Lithe: Lightweight Secure CoAP for the Internet of Things

S. Raza, H. Shafagh, etc. IEEE Sensors 2013, Volume 13

Mahmoud Kalash

28 March 2016

Summary:

- IEEE Sensors journal 2013.
- Security problem in IOT.
- Secure communication protocol in in resource-constrained IOT environments.
- Implementation and evaluation.

Outline

- Introduction
- Background
  - CoAP and DTLS
  - 6LoWPAN
- DTLS Compression
  - DTLS-6LoWPAN Integration
  - 6LoWPAN-NHC for the Record and Handshake Headers
  - 6LoWPAN-NHC for ClientHello / ServerHello
  - 6LoWPAN-NHC for other Handshake Messages
- Implementation
- Evaluation
  - Packet Size Reduction
  - RAM and ROM Requirement
  - Run-Time Performance
- Future work

Outline

- **Introduction**
- Background
  - CoAP and DTLS
  - 6LoWPAN
- DTLS Compression
  - DTLS-6LoWPAN Integration
  - 6LoWPAN-NHC for the Record and Handshake Headers
  - 6LoWPAN-NHC for ClientHello / ServerHello
  - 6LoWPAN-NHC for other Handshake Messages
- Implementation
- Evaluation
  - Packet Size Reduction
  - RAM and ROM Requirement
  - Run-Time Performance
- Future work

Introduction

- **6LoWPAN** (IPv6 over Low power Wireless Personal Area Network) enables IPv6 in low-power and lossy wireless networks such as WSNs.
  - 6LoWPAN defines *header compression mechanisms.*
- HTTP is inefficient in lossy and constrained IOT environment(Low power radios).
- The Internet Engineering Task Force (IETF®).
- **CoAP** (Constrained Application Protocol)
  - Simplicity.
  - Low overhead.
  - Multicast support.

Introduction

- ***DTLS*** (Datagram Transport Layer Security) is used by CoAP as the security protocol
  - key management.
  - data encryption.
  - integrity protection.

- ***CoAPs*** is CoAP with DTLS support, similar to HTTPs.
  - **Problem**: DTLS is inefficient or constrained IOT devices.
  - **Solution:** Apply the 6LoWPAN header compression mechanisms to compress DTLS header.

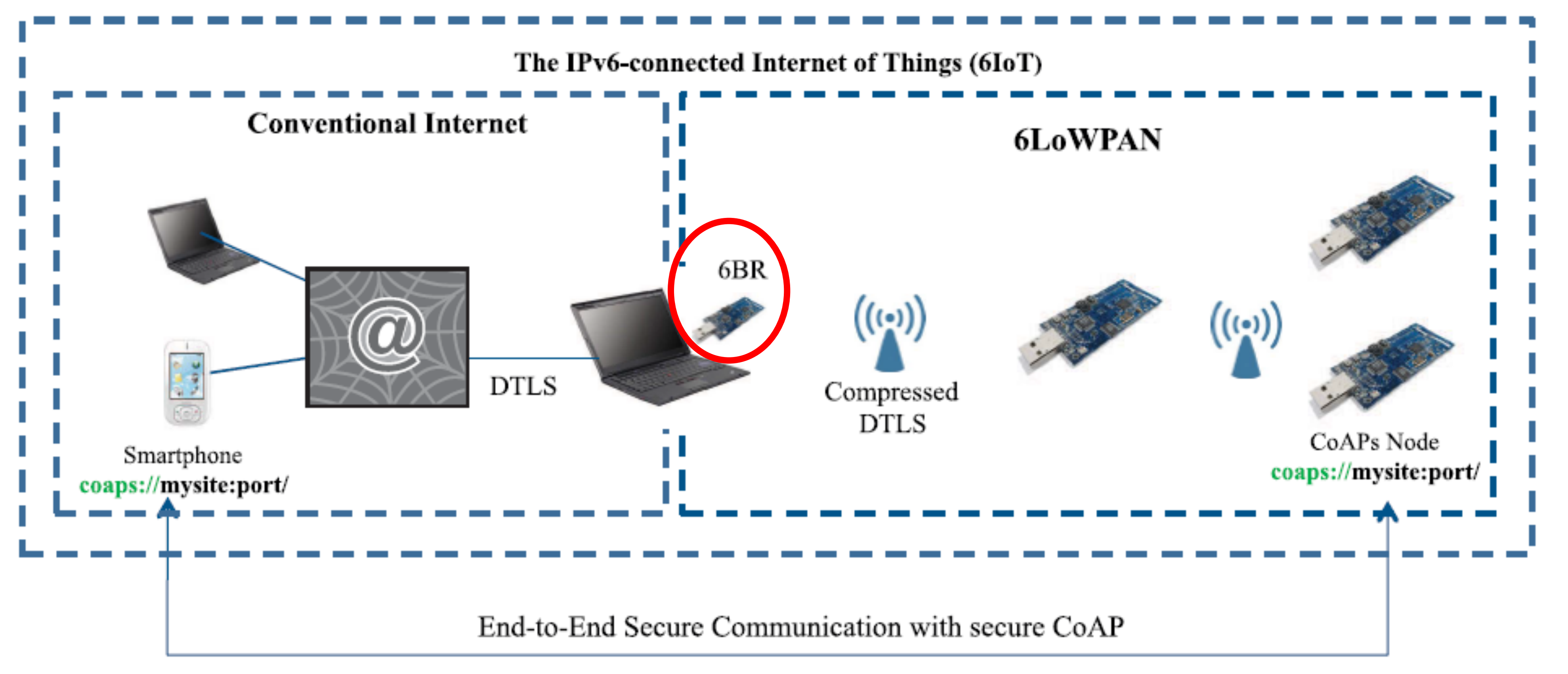
Introduction: Lithe

- **Lithe** is the proposed solution in this paper.

- **Lithe:** a lightweight CoAPs by compressing the DTLS protocol with 6LoWPAN header compression mechanisms.
  - To achieve energy efficiency by reducing the message size;
  - To avoid 6LoWPAN fragmentation as 6LoWPAN protocol is vulnerable to fragmentation attacks.

E2E Communication with CoAPs

- **6BR:** 6LoWPAN Border Router is used between 6LoWPAN networks and the Internet to compress/decompress or/and fragment/reassemble messages before forwarding between the two realms.



Outline

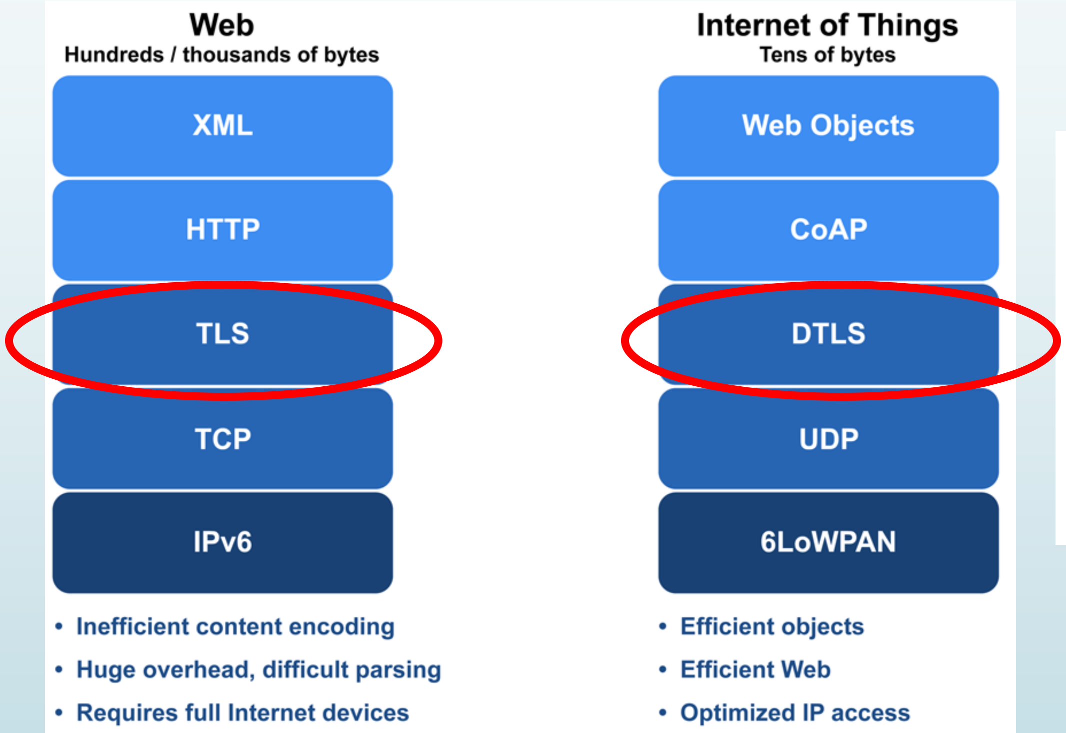
- Introduction
- **Background**
  - CoAP and DTLS
  - 6LoWPAN
- DTLS Compression
  - DTLS-6LoWPAN Integration
  - 6LoWPAN-NHC for the Record and Handshake Headers
  - 6LoWPAN-NHC for ClientHello / ServerHello
  - 6LoWPAN-NHC for other Handshake Messages
- Implementation
- Evaluation
  - Packet Size Reduction
  - RAM and ROM Requirement
  - Run-Time Performance
- Future work

Background

- **Goal**:
- To enable secure and efficient communication among IoT devices that utilize the CoAP protocol.

