Freie Universität Berlin

HAW HAMBURG

IoT Content Object Security with OSCORE and NDN: A First Experimental Comparison

IFIP Networking 2020, Paris

Cenk Gündoğan[1] Christian Amsüss

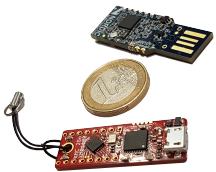
Thomas C. Schmidt[1] Matthias Wählisch[2]

[1]HAW Hamburg [2]Freie Universität Berlin

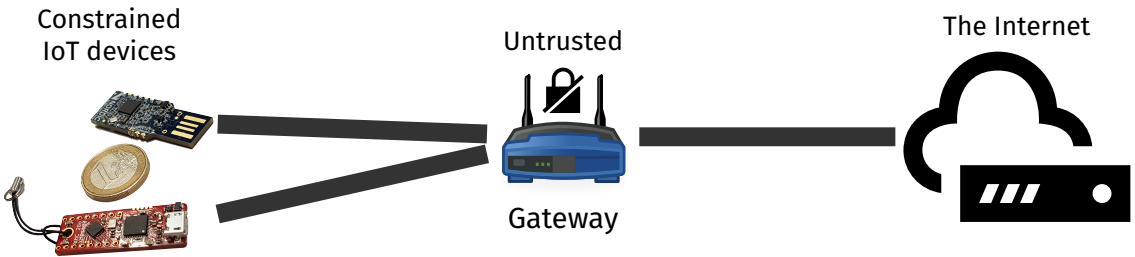
cenk.guendogan@haw-hamburg.de christian@amsuess.com t.schmidt@haw-hamburg.de m.waehlisch@fu-berlin.de

Common IoT Deployments

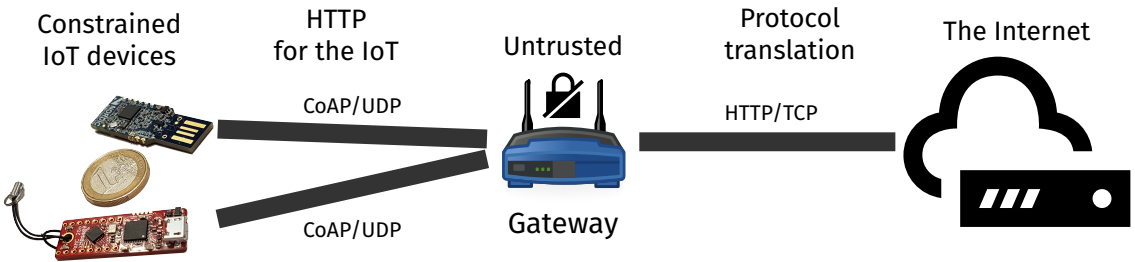
Constrained
IoT devices



Common IoT Deployments



Common IoT Deployments

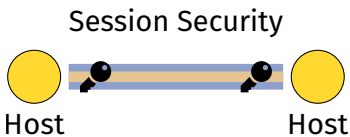


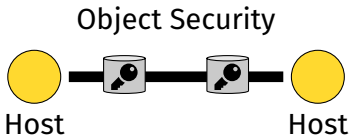
Common IoT Deployments

We need **content object security**
for end-to-end protection

Content Object Security

- Prominent feature in information-centric architectures





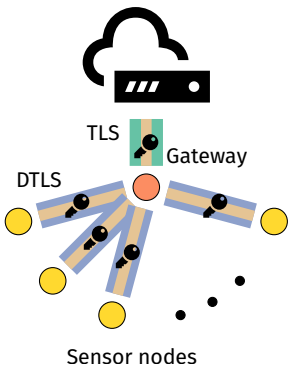
- Content objects securely cacheable
- Slowly transitions into host-centric world

