SNS COLLEGE OF ENGINEERING

Kurumbapalayam(Po),
Coimbatore – 641 107

An Autonomous Institution

DEPARTMENT OF COMPUTER SCIENCE AND DESIGN

Course Code and Name : 19IT503 Internet of Things

Unit 2 – FUNDAMENTAL MECHANISMS & KEY TECHNOLOGIES

Fundamental IoT Mechanism and Key Technologies

- Some fundamental issues and technologies that have to be considered in the context of Internet of things (IoT) design and deployment.

Identification of IoT Object and Services

- Identification codes can be classified as
  - (i) object IDs (OIDs)
  - (ii) communication IDs.
- Examples
  - radio frequency identification (RFID)/electronic product code (EPC), content ID,1
  - telephone number, and uniform resource identifier (URI)/uniform resource locator (URL);
  - media access control (MAC) address, network layer/IP address, and session/protocol ID.

Identification of IoT Object and Services

- All objects to have a permanent unique identifier, an OID.
- All end-point network locations and/or intermediary-point network locations to have a durable, unique network address (NAdr) using IPv6
- When objects that have enough intelligence to run a communications protocol stack (so that they can communicate), are placed on a network, these objects can be tagged with a NAdr.

Identification of IoT Object and Services

- Every object then has a tuple (OID, NAdr) that is always unique, although the second entry (NAdr) of the tuple may change with time, location, or situation.
- In a stationary, non-variable, or mostly static environment, assigns the OID to be identical to the NAdr where the object is expected to attach to the network; that is, the object tuple (NAdr, NAdr).
    - In case the object moved, the OID could then be refreshed to the address of the new location; that is, the object tuple (NAdr', NAdr').
- In general trend toward object mobility, giving rise to a dynamic environment; hence, to retain maximal flexibility, it is best to separate, in principle, the OID from the NAdr

Identification of IoT Object and Services

- Identification scheme is that it affords global uniqueness.
- It is useful to have mechanisms for hierarchical grouping to deal with large populations.
- Modern layered communication architectures also require addressing and processing capabilities at several layers,
  - For example, at the Data Link Layer, at the Network Layer, at the Transport (Protocol ID), and at the session/application layer.
- Some argue that different identification schemes are required for different applications.
  - For example, the information related to things such as books, medicine, and clothes may not require global identification because revocation lists are required

Identification of IoT Object and Services

- An EPC (electronic product code) is a number assigned to an RFID tag representative of an actual EPC.
- Each number is encoded with a header, identifying the particular EPC version used for coding the entire EPC number.
- An EPC manager number is defined, allowing individual companies or organizations to be uniquely identified; an object class number is present, identifying objects used within this organization, such as product types EPC uses a numerical system for product identification, but its capabilities are much greater.
- An EPC is actually a number that can be associated with specific product information, such as date of manufacture and origin and destination of shipment. This provides significant advantages for businesses and consumers.
- The EPC is stored on an RFID tag, which transmits data when prompted by a signal emitted by a special reader.
- Note that EPC and RFID are not interchangeable

Identification of IoT Object and Services

- OID may be replaced by object naming
- Domain name system (DNS) is a mechanism for Internet-based naming
- In the IoT context, the advantages of identifying information by name, not by node address.
- DNS is used to map the "human-friendly" host names of computers to their corresponding "machine friendly" IP addresses. E.g. www.google.com
- Object name service (ONS) will also be important in the IoT to map the "thing-friendly" names of object which may belong to heterogeneous name spaces (e.g., EPC, uCode, and any other self-defined code) on different networks (e.g., TCP/IP network) into their corresponding "machine-friendly" addresses or other related information of another TCP/IP network.
- This naming system can used for set of systems as object name should disclose its identity.

Identification of IoT Object and Services

- For some applications, especially where there is a need for simple end-user visibility of a small set of objects (i.e., where the objects are few and discretely identifiable a home’s thermostat, a home’s refrigerator, a home’s lighting system, a pet of the owner), the object may be identified through Web Services (WSs).
- WSs provide standard infrastructure for data exchange between two different distributed applications.
- Lightweight WS protocols for the representational state transfer (REST) interface may be useful in this context.
- REST is a software architecture for distributed systems to implement WSs. REST is good compared to simple object access protocol (SOAP) and web services description language (WSDL) due to its relative simplicity.

Identification of IoT Object and Services

- IoT objects and IoT applications (e.g., grid control, home control, traffic control, and medical monitoring), security and privacy in communications and services become absolutely critical.
- Strong authentication, encryption while transmitting, and also encryptions for data at rest is ideal; however, the computational requirements for encryption can be significant.
- At the central/authenticating site, rapid authentication support is desirable; otherwise objects would not be able to authenticate in large-population environments.

Identification of IoT Object and Services

- Tracking the object using GPS is costly
- But worst when tracking more than one object.
- There is a need to maintain ubiquitous and seamless communication while tracking the location of objects.

Identification of IoT Object and Services

- Capabilities for scalability are important in order to be able to support an IoT environment where there is a large population that is highly distributed.
- **Locator/identifier separation.**
  - Basic idea behind the separation is that the Internet architecture combines two functions, routing locators (where one is attached to the network) and identifiers (where one is located), in one number space: the IP address.
  - Proponents of the separation architecture postulate that splitting these functions apart will yield several advantages, including improved scalability for the routing system.
  - The separation aims to decouple locators and identifiers, thus allowing for efficient aggregation of the routing locator space and providing persistent identifiers in the identifier space.
  - The protocol called locator/ID separation protocol (LISP)

Identification of IoT Object and Services

- LISP aims for an incrementally deployable protocol.
- The LISP WG (Working Group) frame that include
  - (i) an architecture description, (ii) deployment models,
  - (iii) a description of the impacts of LISP, (iv) LISP security threats and solutions,
  - (v) allocation of end-point identifier (EID) space, (vi) alternate mapping system designs
  - (vii) data models for management of LISP.

Structural Aspects of the IoT

- Structural Issue related to
  - Environment Characteristics
  - Traffic Characteristics
  - Scalability
  - Interoperability
  - Security and Privacy
  - Open Architecture

Structural Aspects of the IoT Environment Characteristics

- Most (but certainly not all) IoT/machine-to-machine (M2M) nodes have design constraints:
    - Low power (with the requirement that they will run potentially for years on batteries)
    - Low cost (total device cost in single-digit dollars or triple digit rupee)
    - Significantly more devices than in a LAN environment
    - Severely limited code and RAM space (e.g., generally desirable to fit the required code—MAC, IP, and anything else needed to execute the embedded application—in, for example, 32K of flash memory, using 8-bit microprocessors)
    - Unobtrusive but very different user interface for configuration (e.g., using gestures or interactions involving the physical world)
    - Requirement for simple wireless communication technology. In particular, the IEEE 802.15.4 standard is very promising for the lower (physical and link) layers

Structural Aspects of the IoT

Traffic Characteristics

- The characteristics of IoT/M2M communication is different from other types of networks or applications.
  - For example, cellular mobile networks are designed for human communication and communication is connection centric; it entails interactive communication like
    - between humans (voice, video), or data communication involving humans (web browsing, file downloads, and so on).
    - It follows that cellular mobile networks are optimized for traffic characteristics of human-based communication and applications.
- But in IoT, M2M the expectation is that there are many devices, there will be long idle intervals, transmission entails small messages, there may be relaxed delay requirements, and device energy efficiency is paramount.

Structural Aspects of the IoT

Scalability

- The application and its a desire over time for the service decides the Scalability.
- When contemplating expansion, one wants to be able to build on previously deployed technology (systems, protocols), without having to scrap the system and start from scratch.
- The efficiency of a larger system should be better than the efficiency of a smaller system.
- This is what is meant by scalability.
- The goal is to make sure that capabilities such as addressing, communication, and service discovery, among others, are delivered efficiently in both small and large scale.

Structural Aspects of the IoT

Interoperability

- Applications, technology suppliers, and stakeholders, it is desirable to develop and/or re-use a core set of common standards.
- To the degree possible, existing standards may prove advantageous to a rapid and cost-effective deployment of the technology.

Structural Aspects of the IoT
Security and Privacy

- IoT relates to electric power distribution, goods distribution, transport and traffic management, e-health, and other key applications, as noted earlier
- It is critical to maintain system-wide confidentiality, identity integrity, and trustworthiness.

Structural Aspects of the IoT

Open Architecture

- The goal is to support a wide range of applications using a common infrastructure, preferably based on a service-oriented architecture (SOA) over an open service platform, and utilizing overly networks (these being logical networks defined on top of a physical infrastructure)

Key IoT Technologies

- List of Technologies are:
  - Device Intelligence
  - Communication Capabilities
  - Mobility Support
  - Device Power
  - Sensor Technology
  - RFID Technology
  - Satellite Technology

Key IoT Technologies
Device Intelligence

- Device Intelligence
  - In order for the IoT to become a reality,
- Objects should be able to intelligently sense and interact with the environment
- Possibly store some passive or acquired data
- Communicate with the world around them
  - Object-to-gateway device communication or even direct object-to-object communication is desirable

Key IoT Technologies
Device Intelligence

- These intelligent capabilities are necessary to support ubiquitous networking to provide seamlessly interconnection between humans and objects
  - Some have called this mode of communication *Any Services*, *Any Time*, *Any Where*, *Any Devices*, *and Any Networks* (also known as “5-Any”)

Key IoT Technologies

Communication Capabilities

- It is highly desirable for objects to support ubiquitous end-to-end communications
- To achieve ubiquitous connectivity for human-to-object & object-to-object communications, networking capabilities will need to be implemented in the objects ("things")
- IP is considered to be key capability for IoT objects
- Self-configuring capabilities, especially how an IoT device can establish its connectivity automatically without human intervention, are also of interest
- IPv6 auto-configuration & multihoming features are useful, particularly scope-based IPv6 addressing features

Key IoT Technologies

Mobility Support

- Another consideration related to tracking and mobility support of mobile object
- Mobility-enabled architectures & protocols are required
- Some objects move independently, while others will move as one of group
- Therefore, according to the moving feature, different tracking methods are required.
- It is important to provide ubiquitous and seamless communication among objects while tracking the location of objects.
- Mobile IPv6 (MIPv6) offers several capabilities that can address this requirement.

Key IoT Technologies
Device Power

- Related to the powering of the “thing”
- Especially for mobile devices or devices that do not have intrinsic power
- M2M/IoT applications are always constrained by following factors:
  - Devices have ultra-low-power capabilities
  - Devices must be of low cost
  - Devices must have small physical size & light in weight

Key IoT Technologies

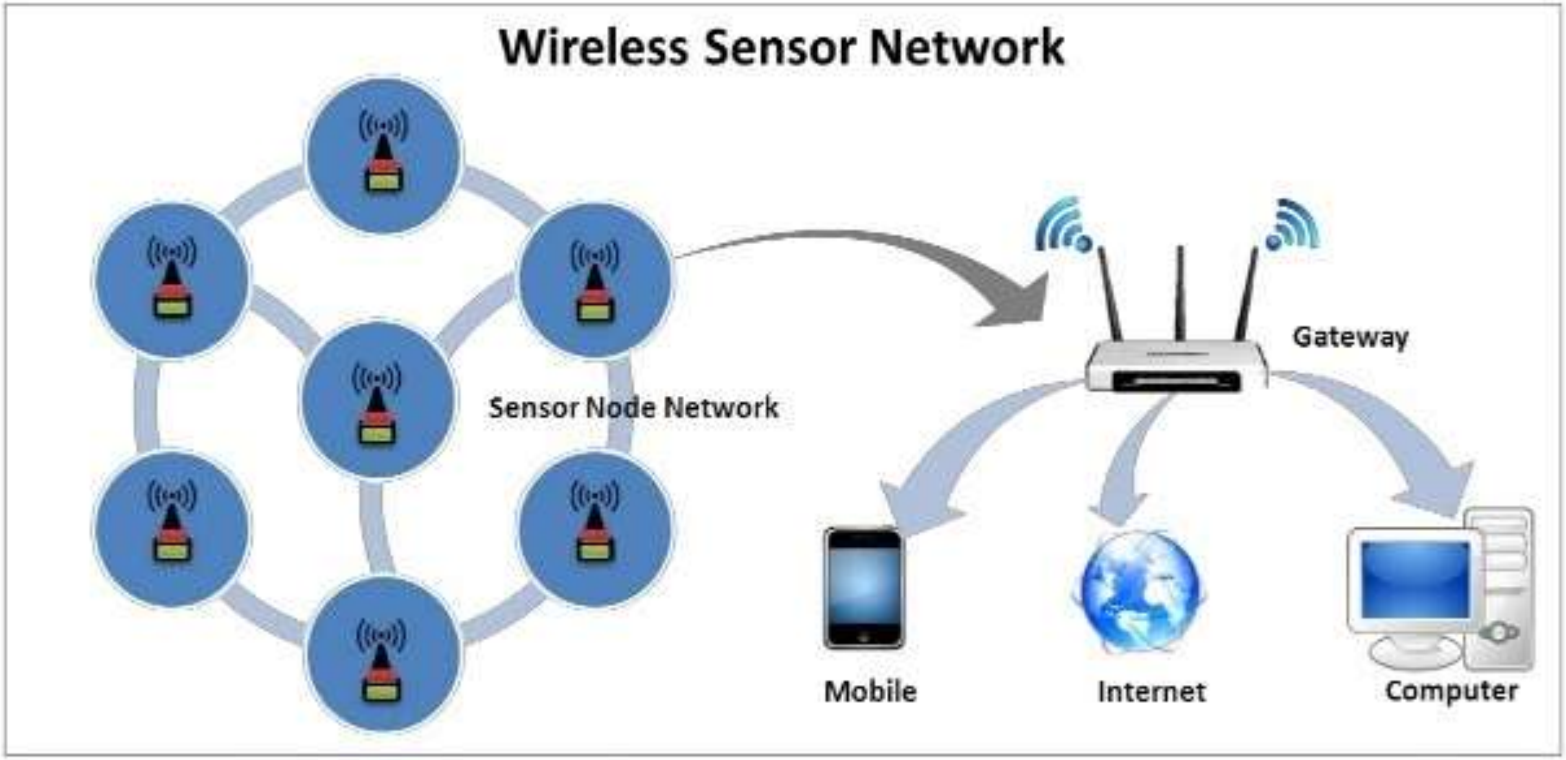
Device Power

- The following factors that must be considered in selecting the most suitable battery for a particular application :
- Operating voltage level
- Load current and profile
- Duty cycle—continuous or intermittent
- Service life
- Physical requirement
  - Size
  - Shape
  - Weight
- Environmental conditions
  - Temperature
  - Pressure
  - Humidity
  - Vibration
  - Shock
  - Pressure
- Safety and reliability
- Shelf life
- Maintenance and replacement
- Environmental impact and recycling capability
- Cost

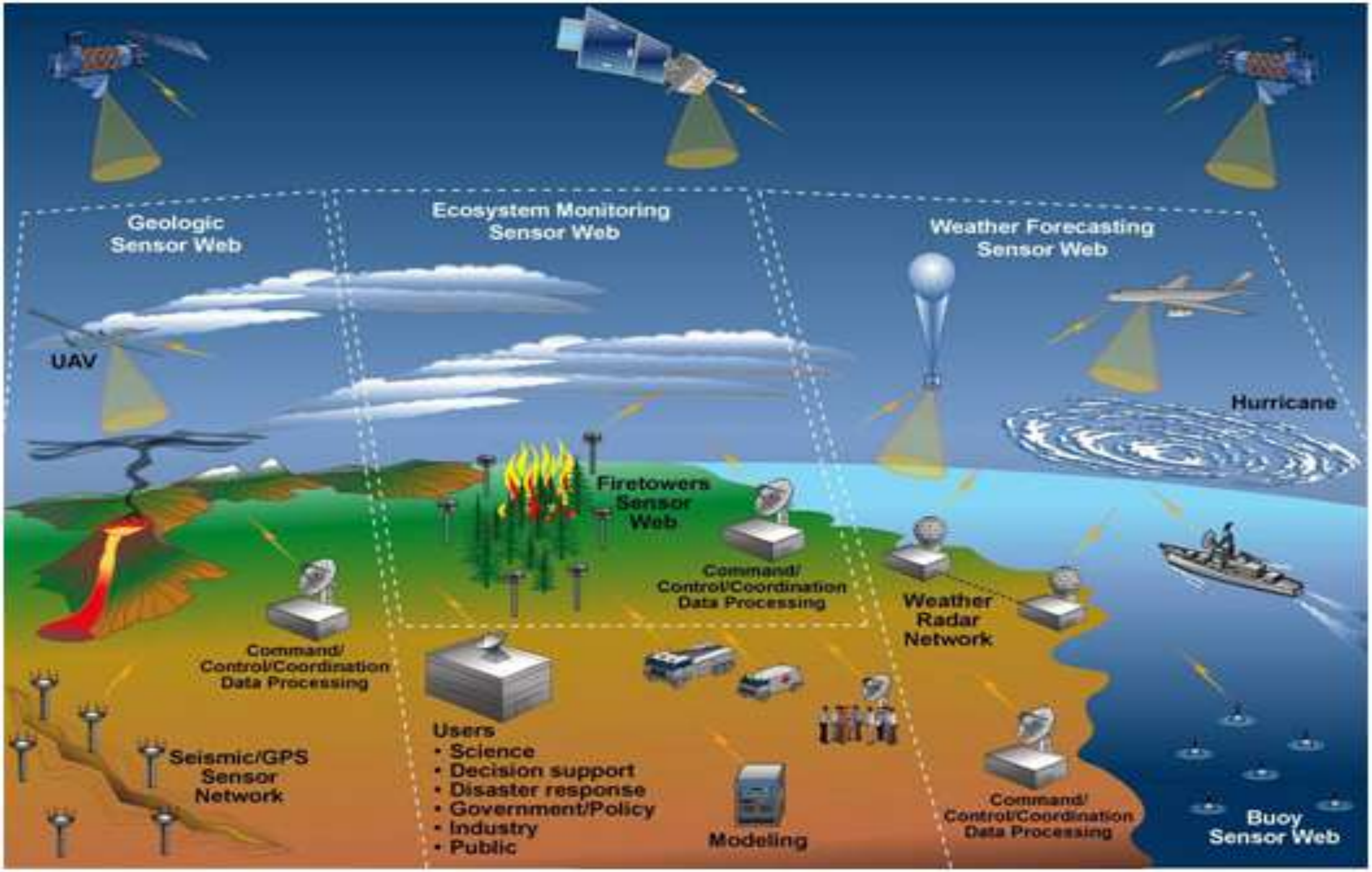
Key IoT Technologies
Sensor Technology

- A sensor network is an infrastructure comprising sensing (measuring), computing, communication, data collection, monitoring, surveillance, and medical telemetry.
- Sensor network technology, specifically, with embedded networked sensing, ships, aircrafts, and buildings can “self-detect” structural faults (e.g., fatigue-induced cracks).
- Earthquake-oriented sensors in buildings can locate potential survivors and can help assess structural damage; tsunami-alerting sensors can certainly prove useful for nations with extensive coastlines.
- Sensors also find extensive applicability in battlefield for reconnaissance and surveillance

