Opcode	
QDCOUNT = 1		ANCOUNT = 1	
NSCOUNT = 0		ARCOUNT = 0	
'\x03'	'w'	'w'	'w'
'\x07'	'e'	'x'	'a'
'm'	'p'	'l'	'e'
'\x02'	'd'	'e'	'\x00'
Type = A(1)		Class = IN(1)	
Offset = 0xc00c		Type = A(1)	
Class = IN(1)		TTL = 300 (MSB)	
TTL = 300 (LSB)		RDLLENGTH = 4	
RD = 192.0.2.7			

(48 bytes)

application/dns+cbor response

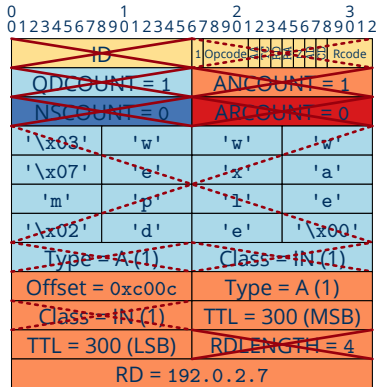
(in CBOR diagnostic notation)

```
[ 0x8000,  
["www.example.de"],  
1, 1 ],  
["www.example.de", 300, 1,  
1,  
ip'192.0.2.7'] ]
```

(45 bytes)

CBOR structure allows for elision

application/dns-message



(48 bytes)

application/dns+cbor response

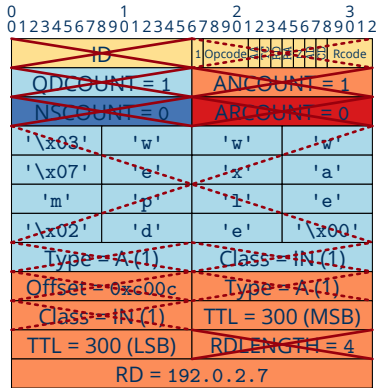
(in CBOR diagnostic notation)

```
[
  0x8000,
  "www.example.de",
  1, 1 ]
[["www.example.de", 300, 1,
  1,
  ip'192.0.2.7']]
]
```

(27 bytes)

CBOR structure allows for elision

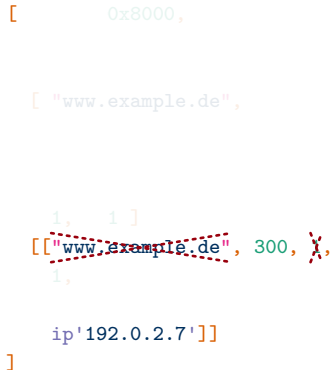
application/dns-message



(48 bytes)

application/dns+cbor response

(in CBOR diagnostic notation)



(27 bytes)

CBOR structure allows for elision

application/dns-message

0	1	2	3
012345678901234567890123456789012			
ID		1 Opcode	
QDCOUNT = 1		ANCOUNT = 1	
NSCOUNT = 0		ARCOUNT = 0	
'\x03'	'w'	'w'	'w'
'\x07'	'e'	'x'	'a'
'm'	'p'	'1'	'e'
'\x02'	'd'	'e'	'\x00'
Type = A(1)		Class = IN(1)	
Offset = 0xc00c		Type = A(1)	
Class = IN(1)		TTL = 300 (MSB)	
TTL = 300 (LSB)		RDLLENGTH = 4	
RD = 192.0.2.7			

(48 bytes)

application/dns+cbor response

(in CBOR diagnostic notation)

```
[ 0x8000,  
  
  ["www.example.de",  
  
   1, 1 ]  
  [ ["www.example.de", 300, 1,  
    1,  
    ip'192.0.2.7' ] ] ]
```

(11 bytes)

CBOR structure allows for elision

application/dns-message

application/dns+cbor response

(in CBOR diagnostic notation)

- Query: `[["www.example.de", 1]]`
(`81 82 6e 7777772e6578616d706c652e6465 01`)
(32 bytes \mapsto 18 bytes)
- Response: `[[[300, ip'192.0.2.7']]]`
(`81 81 82 19 012c 44 c0000207`)
(48 bytes \mapsto 11 bytes)

(48 bytes) (11 bytes)

Packed CBOR allows for value compression

application/dns+cbor response

```
[  
  [ ["www.example.org", 3600, 5, "example.org"] ],  
  [ ["example.org", 3600, 2, "ns1.example.org"],  
    ["example.org", 3600, 2, "ns2.example.org"] ],  
  [ ["example.org", 3600, 1, h'c0002563' ],  
    ["ns1.example.org", 3600, 1, h'c00021ab' ],  
    ["ns1.example.org", 3600, 28, h'20010db8194742a00000000000000000ab' ],  
    ["ns2.example.org", 3600, 1, h'c0002118' ],  
    ["ns2.example.org", 3600, 28, h'20010db8194742a0000000000000000018' ] ]  
]
```

