CoAP

IoT/Web Application Layer Protocols

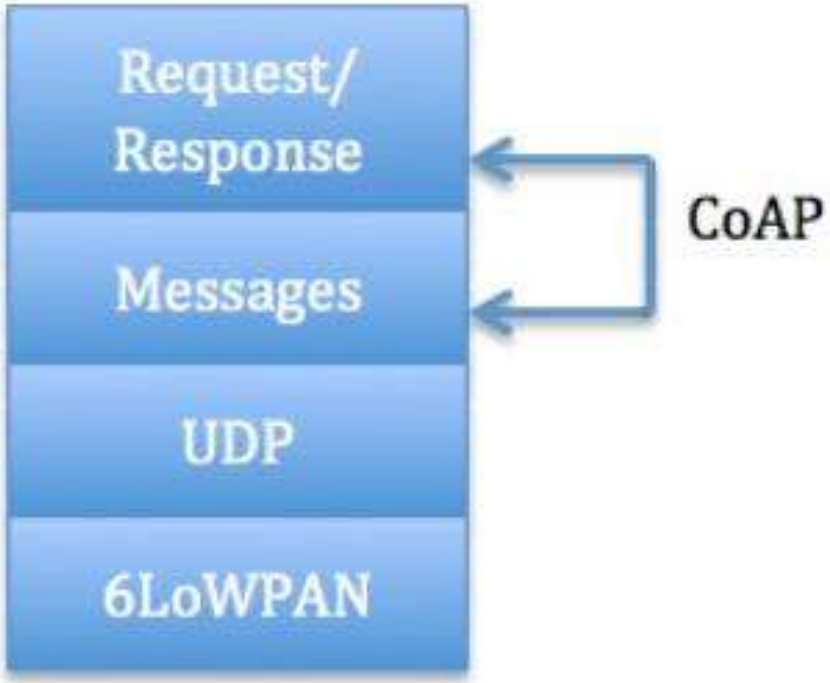
COCSC20 (Internet of Things)

CoAP (Constrained Application Protocol)

- **REST-based** web transfer protocol.
- **manipulates** Web resources using the same methods as HTTP: GET, PUT, POST, and DELETE
- **subset** of HTTP functionality re-designed for low power embedded devices such as sensors (for IoT and M2M).
- Constrained RESTful Environments (CoRE) **workgroup** who has designed CoAP.

CoAP

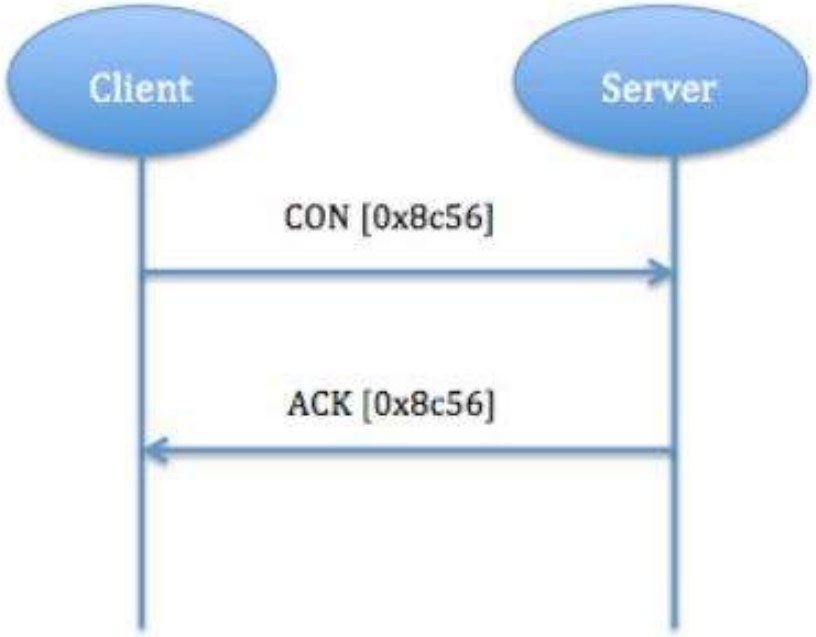
- TCP overhead is too high and its flow control is not appropriate for short-lived transactions.
- UDP has lower overhead and supports multicast.



