# Computer Networks & Internet of Things

## Module

## Internet of Things Protocols


Jaypee Institute of Information Technology, Noida

# Overview

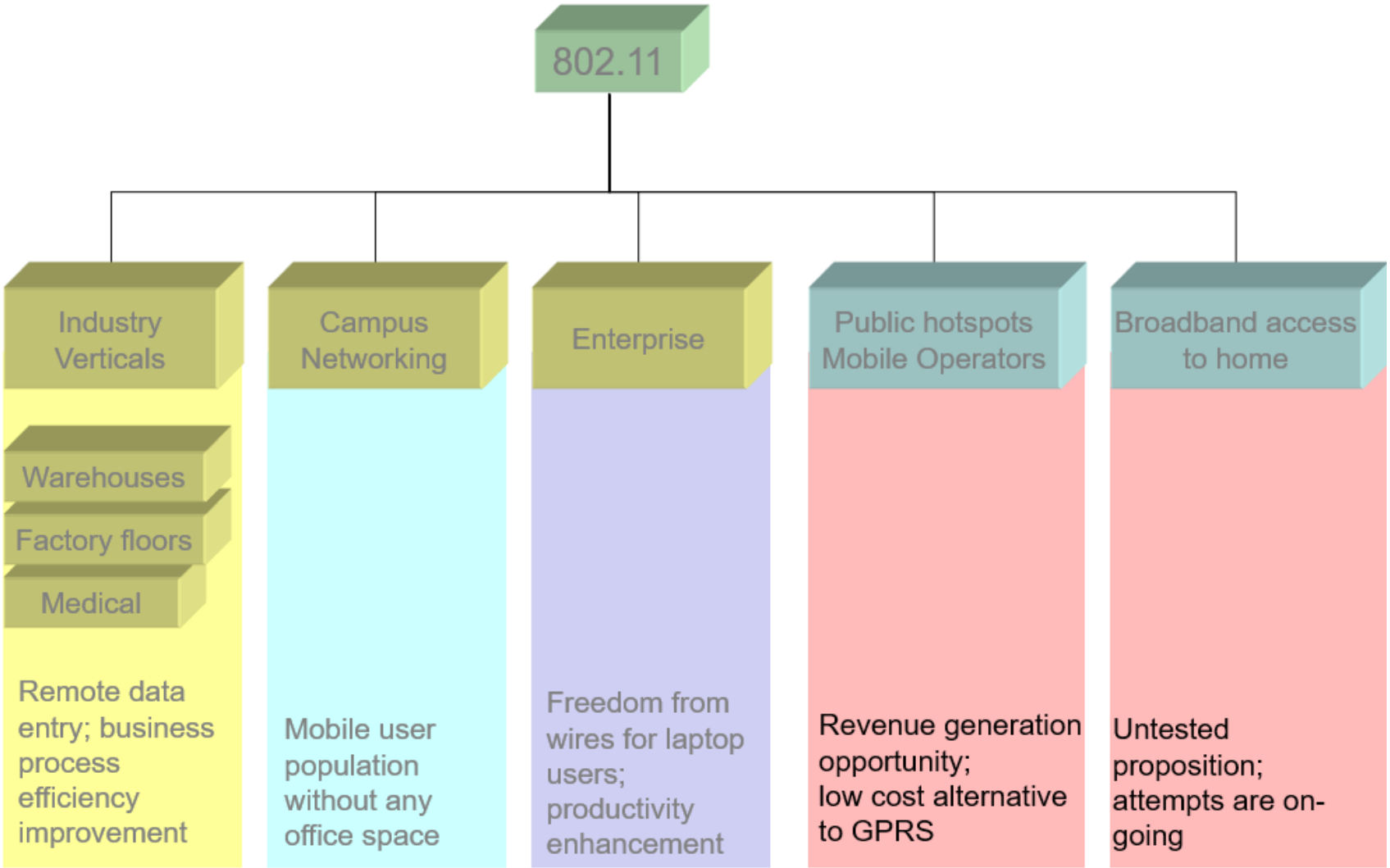
- "Internet of Things" made of two words, **Things** and the **Internet**.
- the **Internet** is used for global communication by enabling information sharing over geographic location in pervasive environment.
- Other one is **Things**, which refer to the actual hardware device to perform the monitoring tasks. Moreover, device is also capable to perform actuating, remote sensing and live monitoring.
- Based on the above, we have two types of IoT protocols:
  1. Communication Protocols
  2. IoT Data Protocols

- **Communication Protocols:**
  IoT network protocols connect medium to high power devices over the network. End-to-end data communication within the network is allowed using this protocol. HTTP, LoRaWAN, Bluetooth, Zigbee are a few popular IoT network protocols.
- **IoT Data Protocols:**
  IoT data protocols connect low power IoT devices. Without any Internet connection, these protocols can provide end-to-end communication with the hardware. Connectivity in IoT data protocols can be done via a wired or cellular network. MQTT, CoAP, AMQP, XMPP, DDS are some popular IoT data protocols.

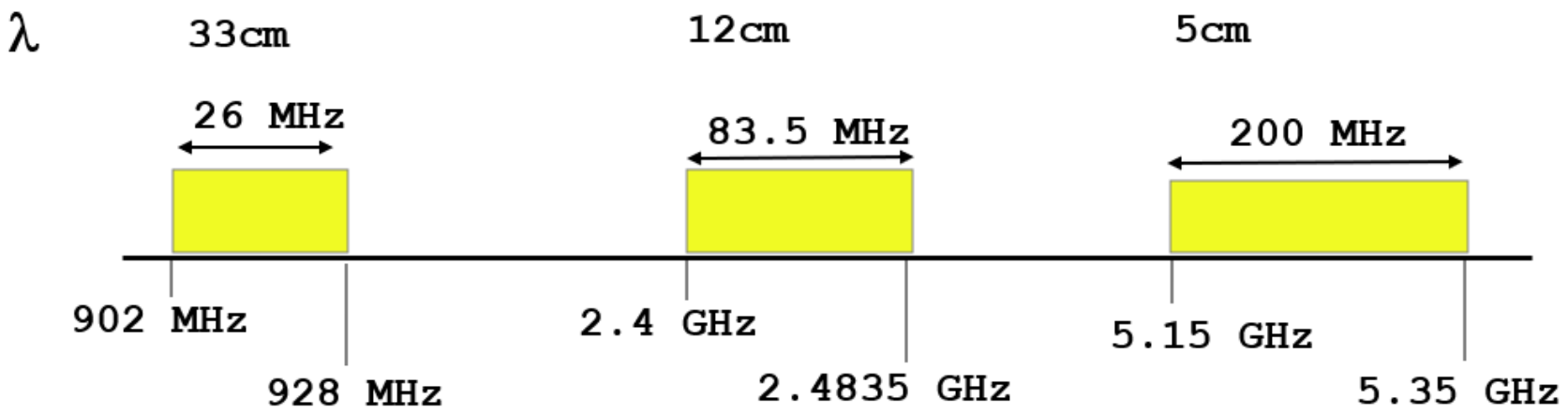
| | |
|---|---|
| CoAP | MQTT |
| UDP | TCP |
| IPV6 | |
| 6LoWPAN | |
| IEEE 802.15.4 (MAC / PHY) | |

- **IPV6**, is an Internet Layer protocol for packet-switched internetworking and provides end-to-end datagram transmission across multiple IP networks.
- **6LoWPAN** is an acronym of IPv6 over Low power Wireless Personal Area Networks. It is an adaption layer for IPv6 over IEEE802.15.4 links. This protocol operates only in the 2.4 GHz frequency range with 250 kbps transfer rate.
- **User Datagram Protocol (UDP)**, A simple OSI transport layer protocol for client/server network applications based on Internet Protocol (IP).
- **UDP** is the main alternative to TCP and one of the oldest network protocols in existence, introduced in 1980.
- **UDP** is often used in applications specially tuned for real-time performance

# IEEE 802.11

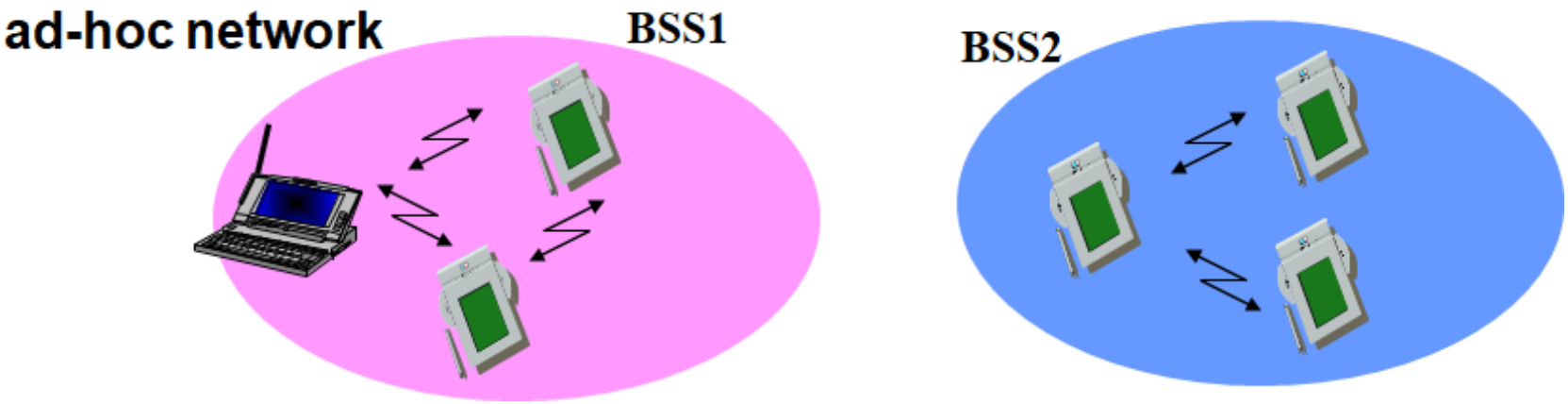

802.11
Industry Verticals
Warehouses
Factory floors
Medical
Remote data entry; business process efficiency improvement
Campus Networking
Mobile user population without any office space
Enterprise
Freedom from wires for laptop users; productivity enhancement
Public hotspots Mobile Operators
Revenue generation opportunity; low cost alternative to GPRS
Broadband access to home
Untested proposition; attempts are on-going