(255 bytes)

Packed CBOR allows for value compression

application/dns+cbor;**packed=1** response

```
[
  "example.org", 3600, 216("."), 218("ns1"), 218("ns2"),
  h'c000', h'20010db8194742a00000000000000000', 229(h'21')],
  [
    [[218("www"), simple(1), 5, simple(0) ]],
    [[simple(0), simple(1), 2, simple(3) ],
     [simple(0), simple(1), 2, simple(4) ]],
    [[simple(0), simple(1), 1, 229(h'2563')],
     [simple(3), simple(1), 1, 231(h'ab')]],
     [simple(3), simple(1), 28, 230(h'ab')]],
     [simple(4), simple(1), 1, 231(h'18')]],
     [simple(4), simple(1), 28, 230(h'18')]]
  ]
]
```

(255 bytes)

Packed CBOR allows for value compression

application/dns+cbor;packed=1 response

```
[
  ["example.org", 3290, 216(" "), 218("ns1"), 218("ns2"),
   h'c000', h'20010db8194742a00000000000000000', 229(h'21')],
  [
    [[218("www"), simple(1), 5, simple(0) ]],
    [[simple(0), simple(1), 2, simple(3) ]],
    [simple(0), simple(1), 2, simple(4) ]],
    [[simple(0), simple(1), 1, 229(h'21')],
     (references packing table to yield reconstructed data item)
     [simple(3), simple(1), 28, 230(h'ab')],
     [simple(4), simple(1), 1, 231(h'18')],
     [simple(4), simple(1), 28, 230(h'18')]]
  ]
]
```

(255 bytes)

Packed CBOR allows for value compression

application/dns+cbor;packed=1 response

```
[
  tag (6)
  ["example.org", 3600, d8d8"."), 218("ns1"), 218("ns2"),
  h'c000', h'20010db8194742a00000000000000000', 229(h'21')],
  [
    [[218("www"), simple(1), 5, simple(0) ]],
    [[simple(0), simple(1), 2, simple(3) ]],
    [simple(0), simple(1), 2, simple(4) ]],
    [[simple(0), simple(1), 1, 229(h'2563')],
     [simple(3), simple(1), 1, 231(h'ab')],
     [simple(3), simple(1), 28, 230(h'ab')],
     [simple(4), simple(1), 1, 231(h'18')],
     [simple(4), simple(1), 28, 230(h'18')]]
  ]
]
```

(255 bytes)

Packed CBOR allows for value compression

application/dns+cbor;packed=1 response

Suffix Reference	
Tag	Idx
216	0
217	1
218	2
219	3
...	

```
[  
  "example.org", 3600, 216("."), 218("ns1"), 218("ns2"),  
  h'c000', h'20010db8194742a00000000000000000', 229(h'21')],  
 [  
   [[218("www"), simple(1), 5, simple(0) ]],  
   [[simple(0), simple(1), 2, simple(3) ]],  
   [simple(0), simple(1), 2, simple(4) ]],  
   [[simple(0), simple(1), 1, 229(h'2563')],  
    [simple(3), simple(1), 1, 231(h'ab')],  
    [simple(3), simple(1), 28, 230(h'ab')],  
    [simple(4), simple(1), 1, 231(h'18')],  
    [simple(4), simple(1), 28, 230(h'18')]]  
 ]  
 ]
```

(255 bytes)

Packed CBOR allows for value compression

application/dns+cbor;packed=1 response

Suffix Reference	
Tag	Idx
216	0
217	1
218	2
219	3
...	

```
[  
  [ "example.org", 3600, 216("."), 218("ns1"), 218("ns2"),  
    h'c000', h'20010db8194742a00000000000000000', 229(h'21') ],  
  [  
    [[ 218("www"), simple(1), 5, simple(0) ]],  
    [[ simple(0), simple(1), 2, simple(3) ]],  
    [ simple(0), simple(1), 2, simple(4) ]],  
    [[ simple(0), simple(1), 1, 229(h'2563') ],  
     [ simple(3), simple(1), 1, 231(h'ab') ],  
     [ simple(3), simple(1), 28, 230(h'ab') ],  
     [ simple(4), simple(1), 1, 231(h'18') ],  
     [ simple(4), simple(1), 28, 230(h'18') ] ] ] ]
```