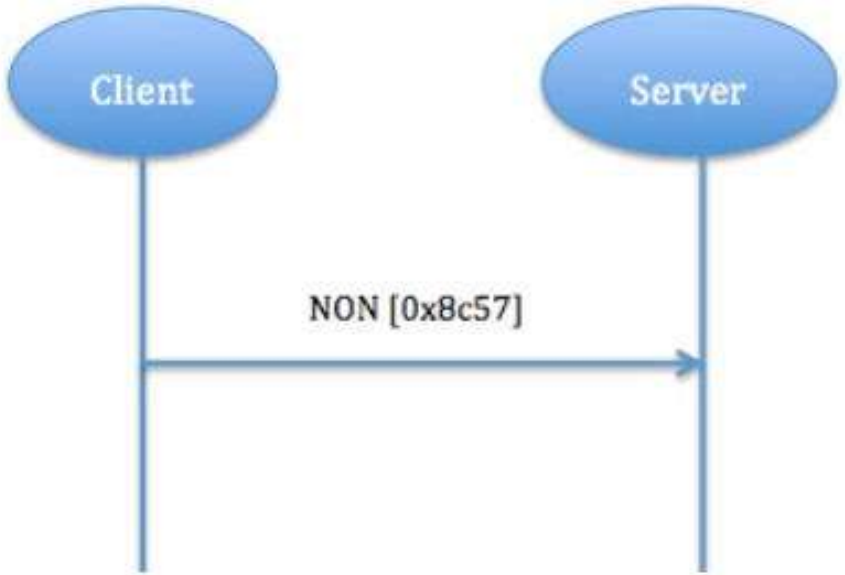
CoAP: Message layer

- Four message types:
  - **Confirmable (00)** – requires an ACK
  - **Non-confirmable (01)** – no ACK needed
  - **Acknowledgement** (10)– ACKs a Confirmable
  - **Reset** (11)- indicates a Confirmable message has been received but context is missing for processing