- Wireless LAN standard defined in the unlicensed spectrum (2.4 GHz and 5 GHz U-NII bands).
- Standards covers the MAC sublayer and PHY layers
- Three different physical layers in the 2.4 GHz band
  - FHSS, DSSS and IR
- OFDM based Phys layer in the 5 GHz band (802.11a)

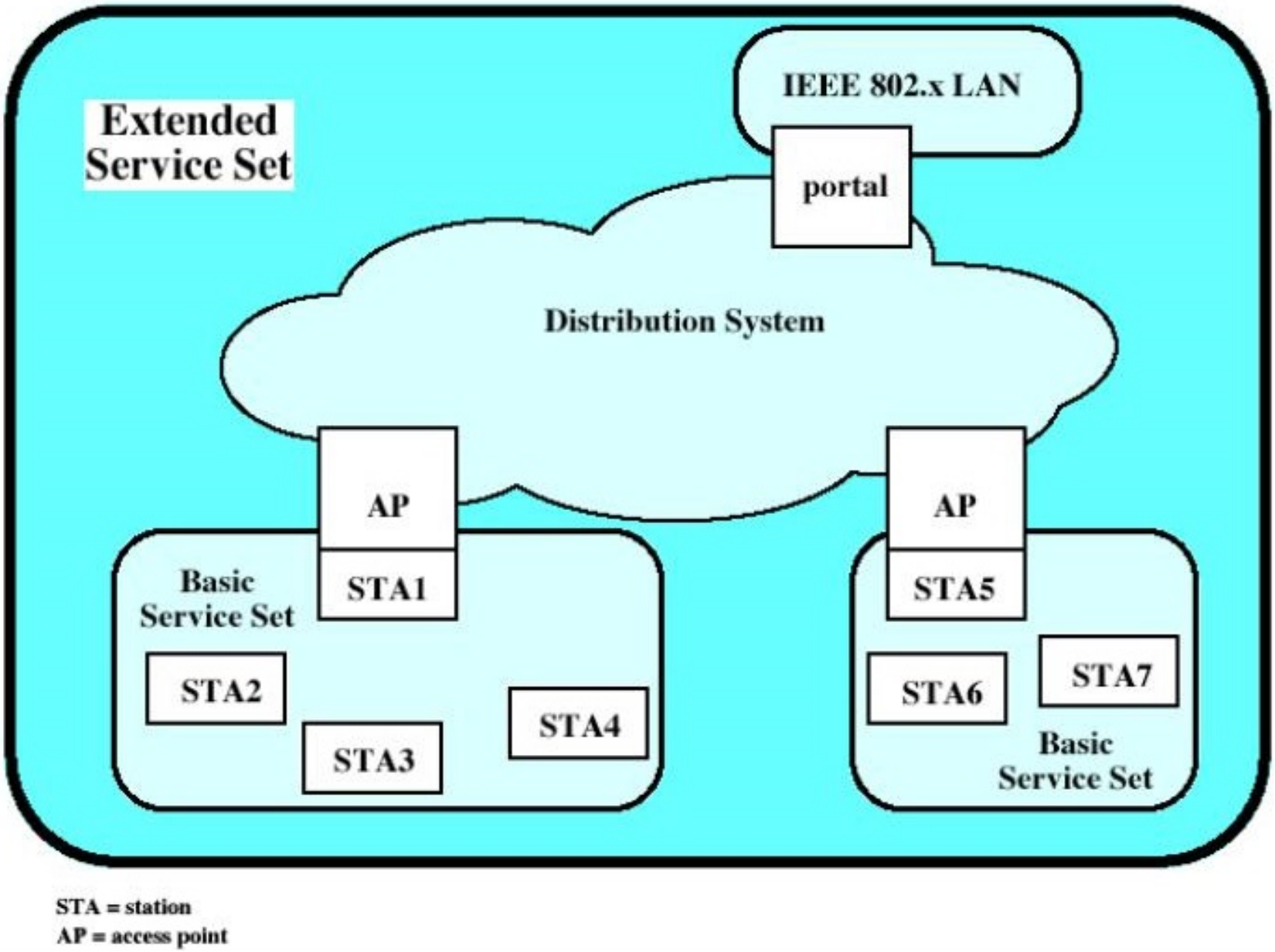


# IEEE 802.11 components

- The basic service set (BSS) is the basic building block of an IEEE 802.11 LAN
- The ovals can be thought of as the coverage area within which member stations can directly communicate
- The Independent BSS (IBSS) is the simplest LAN. It may consist of as few as two stations

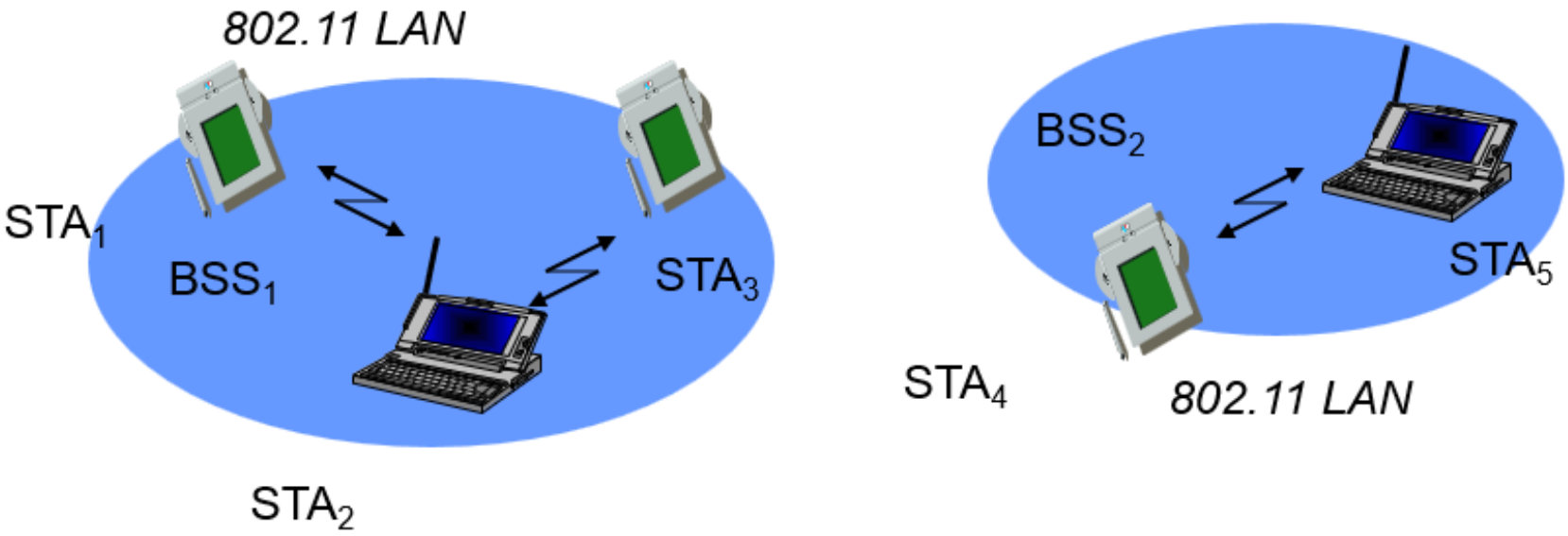


# IEEE 802.11 Architecture



# IEEE 802.11 ad-hoc network

- Direct communication within a limited range
  - Station (STA) : terminal with access mechanisms to the wireless medium
  - Basic Service Set (BSS) : group of stations using the same radio frequency



# LoRaWAN

- LoRaWAN is a low-power, wide area networking protocol built on top of the LoRa radio modulation technique. It wirelessly connects devices to the internet and manages communication between end-node devices and network gateways.
- Usage of LoRaWAN in industrial spaces and smart cities is growing because it is an affordable long-range, bi-directional communication protocol with very low power consumption devices can run for ten years on a small battery.
- It uses the unlicensed ISM (Industrial, Scientific, Medical) radio bands for network deployments.
- An end device can connect to a network with LoRaWAN in two ways:
  1. Over-the-air Activation (OTAA): A device has to establish a network key and an application session key to connect with the network.
  2. Activation by Personalization (ABP): A device is hardcoded with keys needed to communicate with the network, making for a less secure but easier connection.