(255 bytes)

Packed CBOR allows for value compression

application/dns+cbor;packed=1 response

Suffix Reference	
Tag	Idx
216	0
217	1
218	2
219	3
...	

```
[  
  "example.org", 3600, 216("."), 218("ns1"), 218("ns2"),  
  h'c000', h'20010db8194742a00000000000000000', 229(h'21')],  
  [  
    [[218("www"), simple(1), 5, simple(0) ]],  
    [[simple(0), simple(1), 2, simple(3) ]],  
    [simple(0), simple(1), 2, simple(4) ]],  
    [[simple(0), simple(1), 1, 229(h'2563')],  
     [simple(3), simple(1), 1, 231(h'ab')],  
     [simple(3), simple(1), 28, 230(h'ab')],  
     [simple(4), simple(1), 1, 231(h'18')],  
     [simple(4), simple(1), 28, 230(h'18')]]  
  ]  
]
```

Prefix Reference	
Tag	Idx
224	0
225	1
226	2
227	3
228	4
229	5
230	6
231	7

(255 bytes)

Packed CBOR allows for value compression

application/dns+cbor;packed=1 response

Suffix Reference	
Tag	Idx
216	0
217	1
218	2
219	3
...	

```
[  
  ["example.org", 3600, 216("."), 218("ns1"), 218("ns2"),  
   h'c000', h'20010db8194742a00000000000000000', 229(h'21')],  
  [↑──────────────────────────────────────────────────────────────────────────┐  
   h'c00021'  
   └──────────────────────────────────────────────────────────────────────────┘  
   [[218("www"), simple(1), 5, simple(0) ]],  
   [[simple(0), simple(1), 2, simple(3) ]],  
   [simple(0), simple(1), 2, simple(4) ]],  
   [[simple(0), simple(1), 1, 229(h'2563')],  
    [simple(3), simple(1), 1, 231(h'ab')],  
    [simple(3), simple(1), 28, 230(h'ab')],  
    [simple(4), simple(1), 1, 231(h'18')],  
    [simple(4), simple(1), 28, 230(h'18')]]  
]
```

Prefix Reference	
Tag	Idx
224	0
225	1
226	2
227	3
228	4
229	5
230	6
231	7

(255 bytes)

Packed CBOR allows for value compression

application/dns+cbor;packed=1 response

Suffix Reference	
Tag	Idx
216	0
217	1
218	2
219	3
...	

```
[  
  "example.org", 3600, 216("."), 218("ns1"), 218("ns2"),  
  h'c000', h'20010db8194742a00000000000000000', 229(h'21')],  
 [  
  [[218("www"), simple1(1), 5, simple0(0) ]],  
  [[simple0(0), simple1(1), 2, simple3(3) ]],  
  [simple0(0), simple1(1), 2, simple4(4) ]],  
  [[simple0(0), simple1(1), 1, 229(h'2563')],  
   [simple3(3), simple1(1), 1, 231(h'ab')],  
   [simple3(3), simple1(1), 28, 230(h'ab')],  
   [simple4(4), simple1(1), 1, 231(h'18')],  
   [simple4(4), simple1(1), 28, 230(h'18')]]  
 ]  
 ]
```

Prefix Reference	
Tag	Idx
224	0
225	1
226	2
227	3
228	4
229	5
230	6
231	7

(255 bytes)

Packed CBOR allows for value compression

application/dns+cbor;packed=1 response

Suffix
Reference

Tag	Idx
216	0
217	1
218	2
219	3
...	

Value
Reference
with
Simple
Values

```
[  
  ["example.org", 3600, 216("."), 218("ns1"), 218("ns2"),  
   h'c000', h'20010db8194742a00000000000000000'], 229(h'21')],  
  [  
    [[218("www"), simple1(1), 5, simple0(0) ]],  
    [[simple0(0), simple1(1), 2, simple3(3) ]],  
    [simple0(0), simple1(1), 2, simple4(4) ]],  
    [[simple0(0), simple1(1), 1, 229(h'2563')],  
     [simple3(3), simple1(1), 1, 231(h'ab')],  
     [simple3(3), simple1(1), 28, 230(h'ab')],  
     [simple4(4), simple1(1), 1, 231(h'18')],  
     [simple4(4), simple1(1), 28, 230(h'18')]]  
  ]  
]
```

(255 bytes)

Prefix
Reference

Tag	Idx
224	0
225	1
226	2
227	3
228	4
229	5
230	6
231	7

Packed CBOR allows for value compression

application/dns+cbor;packed=1 response

Suffix
Reference

Tag	Idx
216	0
217	1
218	2
219	3
...	

