Internet of Things
IO 4041
Application Layer Protocols

COnstrained Application Protocol (CoAP)

CoAP

- A specialized web transfer protocol for use with constrained nodes and constrained (e.g., low-power, lossy) networks.
  - Nodes often have 8-bit microcontrollers with small amounts of ROM and RAM, while
  - constrained networks such 6LoWPANs often have high packet error rates and a typical throughput of 10s of kbit/s.
- provides a request/response interaction model between application endpoints,
- supports built-in discovery of services and resources, and includes key concepts of the Web such as URIs and Internet media types.

CoAP Features

- designed to easily interface with HTTP for integration with the Web while meeting specialized requirements
  - such as multicast support, very low overhead, and simplicity for constrained environments.
  - supports the basic methods of **POST (**create**), GET (read), PUT** (update/replace), **DELETE (**delete**)**, which are easily mapped to HTTP

CoAP Features

- designed for fulfilling requirements of machine- to-machine (M2M) applications such as smart energy and building automation

- Low header overhead and parsing complexity.

CoAP: Features

- UDP binding with optional **reliability** supporting unicast and multicast requests.

To achieve optional reliability (like in TCP)

- CoAP defines *confirmable messages* and *non-confirmable messages*
  - to define its own reliability mechanism.
- The former requires an ACK while
- the latter does not require any kind of ACK.

CoAP: Features

- Asynchronous message exchanges.
  - **Asynchronous is a communication method wherein the system puts a message in a message queue and does not require an immediate response to continue processing.**
  - Examples include a request for information, explanation or data needed but not needed immediately.

CoAP: Features

Uniform Resource Identifier (**URI)** **and Content-type support**

- A URI is a string of characters that uniquely *identifies a resource by name, location, or both on the internet.*
- **URI** is like name, and a **URL** is a specific subtype of URI i.e., like name combined with address.
- The **Content-Type** header is used to indicate the media type of the resource.
  - The media type is a string sent along with the file indicating the format of the file.
  - For example, for image file its media type will be like image/png or image/jpg.
  - it tells about the type of returned content, to the client.
- Simple proxy and caching capabilities.

CoAP: Terminology

- **Endpoint**: An entity participating in the CoAP protocol
- **Sender**: The originating endpoint of a message.

- **Recipient:** The destination endpoint of a message

- **Client:** The originating endpoint of a request; the destination endpoint of a response.

- **Server:** The destination endpoint of a request; the originating endpoint of a response.

CoAP: Terminology

- **Origin Server:** The server on which a given resource resides or is to be created.
- **Intermediatory:** An endpoint that acts both as a server and as a client towards an origin server (possibly via further intermediaries).
  - A common form of an intermediary is a proxy;
- **Proxy:** An intermediary that mainly is concerned with forwarding requests and relaying back responses
  - two common forms of proxy: forward-proxy and reverse-proxy
  - sometime, a single endpoint might act as an origin server, forward-proxy, or reverse-proxy, switching behavior based on the nature of each request.

CoAP: Terminology

- **Confirmable messages:** messages which require ACK.
- **Non-confirmable messages:** messages not requiring ACK i.e., repeated readings from sensors
- **ACK message:** ACKing arrival of conformation message
- **Reset message:** indicates that a specific message (Confirmable or Non-confirmable) was received, but some context is missing to properly process it.
- **Piggybacked Response:** included in a CoAP ACK message that is sent to acknowledge receipt of the Request for this Response
- **Separate Response:** When a Confirmable message carrying a request is acknowledged with an Empty message. a Separate Response is sent in a separate message exchange
- **Empty message:** A message with a Code of 0.00; neither a request nor a response. An Empty message only contains header

Client Server

HTTP	CoAP
request/response model	request/response model
synchronous	Asynchronous
ASCII based (more complex client)	Binary (simple client)
connection oriented via TCP	connection less via UDP
more bytes to pay on data transfer	Less bytes to transfer