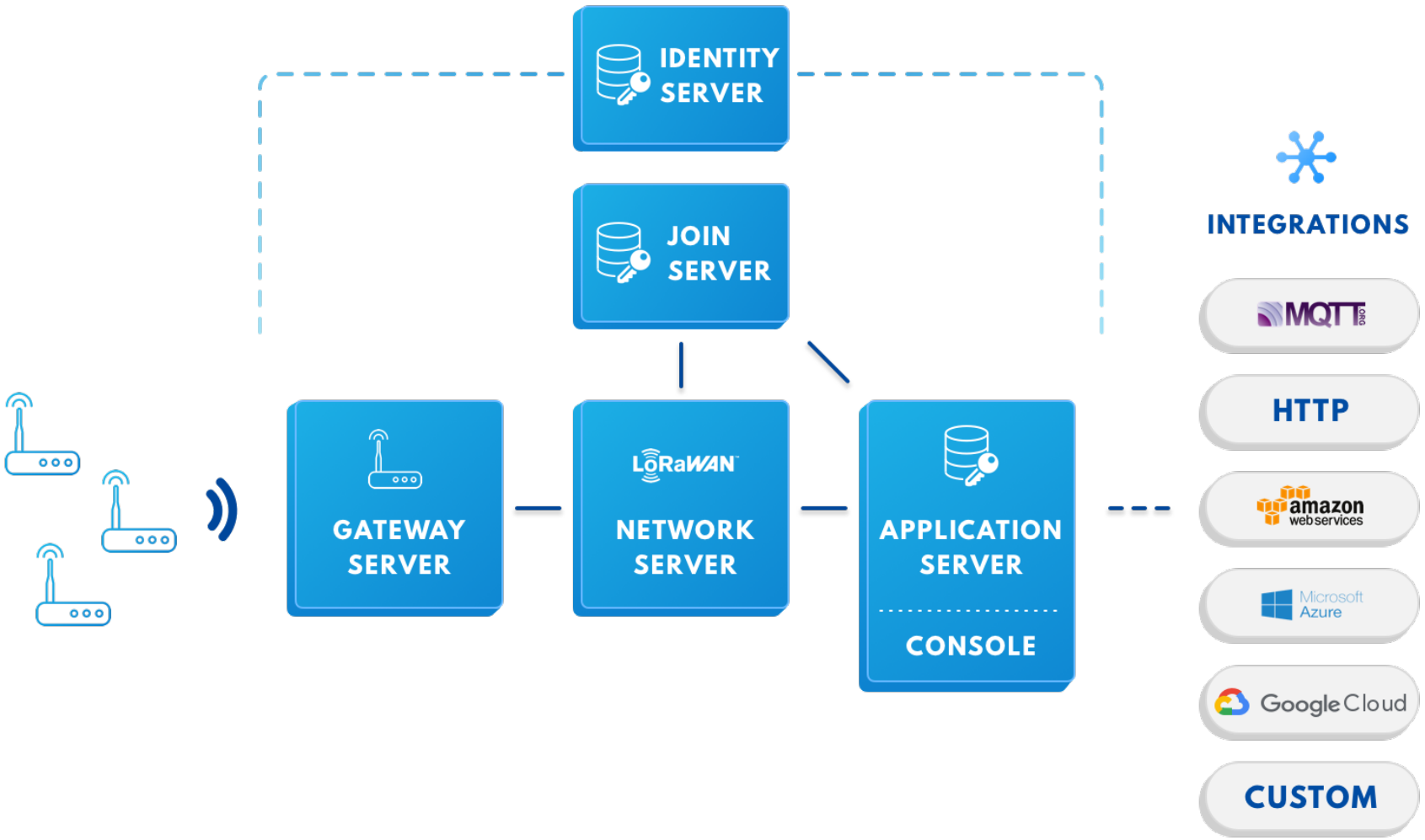
# What is LoRaWAN ?

- Wireless Network for Internet of Things.
- Adds address, mobility, & localization.
- multiple base station receive & process packets.
- Adaptive data rate scheme to improve performance.
- Multiple levels of encryption (Network & Application).
- Supports time slot scheduling of device transmission.

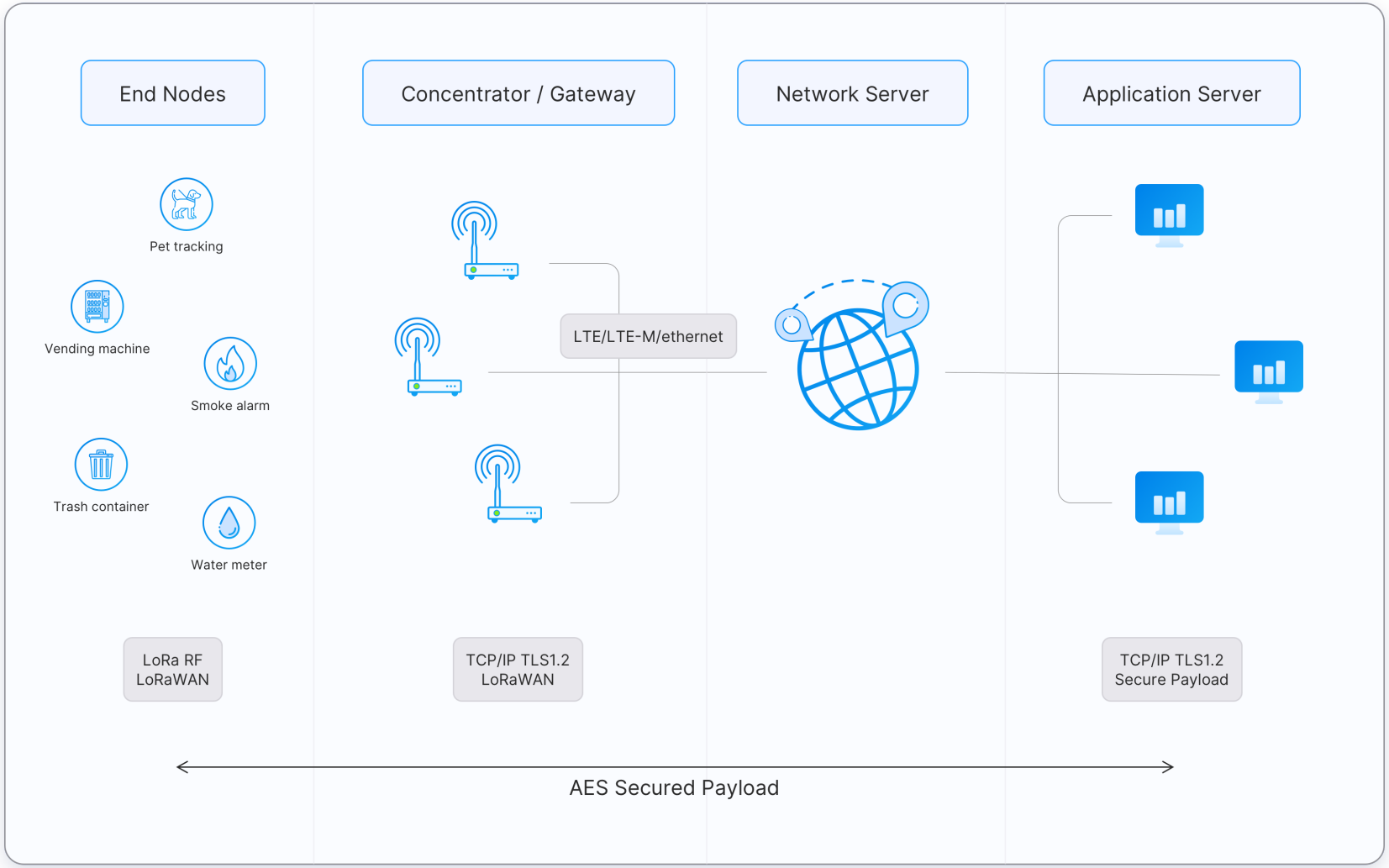
# What is LoRa ?

- PHY Radio protocol for Internet of Things.
- Derivative of chirp spread spectrum.
- Proprietary to semtech.
- Designed for long range, low power, low data rate communication.
- Star topology (not mesh or p2p).

# LoRaWAN Components



# LoRaWAN Architecture



# LoRaWAN End Devices

- A LoRaWAN-enabled end device is a sensor or an actuator which is wirelessly connected to a LoRaWAN network through radio gateways using LoRa RF Modulation.
- Sensors or actuators send LoRa modulated wireless messages to the gateways or receive messages wirelessly back from the gateways.
- In the majority of applications, an end device is an autonomous, often battery-operated sensor that digitizes physical conditions and environmental events. Typical use cases for an actuator include: street lighting, wireless locks, water valve shut off, leak prevention, among others.
- When they are being manufactured, LoRa-based devices are assigned several unique identifiers. These identifiers are used to securely activate and administer the device, to ensure the safe transport of packets over a private or public network and to deliver encrypted data to the Cloud.