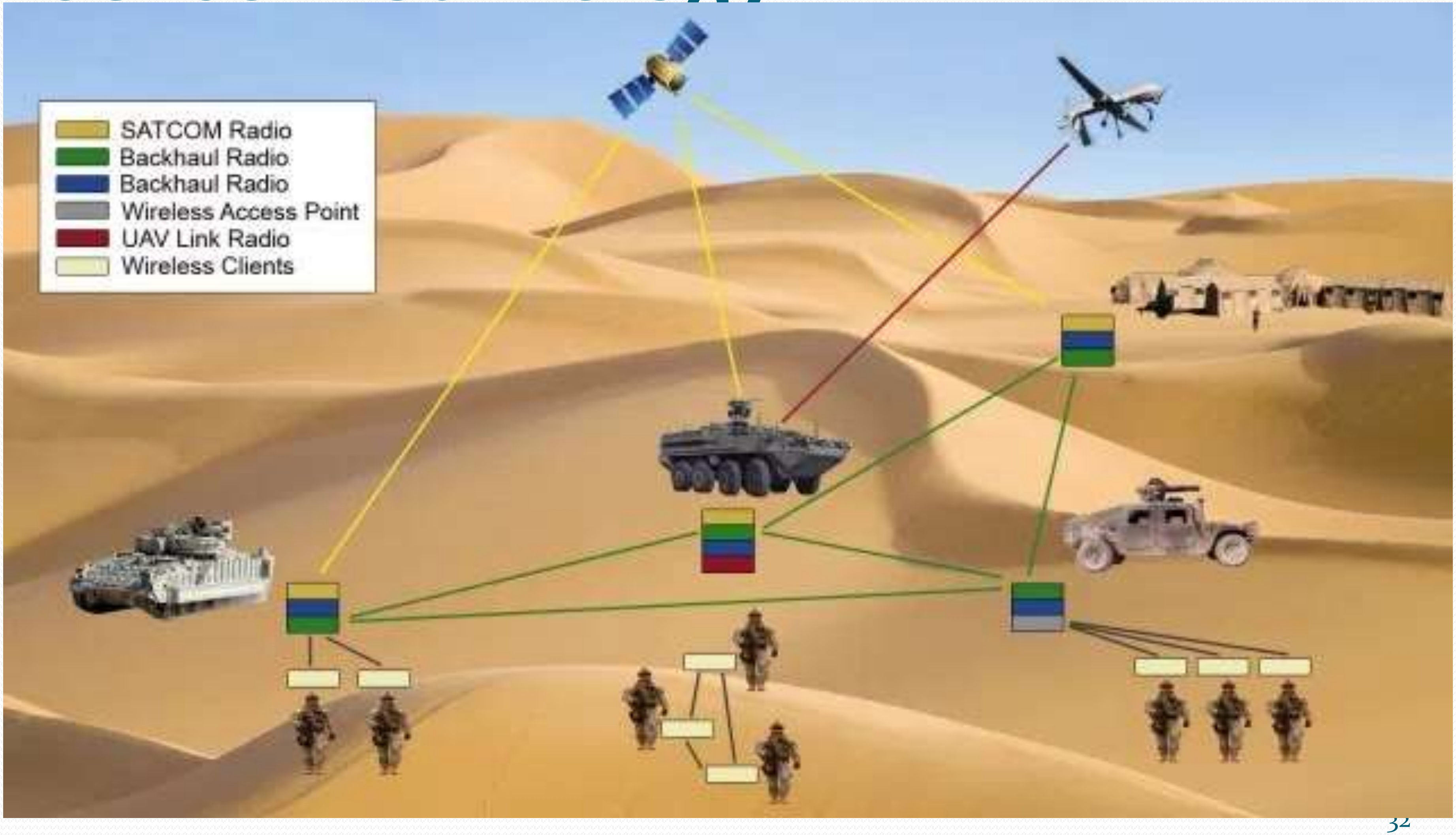
Key IoT Technologies
Sensor Technology



Key IoT Technologies


Geologic Sensor Web
Ecosystem Monitoring Sensor Web
Weather Forecasting Sensor Web
UAV
Hurricane
Firetowers Sensor Web
Command/ Control/Coordination Data Processing
Weather Radar Network
Command/ Control/Coordination Data Processing
Users
• Science
• Decision support
• Disaster response
• Government/Policy
• Industry
• Public
Seismic/GPS Sensor Network
Modeling
Command/ Control/Coordination Data Processing
Buoy Sensor Web

Key IoT Technologies
Sensor Technology



Key IoT Technologies
Sensor Technology

- There are four basic components in a sensor network:
  - (i) an assembly of distributed or localized sensors
  - (ii) an interconnecting network (usually, but not always, wirelessbased)
  - (iii) a central point of information clustering
  - (iv) a set of computing resources at the central point (or beyond) to handle data correlation, event-trending, querying, and data mining.
- Because the interconnecting network is generally wireless, these systems are known as wireless sensor networks (WSNs).
- WSN have the potentially large quantity of data collected, algorithmic methods for data management play an important role in sensor networks.
- In-network processing is desirable in sensor networks; furthermore, node power (and/or battery life) is a key design consideration.

Key IoT Technologies
Sensor Technology

- Sensors can be described as "smart" inexpensive devices equipped with multiple on-board sensing elements:
  - they are low cost, low power, untethered multifunctional nodes that are logically homed to a central sink node.
- Sensor utilize the Internet or some other network for long-haul delivery of information to a point (or points) of final data aggregation and analysis.
- Sensors are typically internetworked via a series of multihop short-distance low power wireless links called "sensor field".
- Sensors are typically deployed in a high density manner and in large quantities:
  - a WSN consists of densely distributed nodes that support sensing,
  - signal processing, embedded computing, and connectivity;
  - sensors are logically linked by self-organizing means (sensors that are deployed in short-hop point-to-point master-slave pair arrangements are also of interest).

Key IoT Technologies
Sensor Technology

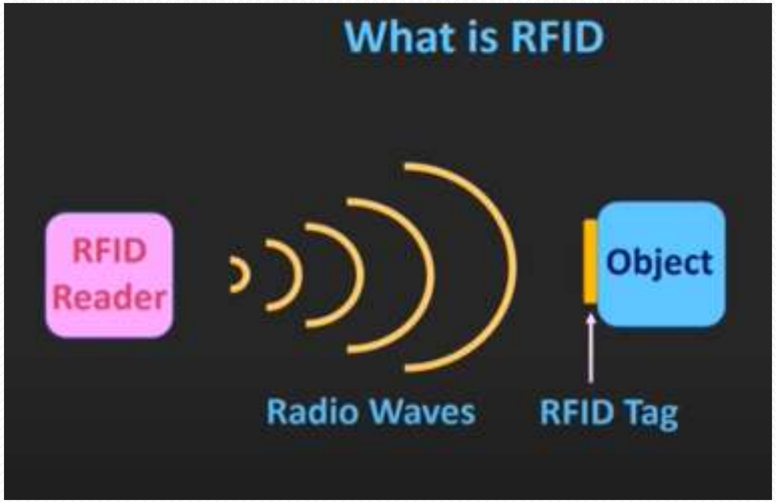
- New wireless design methodologies are needed across a set of disciplines, information transport, network and operational management, confidentiality, integrity, availability, and in-network/local processing, low battery status, other wireless sensor malfunction and lightweight protocol stack.
- Physical size can range from nanoscopic-scale devices to mesoscopic-scale devices at one end; from microscopic-scale devices to macroscopic-scale devices at the other end.
  - Nanoscopic (nanoscale) in the order of 1–100 nm in diameter;
  - Mesoscopic scale refers to objects between 100 and 10,000 nm in diameter
  - The microscopic scale ranges from 10 to 1000 microns
  - The macroscopic scale is at the millimeter-to-meter range.
- Biological sensors, small passive microsensors (such as “smart dust”), and “lab-on-a-chip” assemblies
- The miniaturized ones that are directly embedded in some physical infrastructure, as “microsensors.”

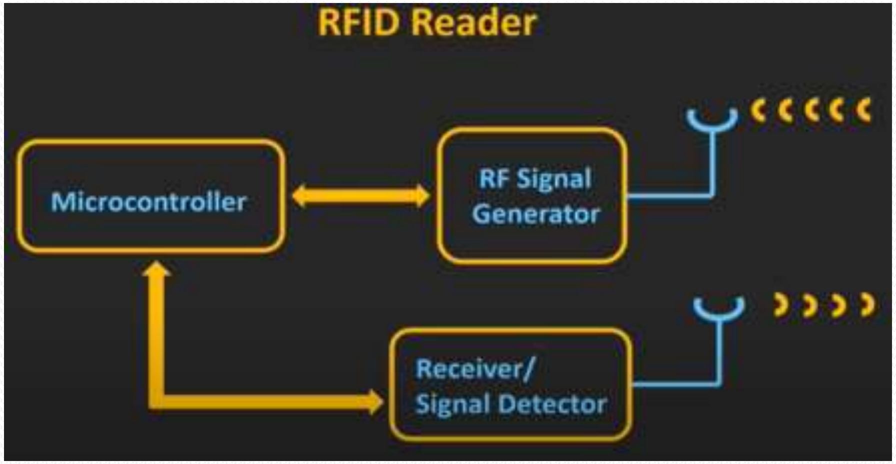
Key IoT Technologies
Sensor Technology

- Sensors may be passive and/or be self-powered; further along in the power consumption chain, some sensors may require relatively low power from a battery or high power.
- Low power consumption for transmission over low bandwidth channels and low power-consumption logic to pre-process and/or compress data.
- Power efficiency in WSNs is generally accomplished in three ways:
  - (i) Low duty cycle operation
  - (ii) Local/in-network processing to reduce data volume (and, hence, transmission time)
  - (iii) Multihop networking (this reduces the requirement for long-range transmission since signal path loss is an inverse power with range/distance) each node in the sensor network can act as a repeater, thereby reducing the link range coverage required, and, in turn, the transmission power

Key IoT Technologies
RFID Technology

- RFIDs are electronic devices associated with objects ("things") that transmit their identity (usually a serial number) via radio links.
- RFID tags are devices that typically have a read-only chip that stores a unique number but has no processing capability.