Outline

Session Security vs. Object Security

Session Security: CoAP over DTLS 1.2

CoAP over DTLS 1.2



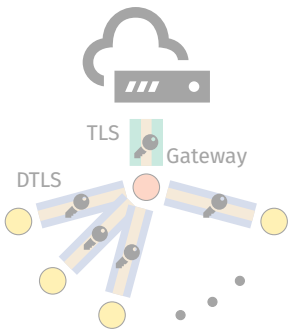
host-centric

HTTP	CoAP
TLS TCP	DTLS UDP
IPv6 6LoWPAN	
802.15.4, BLE, LoRa, …	

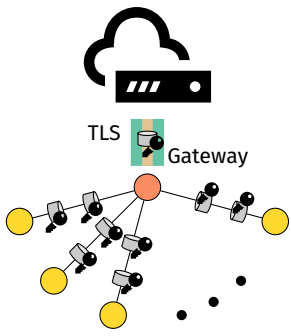
E2E protection is harmed

Object Security: CoAP + OSCORE

CoAP over DTLS 1.2



OSCORE



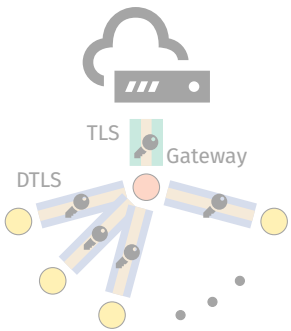
host-centric

HTTP	CoAP **OSCORE**
TLS TCP	DTLS UDP
IPv6 6LoWPAN	
802.15.4, BLE, LoRa, …	

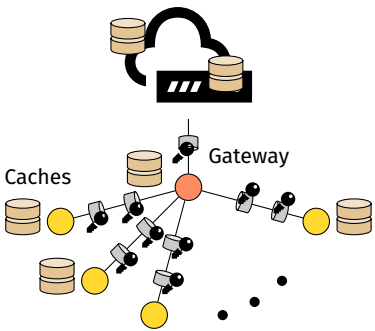
E2E protection is preserved

Object Security: Named Data Networking

CoAP over DTLS 1.2



NDN



information-centric

Applications
File, Stream
Security
NDN
ICNLoWPAN
802.15.4, BLE, LoRa, …

E2E protection is preserved

Comparison of Security Properties

	CoAP		NDN
	DTLS	OSCORE	Protected
Request Message			
Integrity	✓	✓	(✓)
Authenticity	✓	✓	(✓)
Confidentiality	✓	✓	✘*
Response Message			
Integrity	✓	✓	✓
Authenticity	✓	✓	✓
Confidentiality	✓	✓	✘*

* provided on application layer

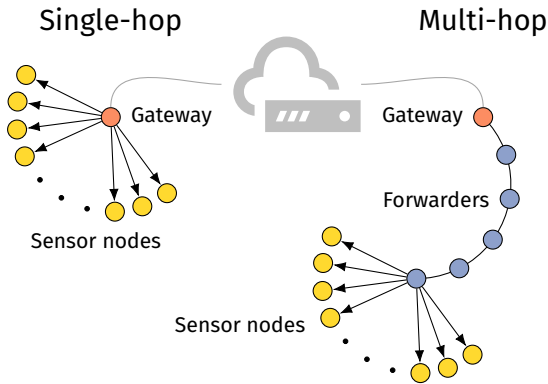
Research Question

Is OSCORE the better alternative
for secure networking in the IoT?

Protocol Performance

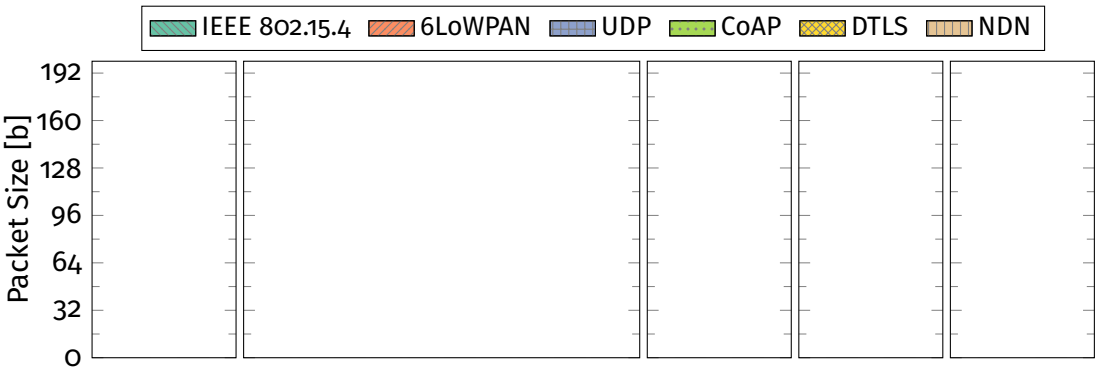
Testbed Setup

Hardware	M3 node in IoT Lab testbed, IEEE 802.15.4
Software	RIOT with tinyDTLS, libOSCORE, CCN-lite
Topology	Single- & Multi-hop
Scenario	Gateway requests 2-byte temperature every $\approx$ 2 s