CoAP

- CoAP is a web protocol that runs over the UDP for IOT.
- Datagram Transport Layer Security (DTLS) is used to protect CoAP transmission.
- Similar to HTTPs (TLS-secured HTTP), CoAPs is DTLS-secured CoAP.
- **Coaps://myIPv6Address:port/MyResource**



Web
Hundreds / thousands of bytes

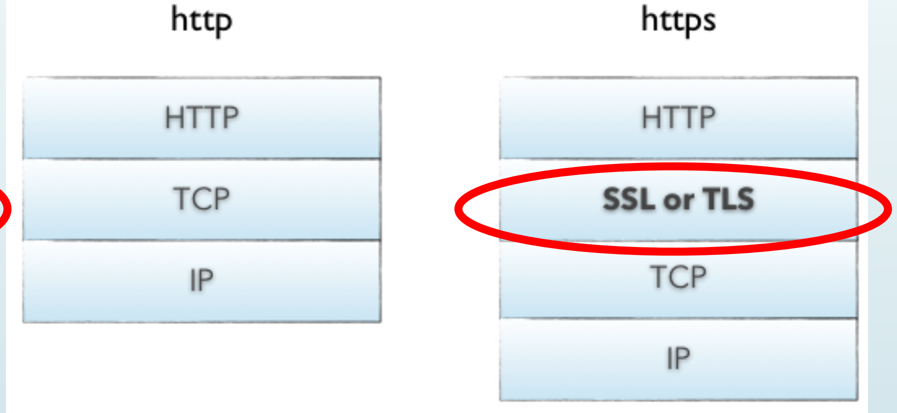
XML
HTTP
TLS
TCP
IPv6

- Inefficient content encoding
- Huge overhead, difficult parsing
- Requires full Internet devices