Value
Reference
with
Simple
Values

```
[  
  ["example.org", 3600, 216("."), 218("ns1"), 218("ns2"),  
   h'c000', h'20010db8194742a00000000000000000'],  
  [  
    [[218("www"), simple(1), 5, simple(0) ]],  
    [[simple(0), simple(1), 2, simple(3) ]],  
    [simple(0), simple(1), 2, simple(4) ]],  
    [[simple(0), simple(1), 1, 229(h'2563')],  
     [simple(3), simple(1), 1, 231(h'ab')],  
     [simple(3), simple(1), 28, 230(h'ab')],  
     [simple(4), simple(1), 1, 231(h'18')],  
     [simple(4), simple(1), 28, 230(h'18')]]  
  ]  
]
```

(255 bytes)

Prefix
Reference

Tag	Idx
224	0
225	1
226	2
227	3
228	4
229	5
230	6
231	7

Packed CBOR allows for value compression

application/dns+cbor;packed=1 response

Suffix
Reference

Tag	Idx
216	0
217	1
218	2
219	3
...	

Value
Reference
with
Simple
Values

```
[  
  "example.org", 3600, 216("."), 218("ns1"), 218("ns2"),  
  h'c000', h'20010db8194742a000000000000000', 229(h'21')],  
  [  
    [[218("www"), simple(1), 5, simple(0) ]],  
    [[simple(0), simple(1), 2, simple(3) ]],  
    [simple(0), simple(1), 2, simple(4) ]],  
    [[simple(0), simple(1), 1, 229(h'2563')],  
     [simple(3), simple(1), 1, 231(h'ab')],  
     [simple(3), simple(1), 28, 230(h'ab')],  
     [simple(4), simple(1), 1, 231(h'18')],  
     [simple(4), simple(1), 28, 230(h'18')]]  
  ]  
]
```

Prefix
Reference

Tag	Idx
224	0
225	1
226	2
227	3
228	4
229	5
230	6
231	7

(119 bytes)

Packed CBOR allows for value compression

application/dns+cbor;packed=1 response

- Packed CBOR removes redundancies
(255 bytes \mapsto 119 bytes)

- Drawbacks:

No in-line streaming

– Whole object needs to be known

Complex state handling

– Remember values
– Sort by size and occurrences

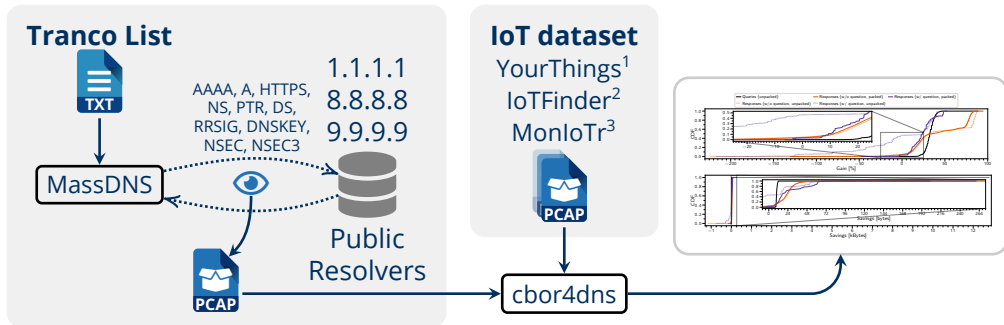
Not universal

– Packing table adds overhead even if empty

Values

(119 bytes)