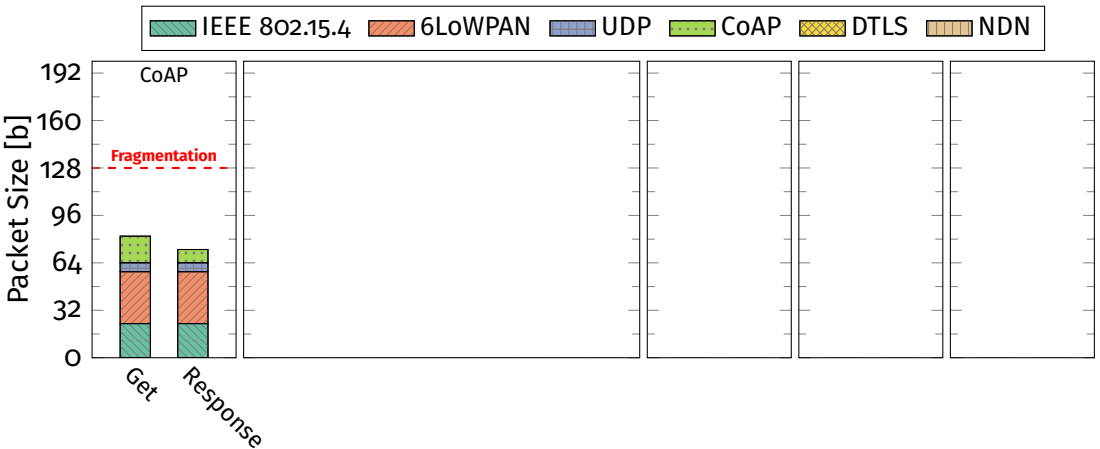
Packet Structure Dissection

- Maximum frame size for IEEE 802.15.4 is 127 bytes



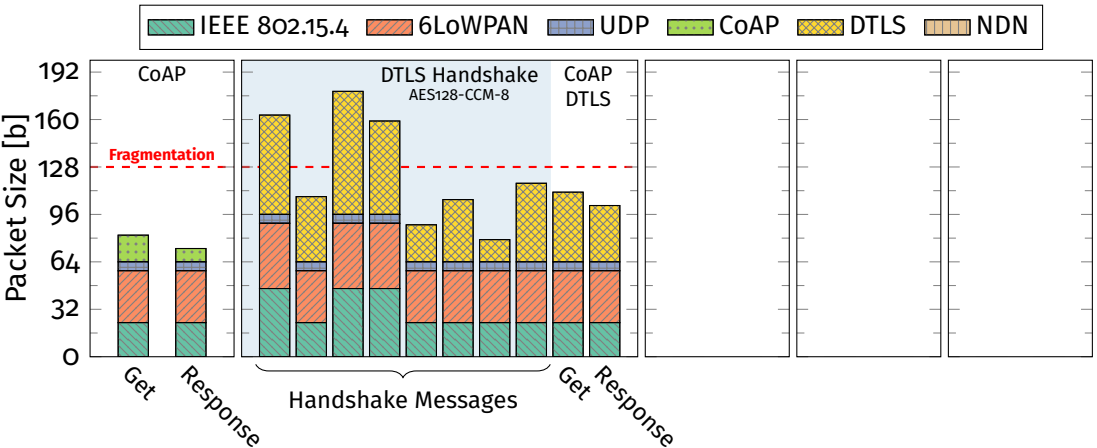
Packet Structure Dissection

- Maximum frame size for IEEE 802.15.4 is 127 bytes



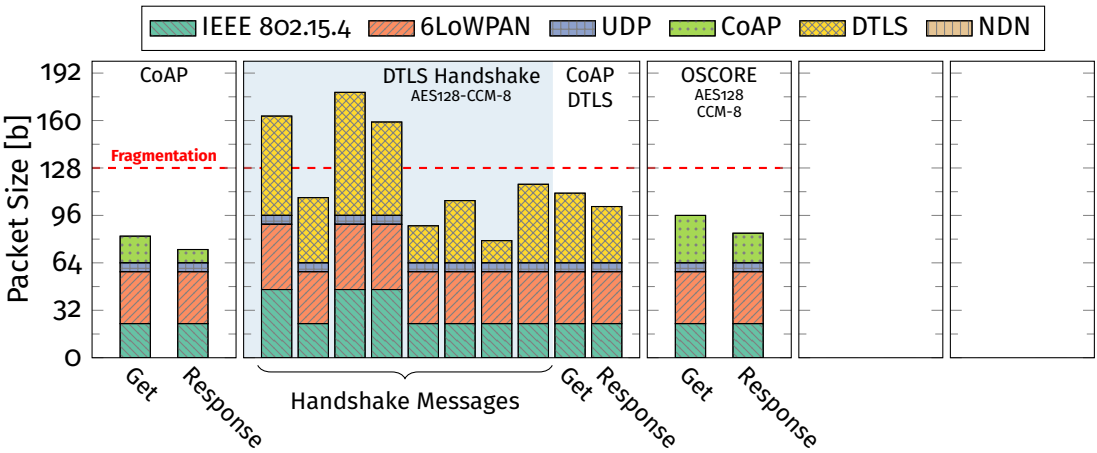
Packet Structure Dissection

- Maximum frame size for IEEE 802.15.4 is 127 bytes



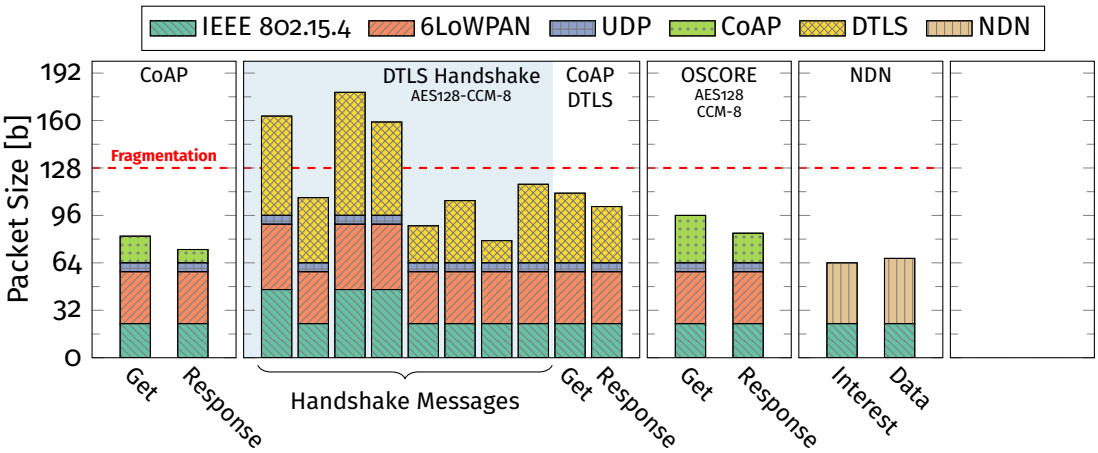
Packet Structure Dissection

- Maximum frame size for IEEE 802.15.4 is 127 bytes



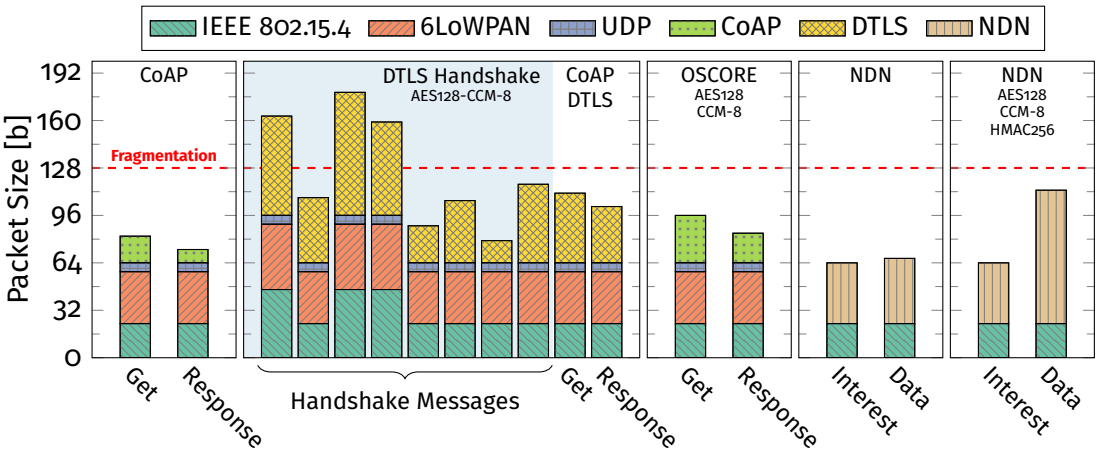
Packet Structure Dissection

- Maximum frame size for IEEE 802.15.4 is 127 bytes



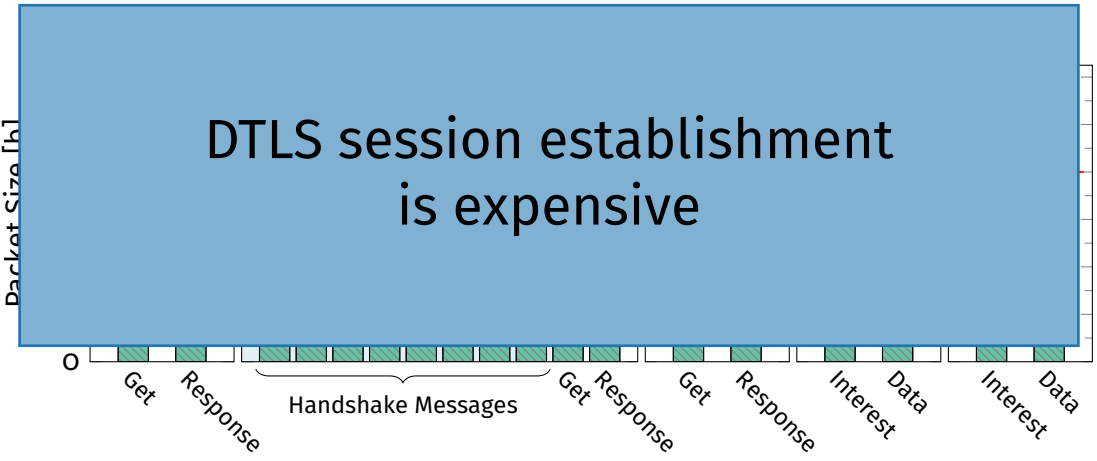
Packet Structure Dissection

- Maximum frame size for IEEE 802.15.4 is 127 bytes



Packet Structure Dissection

- Maximum frame size for IEEE 802.15.4 is 127 bytes



Security Protocol Overhead

- Protocol overhead in bytes compared to unsecured protocol variants

	CoAP				**NDN**	
	DTLS		OSCORE		Protected	
	Request	Response	Request	Response	Request	Response
Structure						
Context ID						
Nonce						
MAC						

Security Protocol Overhead

- Protocol overhead in bytes compared to unsecured protocol variants

	CoAP				**NDN**	
	DTLS		OSCORE		Protected	
	Request	Response	Request	Response	Request	Response
Structure	11	11				
Context ID	2	2				
Nonce	8	8				
MAC	8	8				

Security Protocol Overhead

- Protocol overhead in bytes compared to unsecured protocol variants

	CoAP				**NDN**	
	DTLS		OSCORE		Protected	
	Request	Response	Request	Response	Request	Response
Structure	11	11	4	3		
Context ID	2	2	1	0		
Nonce	8	8	1	0		
MAC	8	8	8	8		

Security Protocol Overhead

- Protocol overhead in bytes compared to unsecured protocol variants

	CoAP				**NDN**	
	DTLS		OSCORE		Protected	
	Request	Response	Request	Response	Request	Response
Structure	11	11	4	3	–	5
Context ID	2	2	1	0	–	1
Nonce	8	8	1	0	–	0
MAC	8	8	8	8	–	40