Internet of Things
Tens of bytes

Web Objects
CoAP
DTLS
UDP
6LoWPAN

- Efficient objects
- Efficient Web
- Optimized IP access



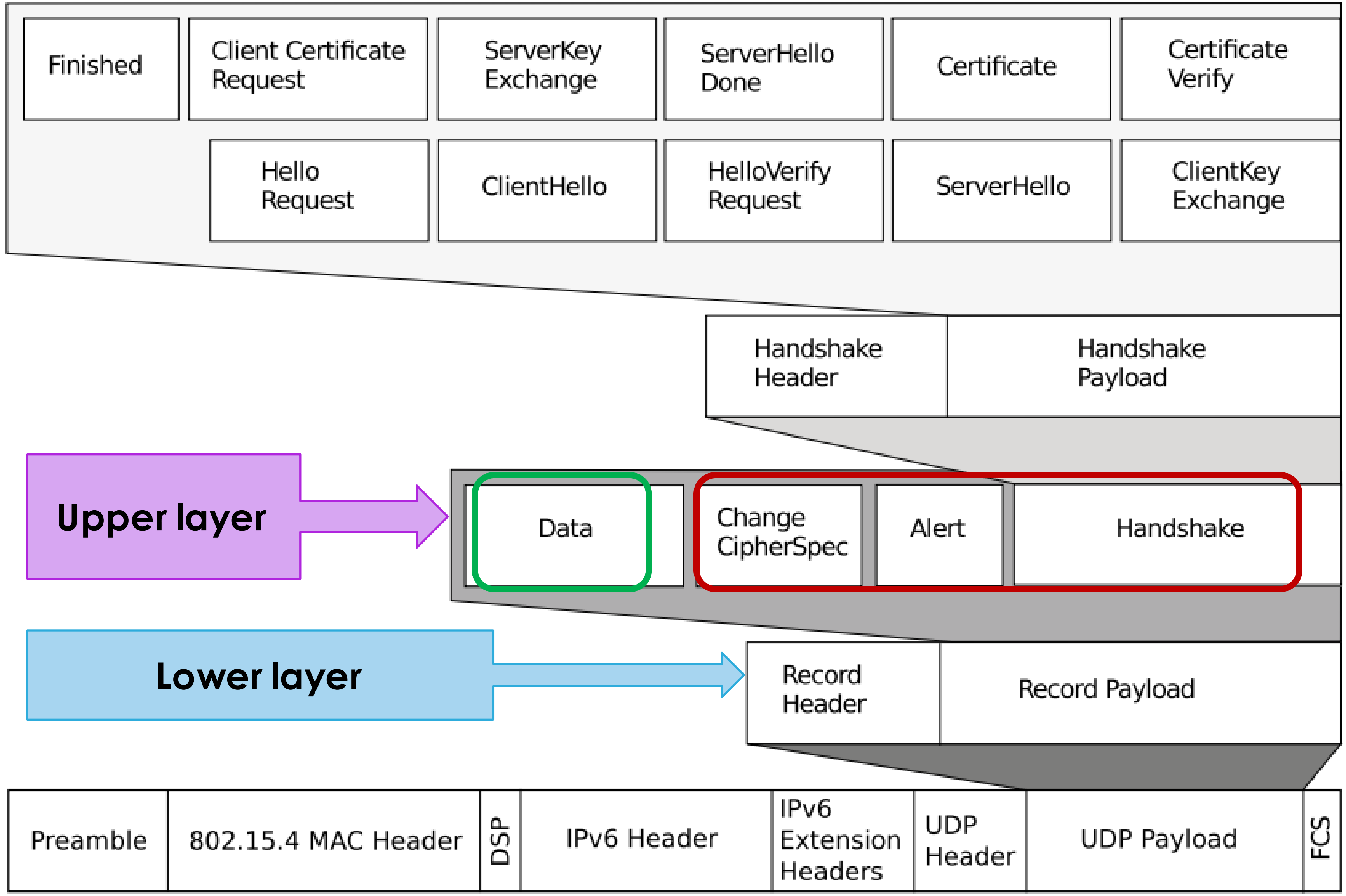
http

HTTP
TCP
IP

https

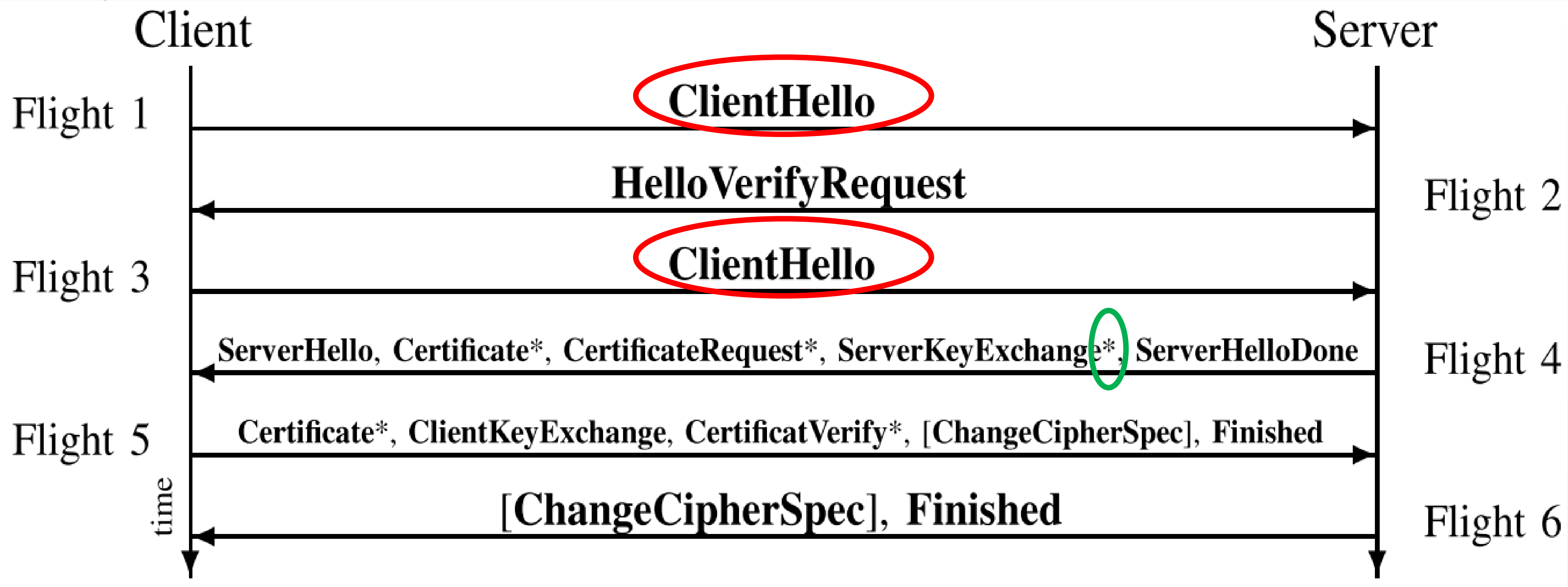
HTTP
SSL or TLS
TCP
IP

Layout of a packet secured with **DTLS**



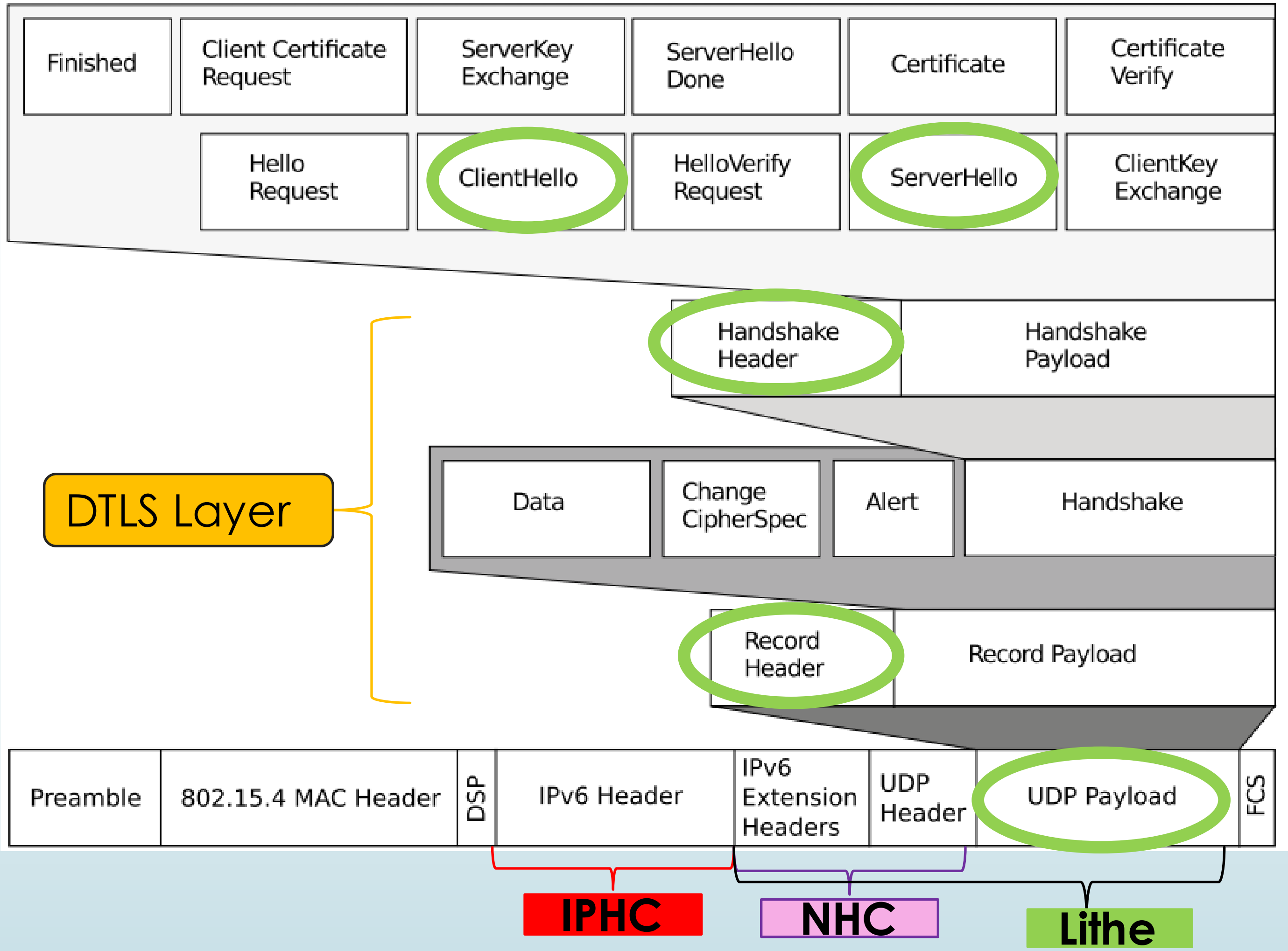
DTLS-Handshake Process

The handshake messages are used to negotiate: security keys, encryption algorithms and compressing methods.


Client
Server
Flight 1
ClientHello
HelloVerifyRequest
Flight 2
Flight 3
ClientHello
ServerHello, Certificate*, CertificateRequest*, ServerKeyExchange*, ServerHelloDone
Flight 4
Flight 5
Certificate*, ClientKeyExchange, CertificatVerify*, [ChangeCipherSpec], Finished
[ChangeCipherSpec], Finished
Flight 6
time

6LoWPAN

- ***This paper is limited to the header compression process only.***



Outline

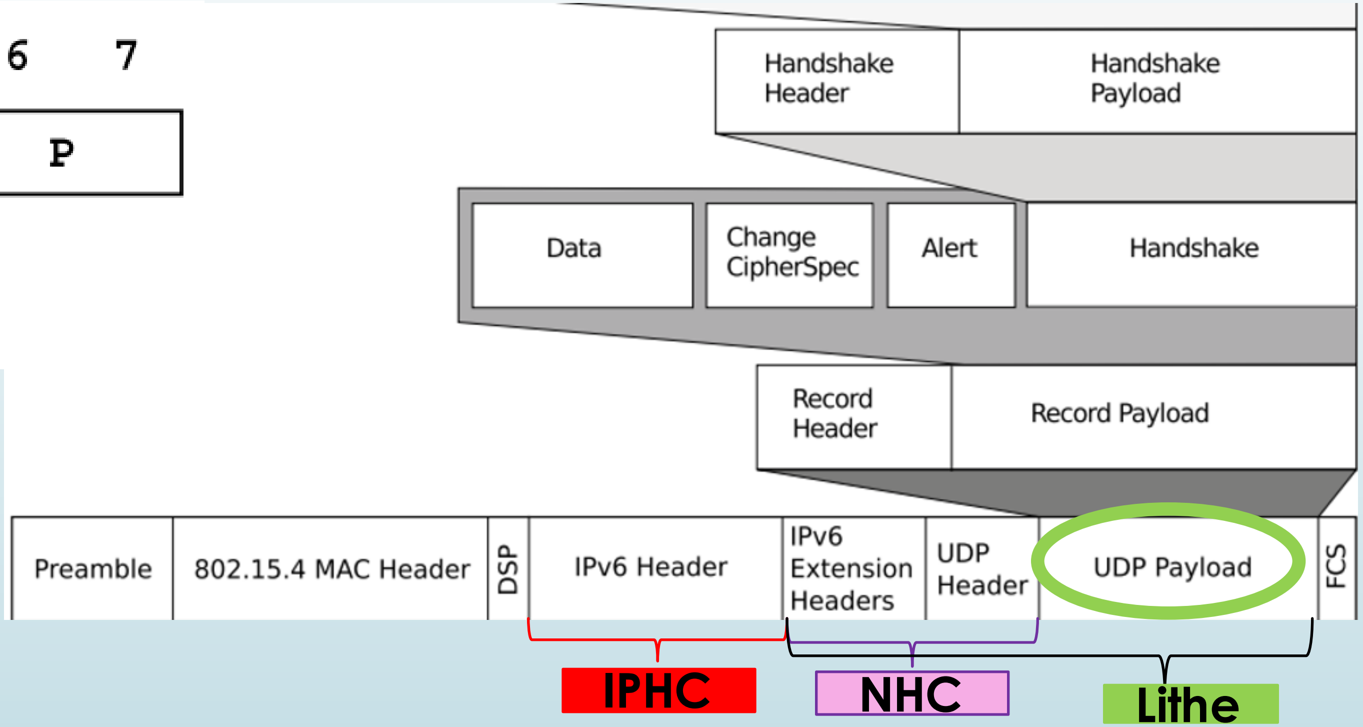
- Introduction
- Background
  - CoAP and DTLS
  - 6LoWPAN
- **DTLS Compression**
  - DTLS-6LoWPAN Integration
  - 6LoWPAN-NHC for the Record and Handshake Headers
  - 6LoWPAN-NHC for ClientHello / ServerHello
  - 6LoWPAN-NHC for other Handshake Messages
- Implementation
- Evaluation
  - Packet Size Reduction
  - RAM and ROM Requirement
  - Run-Time Performance
- Future work

DTLS-6LoWPAN Integration

- Apply 6LoWPAN header compression mechanism to compress headers in the UDP payload.
- The **ID bits in the NHC for UDP defined in 6LoWPAN**:
  - 11110 means the UDP payload is not compressed;
  - **11011** means the UDP payload is compressed with **6LoWPAN-NHC**.

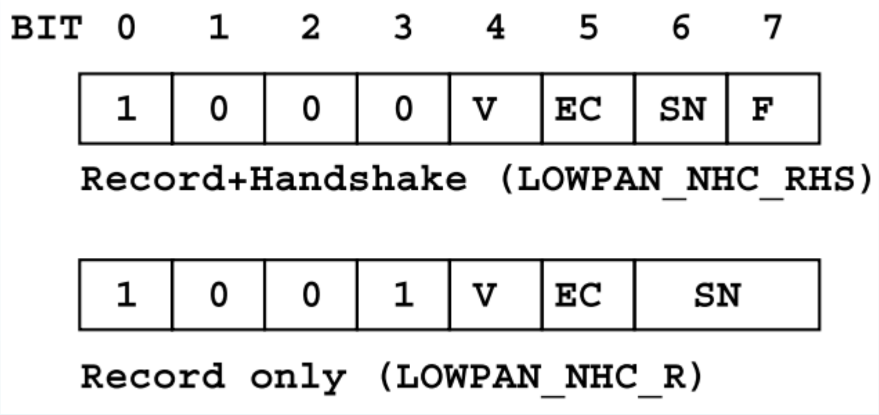
BIT	0	1	2	3	4	5	6	7
	1	1	0	1	1	C	P	

C: Checksum
P: Ports



6LoWPAN-NHC-R & 6LoWPAN-NHC-RHS

- First 4 bits represent the ID field:
  - 1000 – 6LoWPAN-NHC-RHS
  - 1001 – 6LoWPAN-NHC-R
- Version (v): DTLS version
  - 0 – omit version field (16 bits)
- Epoch (EC):
  - 0, 8 bit epoch is used and the left most 8 bits are omitted.
  - 1, all16 bit epoch is used.
- Sequence Number (SN):
  - 0, 16 bit SN, omit 32 bits
  - 1, 48 bit SN
- Fragment (F):
  - 0, not fragment.
    - Omit 2 x (*offset* + *length*) 6 bytes.
  - 1, fragment applied.