Key IoT Technologies
RFID Technology

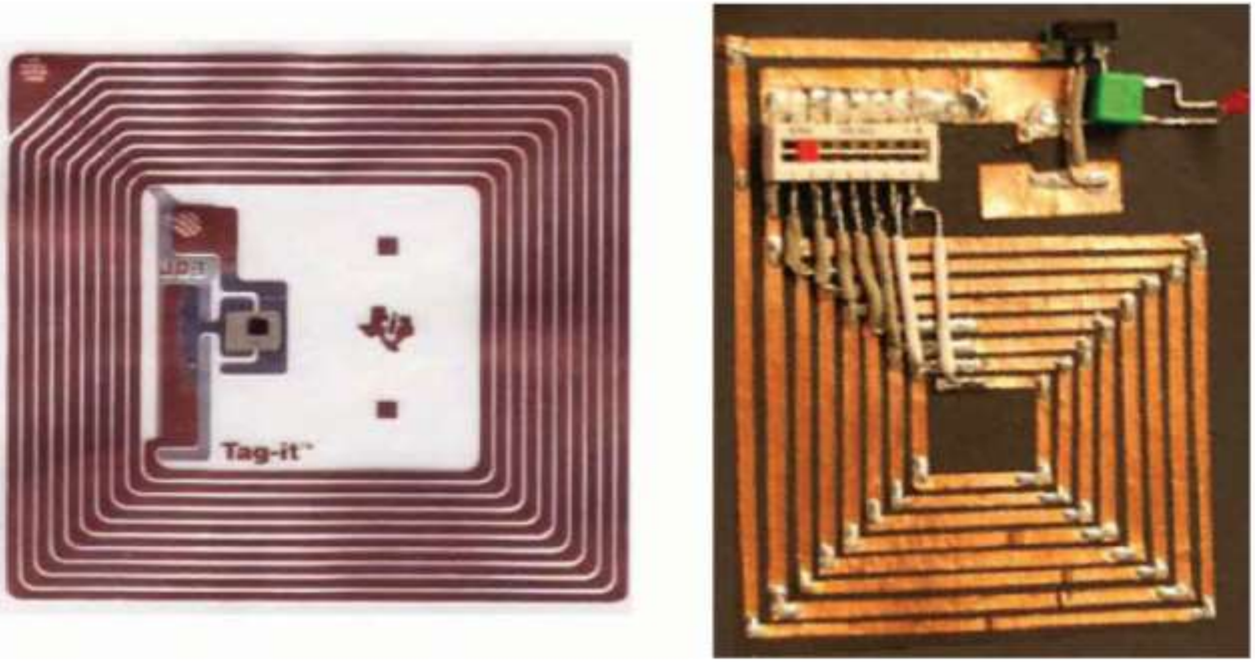
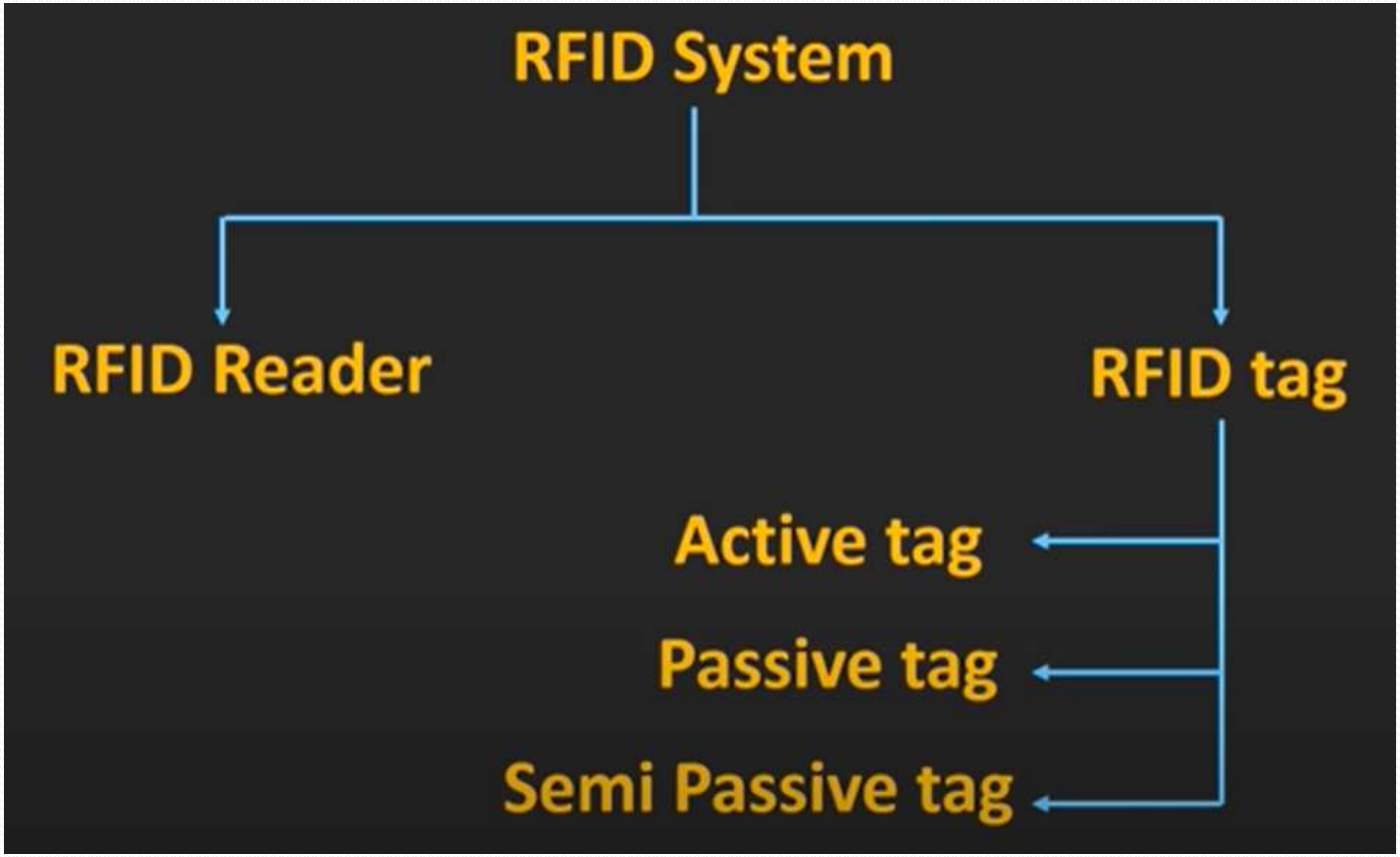
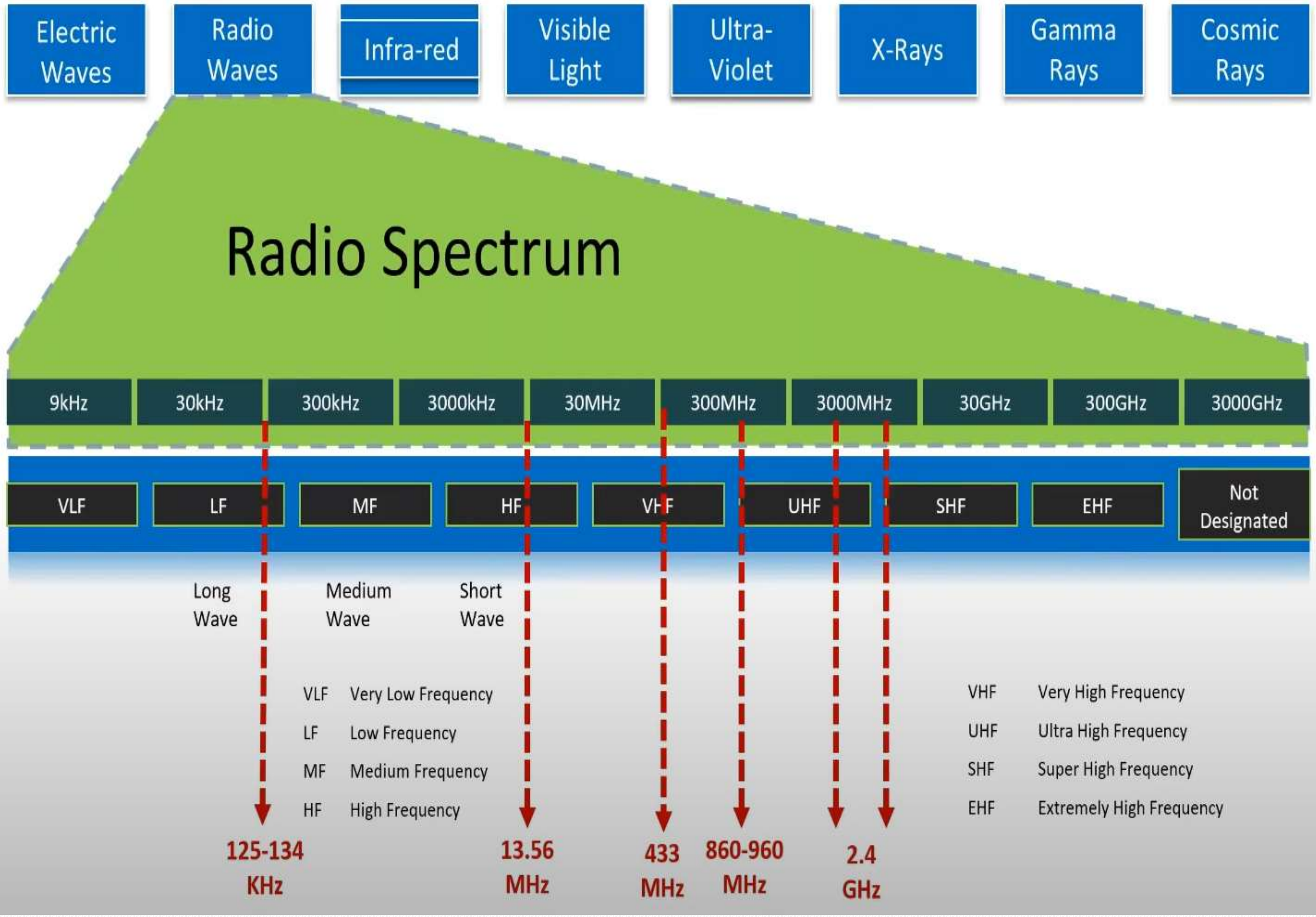


FIGURE 4.1 Illustrative examples of RFIDs.

RFID System


RFID System
RFID Reader
RFID tag
Active tag
Passive tag
Semi Passive tag



Electric Waves | Radio Waves | Infra-red | Visible Light | Ultra-Violet | X-Rays | Gamma Rays | Cosmic Rays

Radio Spectrum

9kHz | 30kHz | 300kHz | 3000kHz | 30MHz | 300MHz | 3000MHz | 30GHz | 300GHz | 3000GHz

VLF | LF | MF | HF | VHF | UHF | SHF | EHF | Not Designated

Long Wave

Medium Wave

Short Wave

VLF	Very Low Frequency
LF	Low Frequency
MF	Medium Frequency
HF	High Frequency

VHF	Very High Frequency
UHF	Ultra High Frequency
SHF	Super High Frequency
EHF	Extremely High Frequency

125-134 KHz

13.56 MHz

433 MHz

860-960 MHz

2.4 GHz

Key IoT Technologies
RFID Technology

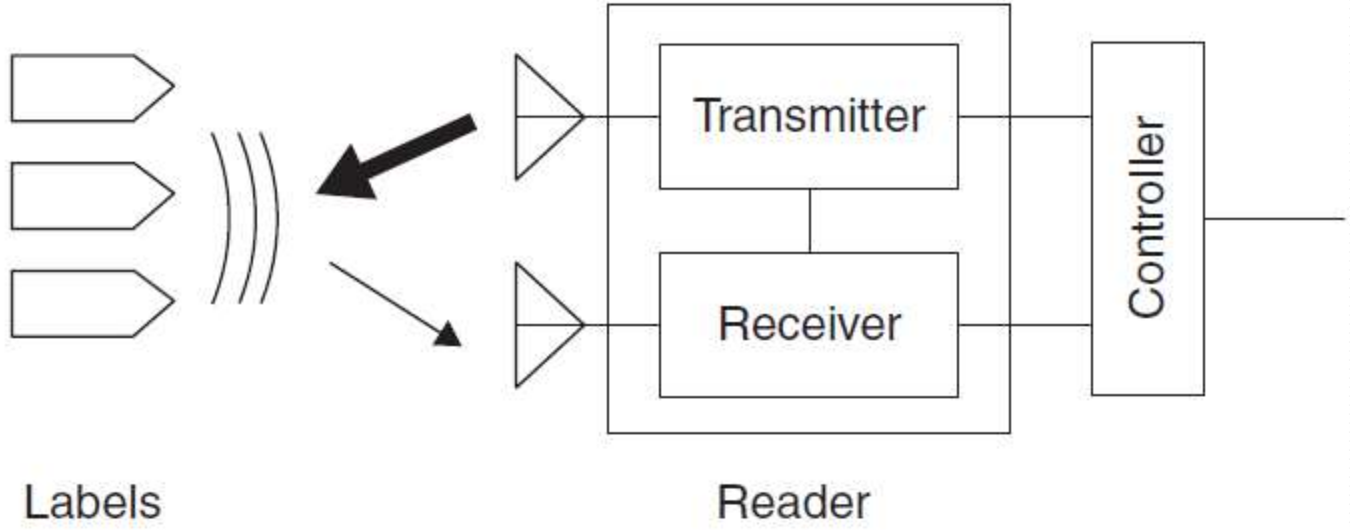
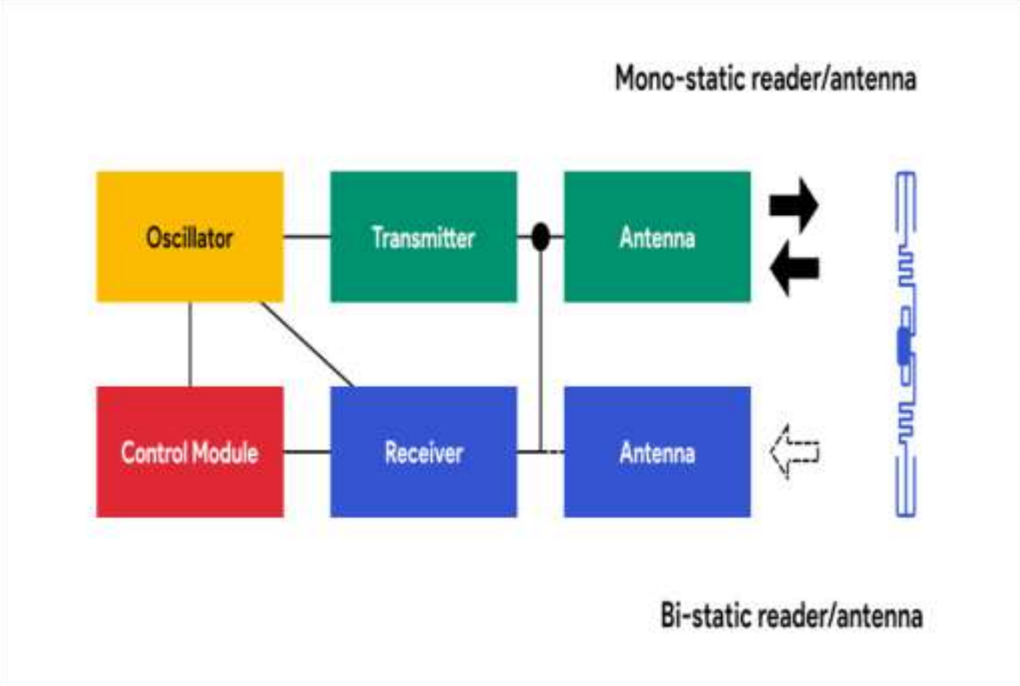


FIGURE 4.2 RFID reader operation.

RFID-Interrogator



Key IoT Technologies
RFID Technology

- There are a number of standards for RFIDs. Some of the key ones include the following:
  - The ISO 14443
    - operating frequency of 13.56 MHz that embed a CPU; power consumption is about 10mW; data throughput is about 100 Kbps and the maximum working distance (from the reader) is around 10 cm.
  - The ISO 15693
    - operating at 13.56 MHz frequency, but it enables working distances as high as 1 m, with a data throughput of a few Kbps.
  - The ISO 18000
    - with frequency such as 135 KHz, 13.56 MHz, 2.45 GHz, 5.8 GHz, 860–960 MHz, and 433 MHz.
    - The ISO 18000–6 standard uses the 860–960MHz range and is the basis for the Class-1 Generation-2 UHF RFID, introduced by the EPCglobal Consortium.

Key IoT Technologies

RFID Technology

- Typically, EPC codes used for active RFIDs or IP addresses are transmitted in clear form
  - Provide strong privacy for the IoT.
  - The host identity protocol (HIP) with this protocol, active RFIDs do not expose their identity in clear text, but protect the identity value (e.g., an EPC) using cryptographic procedures.

Key IoT Technologies
RFID Technology

- An RFID system is logically comprising several layers, as follows:
  - the tag layer,
  - the air interface (also called media interface) layer,
  - the reader layer;
- Tag (device) layer:
  - Architecture and EPCglobal Gen2 tag finite state machine
- Media interface layer:
  - Frequency bands, antennas, read range, modulation, encoding, data rates
- Reader layer:
  - Architecture, antenna configurations, Gen2 sessions, Gen2

Key IoT Technologies
RFID Technology- standards in the EPCglobal environment

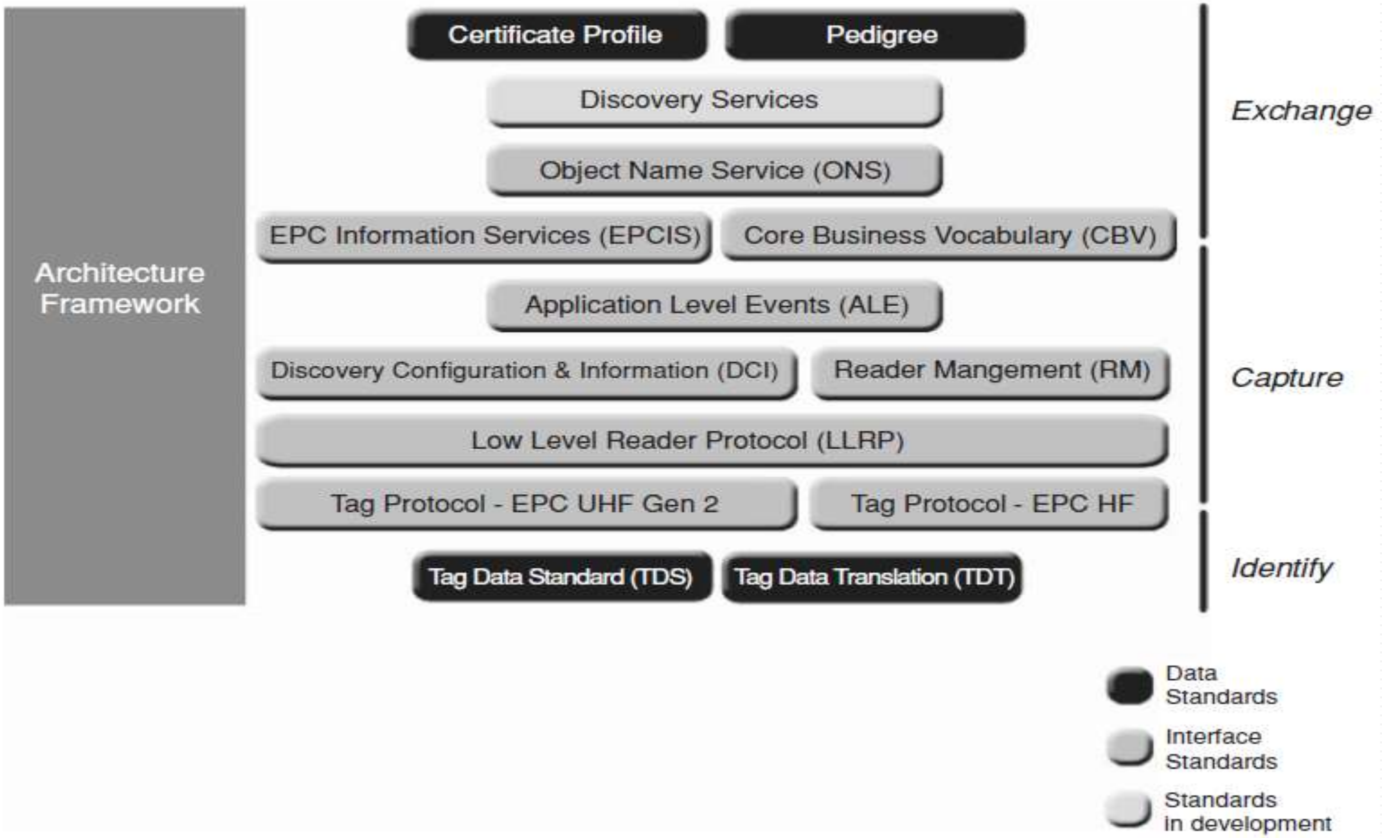

Architecture Framework
Certificate Profile
Pedigree
Discovery Services
Object Name Service (ONS)
EPC Information Services (EPCIS)
Core Business Vocabulary (CBV)
Application Level Events (ALE)
Discovery Configuration & Information (DCI)
Reader Mangement (RM)
Low Level Reader Protocol (LLRP)
Tag Protocol - EPC UHF Gen 2
Tag Protocol - EPC HF
Tag Data Standard (TDS)
Tag Data Translation (TDT)
Exchange
Capture
Identify
Data Standards
Interface Standards
Standards in development

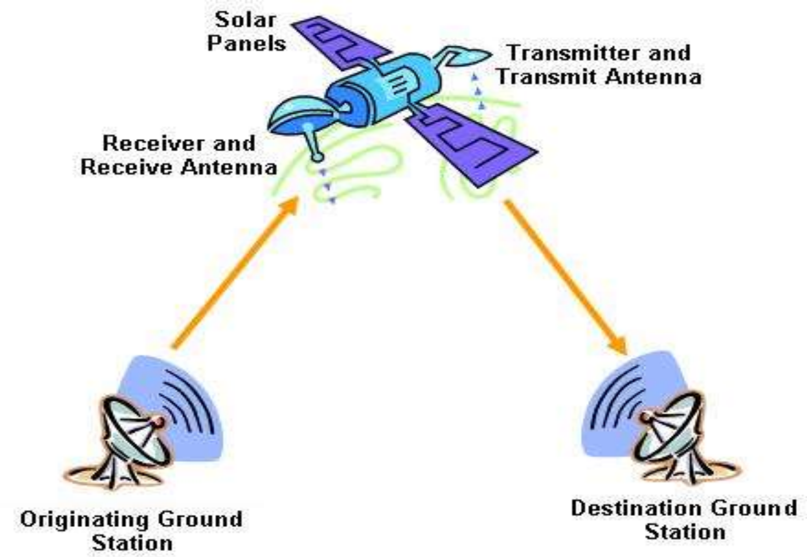
FIGURE 4.4 Standards that comprise the EPCglobal environment.