CBOR as DNS Message Format: Method

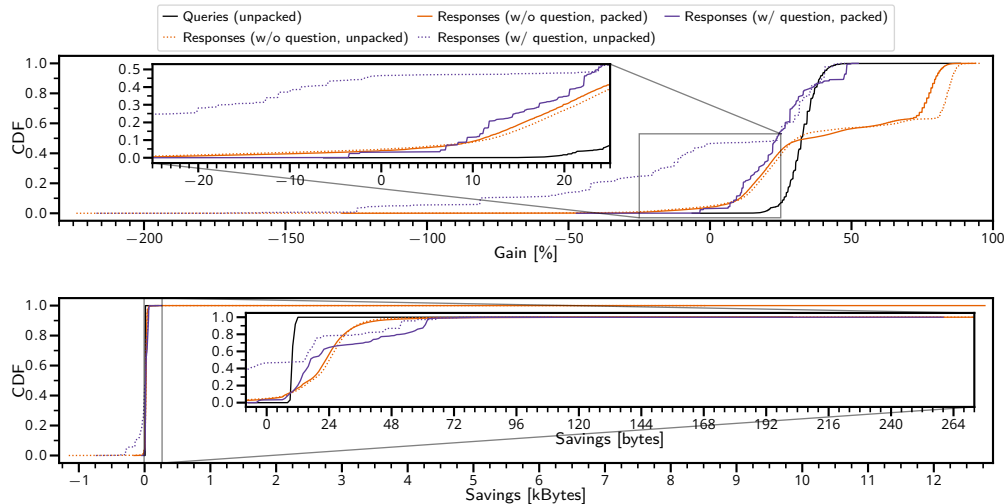


¹O. Alrawi, C. Lever, M. Antonakakis, and F. Monrose. 2019. **SoK: Security Evaluation of Home-Based IoT Deployments**. In *IEEE S&P 2019*. 1362–1380.

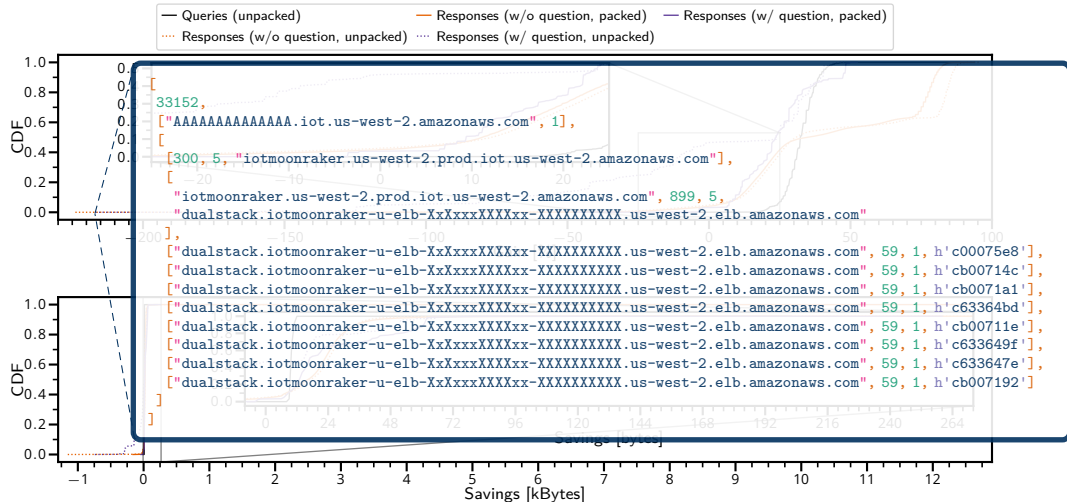
²R. Perdisci, T. Papastergiou, O. Alrawi, and M. Antonakakis. 2020. **IoTFinder: Efficient Large-Scale Identification of IoT Devices via Passive DNS Traffic Analysis**. In *IEEE EuroS&P 2020*. 474–489.

³J. Ren, D.J. Dubois, D. Choffnes, A.M. Mandalari, R. Kolcun, and H. Haddadi. 2019. **Information Exposure for Consumer IoT Devices: A Multidimensional, Network-Informed Measurement Approach**. In *Proc. of the Internet Measurement Conference (IMC)*. ACM.