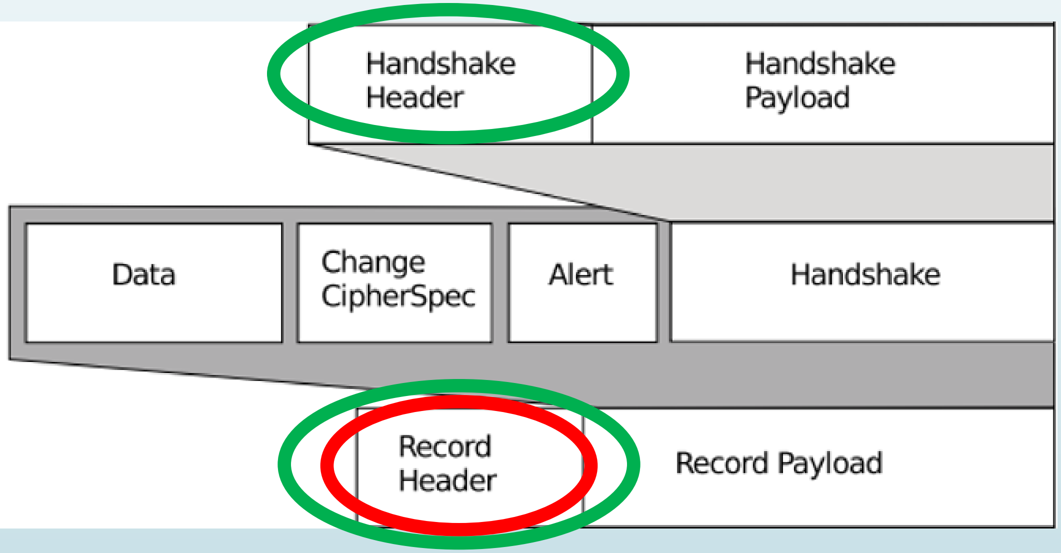


BIT	0	1	2	3	4	5	6	7
	1	0	0	0	V	EC	SN	F

Record+Handshake (LOWPAN_NHC_RHS)

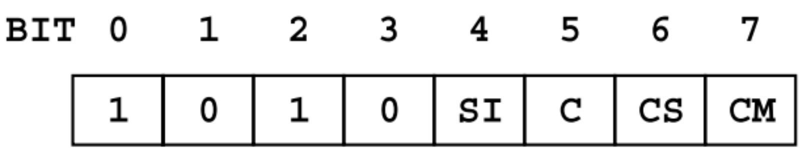
	1	0	0	1	V	EC	SN

Record only (LOWPAN_NHC_R)

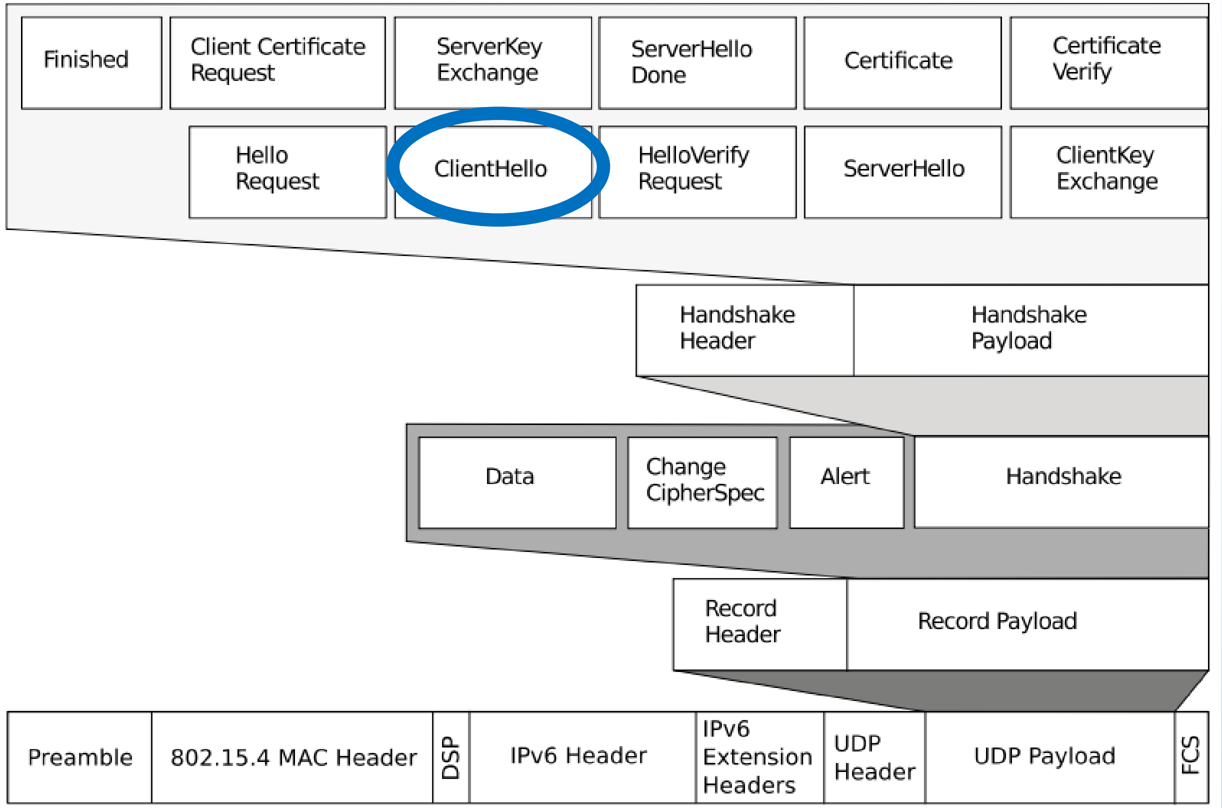


6LoWPAN-NHC-CH

- **First 4 bits is ID, 1010**
- When the parameter is set to 0, the corresponding field is omitted.
  - Session ID (SI): omit 8 bits
  - Cookie (C): omit 16 bits
  - Cipher Suites (CS): omit 16bits
  - Compression Method (CM): Omit 8 bits



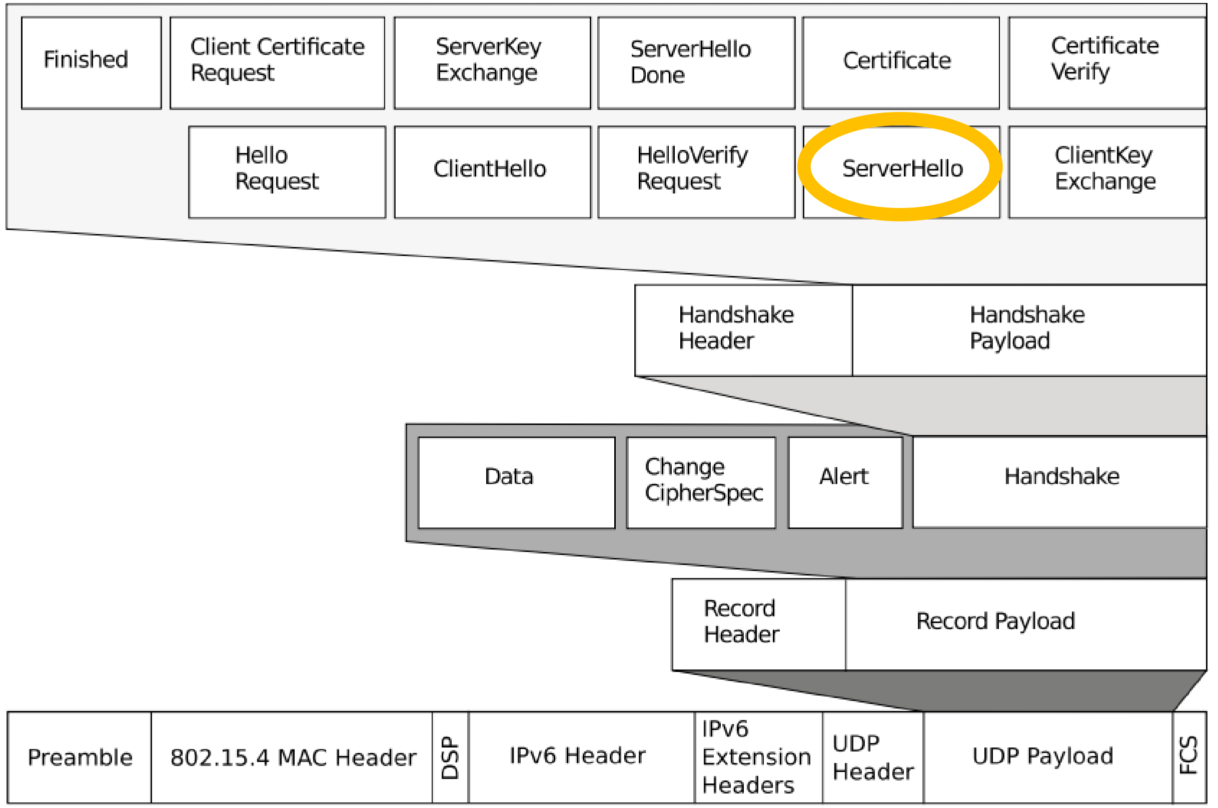
BIT	0	1	2	3	4	5	6	7
	1	0	1	0	SI	C	CS	CM



6LoWPAN-NHC-SH

- Similar to ClientHello except:
  - **ID field is 1011**
  - V (Server DTLS Version): 0 - DTLS 1.0, omit 16 bits