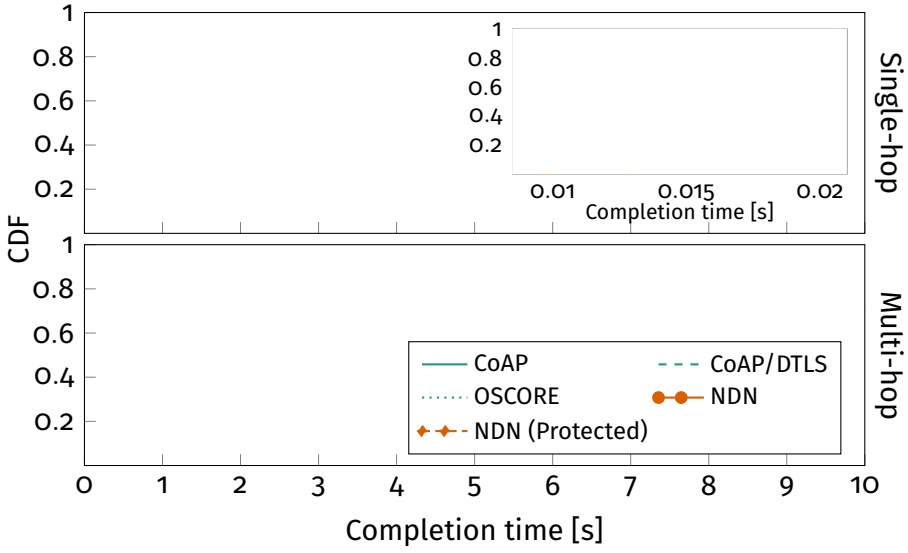
Security Protocol Overhead

- Protocol overhead in bytes compared to unsecured protocol variants

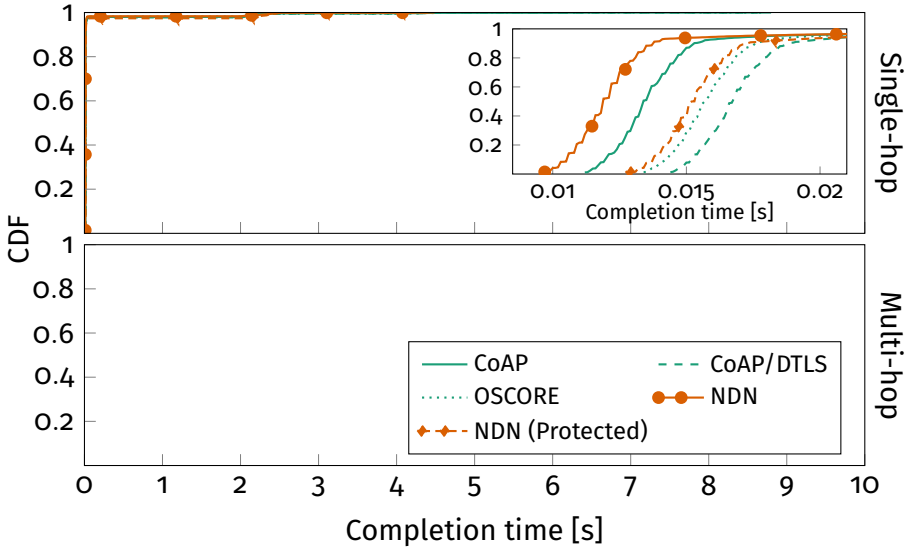
OSCORE leverages CoAP features to reduce overhead

