- IoT Protocols
- CoAP – Constrained Application Protocol
- XMPP – Extensible Messaging and Presence Protocol

- CoAP – Constrained Application Protocol

- Introduction
- CoAP Position in the Protocol Stack
- CoAP Message Types
- CoAP Request-Response Model
- CoAP Features

- XMPP - Extensible Messaging and Presence Protocol

- Introduction
- Highlights of the XMPP protocol
- Core XMPP Technologies
- Weaknesses of XMPP
- Applications of XMPP

CoAP

Introduction

- ✓ CoAP – **Constrained Application Protocol**.
- ✓ **Web transfer protocol** for use with constrained nodes and networks.
- ✓ **Designed for Machine to Machine** (M2M) applications such as smart energy and building automation.
- ✓ Based on **Request-Response model** between end-points
- ✓ Client-Server interaction is **asynchronous over a datagram oriented transport protocol** such as UDP

Source: Z. Shelby , K. Hartke, C. Bormann**, "The Constrained Application Protocol (CoAP)",** Internet Engineering Task Force (IETF), Standards Track, 2014

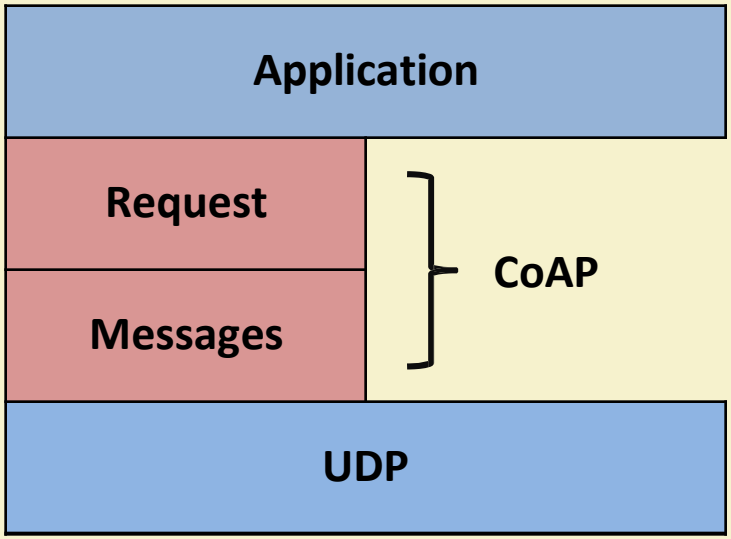
- ✓ The Constrained Application Protocol (CoAP) is a session layer protocol designed by IETF Constrained RESTful Environment (CoRE) working group to provide lightweight RESTful (HTTP) interface.
- ✓ Representational State Transfer (REST) is the standard interface between HTTP client and servers.
- ✓ Lightweight applications such as those in IoT, could result in significant overhead and power consumption by REST.
- ✓ CoAP is designed to enable low-power sensors to use RESTful services while meeting their power constraints.

Source: Z. Shelby , K. Hartke, C. Bormann**, "The Constrained Application Protocol (CoAP)",** Internet Engineering Task Force (IETF), Standards Track, 2014

- ✓ Built over UDP, instead of TCP (which is commonly used with HTTP) and has a light mechanism to provide reliability.
- ✓ CoAP architecture is divided into two main sub-layers:
  - Messaging
  - Request/response.
- ✓ The messaging sub-layer is responsible for reliability and duplication of messages, while the request/response sub-layer is responsible for communication.
- ✓ CoAP has four messaging modes:
  - Confirmable
  - Non-confirmable
  - Piggyback
  - Separate