BIT	0	1	2	3	4	5	6	7
	1	0	1	1	V	SI	CS	CM



Finished

Client Certificate Request

ServerKey Exchange

ServerHello Done

Certificate

Certificate Verify

Hello Request

ClientHello

HelloVerify Request

ServerHello

ClientKey Exchange

Handshake Header

Handshake Payload

Data

Change CipherSpec

Alert

Handshake

Record Header

Record Payload

Preamble

802.15.4 MAC Header

DSP

IPv6 Header

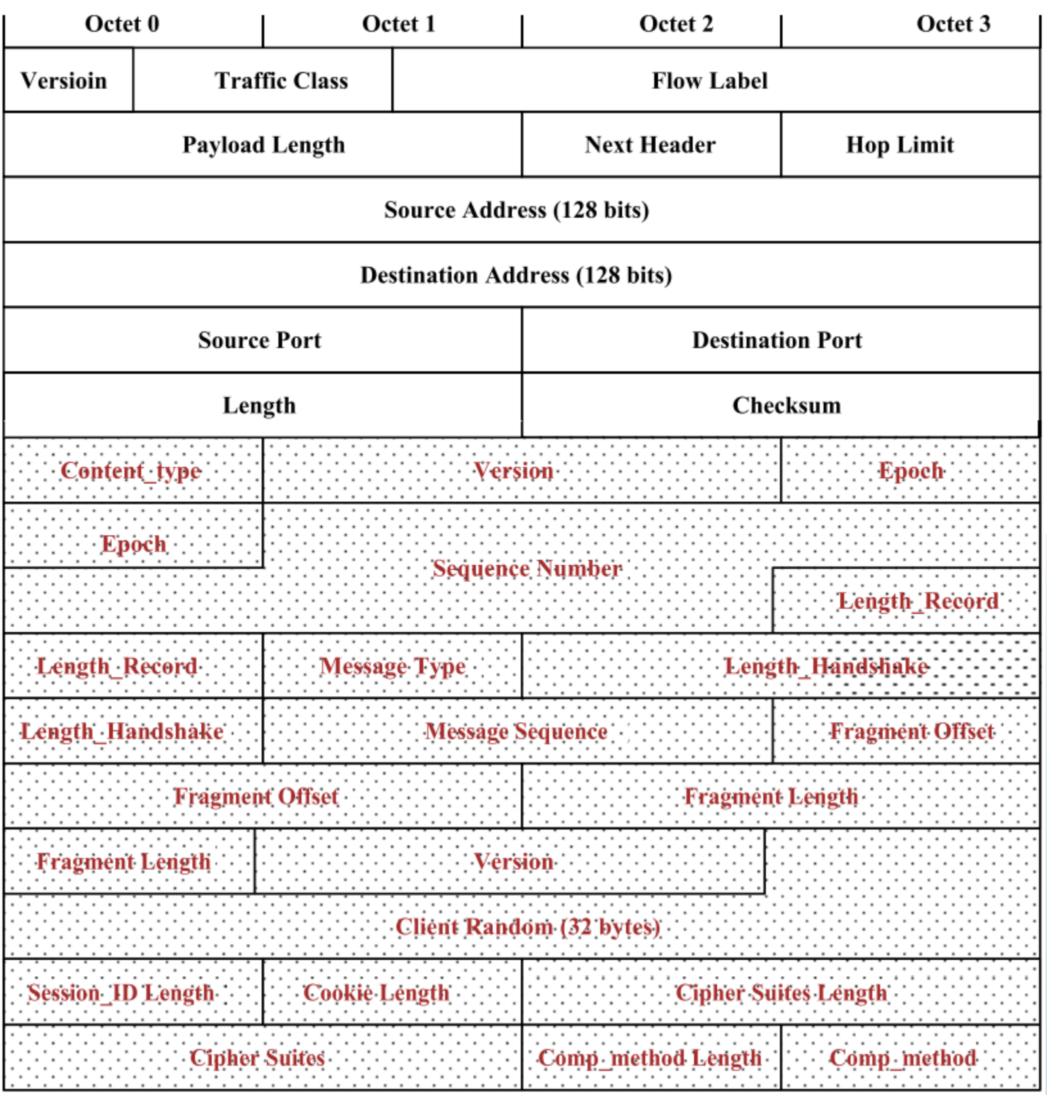
IPv6 Extension Headers

UDP Header

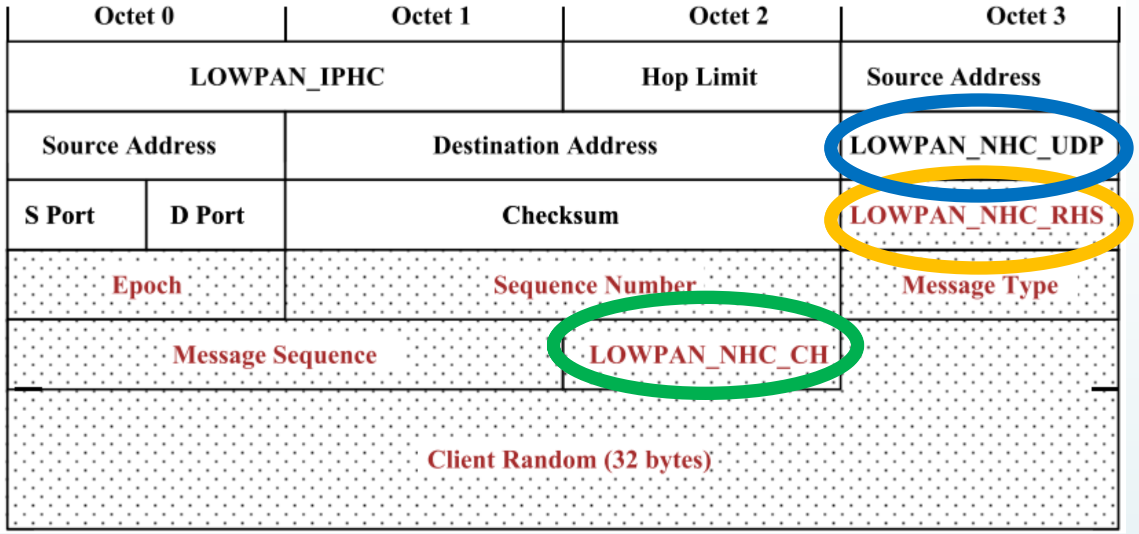
UDP Payload

FCS

6LoWPAN-NHC for ClientHello



Octet 0	Octet 1	Octet 2	Octet 3
Versioin	Traffic Class	Flow Label	
Payload Length		Next Header	Hop Limit
Source Address (128 bits)			
Destination Address (128 bits)			
Source Port		Destination Port	
Length		Checksum	
Content_type	Version		Epoch
Epoch	Sequence Number		
			Length_Record
Length_Record	Message Type	Length_Handshake	
Length_Handshake	Message Sequence		Fragment Offset
Fragment Offset		Fragment Length	
Fragment Length	Version		
Client Random (32 bytes)			
Session_ID Length	Cookie Length	Cipher Suites Length	
Cipher Suites		Comp_method Length	Comp_method



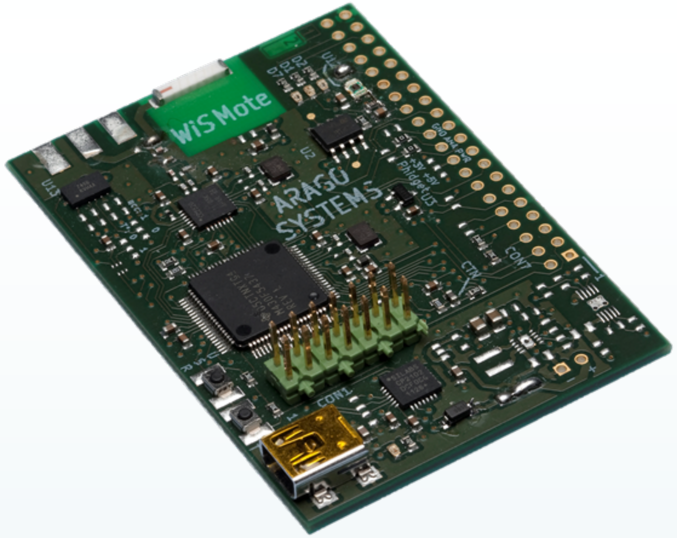
Octet 0	Octet 1	Octet 2	Octet 3
LOWPAN_IPHC		Hop Limit	Source Address
Source Address	Destination Address		LOWPAN_NHC_UDP
S Port D Port	Checksum		LOWPAN_NHC_RHS
Epoch	Sequence Number		Message Type
Message Sequence		LOWPAN_NHC_CH	
Client Random (32 bytes)			