CoAP

- Lightweight, efficient protocol
  - For constrained node networks
  - Over UDP
  - Four types of Messages
    - Confirmable (CON)
    - Non-confirmable (NON)
    - ACKnowledgement
    - Reset

CoAP

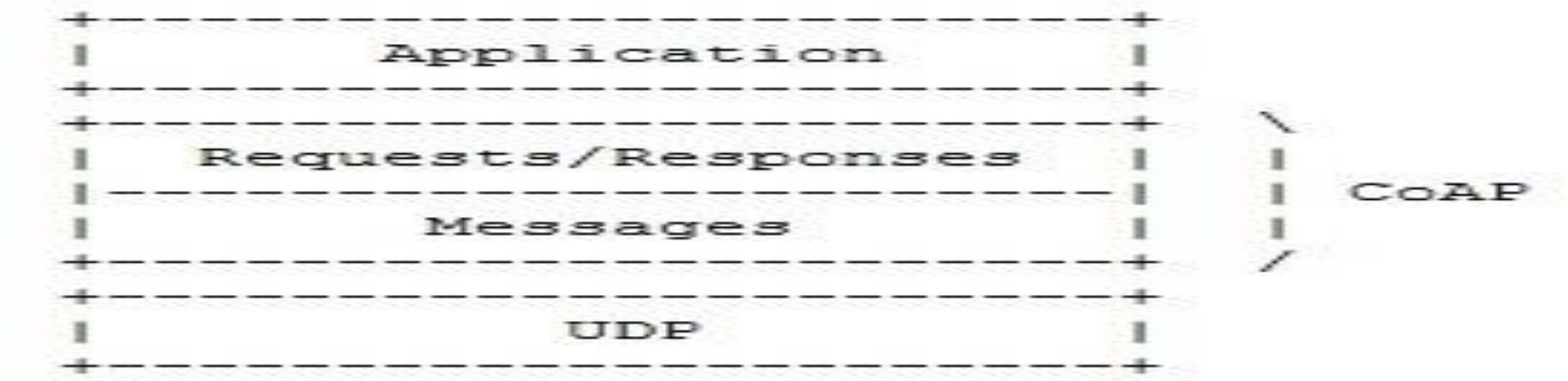
- The basic exchanges of the four types of messages are somewhat orthogonal to the request/response interactions;
- requests can be carried in Confirmable and Non- confirmable messages, and
- responses can be carried in these as well as piggybacked in Acknowledgement messages.

CoAP

Logically, CoAP uses a two-layer approach – abstract layers (actually, it is a single layer)

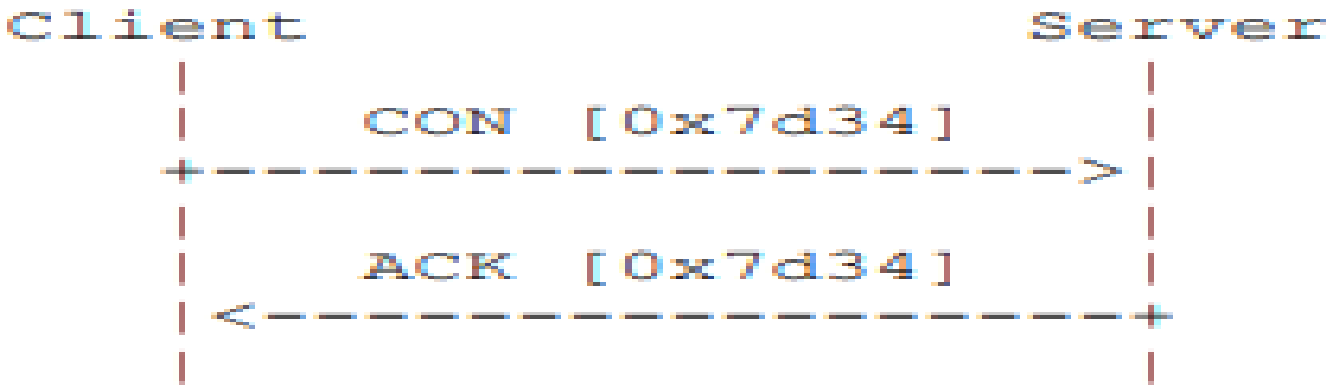
- a CoAP messaging layer used to deal with UDP and the asynchronous nature of the interactions, and