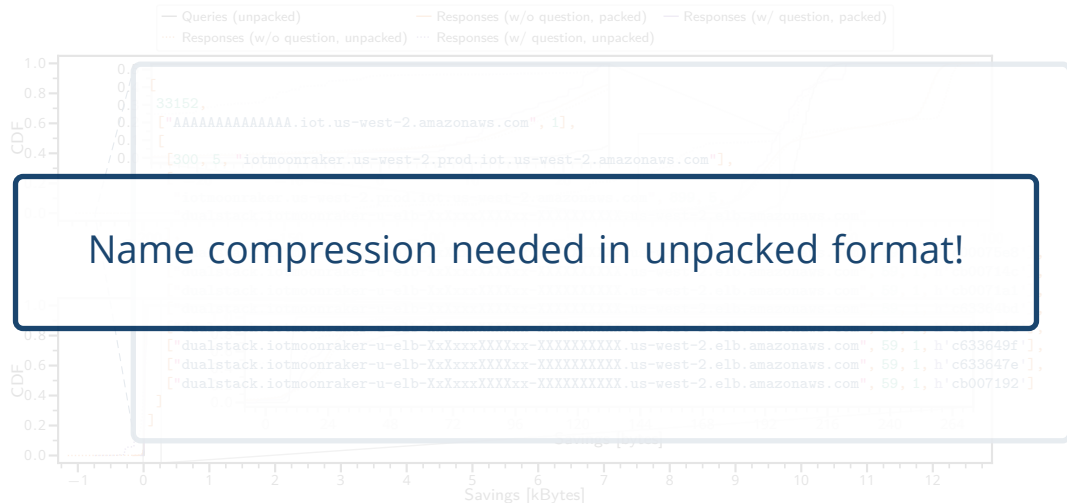
Gain and Byte Savings CBOR-encoded DNS



Gain and Byte Savings CBOR-encoded DNS



Gain and Byte Savings CBOR-encoded DNS



Two Proposals for Name Compression

1. Packed Lite
2. Name Component Referencing

Packed Lite

```
[  
  [ ["www.example.org", 3600, 5, "example.org"] ],  
  [ ["example.org", 3600, 2, "ns1.example.org"],  
    ["example.org", 3600, 2, "ns2.example.org"] ],  
  [ ["example.org", 3600, 1, h'c0002563' ],  
    ["ns1.example.org", 3600, 1, h'c00021ab' ],  
    ["ns1.example.org", 3600, 28, h'20010db8194742a0000000000000000ab' ],  
    ["ns2.example.org", 3600, 1, h'c0002118' ],  
    ["ns2.example.org", 3600, 28, h'20010db8194742a000000000000000018' ] ]  
]
```

Packed Lite

```
[  
  ["example.org", 3600, 216("."), 218("ns1"), 218("ns2"),  
   h'c000', h'20010db8194742a00000000000000000', 229(h'21')],  
  [  
    [[218("www"), simple(1), 5, simple(0) ]],  
    [[simple(0), simple(1), 2, simple(3) ]],  
    [simple(0), simple(1), 2, simple(4) ]],  
    [[simple(0), simple(1), 1, 229(h'2563')],  
     [simple(3), simple(1), 1, 231(h'ab')],  
     [simple(3), simple(1), 28, 230(h'ab')],  
     [simple(4), simple(1), 1, 231(h'18')],  
     [simple(4), simple(1), 28, 230(h'18')]]  
  ]  
]
```

Packed Lite

Only put strings and references to strings in packing table

```
[  
  ["example.org",      216("."), 217("ns1"), 217("ns2")],  
  
  [  
    [[217("www"),      3600, 5, simple(0)    ]],  
    [[simple(0),        3600, 2, simple(2)    ]],  
    [simple(0),        3600, 2, simple(3)    ]],  
    [[simple(0),        3600, 1, h'c0002563' ]],  
    [simple(2),        3600, 1, h'c00021ab' ]],  
    [simple(2),        3600, 28, h'20010db8194742a00000000000000000ab' ],  
    [simple(3),        3600, 1, h'c0002118' ]],  
    [simple(3),        3600, 28, h'20010db8194742a0000000000000000018' ]]  
  ]  
]
```

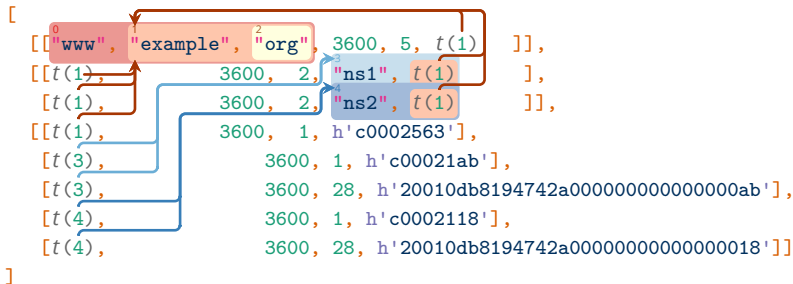
Name Component Referencing

```
[  
  [ "www.example.org", 3600, 5, "example.org" ],  
  [ "example.org", 3600, 2, "ns1.example.org" ],  
  [ "example.org", 3600, 2, "ns2.example.org" ],  
  [ "example.org", 3600, 1, h'c0002563' ],  
  [ "ns1.example.org", 3600, 1, h'c00021ab' ],  
  [ "ns1.example.org", 3600, 28, h'20010db8194742a000000000000000ab' ],  
  [ "ns2.example.org", 3600, 1, h'c0002118' ],  
  [ "ns2.example.org", 3600, 28, h'20010db8194742a00000000000000018' ]  
]
```