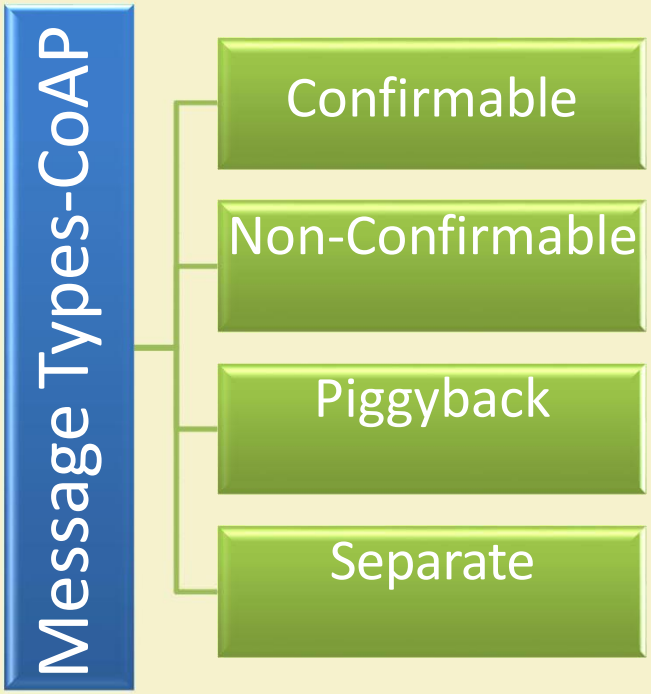
Source: V. Karagiannis, P. Chatzimisios, F. Vazquez-Gallego, and J. Alonso-Zarate, "A survey on application layer protocols for the internet of things," Transaction on IoT and Cloud Computing, vol. 3, no. 1, pp. 11-17, 2015

CoAP Position



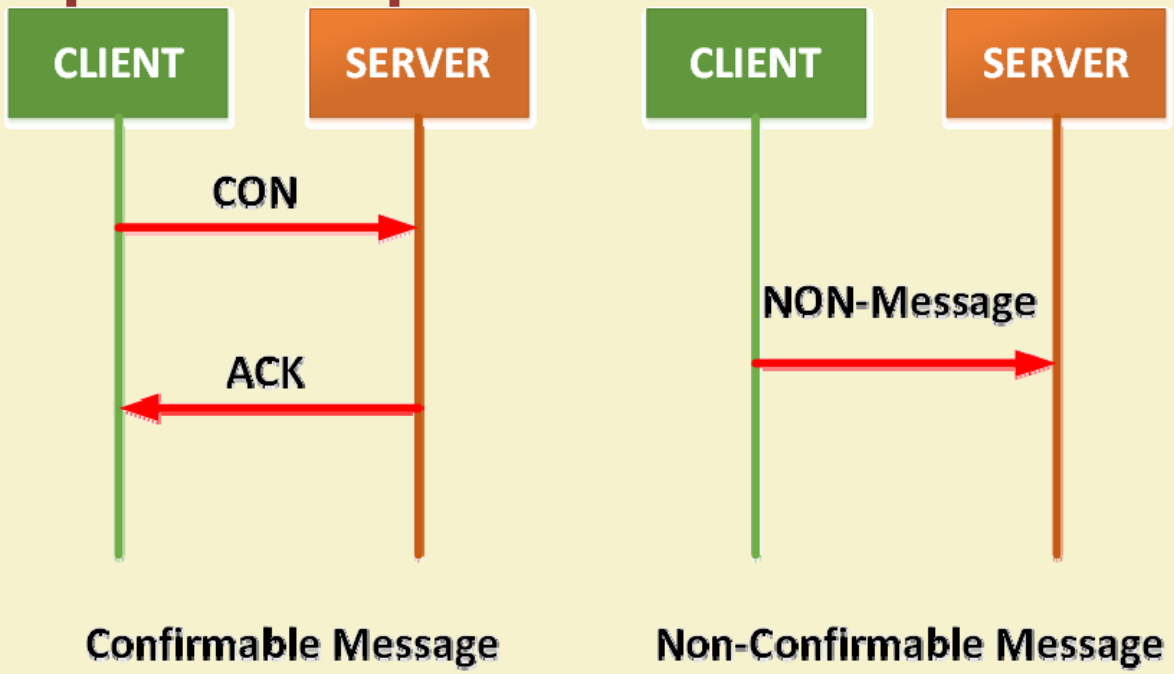
Source: Z. Shelby , K. Hartke, C. Bormann**, "The Constrained Application Protocol (CoAP)",** Internet Engineering Task Force (IETF), Standards Track, 2014

CoAP Message Types



Source: Z. Shelby , K. Hartke, C. Bormann**, "The Constrained Application Protocol (CoAP)",** Internet Engineering Task Force (IETF), Standards Track, 2014