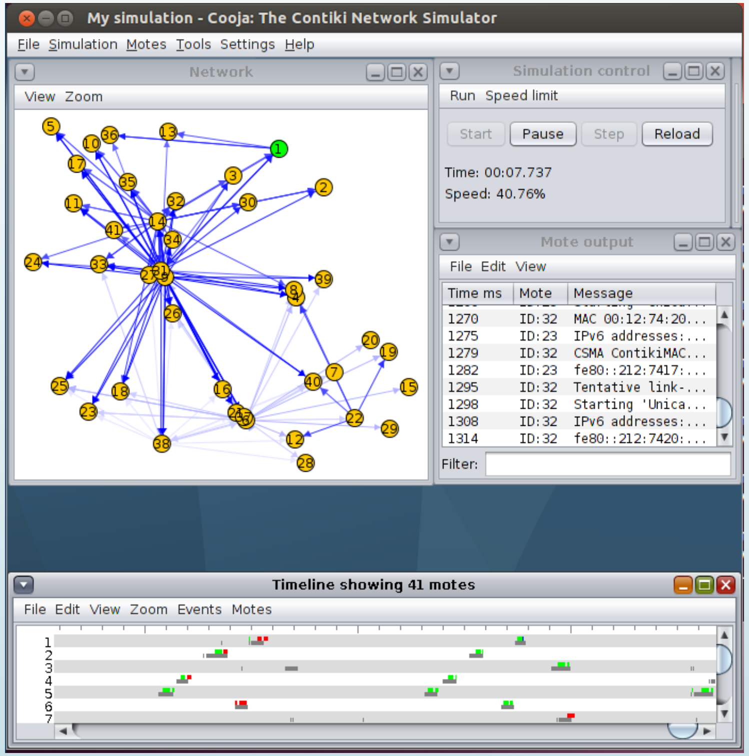
Outline

- Introduction
- Background
  - CoAP and DTLS
  - 6LoWPAN
- DTLS Compression
  - DTLS-6LoWPAN Integration
  - 6LoWPAN-NHC for the Record and Handshake Headers
  - 6LoWPAN-NHC for ClientHello / ServerHello
  - 6LoWPAN-NHC for other Handshake Messages
- **Implementation**
- Evaluation
  - Packet Size Reduction
  - RAM and ROM Requirement
  - Run-Time Performance
- Future work

Implementation

- Lithe was implemented in the Contiki OS.
- Hardware platform: WiSMote.
- Lithe implementation consists of four components:
  - DTLS: open source tinyDTLS.
  - CoAP: default CoAP in Contiki.
  - CoAP-DTLS integration module: Connects the CoAP and DTLS to enable CoAPs.
  - DTLS header compression.




My simulation - Cooja: The Contiki Network Simulator
File Simulation Motes Tools Settings Help
Network
View Zoom
Simulation control
Run Speed limit
Start Pause Step Reload
Mote output
File Edit View
Filter:
Timeline showing 41 motes
File Edit View Zoom Events Motes

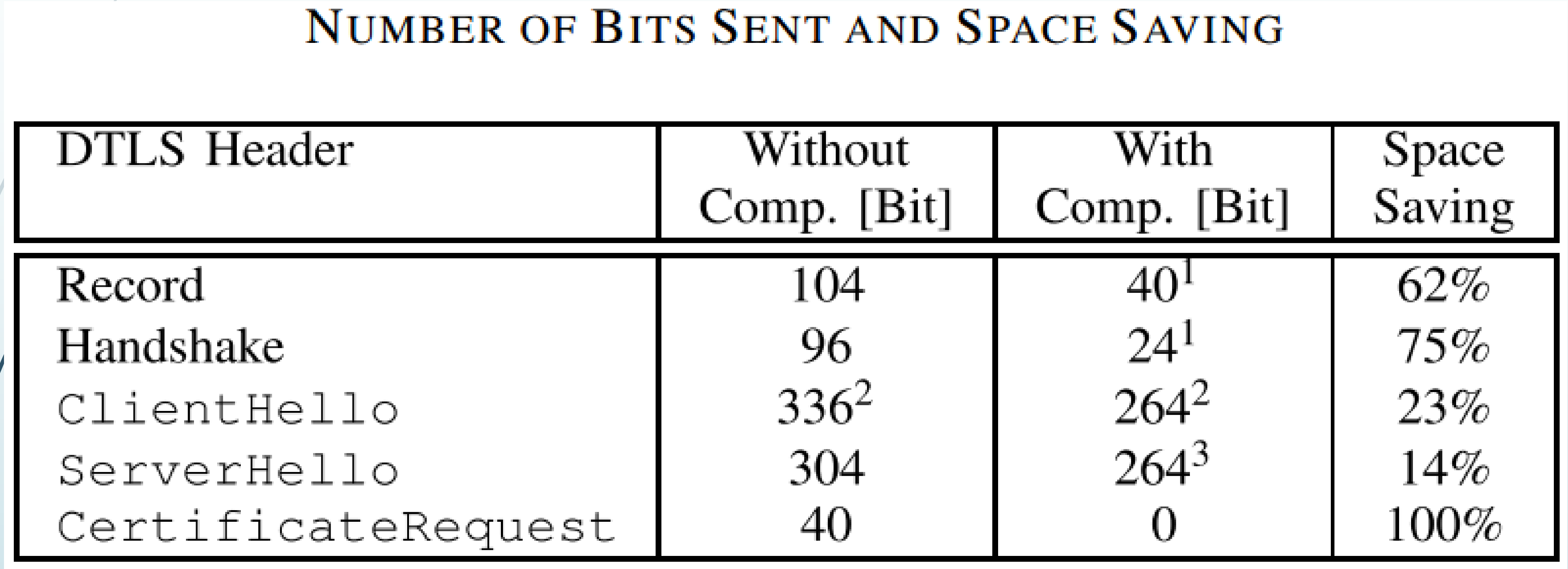
Time: 00:07.737
Speed: 40.76%

```
Time ms   Mote    Message
1270      ID:32   MAC 00:12:74:20...
1275      ID:23   IPv6 addresses:...
1279      ID:32   CSMA ContikiMAC...
1282      ID:23   fe80::212:7417:...
1295      ID:32   Tentative link-...
1298      ID:32   Starting 'Unica...
1308      ID:32   IPv6 addresses:...
1314      ID:32   fe80::212:7420:...
```

Outline

- Introduction
- Background
  - CoAP and DTLS
  - 6LoWPAN
- DTLS Compression
  - DTLS-6LoWPAN Integration
  - 6LoWPAN-NHC for the Record and Handshake Headers
  - 6LoWPAN-NHC for ClientHello / ServerHello
  - 6LoWPAN-NHC for other Handshake Messages
- Implementation
- **Evaluation**
  - Packet Size Reduction
  - RAM and ROM Requirement
  - Run-Time Performance
- Future work