# LoRaWAN Class Devices

- A LoRaWAN-enabled end device is a sensor or an actuator which is wirelessly connected to a LoRaWAN network through radio gateways using LoRa RF Modulation.
- Sensors or actuators send LoRa modulated wireless messages to the gateways or receive messages wirelessly back from the gateways.
- In the majority of applications, an end device is an autonomous, often battery-operated sensor that digitizes physical conditions and environmental events. Typical use cases for an actuator include: street lighting, wireless locks, water valve shut off, leak prevention, among others.
- When they are being manufactured, LoRa-based devices are assigned several unique identifiers. These identifiers are used to securely activate and administer the device, to ensure the safe transport of packets over a private or public network and to deliver encrypted data to the Cloud.

# LoRaWAN Gateways

- It is used to receive messages from end devices and forward them to the Network Server.
- Each gateway is registered (using configuration settings) to a LoRaWAN network server.
- Gateways are connected to the Network Server using a backhaul like Cellular (3G/4G/5G), WiFi, Ethernet, fiber-optic or 2.4 GHz radio links.

## Gateway Working:

A LoRaWAN gateway receives LoRa modulated RF messages from any end device in hearing distance and forwards these data messages to the LoRaWAN network server (LNS), which is connected through an IP backbone. There is no fixed association between an end device and a specific gateway. Instead, the same sensor can be served by multiple gateways in the area.

# LoRaWAN Network Server

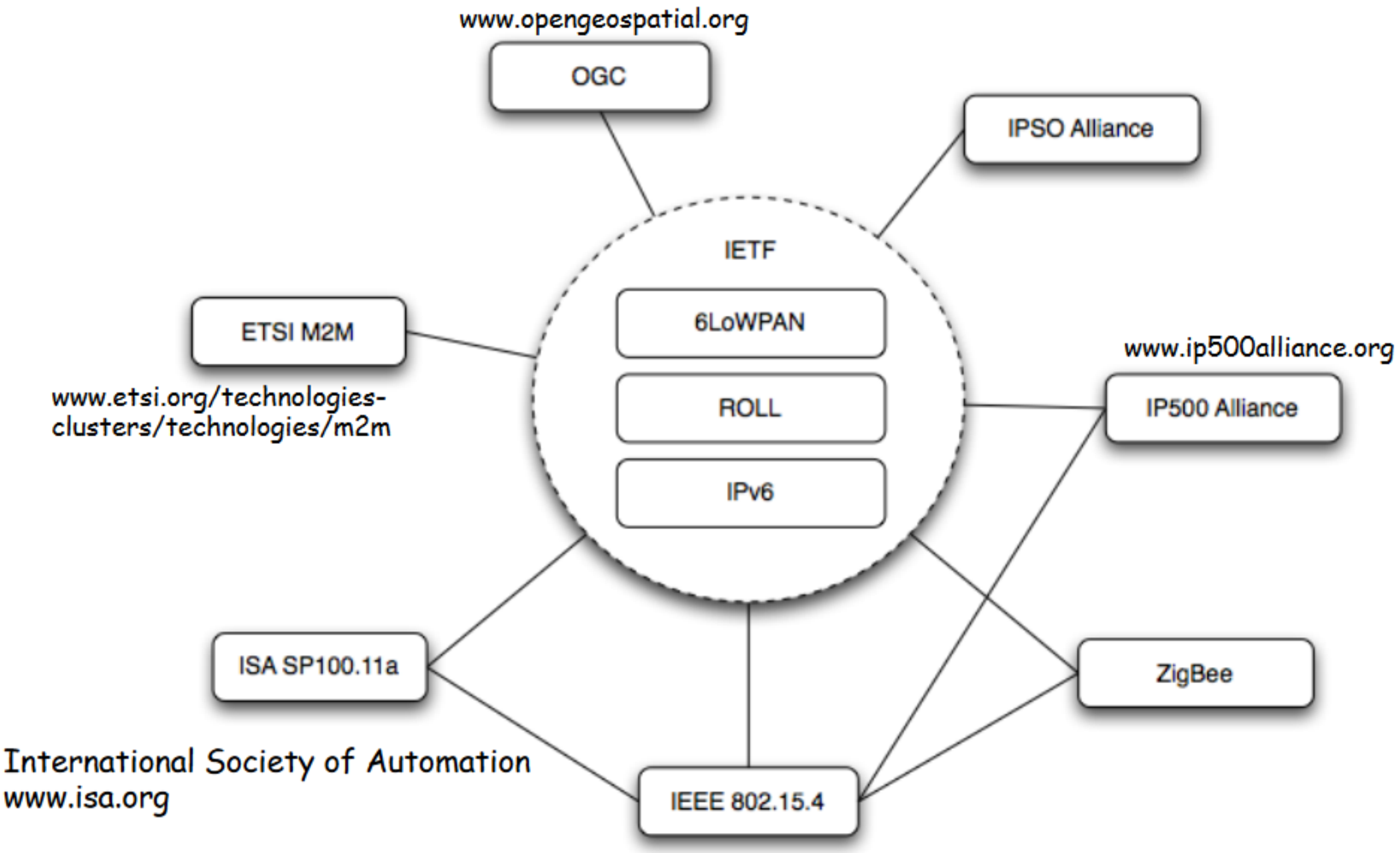
The Network Server manages gateways, end-devices, applications, and users in the entire LoRaWAN network. A typical LoRaWAN Network Server has the following features.

- Establishing secure 128-bit AES connections for the transport of messages between end-devices and the Application Server (end-to-end security).
- Validating the authenticity of end devices and integrity of messages.
- Deduplicating uplink messages.
- Selecting the best gateway for routing downlink messages.
- Sending ADR commands to optimize the data rate of devices.
- Device address checking.
- Providing acknowledgements of confirmed uplink data messages.
- Forwarding uplink application payloads to the appropriate application servers
- Routing uplink application payloads to the appropriate Application Server.
- Forwarding Join-request and Join-accept messages between the devices and the join server
- Responding to all MAC layer commands.

# LoRaWAN Application

Application servers are responsible for securely handling, managing and interpreting sensor application data. They also generate all the application-layer downlink payloads to the connected end devices.

# Standards Relationship



**Definition**

6LoWPAN provides the upper layer system for use with low power wireless communications for IoT and M2M, originally intended for IEEE 802.15.4, it is now used with many other wireless standards.

- The 6LoWPAN system is used for a variety of applications including wireless sensor networks. This form of wireless sensor network sends data as packets and using IPv6 - providing the basis for the name - IPv6 over Low power Wireless Personal Area Networks.
- Devices in the network typically work together to connect the physical environment to real world applications, e.g., wireless sensors networks
- 6LoWPAN provides a means of carrying packet data in the form of IPv6 over IEEE 802.15.4 and other networks. It provides end-to-end IPv6 and as such it is able to provide direct connectivity to a huge variety of networks including direct connectivity to the Internet.