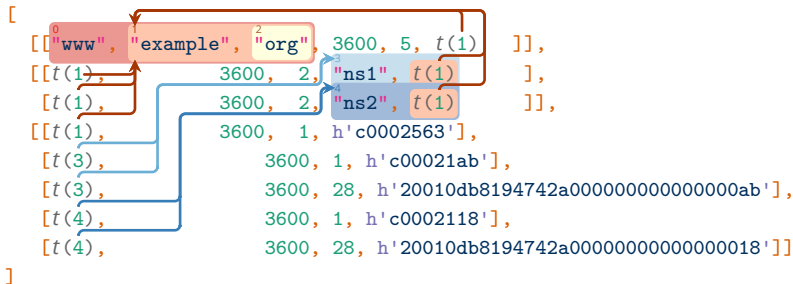
Name Component Referencing

```
[  
  [ "www", "example", "org", 3600, 5, t(1) ],  
  [ t(1), 3600, 2, "ns1", t(1) ],  
  [ t(1), 3600, 2, "ns2", t(1) ],  
  [ t(1), 3600, 1, h'c0002563' ],  
  [ t(3), 3600, 1, h'c00021ab' ],  
  [ t(3), 3600, 28, h'20010db8194742a000000000000000ab' ],  
  [ t(4), 3600, 1, h'c0002118' ],  
  [ t(4), 3600, 28, h'20010db8194742a00000000000000018' ]  
]
```

Name Component Referencing



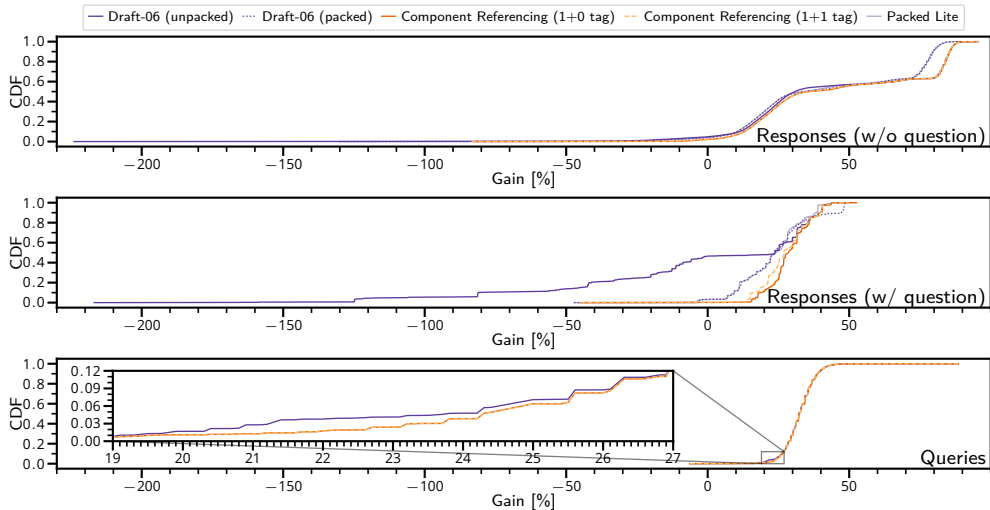
Name Component Referencing



For our evaluation:

- $t = 7$ (c7 \Rightarrow 1+0 (bytes) tag)
- $t = 48$ (d8 30 \Rightarrow 1+1 (bytes) tag)

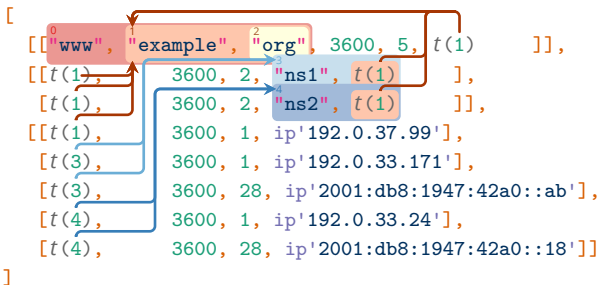
Gain with Name Compression



Component Referencing with Packed CBOR

New idea spun out of IETF 121 (Nov 2024) discussions:

- $t = 7$ (highly contended number allocation) **versus**
 $t = 48$ (larger than DNS name compression)
- Could we not just use `simple(i)` of Packed (*i.e.*, only 1 byte per reference)?