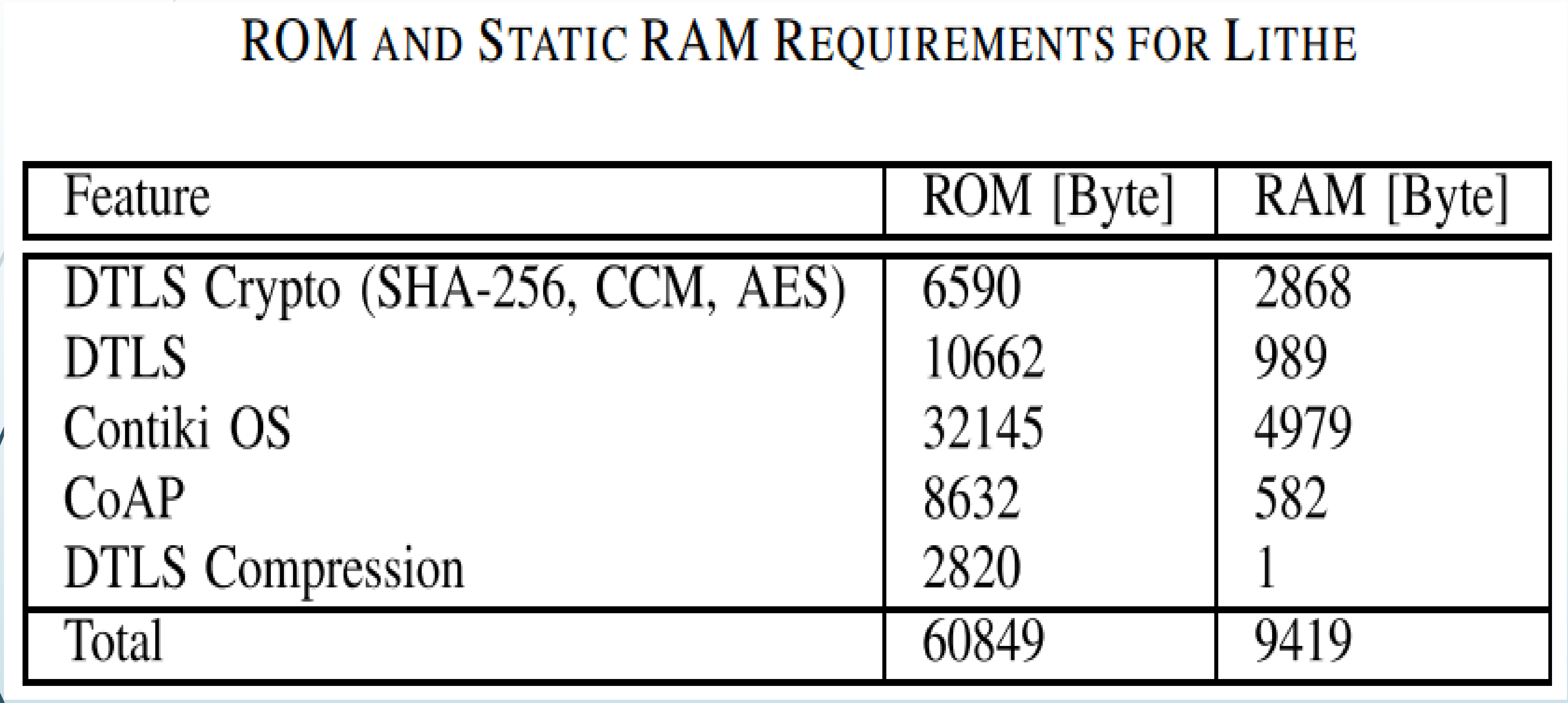
Evaluation - Packet Size Reduction



NUMBER OF BITS SENT AND SPACE SAVING

DTLS Header	Without Comp. [Bit]	With Comp. [Bit]	Space Saving
Record	104	40[1]	62%
Handshake	96	24[1]	75%
`ClientHello`	336[2]	264[2]	23%
`ServerHello`	304	264[3]	14%
`CertificateRequest`	40	0	100%

Evaluation – RAM/ROM Requirement

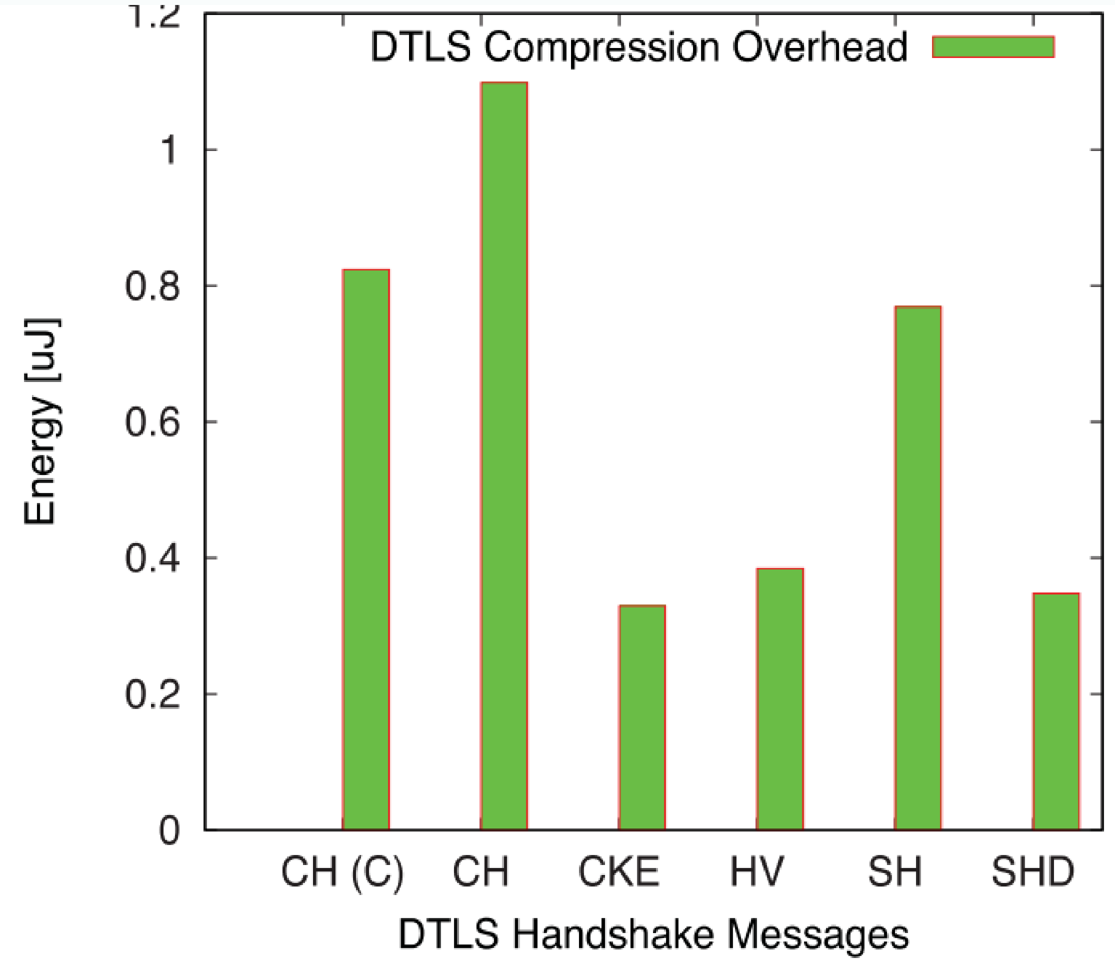


ROM AND STATIC RAM REQUIREMENTS FOR LITHE

Feature	ROM [Byte]	RAM [Byte]
DTLS Crypto (SHA-256, CCM, AES)	6590	2868
DTLS	10662	989
Contiki OS	32145	4979
CoAP	8632	582
DTLS Compression	2820	1
Total	60849	9419

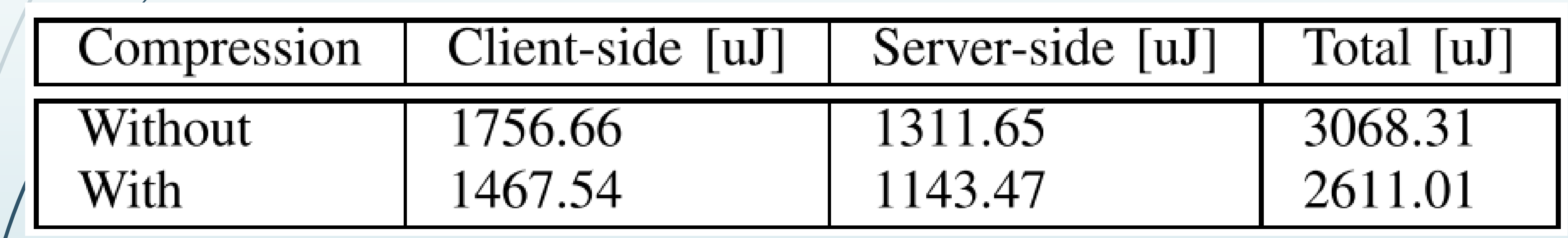
Evaluation - Run-Time Performance

- CH – ClientHello
- CH(C) – ClientHello with Cookie
- CKE – ClientKeyExchange
- HV – HelloVerify
- SH – ServerHello
- SHD - ServerHelloDone