Key IoT Technologies
RFID Technology- standards in the EPCglobal environment

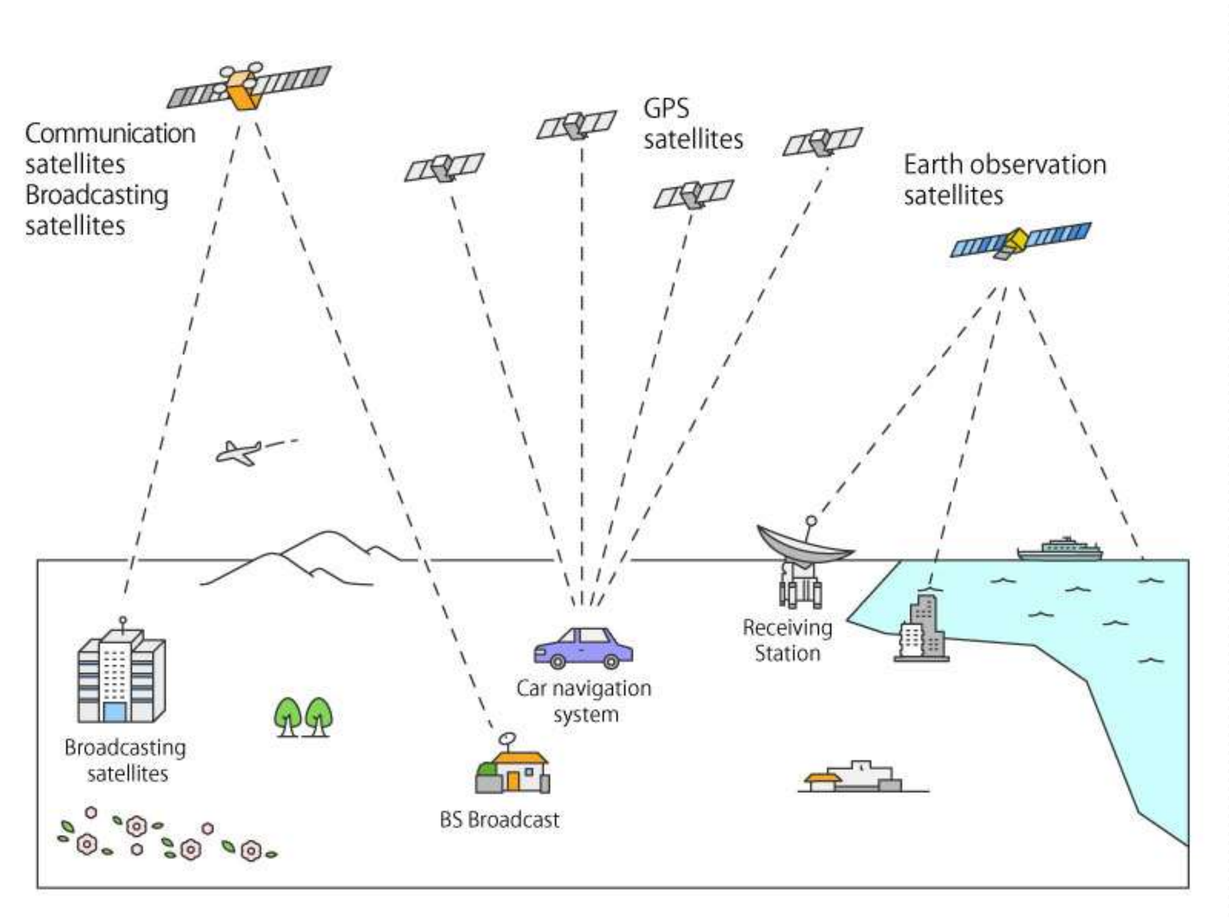
- An interface is the UHF Class-1 Gen-2 tag air interface, which specifies a radio-frequency communications protocol by which an RFID tag and an RFID reader device may interact.
- A component is an RFID tag that is the product of a specific tag manufacturer.
- An EPC Network Service is the ONS, which provides a logically centralized registry through which an EPC may be associated with information services.

Key IoT Technologies
Satellite Technology

- Ability to support mobility in all geographical environments (including Antarctica)
- Global reach
- Offers interesting commercial possibilities


Solar Panels
Transmitter and Transmit Antenna
Receiver and Receive Antenna
Originating Ground Station
Destination Ground Station

Examples



Communication satellites Broadcasting satellites
GPS satellites
Earth observation satellites
Receiving Station
Car navigation system
Broadcasting satellites
BS Broadcast

Evolving IoT Standards

- Covers supporting standards that come into play in the deployment of IoT and machine-to-machine (M2M) services.

Overview and Approaches

- When there is insufficient standardization, capability mismatches between different devices easily arise.
- While IoT systems can utilize existing Internet protocols, power, processing, and capabilities constrained IoT environment can benefit from additional protocols that help optimize the communications and lower the computational requirements.
- Some challenges and modifications because of the larger capability variations than in the current Internet, and because of the fact that there is no human in most applications (M2M), although humans may be in human-to-machine (H2M) situations.

Overview and Approaches

- Standards are particularly critical when there is a requirement to physically or logically connect entities across an interface.
- Some areas requiring standardization which include:
  - Developing IP/routing/transport/web protocols subsets that scale down to IOT devices; specifically, lightweight routing protocols for the IoT;
  - Describing architectures that employ gateways and middleware;
  - Developing mobility management; Internetworking of IoT things
  - Lightweight implementations of cryptographic stacks; and building a suitable security infrastructure: end-to-end security capabilities for the IoT things
  - Developing standards for applications, specifically, data formats;
  - Discouraging on domain-specific solutions.

Overview and Approaches

- Several global organizations are currently working on global M2M standards, layer-specific protocols, optimized architectures, and policy, includes following:
  - The Internet Engineering Task Force (IETF) IPv6 routing protocol for low power and lossy networks (RPL)/routing over low power and lossy networks (ROLL);
  - IETF constrained application protocol (CoAP);
  - IETF constrained RESTful environments (CoRE);
  - IETF IPv6 over low power WPAN (6LoWPAN);
  - 3GPP MTC
  - ETSI M2M.
    - Recall thatM2Minvolves communication without (or only limited) human intervention where the human is not the input agent but possibly (but not always) the output agent.
    - For example, ETSI TS/TR 102 addresses M2M architecture and services (e.g., smart metering, e-health, auto, and city).

Overview and Approaches

- Three major strands of standardization include the following:
  - (i) ETSI: for end-to-end framework for M2M;
  - (ii) 3GPP: to enable operators to support services;
  - (iii) IEEE: to optimize the radio access/physical layer.

IETF IPV6 Routing Protocol for RPL Roll

- IETF- Internet Engineering Task Force
- RPL- Routing Protocol for LLNs
  - LLNs- Low power and Lossy Networks
- ROLL- Routing Over Low power and Lossy networks

IETF IPV6 Routing Protocol for RPL Roll

- Low power and lossy networks (LLNs)
  - A class of networks in which both the routers and their interconnect are constrained.
  - LLN routers typically operate with constraints on processing power, memory, and energy (battery power)
  - their interconnects are characterized by high loss rates, low data rates, and instability. LLNs comprise a few dozen routers up to thousands of routers.
  - Supported traffic flows include
    - point-to-point (between devices inside the LLN),
    - point-to-multipoint (from a central control point to a subset of devices inside the LLN)
    - multipoint-to-point (from devices inside the LLN toward a central control point).
- The IPv6 Routing Protocol for LLNs (RPL) is proposed by the IETF to support multipoint-to-point traffic from devices inside the LLN toward a central control point, as well as point to-multipoint traffic from the central control point to the devices inside the LLN.

IETF IPV6 Routing Protocol for RPL Roll

- LLNs consist largely of constrained nodes
  - with limited processing power, memory, and sometimes energy when they are battery operated or energy scavenging.
- These routers are interconnected by lossy unstable links, resulting in relatively high packet loss rates and typically supporting only low data rates.
- Another characteristic of such networks is that the traffic patterns are not simply point-to-point, but in many cases point-to-multipoint or multipoint-to-point. Furthermore, such networks may potentially comprise up to thousands of nodes.
- To address these issues, the IETF ROLL Working Group has defined application-specific routing requirements for an LLN routing protocol; it has also specified the RPL.

IETF IPV6 Routing Protocol for RPL Roll

- Existing routing protocols include
  - OSPF/IS-IS (open shortest path first/ intermediate system to intermediate system),
  - OLSRv2 (optimized link state routing protocol version 2),
  - TBRPF (topology-based reverse path forwarding),
  - RIP (routing information protocol),
  - AODV (ad hoc on-demand distance vector),
  - DYMO (dynamic MANET on-demand),
  - DSR (dynamic source routing).
- Some of the metrics for IoT applications include the following:
  - Routing state memory space—limited memory resources of low power nodes;
  - Loss response—what happens in response to link failures;
  - Control cost—constraints on control traffic;
  - Link and node cost—link and node properties are considered when choosing routes.
- The existing protocols all fail one or more of these goals for IoT applications.

IETF IPV6 Routing Protocol for RPL Roll

- In order to be use of LLN application domains, RPL separates packet processing and forwarding from the routing optimization objective.
- Examples of such objectives include minimizing energy, minimizing latency, or satisfying constraints.
- Consistent with the layered architecture of IP, RPL does not rely on any particular features of a specific link layer technology.
- RPL is able to operate over a variety of different link layers.

IETF IPV6 Routing Protocol for RPL Roll

- RPL operations, require bidirectional links.
- LLN scenarios, communication links may exhibit asymmetric properties.
  - the reachability of a router needs to be verified before the router can be used as a parent.
- RPL expects an external mechanism to be triggered during the parent selection phase in order to verify link properties and neighbour reachability.
  - Neighbour unreachability detection (NUD) is a mechanism,
  - but alternates are possible, including bidirectional forwarding detection and hints from lower layers via layer 2 triggers.
- In general, a detection mechanism that is reactive to traffic is favored in order to minimize the cost of monitoring links that are not being used.

IETF IPV6 Routing Protocol for RPL Roll

- RPL also expects an external mechanism to access and transport some control information, referred to as the "RPL Packet Information," in data packets.
  - The RPL packet information enables the association of a data packet with an RPL instance and the validation of RPL routing states.
- Example : IPv6 Hop-by-Hop RPL
  - The mechanism is required for all packets except when strict source routing is used which, by nature, prevents endless loops and alleviates the need for the RPL packet information.

IETF IPV6 Routing Protocol for RPL Roll

- RPL provides a mechanism to disseminate information over the dynamically formed network topology to operate autonomously.
- In some applications, RPL assembles topologies of routers that own independent prefixes.
  - A prefix that is owned by a router is advertised as “on-link.”
- RPL have the capability to bind a subnet together with a common prefix and to route within that subnet.
- RPL in particular, disseminate IPv6 neighbour discovery (ND) information prefix information option (PIO) and the route information option (RIO).

IETF IPV6 Routing Protocol for RPL Roll

- Some basic definitions in RPL are as follows :
  - Directed acyclic graph (DAG) is a directed graph with no cycles.
  - Destination-oriented DAG (DODAG) is a DAG rooted at a single destination.
- RPL defines optimization objective when forming paths toward roots based on one or more metrics.
  - Metrics may include both link properties (reliability, latency) and node properties (e.g., powered on not).

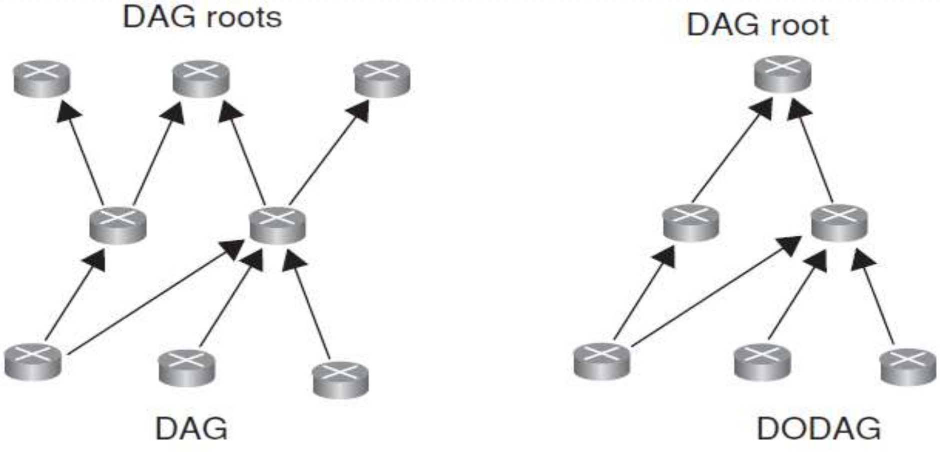
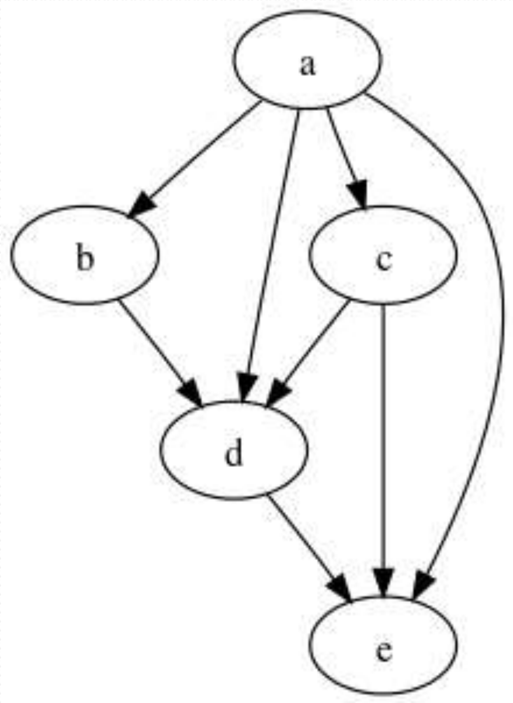
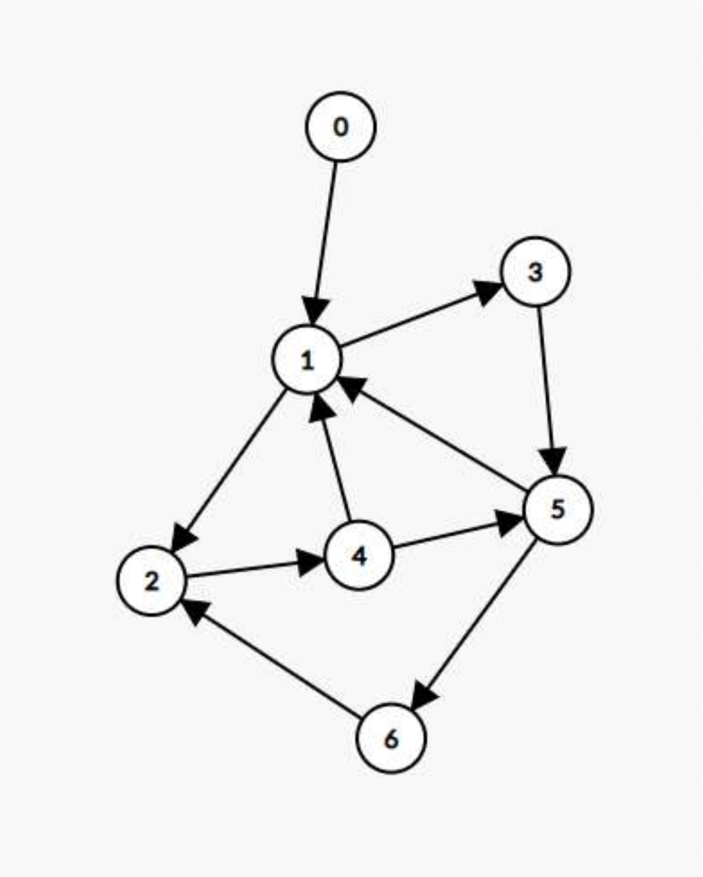