Reliable Message Transport

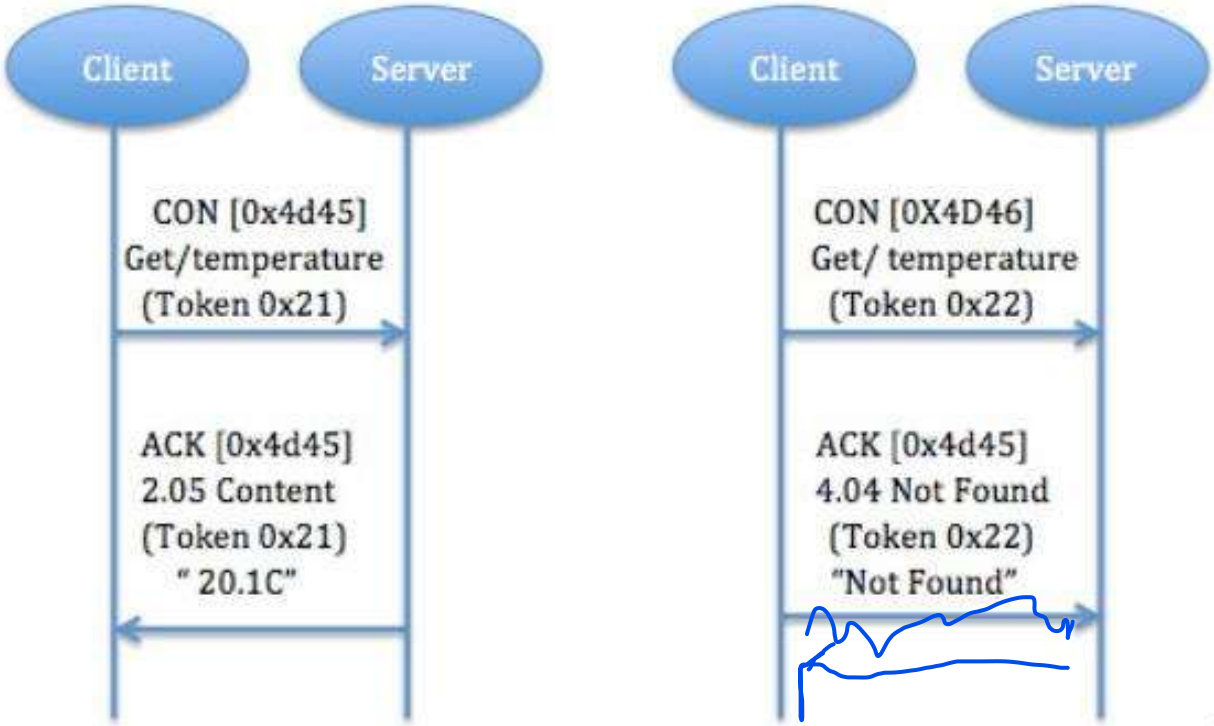


Unreliable Message Transport



CoAP: request/response Layer

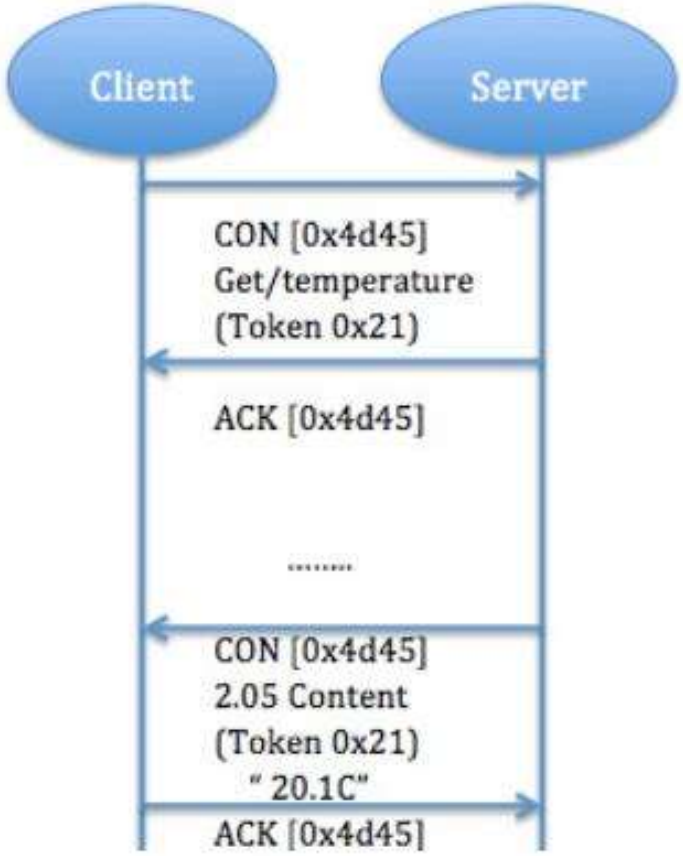