MAC	8	8	8	8	–	40

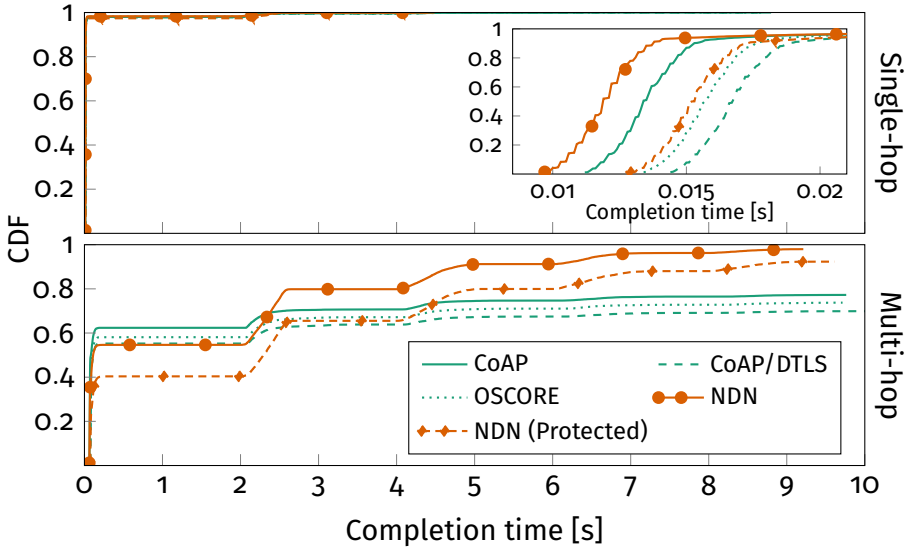
Time to Content Arrival



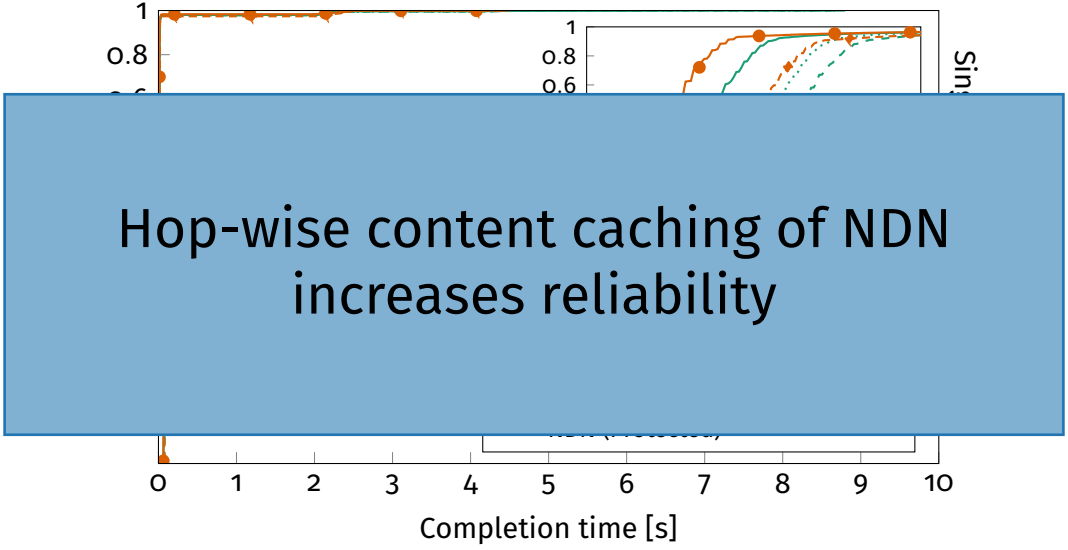
Time to Content Arrival



Time to Content Arrival



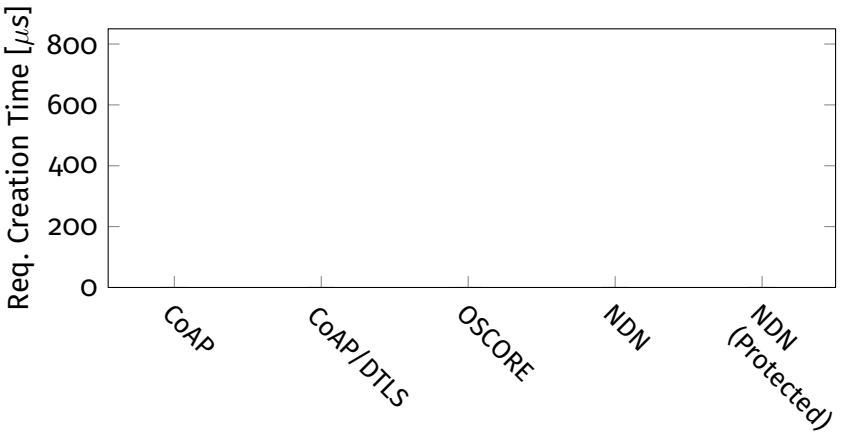
Time to Content Arrival



Hop-wise content caching of NDN increases reliability

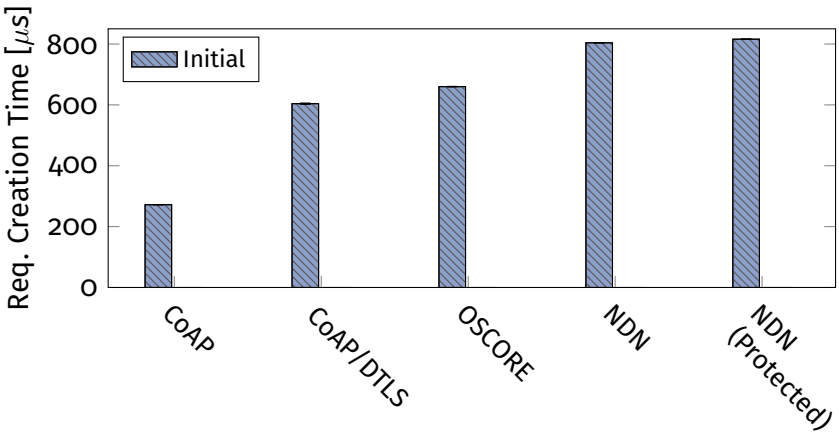
Request Creation Time

- Message retransmissions are frequent in low-power regimes
- **CoAP:** Application layer retransmissions
- **NDN:** Network layer retransmissions



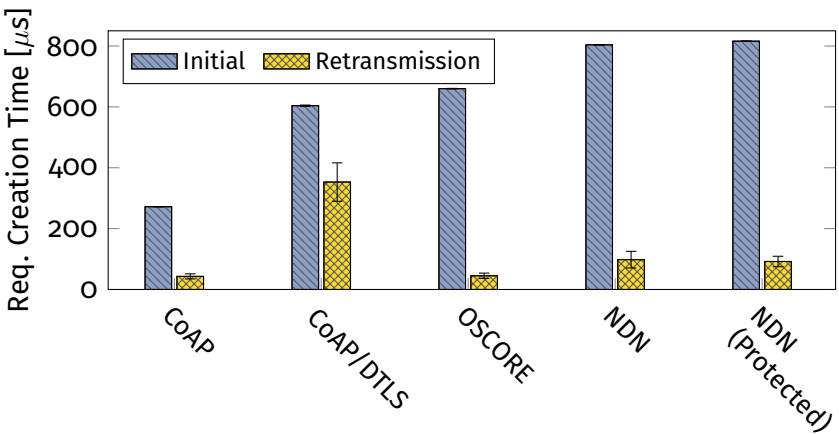
Request Creation Time

- Message retransmissions are frequent in low-power regimes
- **CoAP:** Application layer retransmissions
- **NDN:** Network layer retransmissions