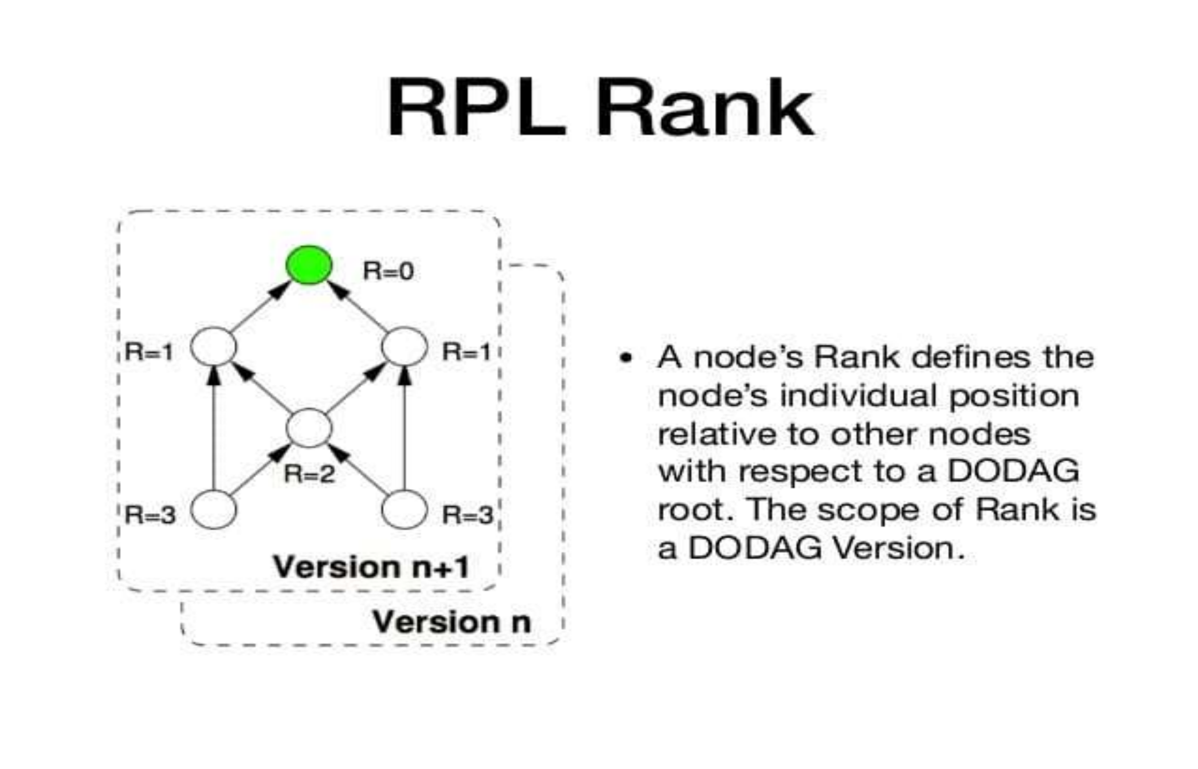
FIGURE 5.1 DAGs and DODAGs.

Example of a directed acyclic and cyclic graph





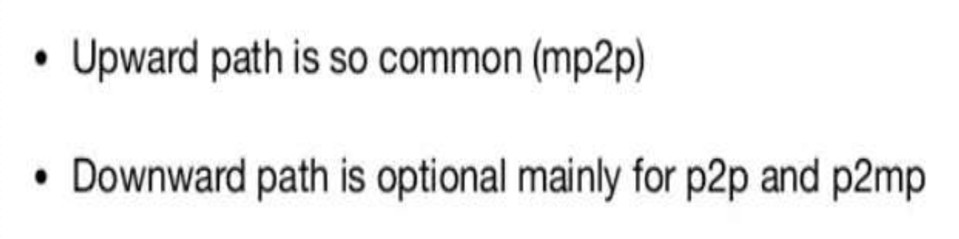
DODAG Rank



RPL Rank

- A node's Rank defines the node's individual position relative to other nodes with respect to a DODAG root. The scope of Rank is a DODAG Version.

Upward—Rank decreases
Downward--- Rank increases



- Upward path is so common (mp2p)
- Downward path is optional mainly for p2p and p2mp

IETF IPV6 Routing Protocol for RPL Roll

- RPL defines a new ICMPv6 (Internet control message protocol version 6) message with three possible types:
  - DAG information object (DIO)—carries information that allows a node to discover an RPL instance, learn its configuration parameters, and select DODAG parents;
  - DAG information solicitation (DIS)—solicit a DODAG information object from an RPL node;
  - Destination advertisement object (DAO)—used to propagate destination information upward along the DODAG.

IETF IPV6 Routing Protocol for RPL Roll

- A node rank defines a node’s relative position within a DODAG with respect to the DODAG root.
- DODAG construction proceeds as follows:
  - Nodes periodically send link-local multicast DIO messages;
  - Stability or detection of routing inconsistencies influence the rate of DIO messages;
  - Nodes listen for DIOs and use their information to join a new DODAG, or to maintain an existing DODAG;
  - Nodes may use a DIS message to solicit a DIO;
  - Based on information in the DIOs, the node chooses parents that minimize path cost to the DODAG root.
- RPL is optimized for many-to-one and one-to-many traffic patterns

Constrained Application Protocol (CoAP)

- Background
- Messaging Model
- Request/Response Model
- Intermediaries and Caching

Constrained Application Protocol (CoAP) Background

- CoAP is a simple application layer protocol targeted to simple electronic devices (e.g., IoT/M2M things) to allow them to communicate interactively over the Internet.
  - CoAP is designed for low power sensors (wireless sensor network [WSN] nodes and actuators.
- CoAP can be seen as a specialized web transfer protocol for use with constrained networks and nodes for M2M applications.
- CoAP operates with HTTP (hypertext transfer protocol) for basic support with the web

Constrained Application Protocol (CoAP) Background

- CoAP protocol are as follows:
  - (i) minimal complexity for the mapping with HTTP;
  - (ii) low header overhead and low parsing complexity;
  - (iii) support for the discovery of resources;
  - (iv) simple resource subscription process;
  - (v) simple caching based on max-age.

Constrained Application Protocol (CoAP) Background

- CoAP makes use of two message types, requests and responses, using a simple binary base header format.
  - Any bytes after the headers in the packet are considered the message body if any.
  - The length of the message body is implied by the datagram length.

Constrained Application Protocol (CoAP) Background

- The constrained nodes for which CoAP is targeted often have 8-bit microcontrollers with small amounts of ROM and RAM, while networks such as 6LoWPAN (IPv6 OVER LOWPOWER WPAN)
- CoAP provides a method/response interaction model between application end-points, supports built-in resource discovery, and includes key web concepts such as URIs (uniform resource identifiers) and content-types.
- CoAP easily translates to HTTP for integration with the web.

Constrained Application Protocol (CoAP) Background

- The use of Web Services (WS) on the Internet has become ubiquitous in most applications; it depends on the fundamental representational state transfer (REST) architecture of the web.

Constrained Application Protocol (CoAP) Background

- CoAP has the following main features:
  - Constrained web protocol fulfilling M2M requirements;
  - UDP (User datagram protocol) binding with optional reliability supporting unicast and multicast requests;
  - Asynchronous message exchanges;
  - Low header overhead and parsing complexity;
  - URI and content-type support;
  - Simple proxy and caching capabilities;
  - A stateless HTTP mapping, allowing proxies to be built providing access to CoAP resources via HTTP in a uniform way or for HTTP simple interfaces to be realized alternatively over CoAP
  - Security binding to datagram transport layer security (DTLS).

Constrained Application Protocol (CoAP) Background

- M2M interactions typically result in a CoAP implementation acting in both client and server roles (called an end-point).
- A CoAP request is equivalent to that of HTTP and is sent by a client to request an action (using a method code) on a resource (identified by a URI) on a server.
- The server then sends a response with a response code; this response may include a resource representation.
- CoAP defines four types of messages:
  - confirmable (CON), non-confirmable (NON), acknowledgement, reset;
- Method codes and response codes included in some of these messages make them carry requests or responses.
- The basic exchanges of the four types of messages are transparent to the request/response interactions.

Constrained Application Protocol (CoAP) Background

- CoAP logically as
  - using a two-layer approach,
  - a CoAP messaging layer used to deal with UDP (User Datagram Protocol)
  - the asynchronous nature of the interactions,
  - the request/response interactions using method and response codes

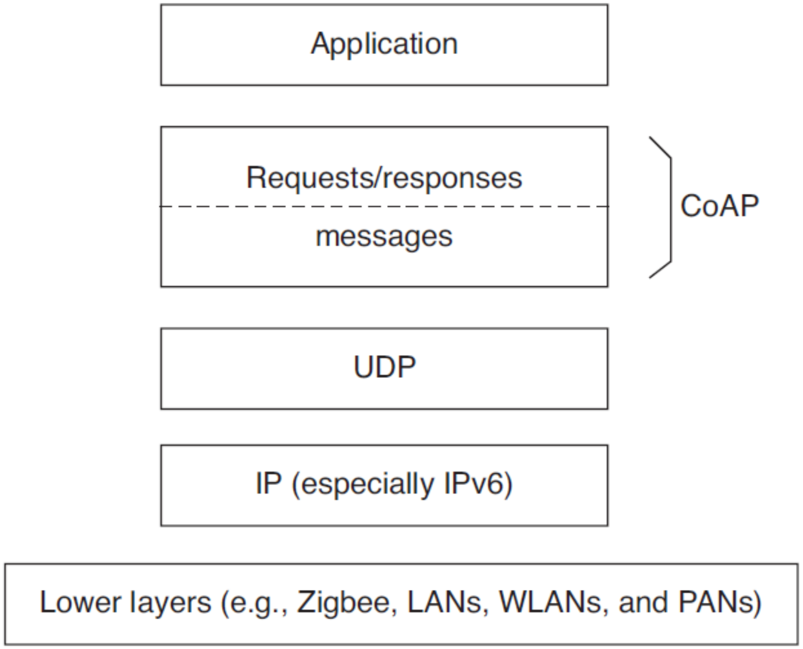


FIGURE 5.2 Abstract layering of CoAP.

Constrained Application Protocol (CoAP) Background

- CoAP is, however, a single protocol, with messaging and request/response just features of the CoAP header.
- Figure depicts the overall protocol stack that is being considered in the CoAP context.

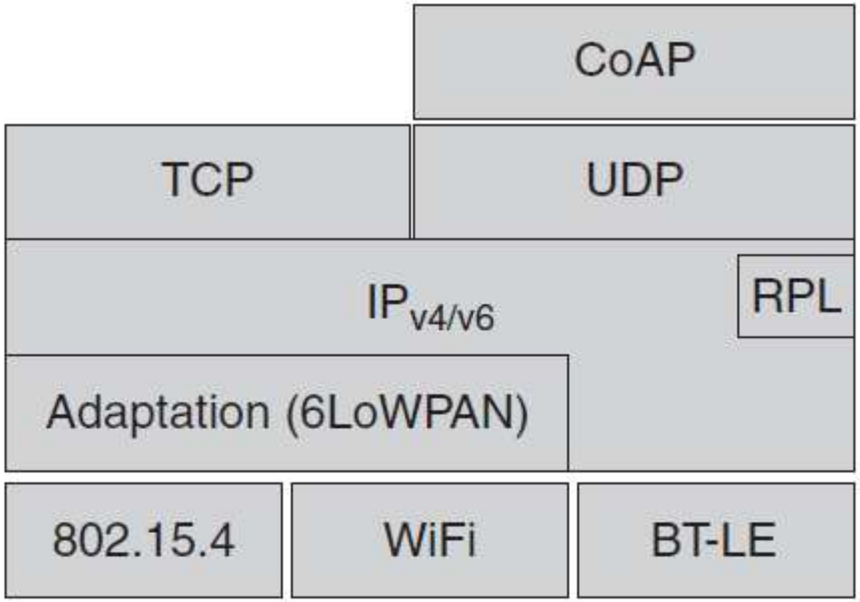


FIGURE 5.3 Overall protocol stack in CoAP's environment.

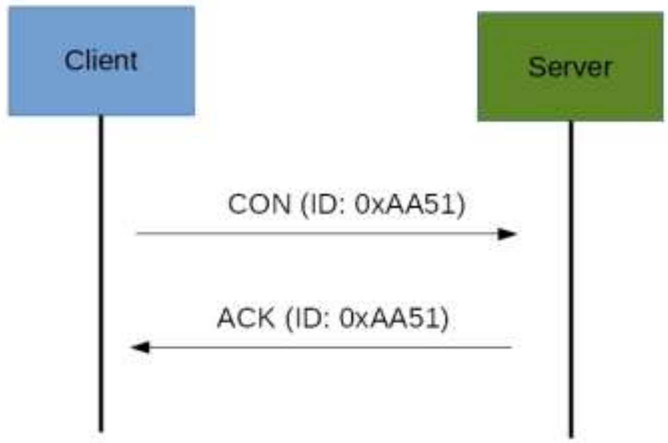
Constrained Application Protocol (CoAP) Messaging Model

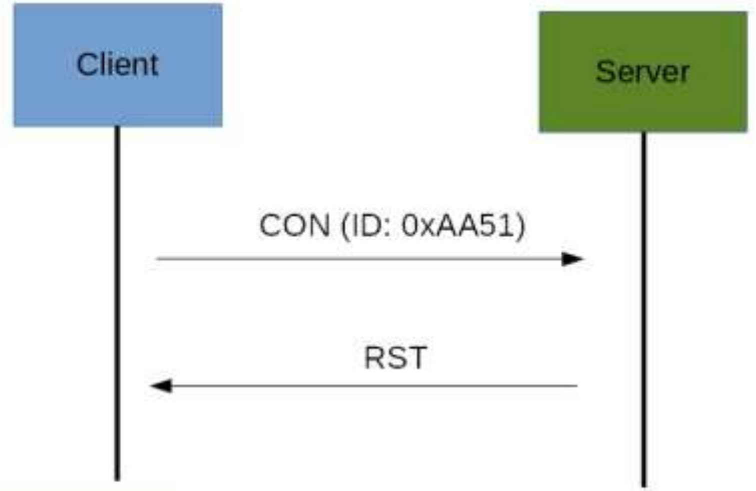
- The CoAP messaging model is based on the exchange of messages over UDP between end-points.
  - It uses a short fixed-length binary header (4 bytes) that may be followed by compact binary options and a payload.
  - This message format is shared by requests and responses.
  - Each CoAP message contains a message ID used to detect duplicates and for optional reliability.

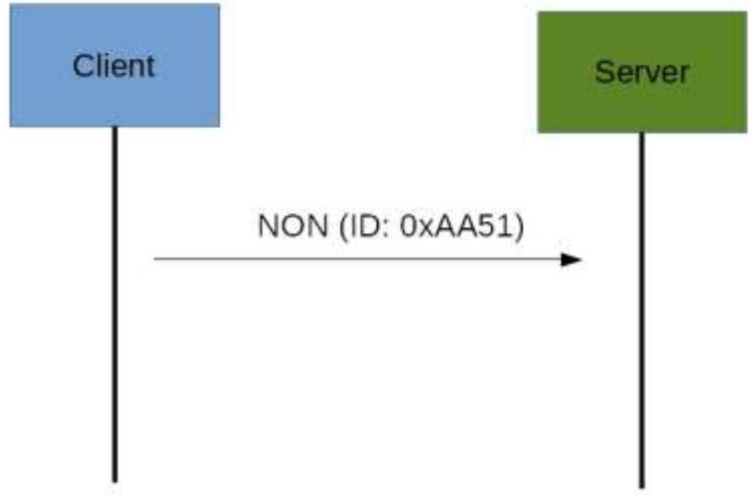
Constrained Application Protocol (CoAP) Messaging Model

- Reliability is provided by marking a message as CON.
- A CON message is retransmitted using a default timeout and exponential back-off between retransmissions, until the recipient sends an acknowledgement message (ACK) with the same message ID from the corresponding end-point.
- When a recipient is not able to process a CON message, it replies with a reset message (RST) instead of an ACK.
- A message that does not require reliable delivery, for example, each single measurement out of a stream of sensor data, can be sent as a NONmessage.
  - These are not acknowledged, but still have a message ID for duplicate detection.
  - When a recipient is not able to process a NON message, it may reply with an RST.
- Since CoAP is based on UDP, it also supports the use of multicast IP destination addresses, enabling multicast CoAP requests.