CoAP Request-Response Model

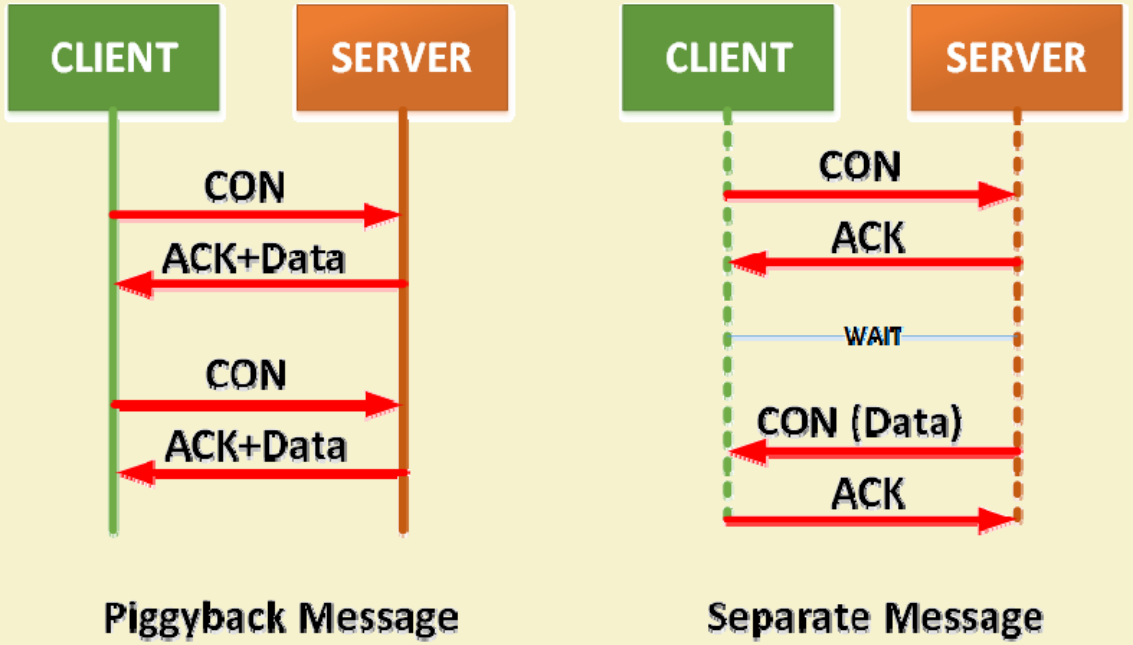


Source: V. Karagiannis, P. Chatzimisios, F. Vazquez-Gallego, and J. Alonso-Zarate, "A survey on application layer protocols for the internet of things," Transaction on IoT and Cloud Computing, vol. 3, no. 1, pp. 11-17, 2015

- ✓ Confirmable and non-confirmable modes represent the reliable and unreliable transmissions, respectively, while the other modes are used for request/response.
- ✓ Piggyback is used for client/server direct communication where the server sends its response directly after receiving the message, i.e., within the acknowledgment message.
- ✓ On the other hand, the separate mode is used when the server response comes in a message separate from the acknowledgment, and may take some time to be sent by the server.
- ✓ Similar to HTTP, CoAP utilizes GET, PUT, PUSH, DELETE messages requests to retrieve, create, update, and delete, respectively

Source: V. Karagiannis, P. Chatzimisios, F. Vazquez-Gallego, and J. Alonso-Zarate, "A survey on application layer protocols for the internet of things," Transaction on IoT and Cloud Computing, vol. 3, no. 1, pp. 11-17, 2015

CoAP Request-Response Model



Piggyback Message

Separate Message

Source: V. Karagiannis, P. Chatzimisios, F. Vazquez-Gallego, and J. Alonso-Zarate, "A survey on application layer protocols for the internet of things," Transaction on IoT and Cloud Computing, vol. 3, no. 1, pp. 11-17, 2015

Features

- ✓ Reduced overheads and parsing complexity.
- ✓ URL and content-type support.
- ✓ Support for the discovery of resources provided by known CoAP services.
- ✓ Simple subscription for a resource, and resulting push notifications.
- ✓ Simple caching based on maximum message age.

Source: "Constrained Application Protocol", Wikipedia (Online)

XMPP

Introduction

- ✓ **XMPP – Extensible Messaging and Presence Protocol.**
- ✓ A communication protocol for **message-oriented middleware** based on XML (Extensible Markup Language).
- ✓ Real-time exchange of structured data.
- ✓ It is an open standard protocol.