Common topologies include star, mesh, and combinations of star and mesh

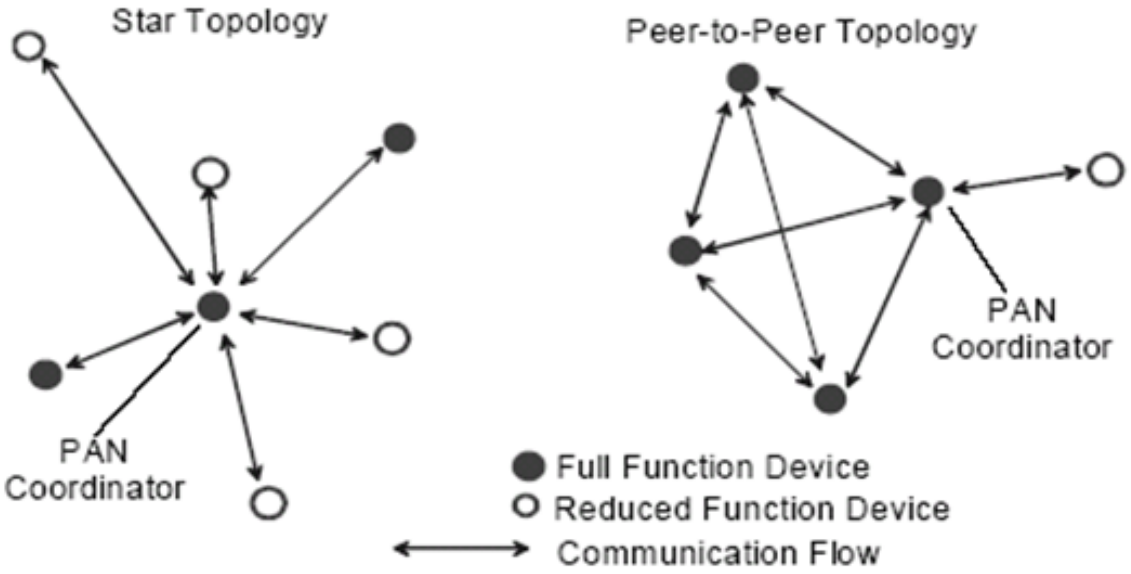


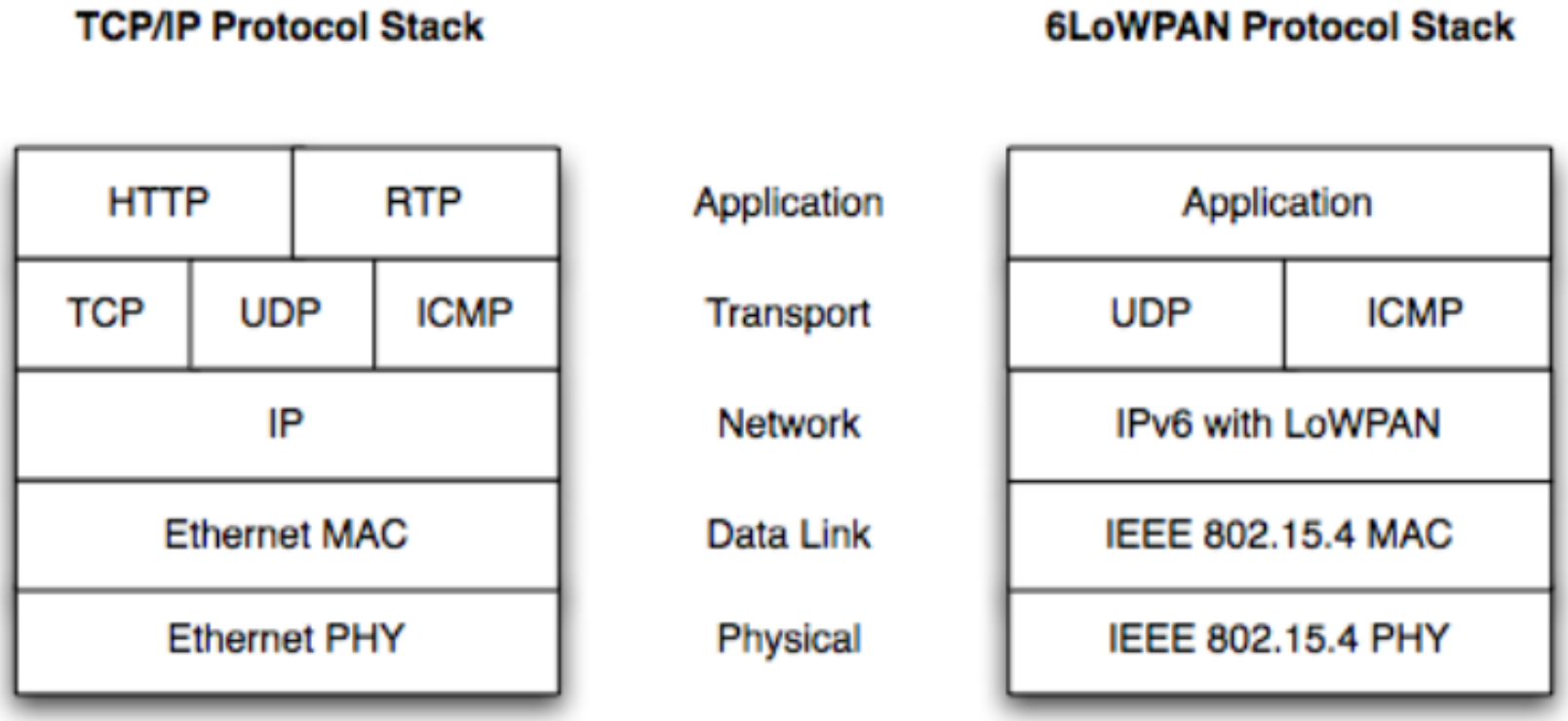

**Figure 1—Star and peer-to-peer topology examples**

# 6LoWPAN MAC Format

<table>
<tr><td>Octets: 2</td><td>1</td><td>0/2</td><td>0/2/8</td><td>0/2</td><td>0/2/8</td><td>variable</td><td>2</td></tr>
<tr><td rowspan="2">Frame control</td><td rowspan="2">Sequence number</td><td>Destination PAN identifier</td><td>Destination address</td><td>Source PAN identifier</td><td>Source address</td><td rowspan="2">Frame payload</td><td rowspan="2">FCS</td></tr>
<tr><td colspan="4">Addressing fields</td></tr>
<tr><td colspan="6">MHR</td><td>MAC payload</td><td>MFR</td></tr>
</table>

**Figure 34—General MAC frame format**

# 6LoWPAN Protocol Stack



# 6LoWPAN Security Perspective

- It is anticipated that the Internet of Things, IoT will offer hackers a huge opportunity to take control of poorly secured devices and also use them to help attack other networks and devices.
- Accordingly security is a major issue for any standard like 6LoWPAN, and it uses AES-128 link layer security which is defined in IEEE 802.15.4. This provides link authentication and encryption.
- Further security is provided by the transport layer security mechanisms that are also included. This is defined in RFC 5246 and runs over TCP.
- For systems where UDP is used the transport layer protocol defined under RFC 6347 can be used, although this may require some specific hardware requirements.

# 6LoWPAN Interoperability Perspective

- One key issue with IoT device is interoperability. It is vital that equipment from different manufacturers operates together.
- When testing for interoperability, it is necessary to ensure that all layers of the OSI stack are compatible. To ensure that this can be achieved there several different specifications that are applicable.
- 6LoWPAN is a wireless / IoT style standard that has quietly gained significant ground. Although initially aimed at usage with IEEE 802.15.4, it is equally able to operate with other wireless standards making it an ideal choice for many applications.

## Definition

It is a routing protocol for low-powered and lossy-network, which is considered a suitable routing protocol for the Internet of Things (IoT).

## Low-Power and Lossy Network

- It is a network typically composed of many embedded devices with limited power, memory, and processing resources interconnected by a variety of links, such as IEEE 802.15.4 or low-power Wi-Fi.
- There is a wide scope of application areas for LLNs, including industrial monitoring, building automation (heating, ventilation, and air conditioning (HVAC), llighting, access control, fire), connected home, health care, environmental monitoring, urban sensor networks, energy management, assets tracking, and refrigeration.

**RPL is a**

- Distance Vector (DV) protocol
- Source Routing Protocol

# Distance Vector Protocol ?

- The term distance vector refers to the fact that the protocol manipulates vectors (arrays) of distances to other nodes in the network
- It is an Intra-domain routing protocol
- Requires that a router inform its neighbors of topology changes periodically
- Have less computational complexity and message overhead
- Distance-vector protocols are based on calculating the Direction and Distance to any link in a network.
  - "Direction" usually means the next hop address and the exit interface.
  - "Distance" is a measure of the cost to reach a certain node.
- The least cost route between any two nodes is the route with minimum distance.
- Each node maintains a vector (table) of minimum distance to every node.
- The cost of reaching a destination is calculated using various route metrics

# Source Routing (path addressing) protocol ?

- Allows a sender of a packet to partially or completely specify the route the packet takes through the network.
- Enables a node to discover all the possible routes to a host.

# RPL Approach

- Construct and maintain a DAG supporting MP2P flows
  - multiple successors when available (vs. Tree)
  - implementation specific metrics and objective functions to find the least cost paths
- Use DAG to constrain & guide computation of routes supporting P2MP flows
- Use MP2P + P2MP as basic P2P support
  - More optimal P2P provisioned with complementary mechanisms

# RPL Approach: the DAG

- Nodes taking up a position in the DAG compute a Depth value, specific to the metric and objective function in use
- Depth value may be used to gauge relative position in the DAG

# RPL Approach: Forwarding

- Forwarding MP2P traffic to nodes of lesser depth avoids loops
  - only occur in presence of depth inconsistency, which is avoided or discovered and resolved
  - Ample redundancy in most networks
- Forwarding traffic to nodes of equal depth (DAG siblings) may be used if forwarding to lesser depth is temporarily failed
  - Increases redundancy, but additional protection against loops, e.g., ids, should be added
- Forwarding MP2P traffic to nodes of deeper depth is unlikely to make forward progress and likely to loop

# RPL Approach: Loops

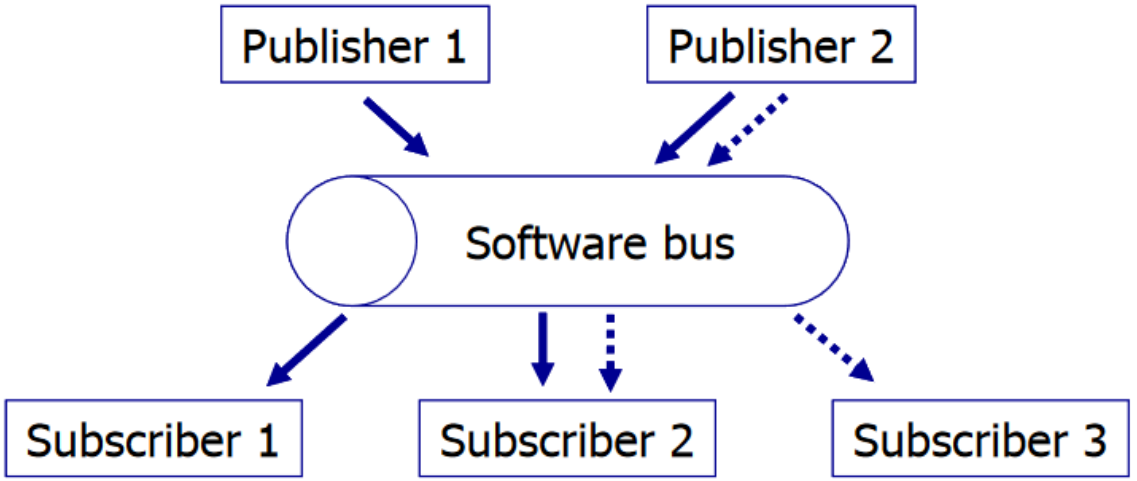
- Deeper nodes may be in the nodes own sub-DAG, which significantly increases the chance of loops
  - Forwarding traffic into ones own sub-DAG means it may come back around!
- RPL constructs and maintains the DAG in a coordinated way that avoids forming loops