CON and NON


Client
Server
CON (ID: 0xAA51)
ACK (ID: 0xAA51)


Client
Server
CON (ID: 0xAA51)
RST


Client
Server
NON (ID: 0xAA51)

Constrained Application Protocol (CoAP) Request/Response Model

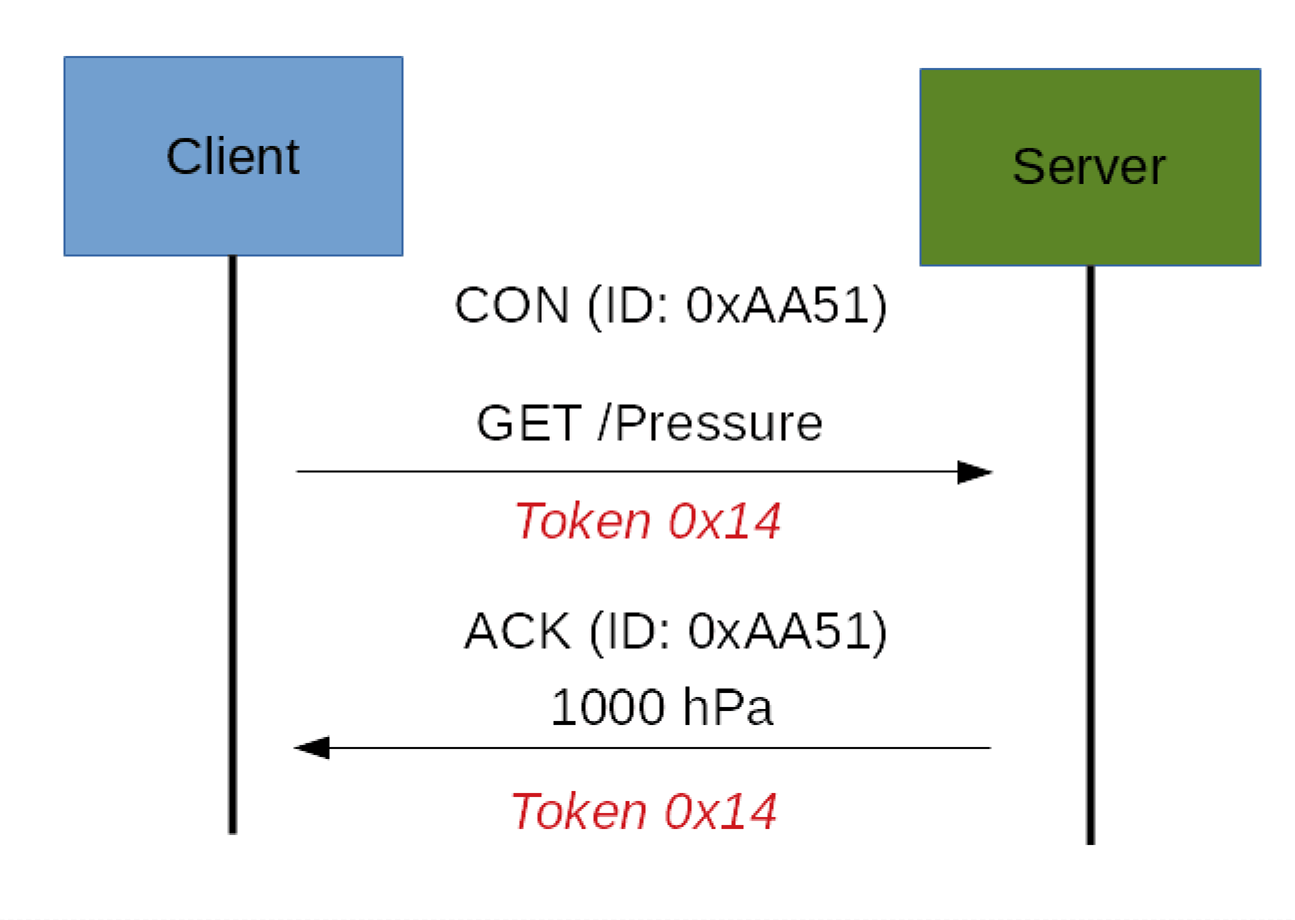
- CoAP messages, which include either a method code or response code, respectively.
- Optional (or default) request and response information, such as the URI (uniform resource identifier) and payload content-type, are carried as CoAP options.
- A token option is used to match responses to requests independent of the underlying messages.

Constrained Application Protocol (CoAP)

Request/Response Model

- A request is carried in a CON (confirmable) or NON (non-confirmable) message, and if immediately available, the response to a request carried in a CON message is carried in the resulting ACK message. This is called a **piggy-backed response**.
- If the server is not able to respond immediately to a request carried in a CON message, it simply responds with an empty ACK message so that the client can stop retransmitting the request.
- When the response is ready, the server sends it in a new CON message (which then in turn needs to be acknowledged by the client). **This is called a separate response.**
- Likewise, if a request is sent in a NON message, then the response is usually sent using a new NON message, although the server may send a CON message.
- CoAP makes use of GET, PUT, POST, and DELETE methods in a similar manner to HTTP.

Request/Response Model


Client

Server

CON (ID: 0xAA51)

GET /Pressure

Token 0x14

ACK (ID: 0xAA51)

1000 hPa

Token 0x14

Constrained Application Protocol (CoAP) Intermediaries and Caching

- The protocol supports the caching of responses in order to efficiently fulfil requests.
- Simple caching is enabled using freshness and validity information carried with CoAP responses.
- A cache could be located in an end-point or an intermediary.

Constrained Application Protocol (CoAP) Intermediaries and Caching

- Proxying is useful in constrained networks for several reasons, including
  - (i) network traffic limiting,
  - (ii) to improve performance,
  - (iii) to access resources of sleeping devices,
  - (iv) for security reasons.
- The proxying of requests on behalf of another CoAP end-point is supported in the protocol.
- The URI of the resource to request is included in the request, while the destination IP address is set to the proxy.

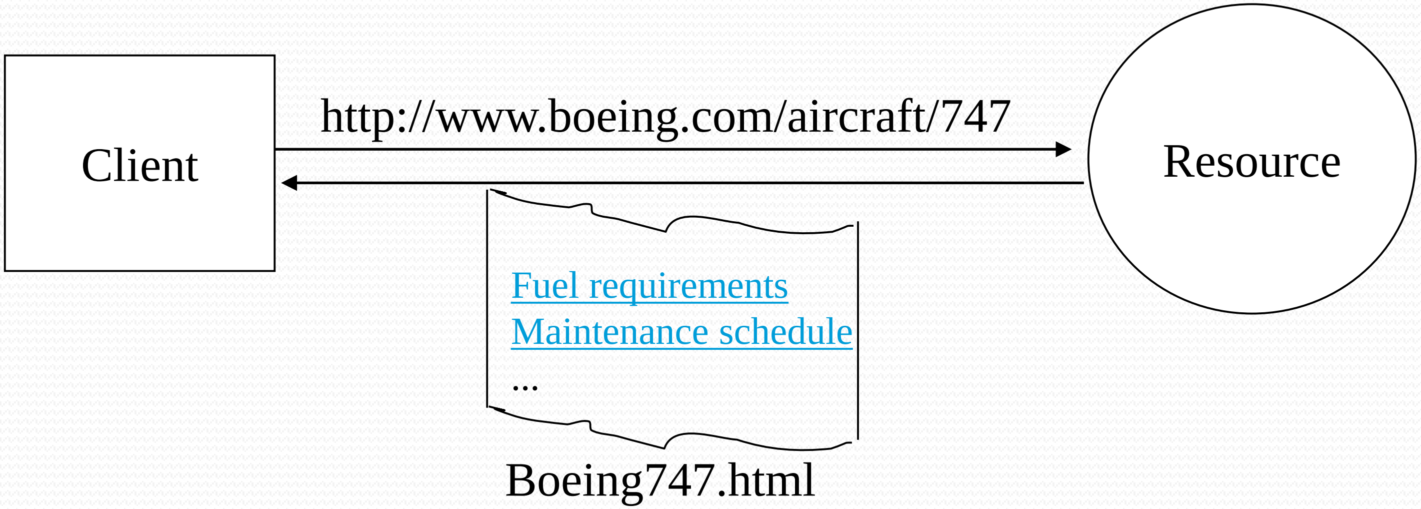
REPRESENTATIONAL STATE TRANSFER (REST)

- As noted, CoAP uses REST techniques. Its like distributed computing.
- REST aims at supporting scalability of component interactions, generality of interfaces, and independent deployment of components.
- It defines a set of architectural principles by which one can design WS that focus on a system’s resources, including how resource states are addressed and transferred over HTTP.
- REST is an architectural style of large-scale networked software that takes advantage of the technologies and protocols of the World Wide Web; it describes how distributed data objects, or resources, can be defined and addressed, stressing the easy exchange of information and scalability.

REPRESENTATIONAL STATE TRANSFER (REST)

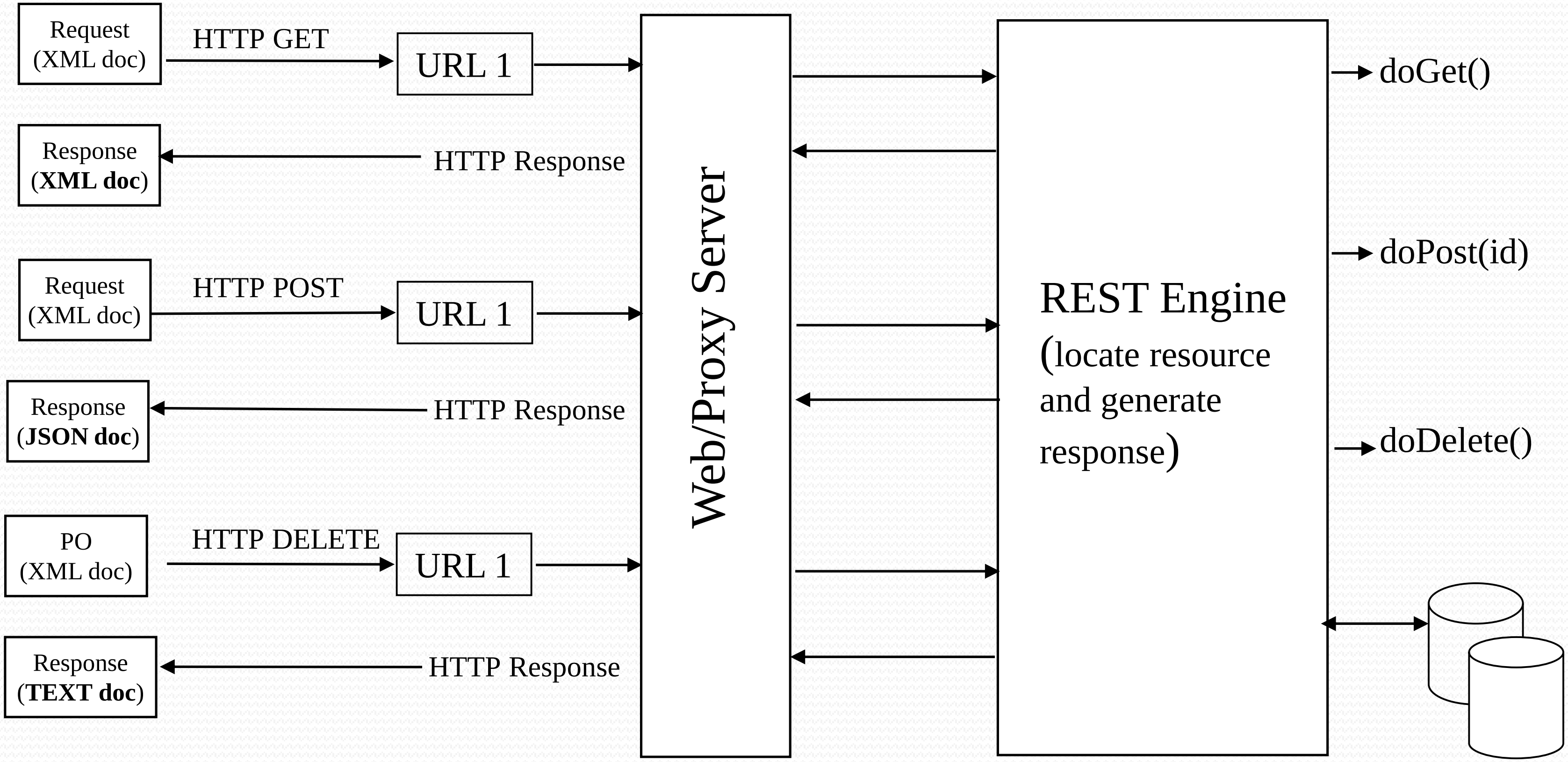
- A REST-based WS follows four basic design principles:
  - Use HTTP methods explicitly.
  - Be stateless.
  - Expose directory structure-like URIs.
  - Transfer XML, JavaScript Object Notation (JSON), or both.

Why is it called "Representational State Transfer"?

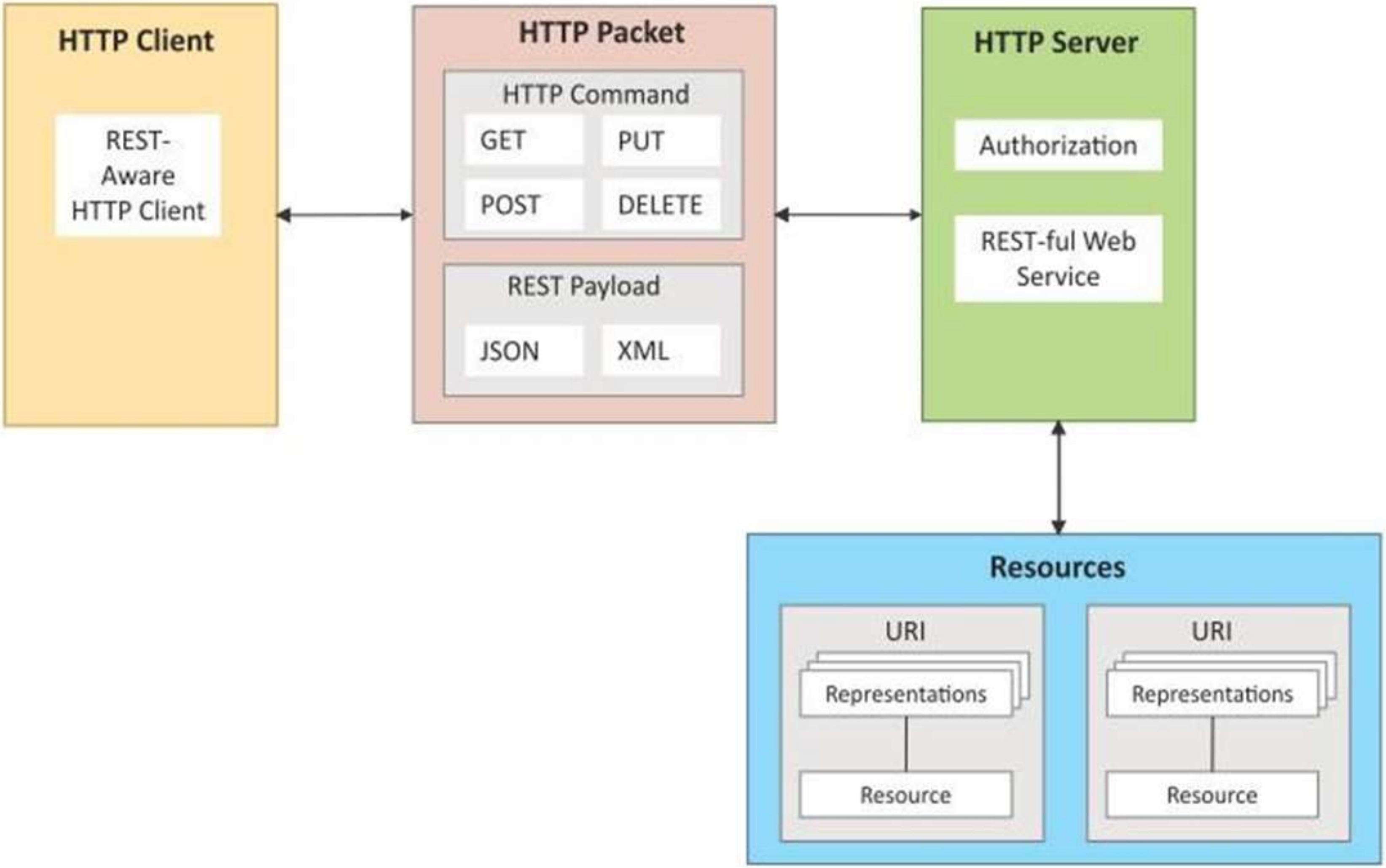


The Client references a Web resource using a URL. A **representation** of the resource is returned (in this case as an HTML document).
The representation (e.g., Boeing747.html) places the client application in a **state**. The result of the client traversing a hyperlink in Boeing747.html is another resource accessed. The new representation places the client application into yet another state. Thus, the client application changes (**transfer**s) state with each resource representation --> Representation State Transfer!

Architecture Style



REST-based Communication APIs



ETSI M2M

- ETSI recently created a dedicated Technical Committee, to develop standard M2M communications.
    - The group seeks to provide an end-to-end view of M2M standardization and is expected to co-operate closely with ETSI’s ongoing activities on next-generation networks (NGNs), radio communications, fiber optics and powerline, as well as collaboration with 3GPP standards group on mobile communication technologies.
- M2M model developed by this group, as defined in various evolving standards, including
    - the ETSI M2M Release 1 standards described in ETSI TS 102 689 (requirements),
    - ETSI TS 102 690 (functional architecture),
    - ETSI TS 102 921 (interface descriptions).