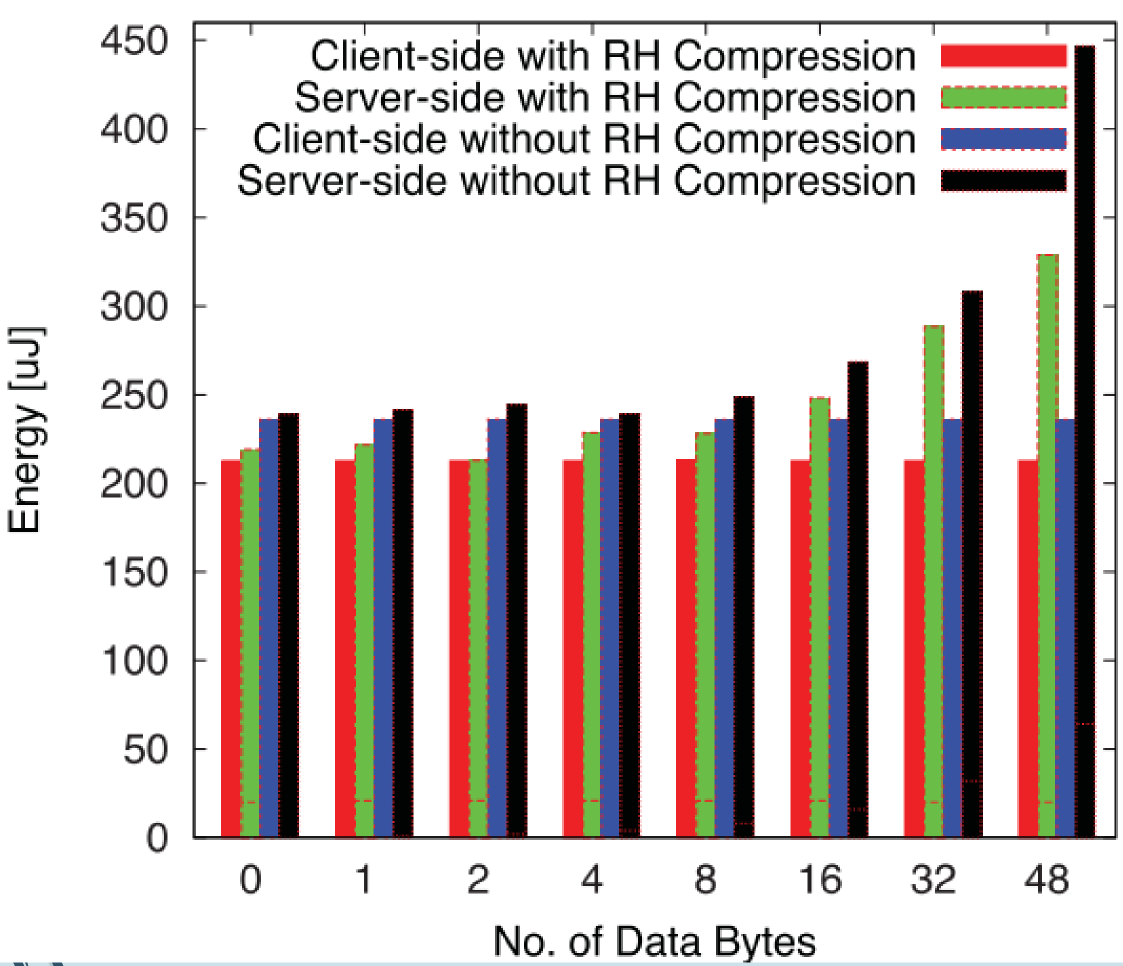
Evaluation - Run-Time Performance

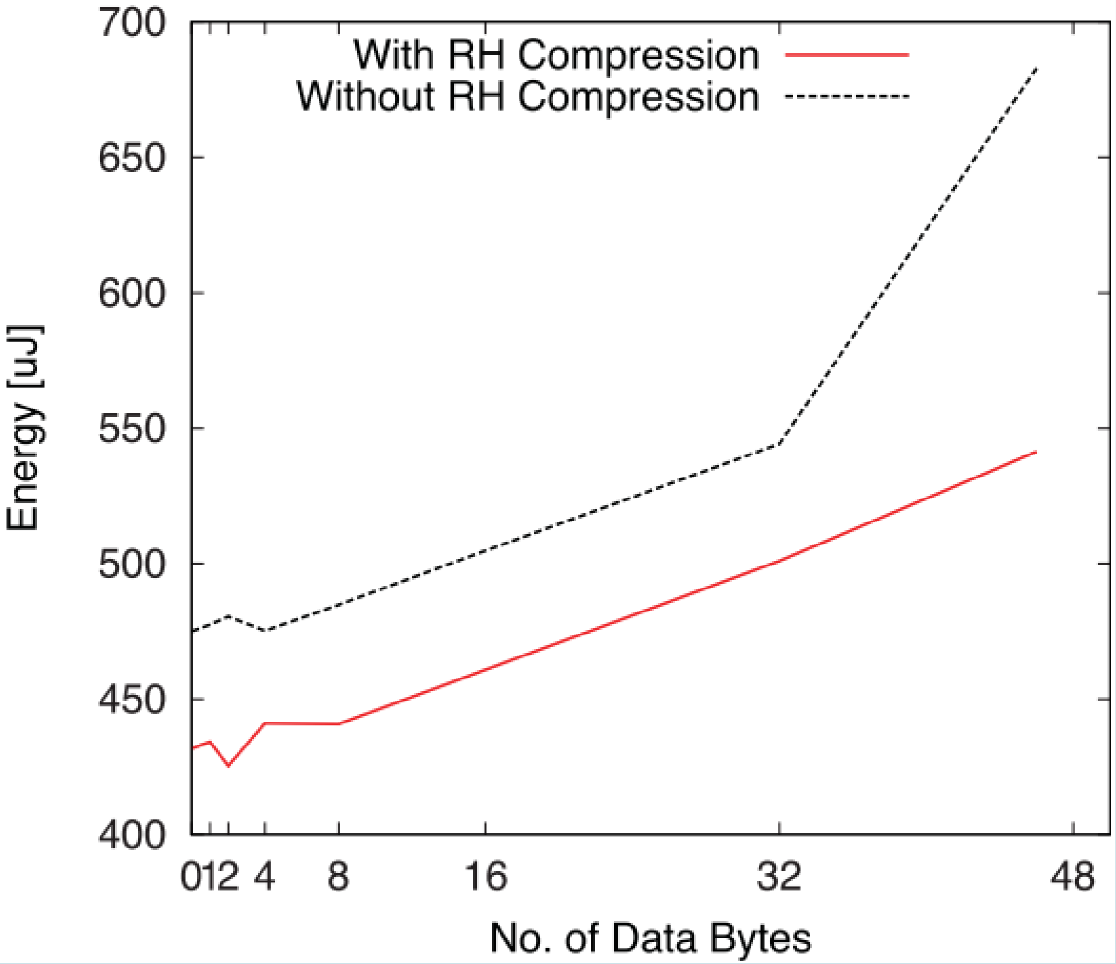
- 15% less energy is used transmit/receive compressed packets.



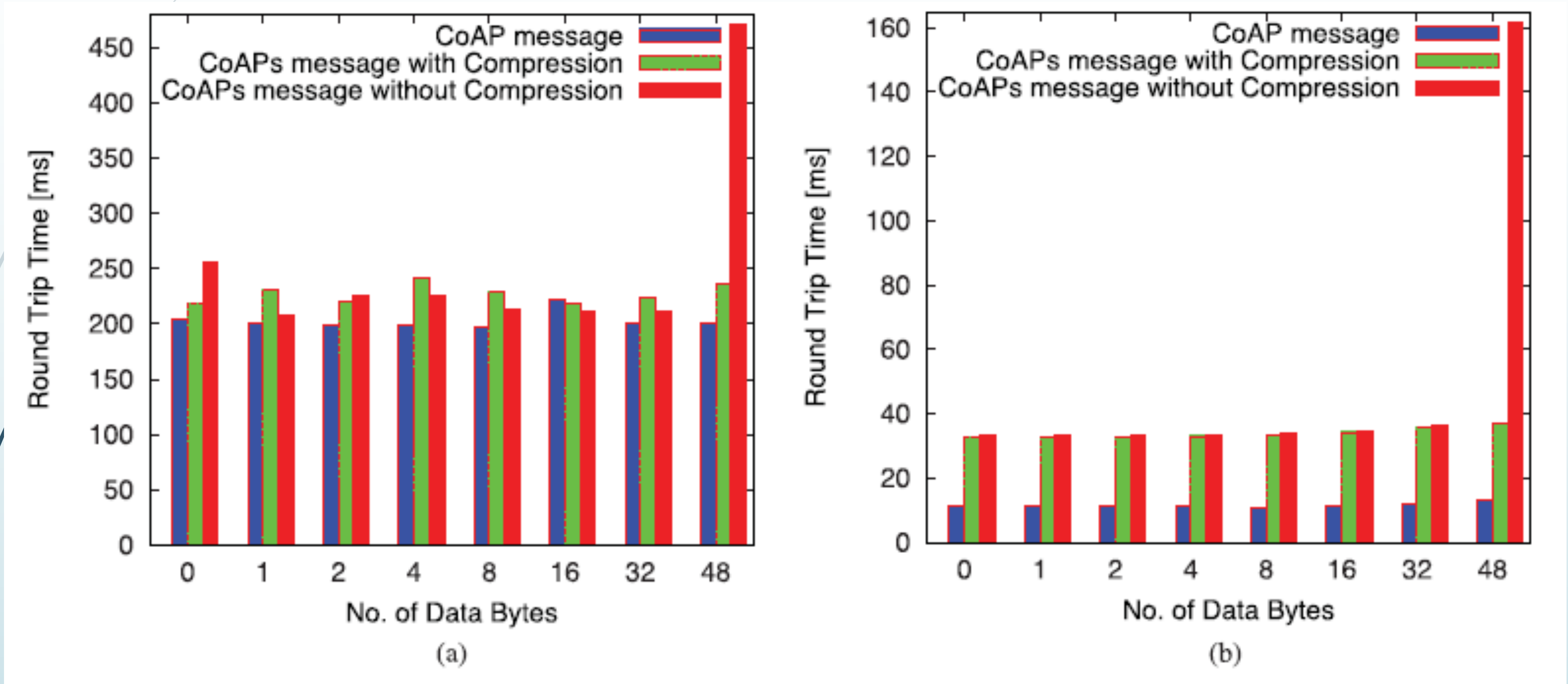
Compression	Client-side [uJ]	Server-side [uJ]	Total [uJ]
Without	1756.66	1311.65	3068.31
With	1467.54	1143.47	2611.01

Evaluation – Energy Consumption





Evaluation – Round Time Trip (RTT)



(a)

(b)

Outline

- Introduction
- Background
  - CoAP and DTLS
  - 6LoWPAN
- DTLS Compression
  - DTLS-6LoWPAN Integration
  - 6LoWPAN-NHC for the Record and Handshake Headers
  - 6LoWPAN-NHC for ClientHello / ServerHello
  - 6LoWPAN-NHC for other Handshake Messages
- Implementation
- Evaluation
  - Packet Size Reduction
  - RAM and ROM Requirement
  - Run-Time Performance
- **Future work**

Future work:

- deploy Lithe in a real world IOT system with a real application scenario.

References

- http://www.ujjwal.com/technical-optimization/https-seo-662/
- https://www.micrium.com/iot/internet-protocols/
- http://www.wismote.com/
- http://contiki-os.blogspot.ca/
- http://s3.amazonaws.com/ppt-download/lithe-150602124705-lva1-app6891.pdf

Thank you