- the request/response interactions using Method and Response Codes



CoAP: Messaging model

- based on the exchange of messages over UDP between endpoints.
- uses a short fixed-length binary header (4 bytes)
  - may be followed by compact binary options and a payload.
- Each message contains a Message ID used to detect duplicates and for optional reliability.
- Reliability is provided by marking a message as Confirmable (CON).



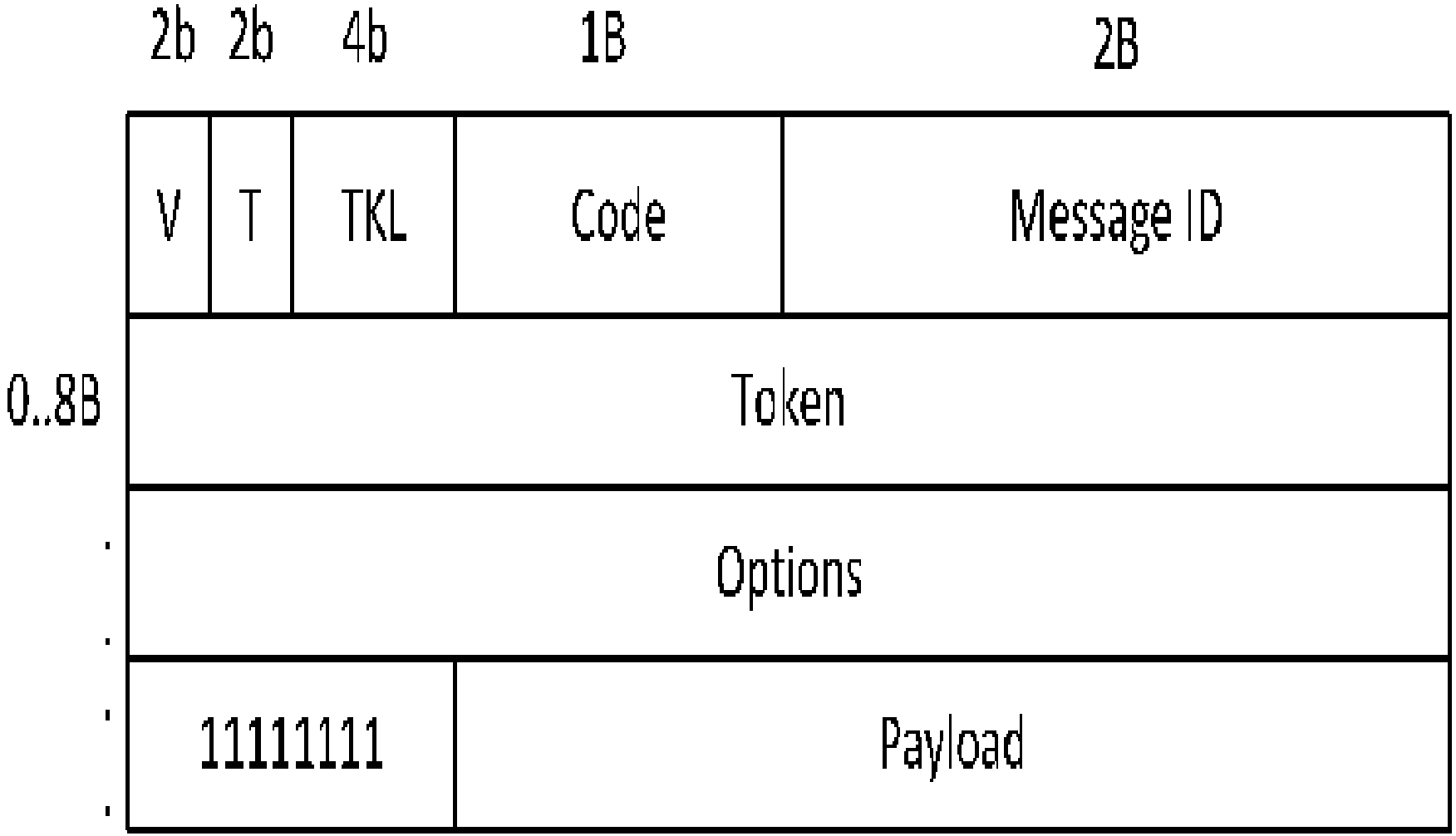
CoAP: Messaging model

- A CON message is retransmitted using a default timeout and exponential back-off between retransmissions,
  - until the recipient sends an ACK with the same Message ID
- When a recipient is not at all able to process a CON message (i.e., not even able to provide a suitable error response),
  - it replies with a Reset message (RST) instead of an ACK.

CoAP: Messaging model

- A message that does not require reliable transmission
  - for example, each single measurement out of a stream of sensor data) can be sent as a Non-confirmable message (NON).
  - These are not ACKed, but still have a Message ID for duplicate detection.
- When a recipient is not able to process a Non-confirmable message,
  - it may reply with a Reset message (RST).

CoAP: Message Fromat



CoAP: Message Fromat

- **CoAP is based on the exchange of compact messages that, by default, are transported over UDP**
- CoAP may also be used over
  - Datagram Transport Layer Security (DTLS).
  - other transports such as SMS, TCP, or SCTP
  - Out of scope here
- CoAP messages are encoded in a simple binary format.
- The message format starts with a fixed-size 4-byte header.
- followed by a variable-length Token value, which can be between 0 and 8 bytes long

CoAP: Message Fromat

- Then comes a sequence of zero or more CoAP Options in Type-Length-Value (TLV) format,