ETSI M2M

- Key elements in the M2M environment include the following :
  - M2M device: A device capable of replying to request for data contained within those device or capable of transmitting data contained within those devices autonomously;
  - M2M area network (device domain): A network that provides connectivity between M2M devices and M2M gateways, for example, a PAN;
  - M2M gateway: A gateway (say a router or higher layer network element) that uses M2M capabilities to ensure M2M devices interworking and interconnection to the communication network;
  - M2M communication networks (network domain): A wider-range network that supports communications between the M2M gateway(s) and M2M application; examples include xDSL, LTE, WiMAX, and WLAN; and
  - M2M applications: Systems that contain the middleware layer where data goes through various application services and is used by the specific business processing engines.

Third Generation Partnership Project Service Requirements for Machine Type Communications (MTC)

- Approach
- Architectural Reference Model for MTC

Third Generation Partnership Project Service Requirements for Machine Type Communications (MTC)
Approach

- Current mobile networks are optimized for human-to-human (H2H) traffic and not for M2M/MTC interactions; hence, optimizations for MTC are advantageous.
- For example, one needs lower costs to reflect lower MTC ARPUs (average revenue per user); also, there is a need to support triggering.
- Hence, 3GPP has started work on M2Mspecification in 2010 for interoperable solutions, particularly in the 3G/4G/LTE context.

Third Generation Partnership Project Service Requirements for Machine Type Communications (MTC)
Approach

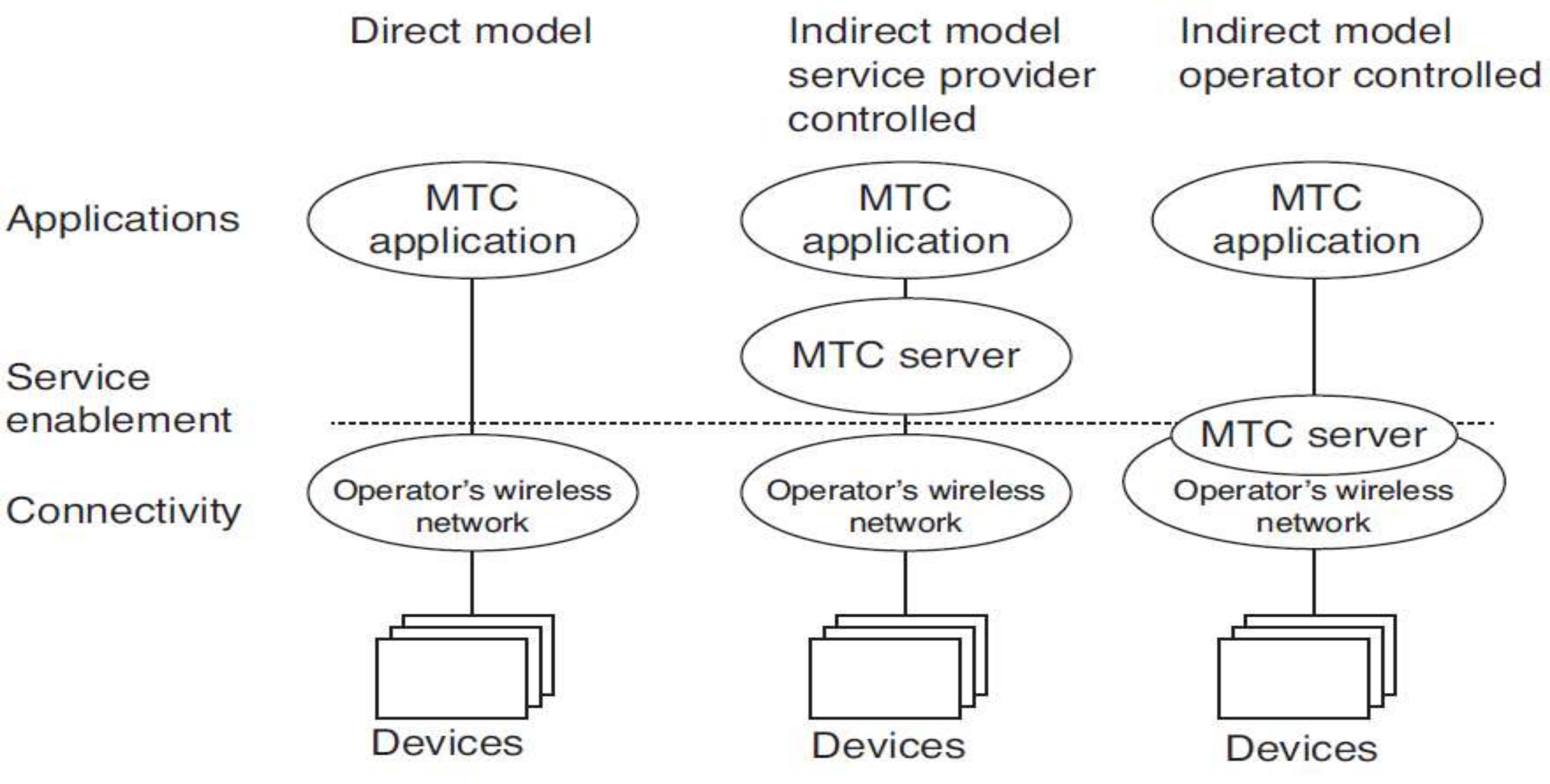
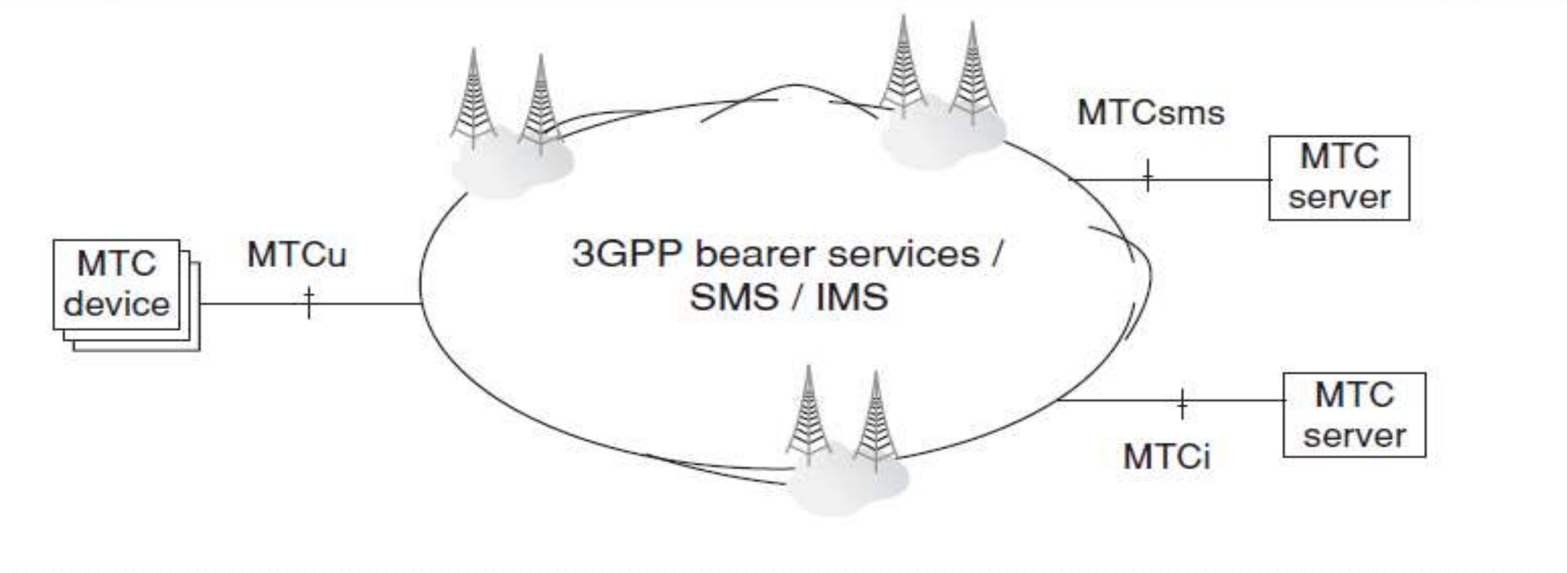

Direct model
Indirect model service provider controlled
Indirect model operator controlled
Applications
Service enablement
Connectivity
MTC application
MTC application
MTC application
MTC server
MTC server
Operator's wireless network
Operator's wireless network
Operator's wireless network
Devices
Devices
Devices

FIGURE 5.4 M2M in 3GPP—service models.

Third Generation Partnership Project Service Requirements for MachIn architectureine Type Communications (MTC) Approach

- , the interfaces are as follows:
  - MTCu: provides MTC devices access to the 3GPP network for the transport of user traffic;
  - MTCi: the reference point for MTC server to connect the 3GPP network via 3GPP bearer service;
  - MTCsms: the reference point for MTC server to connect the 3GPP network via 3GPP SMS.



Third Generation Partnership Project Service Requirements for Machine Type Communications (MTC) Approach

- The key document *3rd Generation Partnership Project Service Requirements for Machine Type Communications*—focused on
  - overload and congestion control,
  - extended access barring (EAB),
  - low priority access,
  - APN (access point name)-based congestion control,
  - downlink throttling.

Third Generation Partnership Project Service Requirements for Machine Type Communications (MTC)

Approach

- For MTC communication, the following communication scenarios are identified:
  - (i) MTC devices communicating with one or more MTC server;
  - (ii) MTC devices communicating with each other.

Third Generation Partnership Project Service Requirements for Machine Type Communications (MTC)
Approach

- For MTC devices communicating with one or more MTC servers, the following use cases exist:
  - (a) MTC server controlled by the network operator; namely the MTC server is located in the operator domain. Here
    - The network operator offers API (e.g., Open Systems Architecture [OSA]) on its MTC server(s)
    - MTC user accesses MTC server(s) of the network operator via API
  - (b) MTC server not controlled by the network operator; namely MTC server is located outside the operator domain. Here
    - The network operator offers the network connectivity to the MTC server(s) located outside of the network operator domain

Third Generation Partnership Project Service Requirements for Machine Type Communications (MTC)
Approach

- MTC applications do not all have the same characteristics.
- This implies that not every system optimization is suitable for every MTC application.
- Therefore, MTC features are to provide structure for the different system optimization possibilities that can be invoked.

Third Generation Partnership Project Service Requirements for Machine Type Communications (MTC) Approach

- The following MTC features have been defined:
  - Low mobility
  - Time controlled
  - Time tolerant
  - Packet switched (PS) only (here the MTC feature PS only is intended for use with MTC devices that only require packet switched services)
  - Small data transmissions
  - Mobile originated only
  - Infrequent mobile terminated
  - MTC monitoring
  - Priority alarm
  - Secure connection
  - Location-specific trigger
  - Network provided destination for uplink data
  - Infrequent transmission

Third Generation Partnership Project Service Requirements for Machine Type Communications (MTC)
Architectural Reference Model for MTC

- *3rd Generation Partnership Project Service Requirements for Machine Type Communications* focuses on numbers and addressing, on improvements of device triggering, and on interfaces between MTC server and mobile network.
- Referring to Figure in next slide,

Third Generation Partnership Project Service Requirements for Machine Type Communications (MTC)

Architectural Reference Model for MTC

MTC-IWF is a new interworking function between (external) MTC server and operator core network handling security, authorization, authentication, and charging.

MTCsp is a new control interface for interactions with MTC server

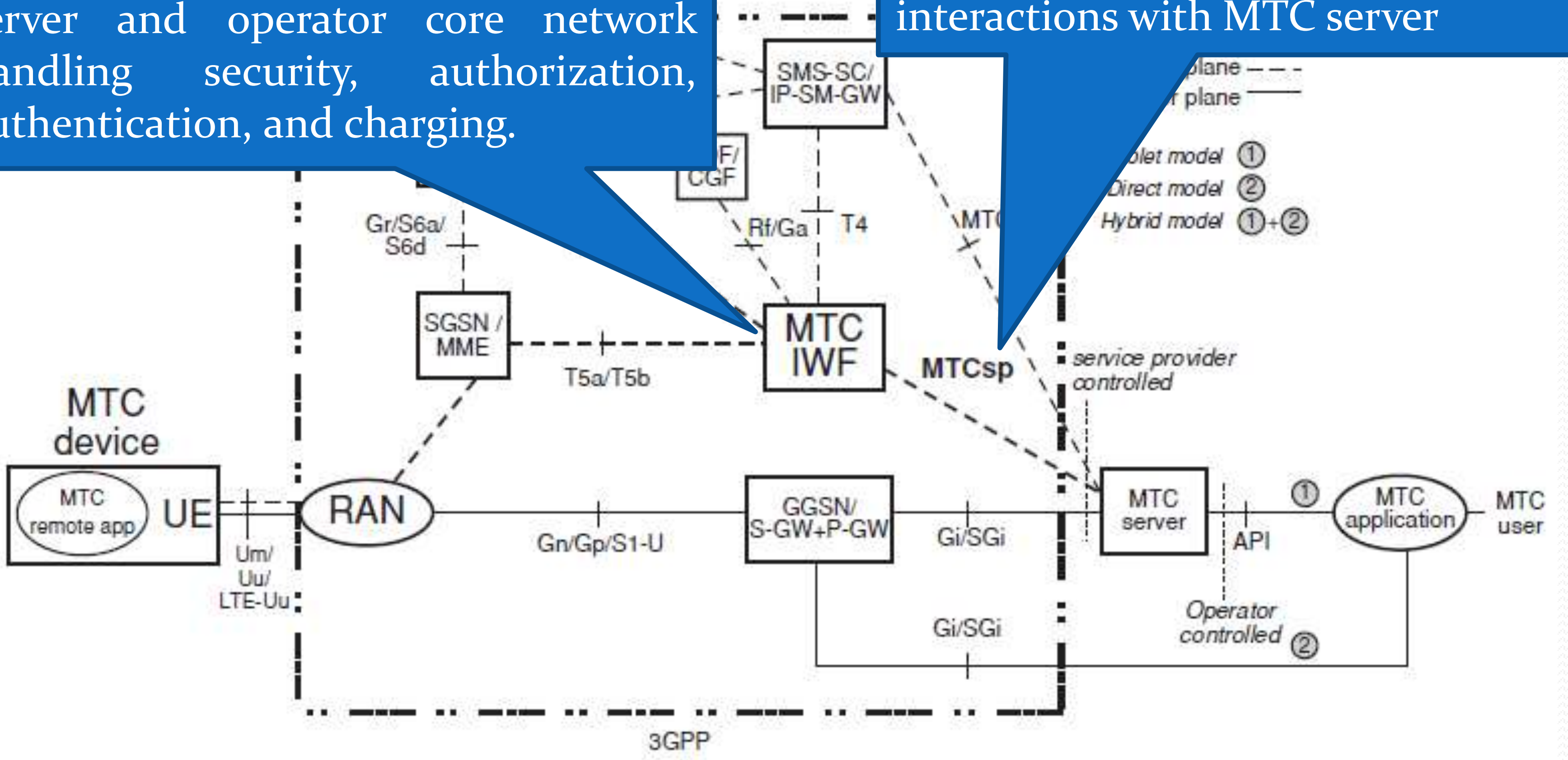


FIGURE 5.5 M2M in 3GPP—Architecture.

https://jwcn-eurasipjournals.springeropen.com/articles/10.1186/s13638-015-0479-y/tables/4

Third Generation Partnership Project Service Requirements for Machine Type Communications (MTC)
Architectural Reference Model for MTC

- The end-to-end application, between the user equipment (UE) used for MTC and the MTC application, uses services provided by the 3GPP system, and optionally services provided by an MTC server.
- The 3GPP system provides transport and communication services (including 3GPP bearer services, IMS, and SMS) including various optimizations that can facilitate MTC.
- UE used for MTC connecting to the 3GPP network (UTRAN, E-UTRAN, GERAN, I-WLAN, and so on) via the Um/Uu/LTE-Uu interface.

Third Generation Partnership Project Service Requirements for Machine Type Communications (MTC)
Architectural Reference Model for MTC

- The architecture encompasses a number of models as follows:
  - Direct model —direct communication provided by the 3GPP operator: The MTC application connects directly to the operator network without the use of any MTC server;
  - Indirect model —MTC service provider controlled communication: The MTC server is an entity outside of the operator domain. The MTCsp and MTCsms are external interfaces (i.e., to a third-party M2M service provider);
  - Indirect model—3GPP operator controlled communication: The MTC server is an entity inside the operator domain. The MTCsp and MTCsms are internal to the public land mobile network (PLMN);
  - Hybrid model: The direct and indirect models are used simultaneously in the hybrid model, for example, connecting the user plane using the direct model and doing control plane signalling using the indirect model.

Third Generation Partnership Project Service Requirements for Machine Type Communications (MTC)
Architectural Reference Model for MTC

- In several countries, regulators have indicated that there are not enough (mobile) numbers available for M2M applications.
- 3GPP postulates that solutions will have to support 100× more M2M devices than devices for H2H communications.
- Proposed solutions include:
  - (i) mid-term solution: special M2M number ranges with longer telephone numbers (e.g., 14 digits);
  - (ii) long-term solution: no longer provide telephone numbers for M2M applications.