Piggy-backed



The successful and failure response results of GET method

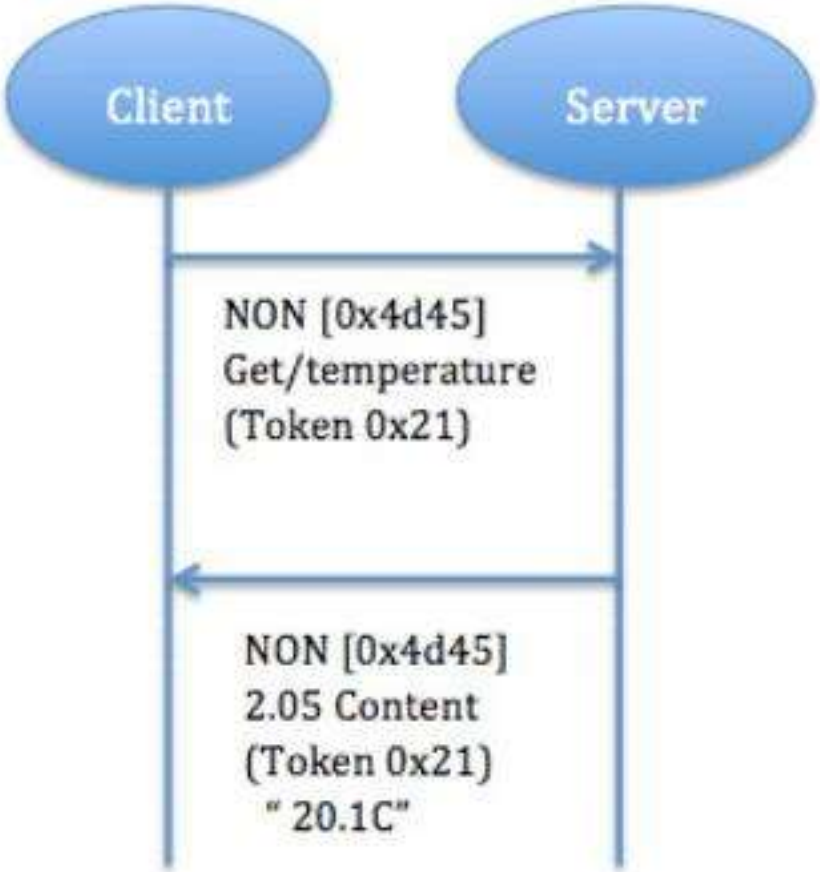
CoAP: request/response Layer

Separate response