# MQTT Protocol : Introduction

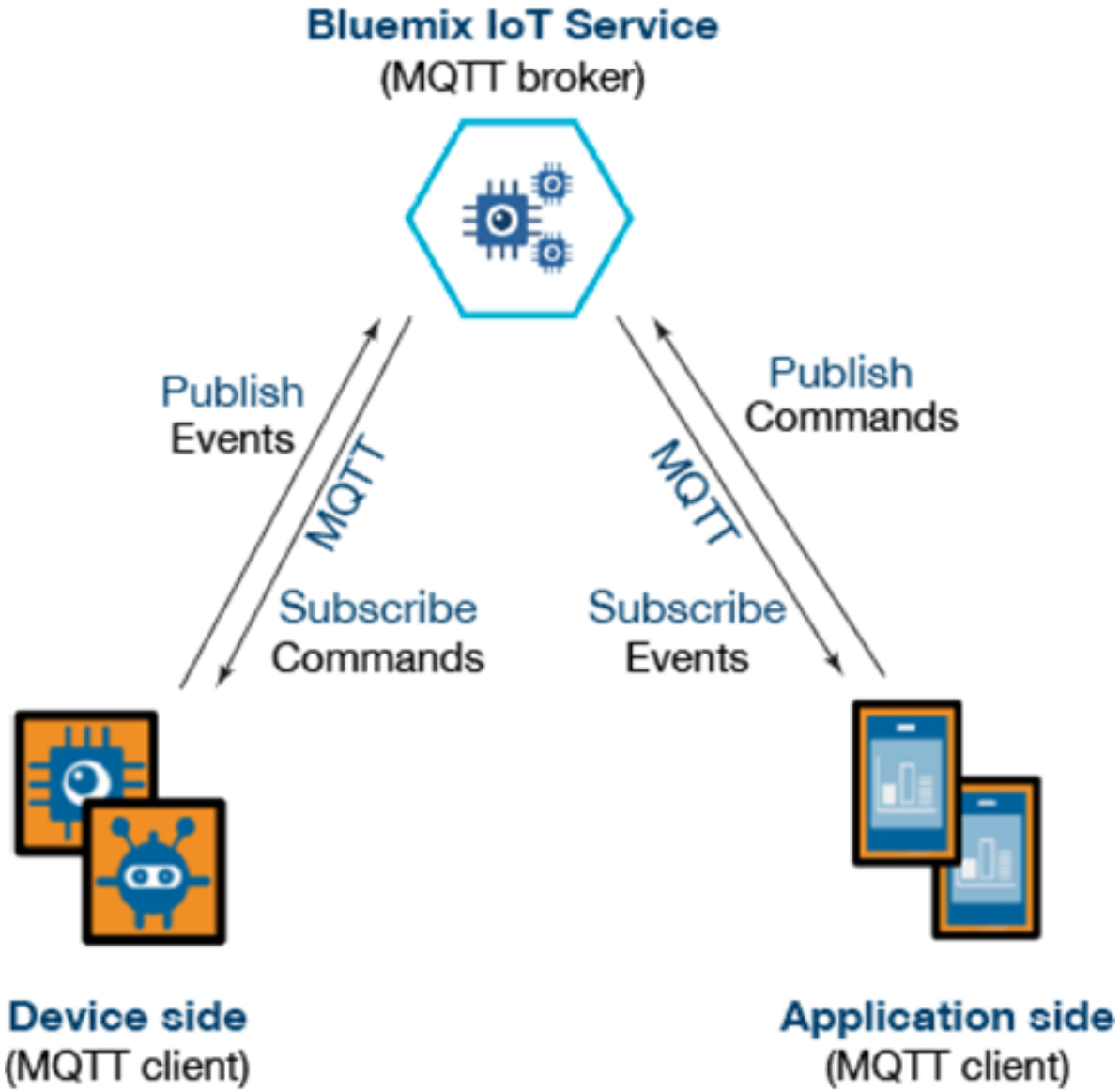
- MQTT was invented by Andy Stanford-Clark (IBM) and Arlen Nipper (Arcom, now Cirrus Link) back in 1999, where their use case was to create a protocol for minimal battery loss and minimal bandwidthconnecting oil pipelines over satellite connection.
- A lightweight publish-subscribe protocol that runs on embedded devices and mobile platforms designed to connect the physical world devices with applications and middleware.
- Designed to provide a low latency two-way communication channel and efficient distribution to one or many receivers.
- **Minimizes the amount of bytes flowing over the wire**
- **Low power usage.**

# MQTT Basic Concept Publish/Subscribe

- Pub/Sub decouples a client, who is sending a particular message (called publisher) from another client (or more clients), who is receiving the message (called subscriber). This means that the publisher and subscriber dont know about the existence of one another.
- There is a third component, called broker, which is known by both the publisher and subscriber, which filters all incoming messages and distributes them accordingly.



# MQTT : Use case



# MQTT : Use case

- Clients connect to a Broker
- Clients subscribe to topics eg,
  - client.subscribe(toggleLight/1)
  - client.subscribe(toggleLight/2)
  - client.subscribe(toggleLight/3)
- Clients can publish messages to topics:
  - client.publish(toggleLight/1, toggle);
  - client.publish(toggleLight/2, toggle);
- All clients receive all messages published to topics they subscribe to
- **Messages can be anything**
  - Text, Images, etc

# What is CoAP ?

## Definition

Constrained Application Protocol (CoAP) is a specialized web transfer protocol for use with constrained nodes and constrained networks in the Internet of Things. CoAP is designed to enable simple, constrained devices to join the IoT even through constrained networks with low bandwidth and low availability. It is generally used for machine-to-machine (M2M) applications such as smart energy and building automation.

- CoAP - Constrained Application Protocol.
- Specialized web transfer protocol.
- Devised for contrained and low power networks.

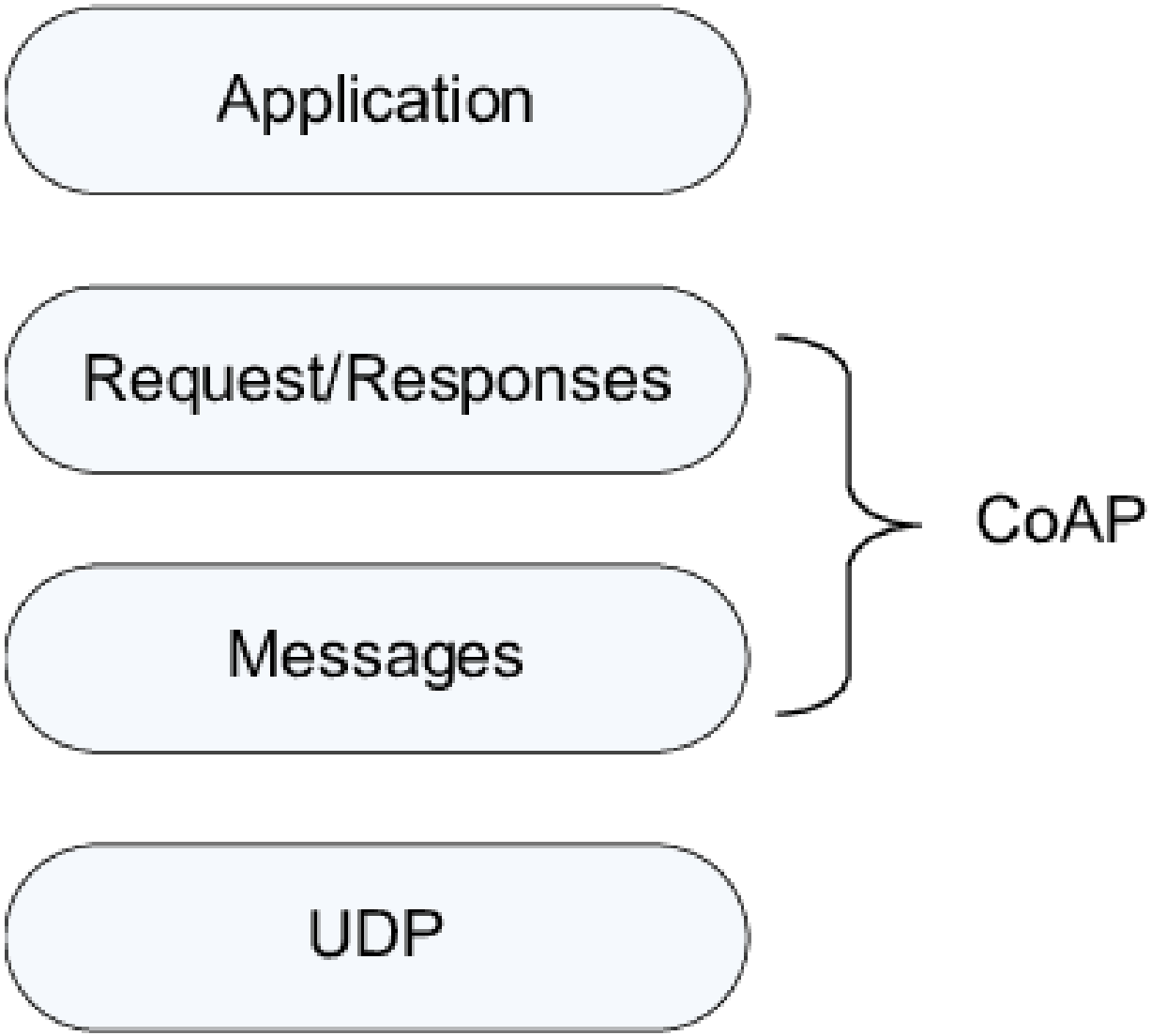
# WHY CoAP ?