Third Generation Partnership Project Service Requirements for Machine Type Communications (MTC)
Architectural Reference Model for MTC

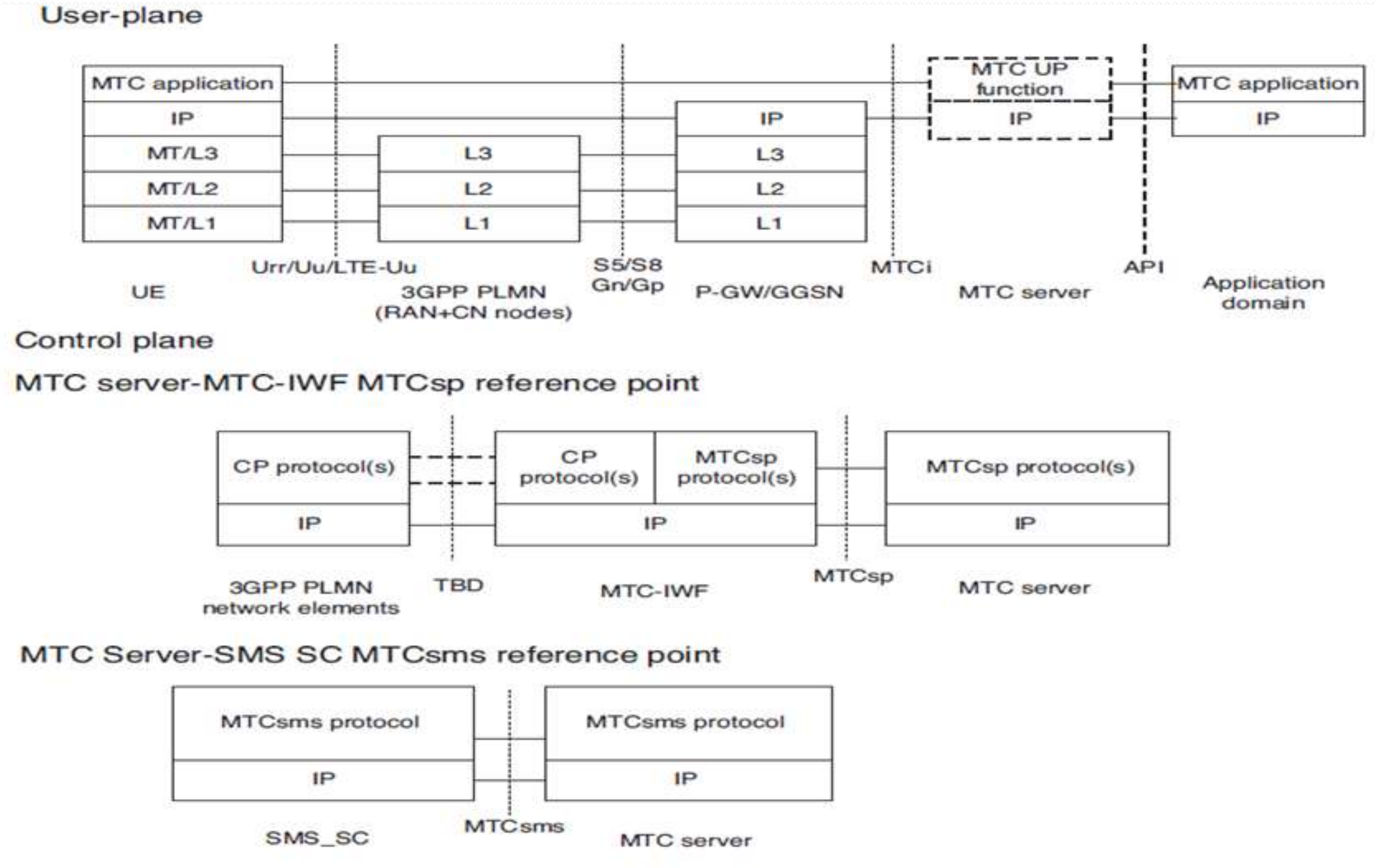


FIGURE 5.6 User and control plane stack for MTC architecture

CENELEC

- European Committee for Electrotechnical Standardization (CENELEC)
  - has adopted the transport profile of Siemens' distribution line carrier communication protocol (CX1) as a standardization proposal.
- The standard aims at supporting open and fault tolerant communication via powerline in intelligent power supply grids.
- As the basis for the transmission protocol, which uses the low voltage network as a communication channel for data of grid sensors and smart meters, the transport profile has been designed to ensure interoperability in accordance with EU Mandate M/441.

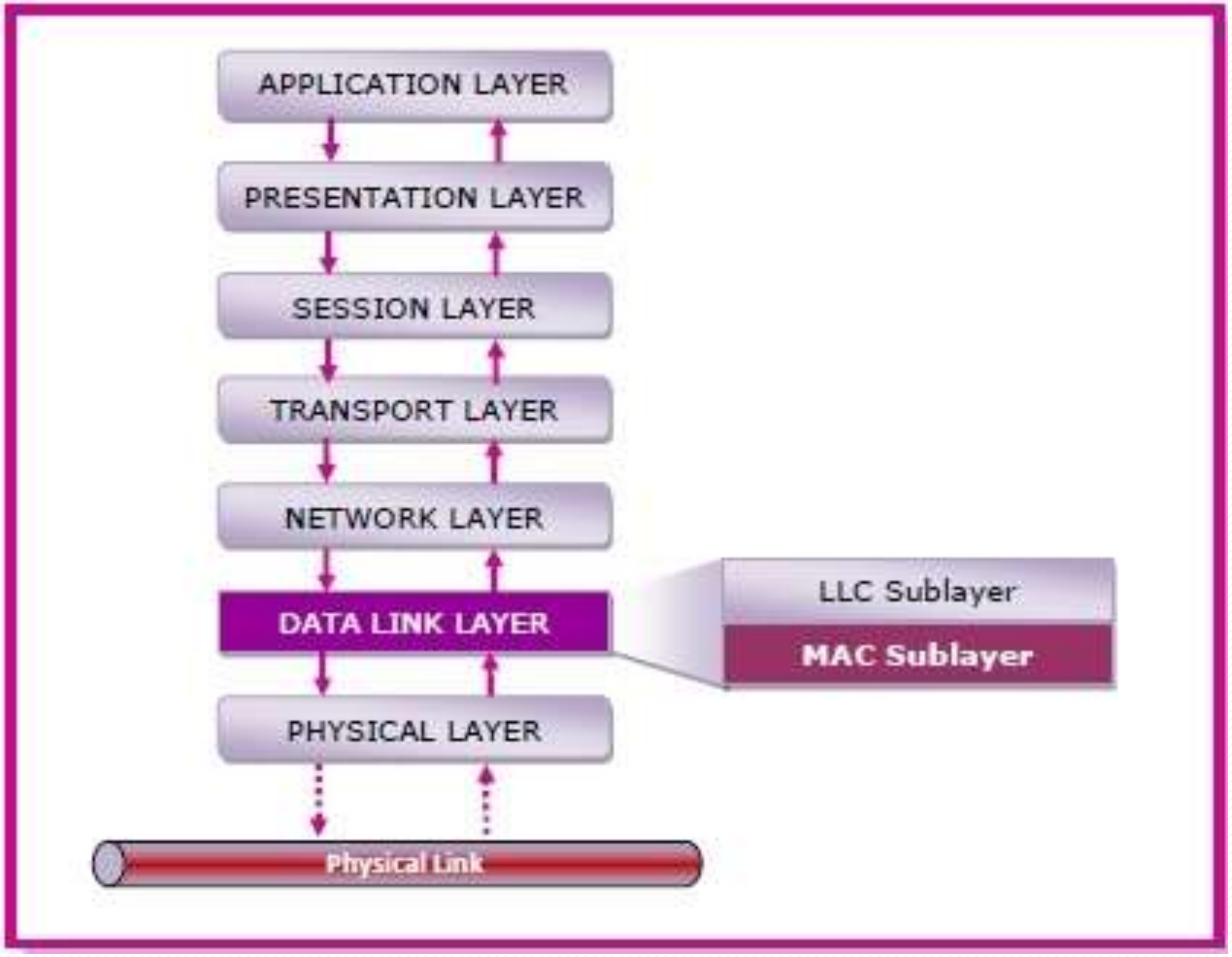
CENELEC

- CENELEC TC 13 was planning to forward the CX1 transport profile to TC 57 of the International Electrotechnical Commission (IEC).
- CX1 is already used to connect meters and other intelligent terminal devices in Siemens’ SG metering systems, such as in the load switching devices that will replace household ripple control receivers.
- The systems collect energy consumption data and network information, which are then relayed to a control center for further processing.
- The communication protocol can handle any change in the physical communication parameters of a low voltage power supply grid, such as signal attenuation, noise, network disruption and signal coupling, as well as operational changes in network configuration.
- The protocol can also be integrated into existing IEC protocol-based network automation and energy management infrastructures.

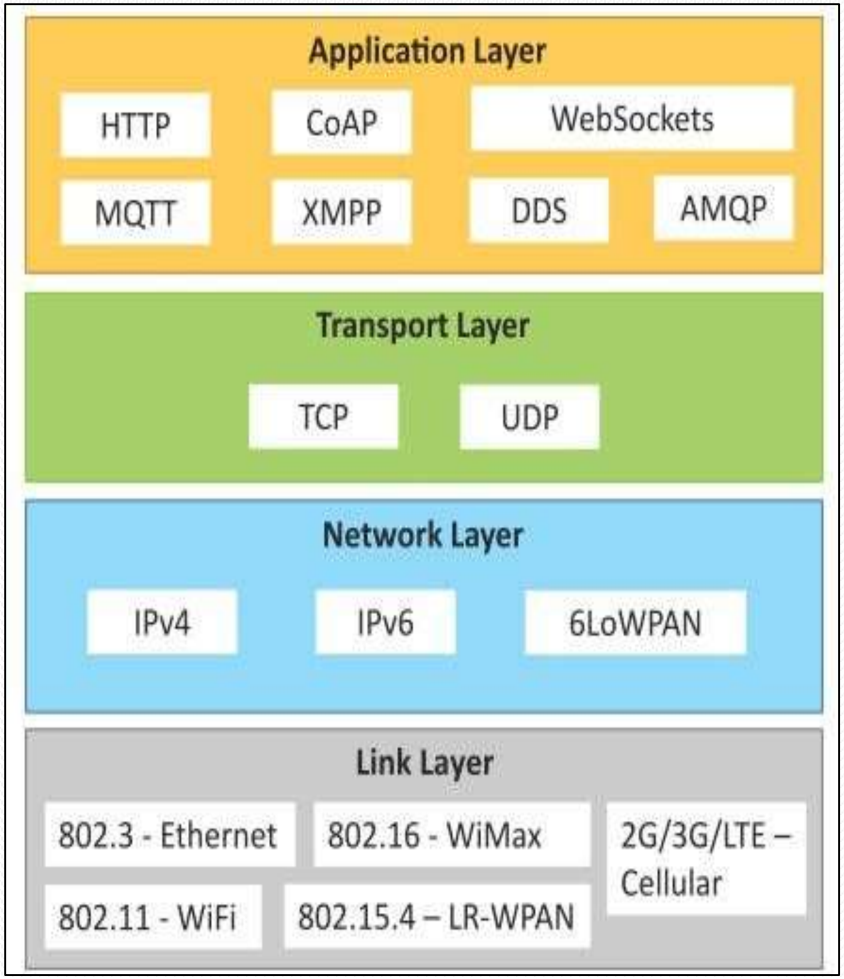
IETF IPv6 Over Lowpower WPAN

- 6LoWPAN is an IPv6 adaption layer for low power wireless PAN (LoWPAN).
- Network with the help of an adaptation layer which sits between the MAC layer and the IP network layer.
- As it should be clear at this juncture, a link in a LoWPAN is characterized as lossy, low power, low bit-rate, short range, with many nodes saving energy with long sleep periods.

OSI Layer


APPLICATION LAYER
PRESENTATION LAYER
SESSION LAYER
TRANSPORT LAYER
NETWORK LAYER
DATA LINK LAYER
LLC Sublayer
MAC Sublayer
PHYSICAL LAYER
Physical Link

IETF IPv6 Over Low power WPAN



Application Layer

HTTP CoAP WebSockets

MQTT XMPP DDS AMQP

Transport Layer

TCP UDP

Network Layer

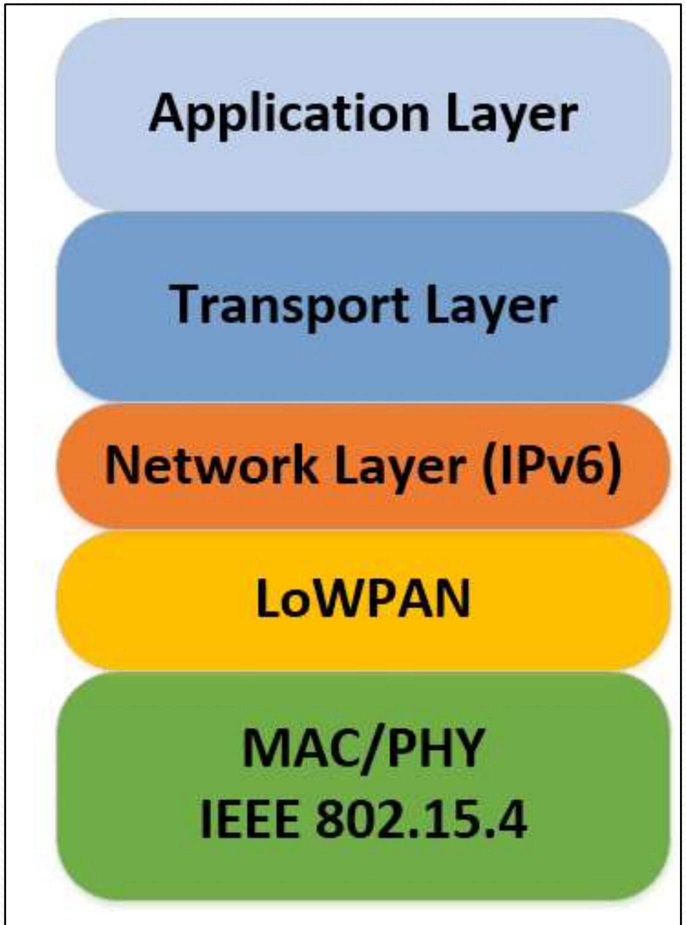
IPv4 IPv6 6LoWPAN

Link Layer

802.3 - Ethernet 802.16 - WiMax 2G/3G/LTE – Cellular

802.11 - WiFi 802.15.4 – LR-WPAN

IOT Protocol Stack



Application Layer

Transport Layer

Network Layer (IPv6)

LoWPAN

MAC/PHY
IEEE 802.15.4

6LoWPAN Protocol Stack

IETF IPv6 Over Low power WPAN

- A LoWPAN is potentially composed of a large number of overlapping radio ranges. Although a given radio range has broadcast capabilities, the aggregation of these is a complex non-broadcast multi access (NBMA) structure with generally no LoWPAN-wide multicast capabilities.
- Link-local scope is in reality defined by reachability and radio strength.
- A LoWPAN to be made up of links with undetermined connectivity properties, along with the corresponding address model assumptions defined therein.

Zigbee IP(ZIP)

- ZigBee is a wireless PAN IEEE 802.15.4 standard
- ZigBee Alliance's ZIP standard, which is a first definition of an open standards-based **IPv6 stack for smart objects.**
- The goal is to extend the use of IP networking into resource-constrained devices over a wide range of low power link technologies.
- **ZIP is a protocol stack based on IETF**- and IEEE-defined standards such as 6LoWPAN and IEEE 802.15.4 to be used for the Smart Energy 2.0 (SE 2.0) profile.

Zigbee IP(ZIP)

- ZIP enables low power 802.15.4 nodes to participate natively with other IPv6-enabled WiFi, Homeplug, and Ethernet nodes without the complexity and cost of application layer gateways.
- To accomplish this, the ZIP stack incorporates a number of standardized IETF protocols including 6LoWPAN for IP header compression and ND (Neighbour Discovery), and RPL for mesh routing.
- ZIP further employs other IETF standards to support network joining procedures, service discovery, and TLS/SSL-based security mechanisms.

IPSO (IP IN SMART OBJECTS)

- The IPSO Alliance is an advocate for IP-networked devices for use in energy, consumer, healthcare, and industrial applications.
- The objective of the Alliance is not to define technologies or standards, but to document the use of IP-based technologies defined at the standard organizations such as IETF with focus on support by the Alliance of various use cases.

IPSO (IP IN SMART OBJECTS)

- Goals include:
  - Promote IP as the premier solution for access and communication for smart objects.
  - Promote the use of IP in smart objects by developing and publishing white papers and case studies and providing updates on standards progress from associations like IETF, among others, and through other supporting marketing activities.
  - Understand the industries and markets where smart objects can have an effective role in growth when connected using the Internet protocol.
  - Organize interoperability tests that will allow members and interested parties to show that products and services using IP for smart objects can work together and meet industry standards for communication.
  - Support IETF and other standards development organizations in the development of standards for IP for smart objects.

Thank you