Source: "XMPP", Wikipedia (Online)

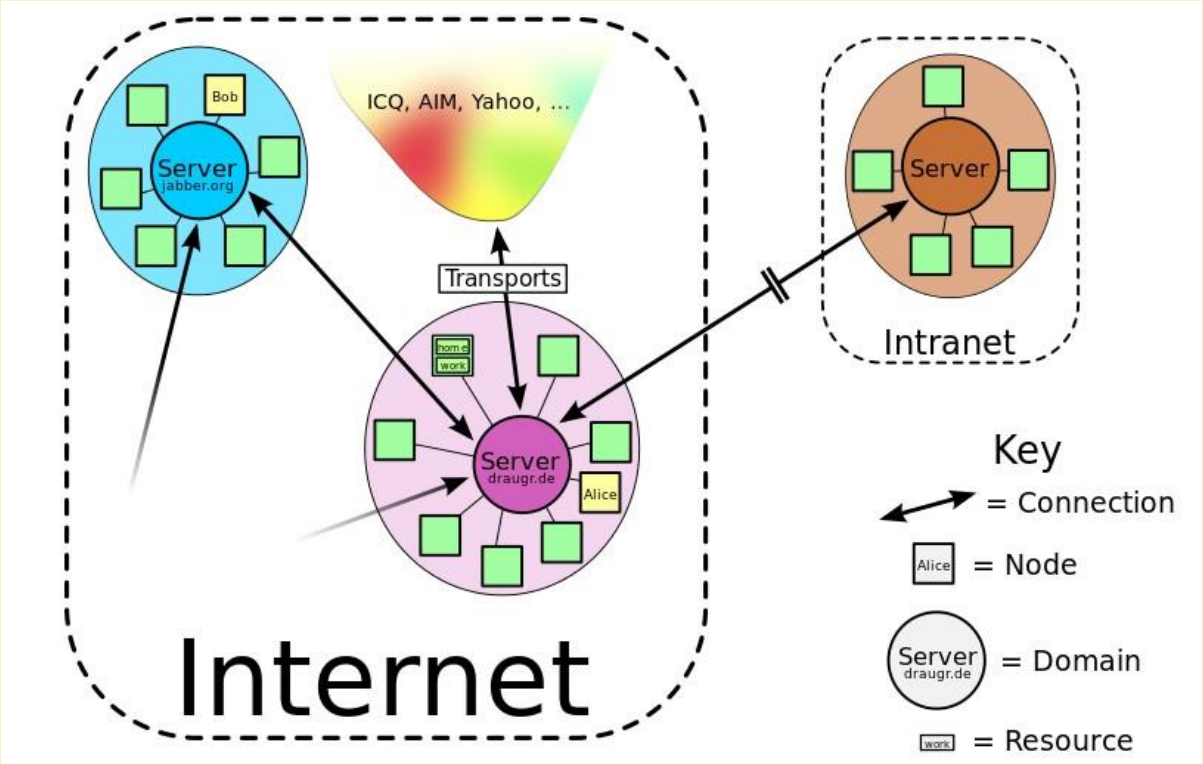
- ✓ XMPP uses a **client-server architecture**.
- ✓ As the model is **decentralized**, no central server is required.
- ✓ XMPP provides for the **discovery of services** residing locally or across a network, and the **availability information** of these services.
- ✓ Well-suited for cloud computing where virtual machines, networks, and firewalls would otherwise present obstacles to alternative service discovery and presence-based solutions.
- ✓ Open means to support machine-to-machine or peer-to-peer communications across a diverse set of networks.

Source: "XMPP", Wikipedia (Online)

Highlights

- ✓ Decentralization – No central server; anyone can run their own XMPP server.
- ✓ Open standards – No royalties or granted permissions are required to implement these specifications
- ✓ Security – Authentication, encryption, etc.
- ✓ Flexibility – Supports interoperability

Source: "XMPP", Wikipedia (Online)



Source: "JabberNetwork.svg", Wikimedia Commons (Online)

Core XMPP Technologies

Core
- information about the core XMPP technologies for XML streaming

Jingle
- multimedia signalling for voice, video, file transfer

Multi-user Chat
- flexible, multi-party communication

PubSub
- alerts and notifications for data syndication

BOSH
- HTTP binding for XMPP

Source: "XMPP: Technology Overview", XMPP.org (Online)

Weaknesses

- ✓ Does not support QoS.
- ✓ Text based communications induces higher network overheads.
- ✓ Binary data must be first encoded to base64 before transmission.

Applications

- ✓ Publish-subscribe systems
- ✓ Signaling for VoIP
- ✓ Video
- ✓ File transfer
- ✓ Gaming
- ✓ Internet of Things applications
  - Smart grid
  - Social networking services