- Current web technologies do not consider memory, energy and computation constraints of embedded devices.
- CoRE group from IETF works on developing RESTful application layer protocol - CoAP1 .
- No consensus on a common application layer due to huge variety of manufacturers of these embedded devices is one of the reasons for this.
- Need for a common application layer for resource constrained devices formed the motivation for CoAP.

# CoAP

- CoAP provides request/response interaction model as in WWW - HTTP.
- CoAP helps in integration with existing web along with meeting special needs of constrained devices.
- CoAP is based on UDP, supports asynchronous messages, low overheads, URI & content type support and provides simple proxy and caching possibilities.

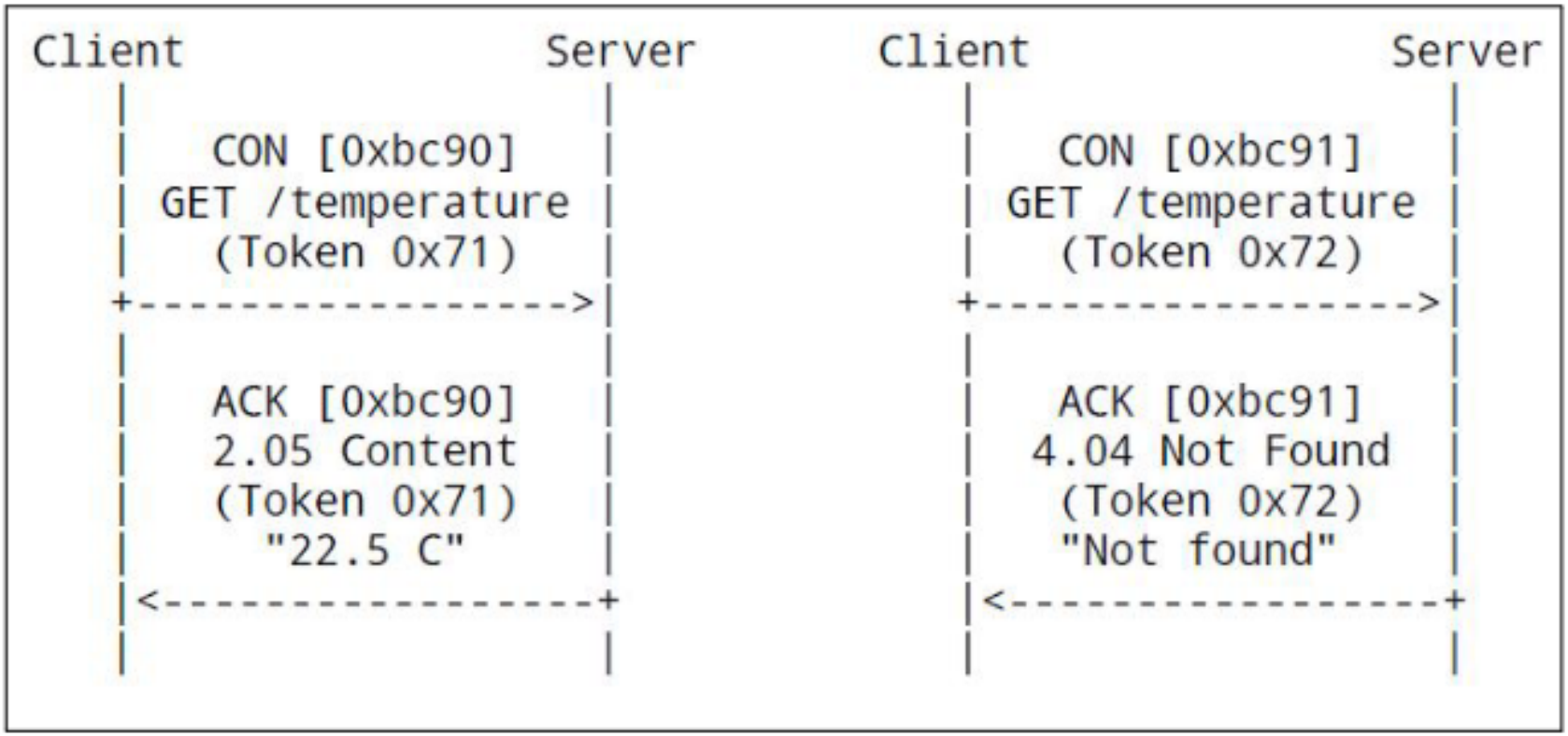
# CoAP : Layering



# CoAP : Layering

- TCP complexities are reduced by using UDP.
- Request Methods: GET, POST, PUT, DELETE.
- Response Methods: 2.xx (success), 4.xx (client error), 5.xx (servererror).
- Message types: Confirmable, Non Confirmable, Acknowledgement and Reset.
- Unicast and multicast requests.
- Resource discovery capability.
- URI representations for resources. coap-URI=coap: // host [ : port ] path-abempty [ ? query ]
- Block transfers for large files.

# CoAP : Communication Example



# CoAP : Message Transmission

CoAP requests and responses are transferred asychnronously wrapped in messages. Due to UDP, messages could be out of order, duplicate or get lost. Thus, it also introduces a reliable lightweight protocol like TCP.

- Stop-and-wait protocol
- Binary exponential back-off for Confirmable messages.
- Duplicate detection for both Confirmable and Non-confirmable messages.

# CoAP : Message Transmission

- Message Transmission is asynchronous between the endpoints.
- A CoAP endpoint is a source or destination of a message.
- Without security endpoints are identified by IP and Port number.
- With security its the security mode: NoSec, PreSharedKey,
- RawPublicKey and Certificate.
- For reliable message transmission, it should be marked as Confirmable in the CoAP header.
- Confirmable message is transmitted at exponentially increasing intervals, until an acknowledgement (or Reset message) is recieved, or attempts get over.
- Therefore, timer and counter are required to handle retransmissions.

# CoAP : Message Transmission

- For unreliable message transmission, acknowledgement is not required for example repeated readings from sensor.
- A message is marked Non-confirmable in this case.
- Reciept of such message cannot be tracked.
- Message deduplication is required for multiple messages recieved.
- In case of reliable messages, each duplicate copy is to be acknowledged by the reciever, but request can be processed only once.
- In case of unreliable messages, each duplicate copy is to be silently ignored by the reciever.

# CoAP : Transmission Parameters

- ACK TIMEOUT - 2 seconds
- ACK RANDOM FACTOR - 1.5
- MAX RETRANSMIT - 4
- NSTART - 1
- DEFAULT LEISURE - 5 seconds
- PROBING RATE - 1 Byte per second

# CoAP : Method Definitions

- GET: Retrieves the information corresponding to the resource in request URI. It is safe and idempotent.
- POST: Requests processing of representation in the response. Neither safe nor idempotent.
- PUT: Resource identified by the request URI be updated or created with the enclosed representation. Not safe but idempotent.
- DELETE: Requests to delete the resource identified by URI. Not safe but idempotent.

# CoAP: Limitations

- End to end secure connection required for CoAP/HTTP mapping at a proxy using DTLS/TLS.
- Securing multicast communications.
- Semantics should be standardized.
- Caching of requests should also be allowed.