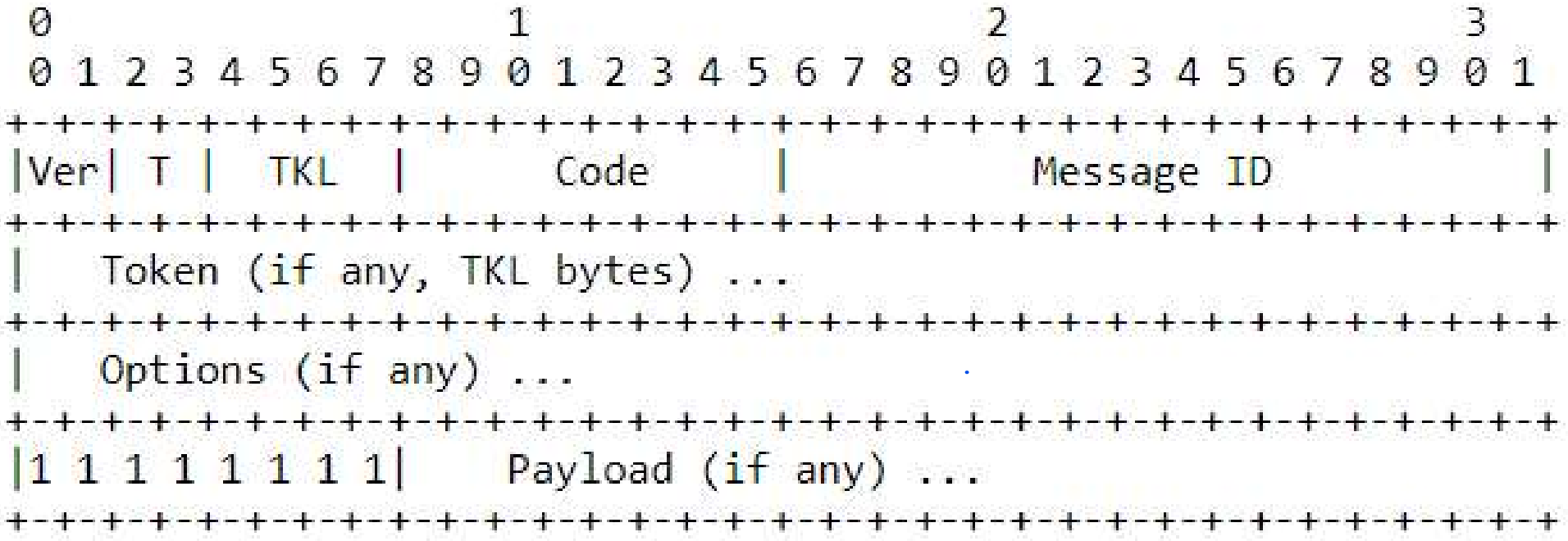


CoAP: request/response Layer

Non confirmable request and response



Message Format



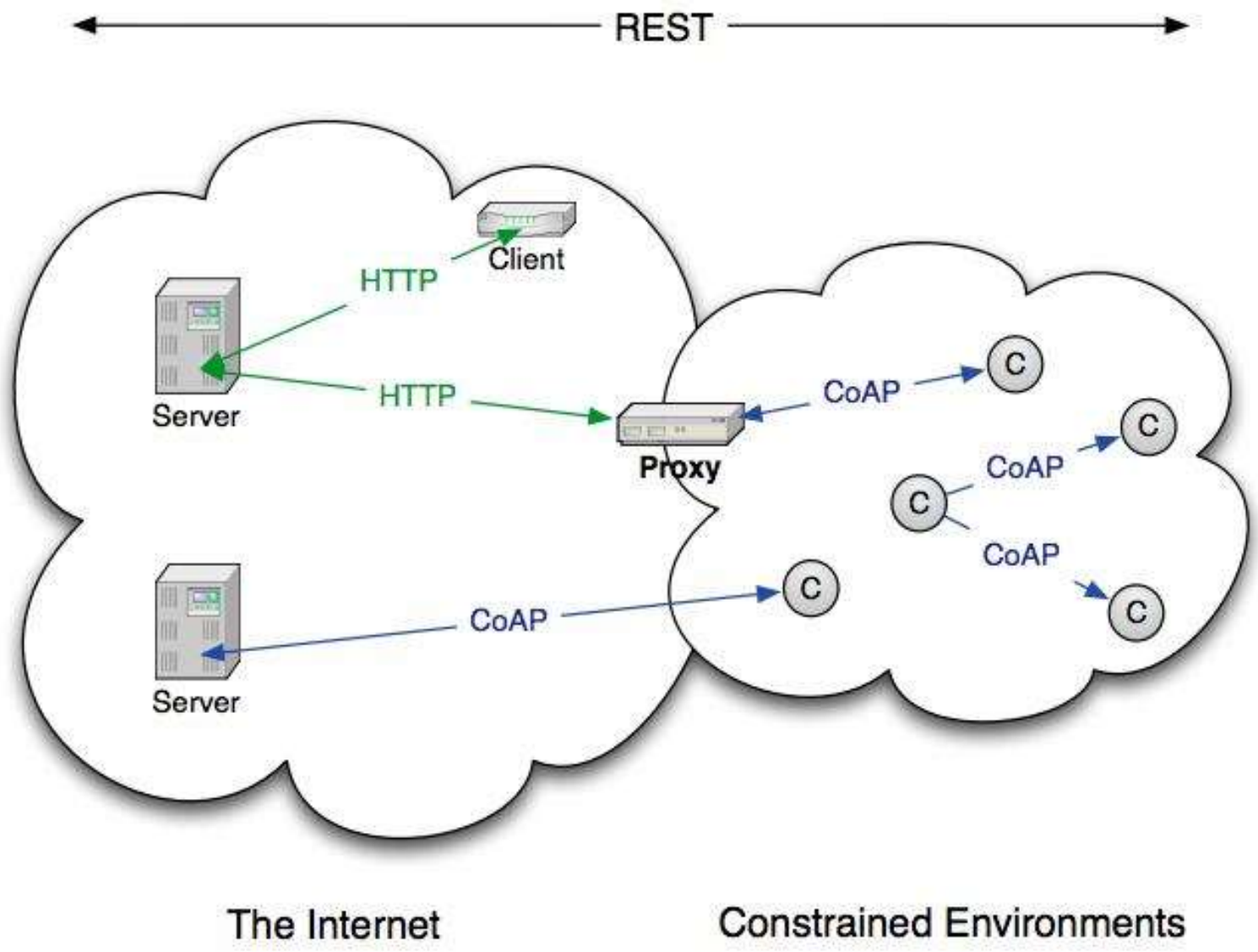
Example:

44 01 C4 09 74 65 73 74 B7 65 78 61 6D 70 6C 65

44 01 C4 09 74 65 73 74 B7 65 78 61 6D 70 6C 65

Field	HEX	Bits	Meaning
Ver	44	01	Version 01, which is mandatory here.
T		00	Type 0: confirmable.
TKL		0100	Token length: 4.
Code	01	000 00001	Code: 0.01, which indicates the GET method.
Message ID	C4 09	2 Bytes equal to hex at left	Message ID. The response message will have the same ID. This can help out identification.
Token	74 65 73 74	4 Bytes equal to hex at left	Token. The response message will have the same token. This can help out identification.
Option delta	B7	1011	Delta option: 11 indicates the option data is Uri-Path.
Option length		0111	Delta length: 7 indicates there are 7 bytes of data following as a part of this delta option.
Option value	65 78 61 6D 70 6C 65	7 Bytes equal to hex at left	Example.

CoAP


REST
Client
HTTP
Server
HTTP
Proxy
CoAP
C
CoAP
C
C
CoAP
C
Server
CoAP
C
The Internet
Constrained Environments

CoAP

- CoAP provides reliability without using TCP as transport protocol.
- CoAP enables asynchronous communication
  - e.g., when CoAP server receives a request which it cannot handle immediately, it first ACKs the reception of the message and sends back the response in an off-line fashion
- Also supports multicast and congestion control.

What CoAP Is

- CoAP is
  - A RESTful protocol
  - Both synchronous and asynchronous
  - For constrained devices and networks
  - Specialized for M2M applications
  - Easy to proxy to/from HTTP

Comparison between CoAP & MQTT

Both used in IoT

- CoAP:
  - One-to-one communication
  - UDP/IP
  - Unreliable
  - Lightweight and easy to implement

- MQTT:
  - Many-to-many communication
  - TCP/IP
  - Focus on message delivery; reliable
  - Higher overheads (protocol data, processing costs)

Performance Evaluation - Background

Criteria	HTTP	CoAP	MQTT
Architecture	Client/Server	Client/Server or Client/Broker	Client/Broker
Abstraction	Request/Response	Request/Response or Publish/ Subscribe	Publish/Subscribe
Header Size	Undefined	4 Byte	2 Byte
Message size	Large and Undefined (depends on the web server or the programming technology)	Small and Undefined (normally small to fit in single IP datagram)	Small and Undefined (up to 256 MB maximum size)
Semantics/Methods	Get, Post, Head, Put, Patch, Options, Connect, Delete	Get, Post, Put, Delete	Connect, Disconnect, Publish, Subscribe, Unsubscribe, Close
Quality of Service (QoS) /Reliability	Limited (via Transport Protocol - TCP)	Confirmable Message or Non-confirmable Message	QoS 0 - At most once QoS 1 - At least once QoS 2 - Exactly once
Transport Protocol	TCP	UDP, TCP	TCP (MQTT-SN can use UDP)
Security	TLS/SSL	DTLS/IPSEC	TLS/SSL
Default Port	80/443 (TLS/SSL)	5683 (UDP)/5684 (DTLS)	1883/8883 (TLS/SSL)

* N. Naik, "Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP," 2017 IEEE International Systems Engineering Symposium (ISSE), Vienna, 2017, pp. 1-7.

Thank You