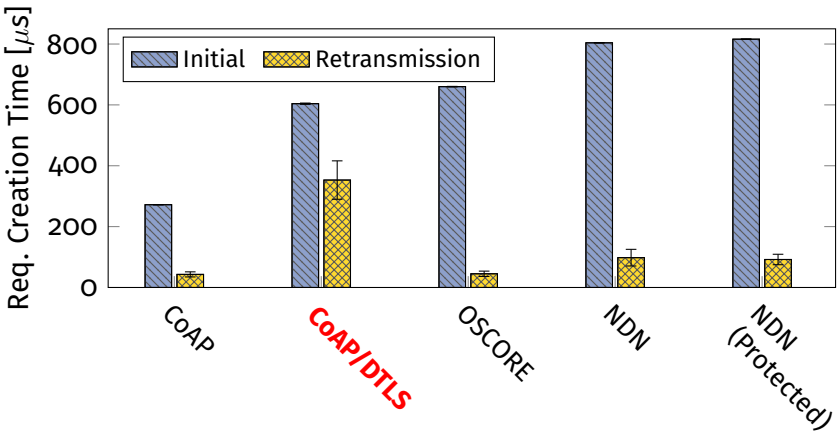
Request Creation Time

- Message retransmissions are frequent in low-power regimes
- **CoAP:** Application layer retransmissions
- **NDN:** Network layer retransmissions



Request Creation Time

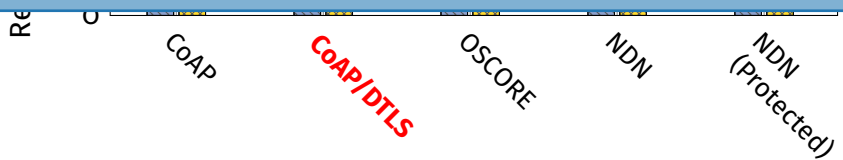
- Message retransmissions are frequent in low-power regimes
- **CoAP:** Application layer retransmissions
- **NDN:** Network layer retransmissions



Request Creation Time

- Message retransmissions are frequent in low-power regimes
- **CoAP:** Application layer retransmissions

DTLS session layer generates higher load on retransmissions

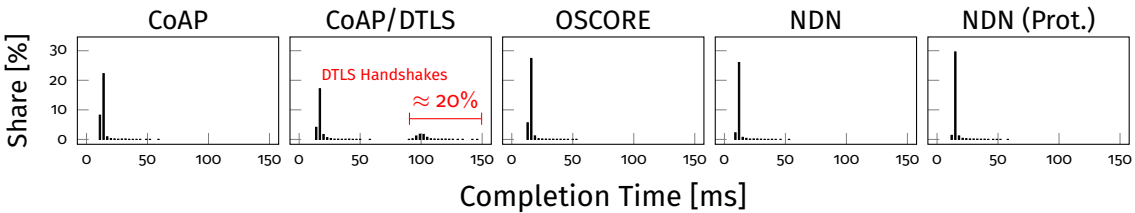


Protocol Latencies

- DTLS: security association with 5-tuple: (IP_{src}, $Port_{src}$, IP_{dst}, $Port_{dst}$, Protocol)
- Frequent endpoint changes or loss of session state leads to handshakes
- **Setup:** requests change endpoint information with probability of 20%

Protocol Latencies

- DTLS: security association with 5-tuple: (IP_{src}, $Port_{src}$, IP_{dst}, $Port_{dst}$, Protocol)
- Frequent endpoint changes or loss of session state leads to handshakes
- **Setup:** requests change endpoint information with probability of 20%



Protocol Latencies

- DTLS: security association with 5-tuple: (IP$_{src}$, Port$_{src}$, IP$_{dst}$, Port$_{dst}$, Protocol)

Conclusion & Outlook

Takeaways

- OSCORE brings a lean object security to the constrained IoT
- NDN has a higher reliability due to hop-wise caching
- CoAP over DTLS 1.2 has an expensive session overhead

Next Steps

- Extend OSCORE with caching capabilities
- Explore a RESTful information-centric Web of Things

Thank You!

We support reproducible research.

https://github.com/inetrg/IFIP-Networking-2020

Backup

IEEE 802.15.4

Low-rate and low-power wireless personal area networks

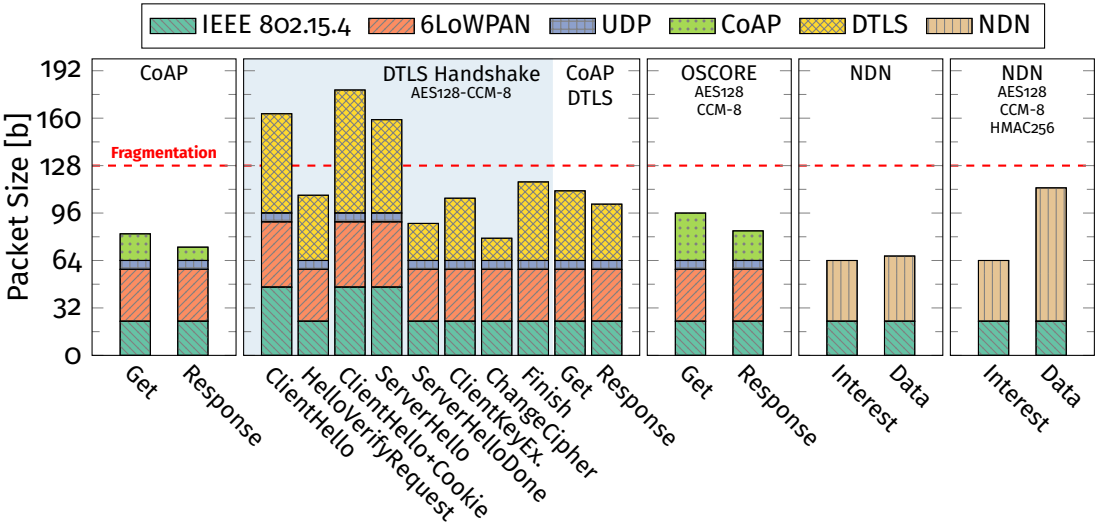
Radio Properties

- Max physical packet size: 127 bytes
- Theoretical bandwidth: 250 kbit/s
- Range: $\approx 10 - 200$ meters