- optionally followed by a payload that takes up the rest of the datagram.

- Version (Ver): 2-bit unsigned integer: CoAP version number (here 01)
  - Messages with unknown version numbers MUST be silently ignored
- Type (T): 2-bits, indicates type of message
  - 0 for CON, 1 for NON, 2 for ACK, 3 for Reset

CoAP: Message Fromat

- Token Length (TKL): 4-bits, Indicates the length of the variable-length Token field (0-8 bytes)
  - Lengths 9-15 are reserved,
  - MUST NOT be sent, and
  - MUST be processed as a message format error.
- Code: 8-bits: divided into 2 parts (c.dd)
  - **Class:** 3 most significant bits
  - Detail: 5 least significant bits
  - c is 0 to 7 (3 bit sub field) and dd is 0 to 31 (5 bit subfield)

CoAP: Message Fromat

- Class can indicate a
  - request (0), a success response (2), a client error response (4), or a server error response (5).
  - All other class values are reserved

- As a special case, Code 0.00 indicates an Empty message.

- In case of a request, the Code field indicates the Request Method;

- in case of a response, a Response Code.

CoAP: Message Fromat

- Message ID: 16-bits
- The header is followed by the Token value, which may be 0 to 8 bytes, as given by the Token Length field.
- The Token value is used to correlate requests and responses.
- The rules for generating a Token and correlating requests and responses are out of scope

CoAP: Message Fromat

- Header and Token are followed by zero or more Options
- An Option can be followed by the end of the message, by another Option, or by the Payload Marker and the payload.
  -
- Following the header, token, and options, if any, comes the optional payload.
  - If present and of non-zero length, it is prefixed by a fixed, one-byte Payload Marker (0xFF),
  - which indicates the end of options and the start of the payload.