Component Referencing with Packed CBOR

New idea spun out of IETF 121 (Nov 2024) discussions:

- $t = 7$ (highly contended number allocation) **versus**
 $t = 48$ (larger than DNS name compression)
- Could we not just use `simple(i)` of Packed (*i.e.*, only 1 byte per reference)?

```
[  
  [0, "www", 1, "example", 2, "org", 3600, 5, simple(1)],  
  [simple(1), 3600, 2, "ns1", simple(1)],  
  [simple(1), 3600, 2, "ns2", simple(1)],  
  [simple(1), 3600, 1, ip'192.0.37.99'],  
  [simple(3), 3600, 1, ip'192.0.33.171'],  
  [simple(3), 3600, 28, ip'2001:db8:1947:42a0::ab'],  
  [simple(4), 3600, 1, ip'192.0.33.24'],  
  [simple(4), 3600, 28, ip'2001:db8:1947:42a0::18']  
]
```

```
[  
  [0, "www", "example", "org"],  
  [1, "example", "org"],  
  [2, "org"],  
  [3, "ns1", simple(1)],  
  [4, "ns2", simple(1)]  
]  
"Virtual" Packing Table
```

Component Referencing with Packed CBOR

New idea spun out of IETF 121 (Nov 2024) discussions:

- $t = 7$ (highly contended number allocation) **versus**

$t = 48$ (larger than DNS name compression)

Could we not just use `simple(i)` of Packed (i.e. only 1 byte per reference)?

Physical attendance at IETF 121

Discussions over coffee with colleagues

Sharpening of ideas

"Virtual" Packing Table

Outline

Motivation

Designing DNS over CoAP

Going to IETF

A New Message Format is Needed

The Classic Internet Can Benefit

Conclusion

Access to the Web Requires ...

Fast Internet connectivity

+

High end devices

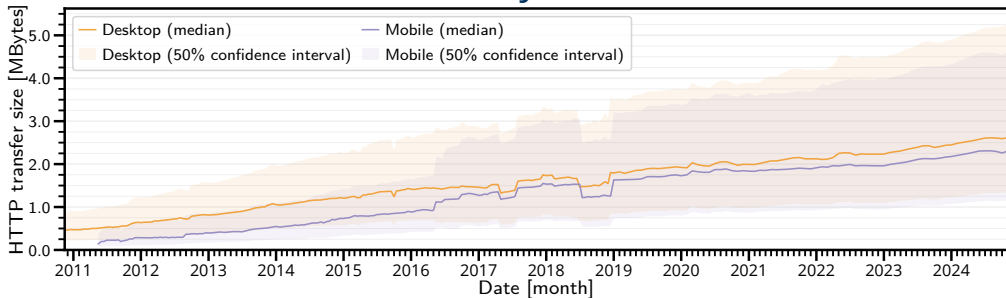
Access to the Web Requires ...

Fast Internet connectivity

+

High end devices

Why?



Source: <https://httparchive.org/reports/state-of-the-web>

Modern Web Fosters Digital Inequality

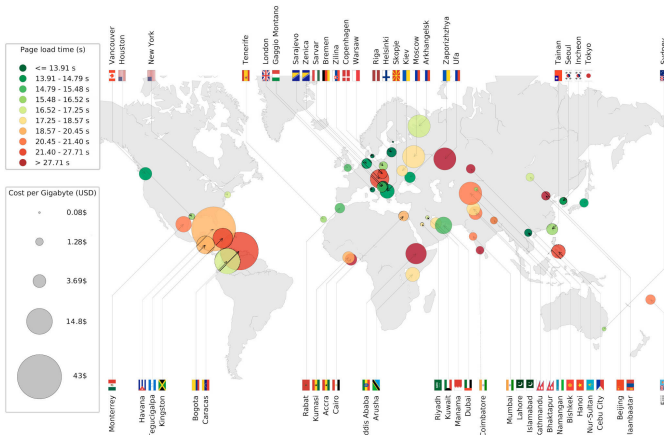


Figure source: M. Chaqfeh, *et al.*. 2023. **Towards a World Wide Web without digital inequality.** <https://doi.org/10.1073/pnas.2212649120>

How to Decrease Latency?

Get object sizes smaller

JSON contributes growing part (4% in Sep 2024) of transfer size

Get message sizes smaller

DNS poses a latency bottleneck to the Web

How to Decrease Latency?

Learn from constrained IoT?

Reminder:

- CBOR is smaller than JSON
- CBOR allows for elision of message fields
- Packed CBOR allows for value compression

“CBORify” JSON, CSS, and HTML as well?

Outline

Motivation

Designing DNS over CoAP

Going to IETF

A New Message Format is Needed

The Classic Internet Can Benefit

Conclusion

Conclusion

- DNS over CoAP (DoC) allows for encrypted DNS in constrained environments
 - Two security modes: DTLS and OSCORE
 - Query via FETCH method: En-route caching at CoAP proxies possible
- Standardized within the core working group of the IETF
- Inspired new DNS message format based on CBOR (ongoing work)
 - Name compression needed: Packed CBOR with virtual Packing Table provides DNS-like mechanism
- The “big Internet” can benefit from CBOR
 - “CBORify” JSON, CSS, and HTML?