# A working example: WSN over CoAP

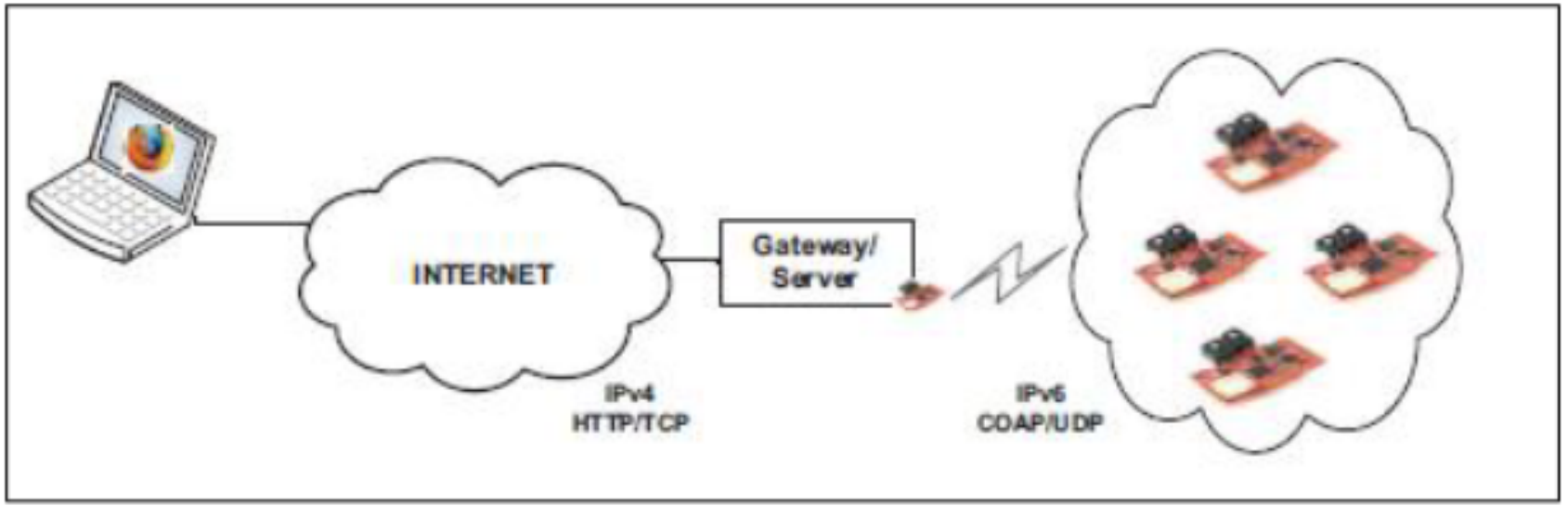



Figure: Integration of web and WSN

W. Colitti, K. Steenhaut, N. De Caro, B. Buta, and V. Dobrota, Rest enabled wireless sensor networks for seamless integration with web applications, in Mobile Adhoc and Sensor Systems (MASS), 2011 IEEE 8th International Conference on, 2011, pp. 867872.

# A working example: WSN over CoAP

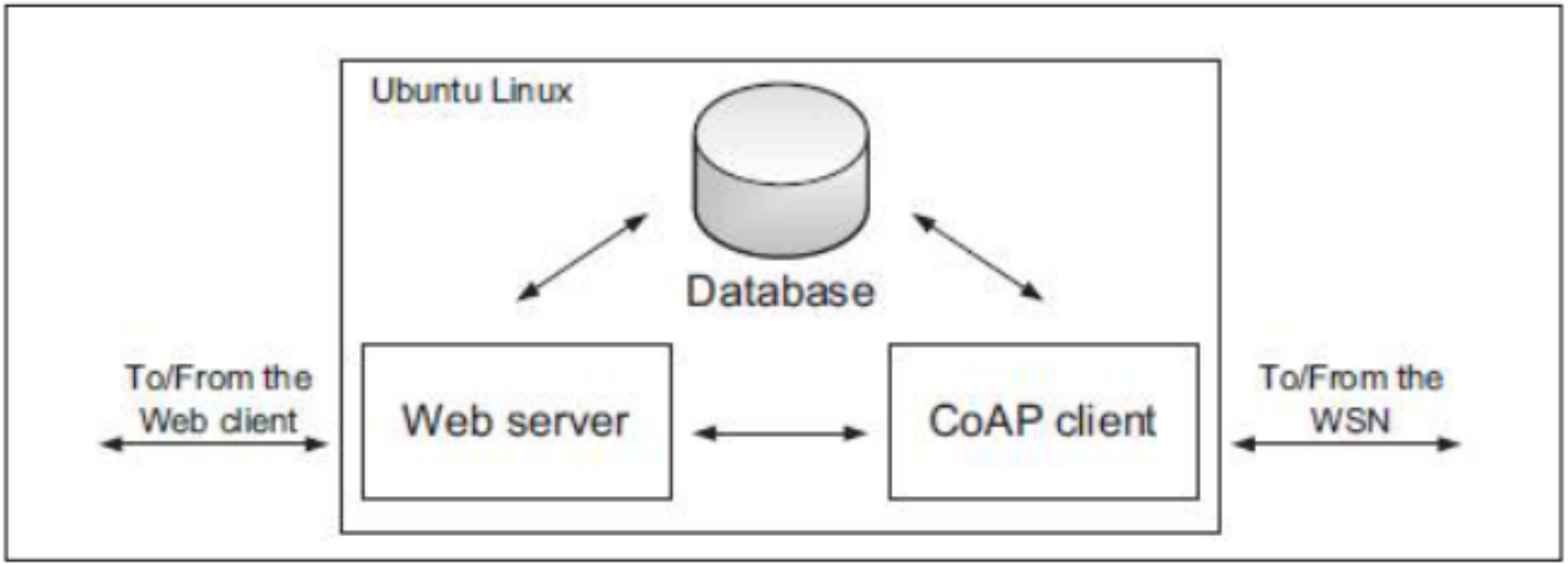



Figure: Gateway Building Blocks

W. Colitti, K. Steenhaut, N. De Caro, B. Buta, and V. Dobrota, Rest enabled wireless sensor networks for seamless integration with web applications, in Mobile Adhoc and Sensor Systems (MASS), 2011 IEEE 8th International Conference on, 2011, pp. 867872.

# A working example: WSN over CoAP

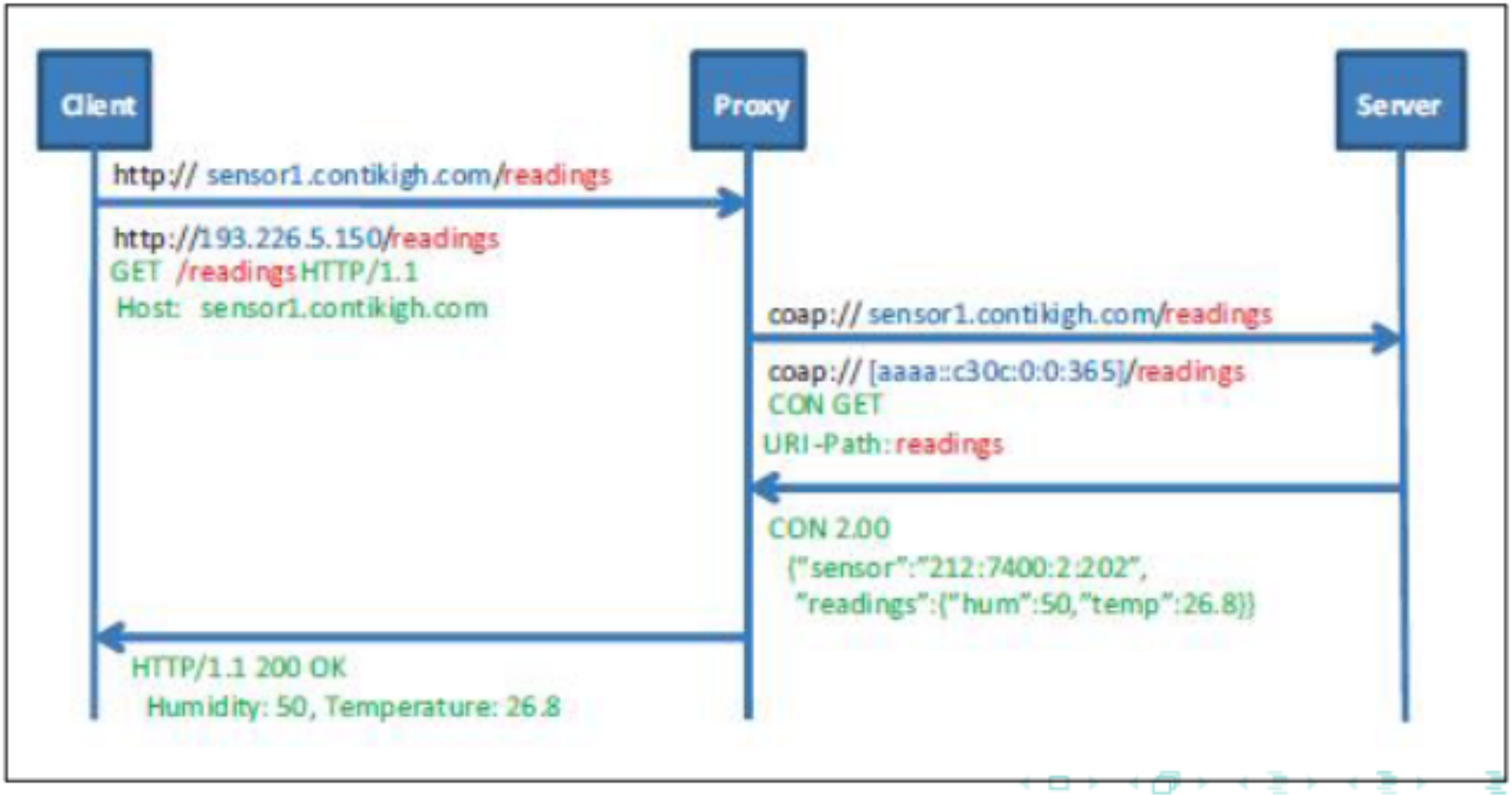



Figure: Message Exchange

W. Colitti, K. Steenhaut, N. De Caro, B. Buta, and V. Dobrota, Rest enabled wireless sensor networks for seamless integration with web applications, in Mobile Adhoc and Sensor Systems (MASS), 2011 IEEE 8th International Conference on, 2011, pp. 867872.

# Thank You!!!