Media Access Control Layers

- Unslotted CSMA/CA + timeout-based acknowledgements
- Time slotted channel hopping (TSCH)

Packet Structure Dissection

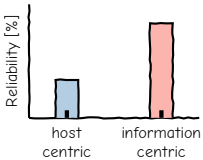


DTLS Enhancements

- Connection Identifiers (draft-ietf-tls-dtls-connection-id-07)
- DTLS 1.3 (draft-ietf-tls-dtls13-38)
  - Optimized record layer encoding and shorter header sizes
  - New handshake pattern with shorter message exchange
  - New session resumption mechanism

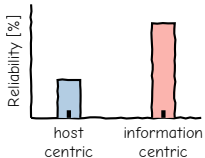
Information-Centric Networking with NDN

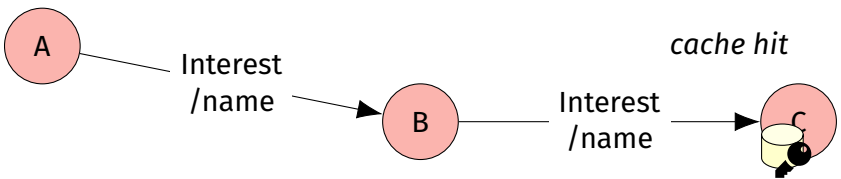
- Name-based routing & hop-wise forwarding
- In-network caching & object security
- Current research indicates higher reliability for IoT



Information-Centric Networking with NDN

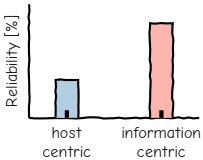
- Name-based routing & hop-wise forwarding
- In-network caching & object security
- Current research indicates higher reliability for IoT

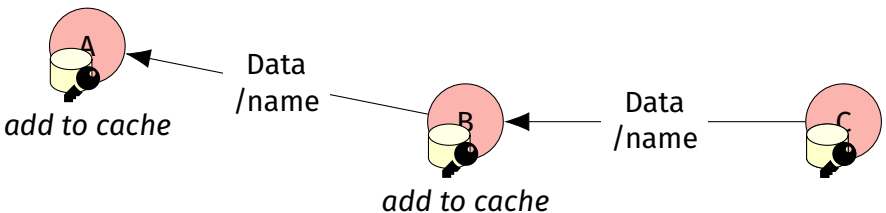




Information-Centric Networking with NDN

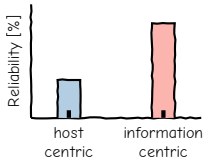
- Name-based routing & hop-wise forwarding
- In-network caching & object security
- Current research indicates higher reliability for IoT

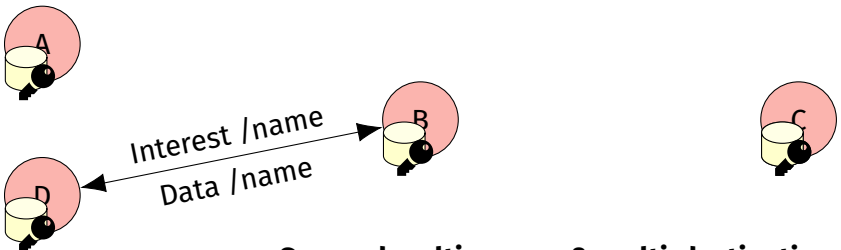




Information-Centric Networking with NDN

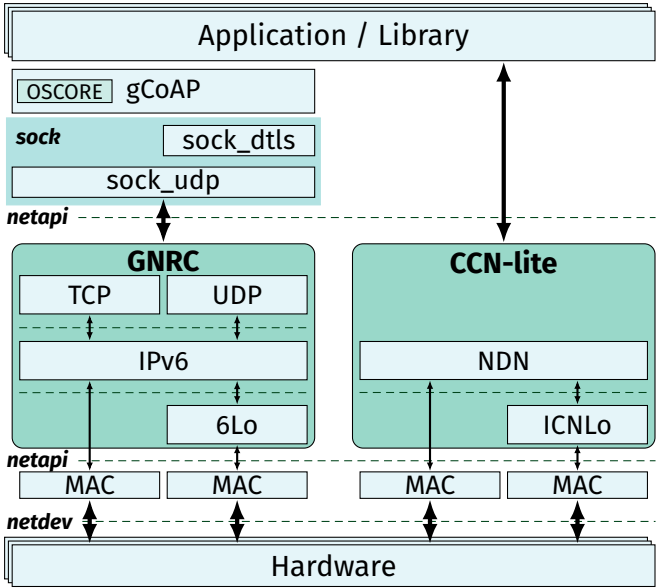
- Name-based routing & hop-wise forwarding
- In-network caching & object security
- Current research indicates higher reliability for IoT





Secured multi-source & multi-destination

RIOT Network Stack



CoAP / DTLS

- gCoAP over sock_dtls
- tinyDTLS package

CoAP / OSCORE

- gCoAP with OSCORE
- libOSCORE package

NDN

- NDN over netapi
- CCN-lite package

Authenticated Encryption with Associated Data

Encryption

Input: plaintext + key + optional plaintext header

Output: ciphertext + authentication tag

Decryption

Input: ciphertext + key + authentication tag + optional plaintext header

Output: plaintext + authentication result