

**NATIONAL & KAPODISTRIAN
UNIVERSITY OF ATHENS
DEPARTMENT OF INFORMATICS AND
TELECOMMUNICATIONS**

INTERNET OF THINGS

INTERNET OF THINGS

- The term **Internet of Things** (IoT) refers to a network that is more and more spread out
- Objects connected to the Internet, identified and recognized
 - Similar to other devices as computers, tablets, smartphones, etc.



INTERNET OF THINGS

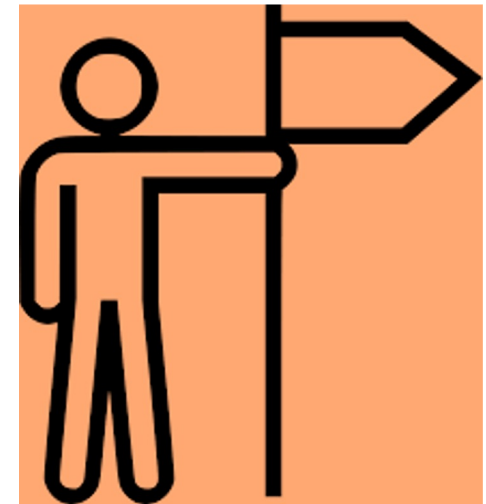
○ Definitions

- Weil & Souissi:

The extension of the current Internet to all objects able to communicate, directly or indirectly, with electronic devices that are themselves connected to the Internet

- IEEE, 2015:

*An IoT [Internet of Things] is a network that connects uniquely identifiable 'Things' to the Internet. The 'Things' have **sensing/actuation** and potential **programmability** capabilities*



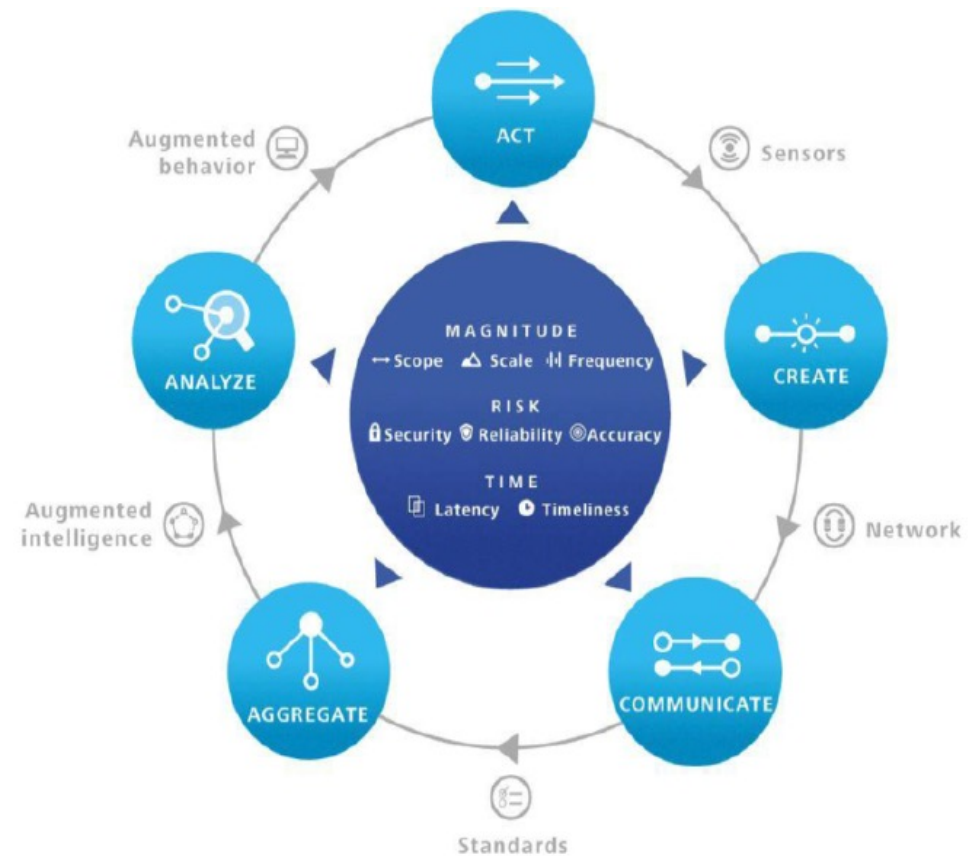
INTERNET OF THINGS

- The technical approach suggests that IoT is an:
 - “[...] extension of the **Internet naming system** and indicates a convergence of digital identifiers [...]”
- The identification methods used for the Internet network can be applied to objects, computers, portable telephones or tablets
- All objects should be identifiable *uniquely* and *automatically*

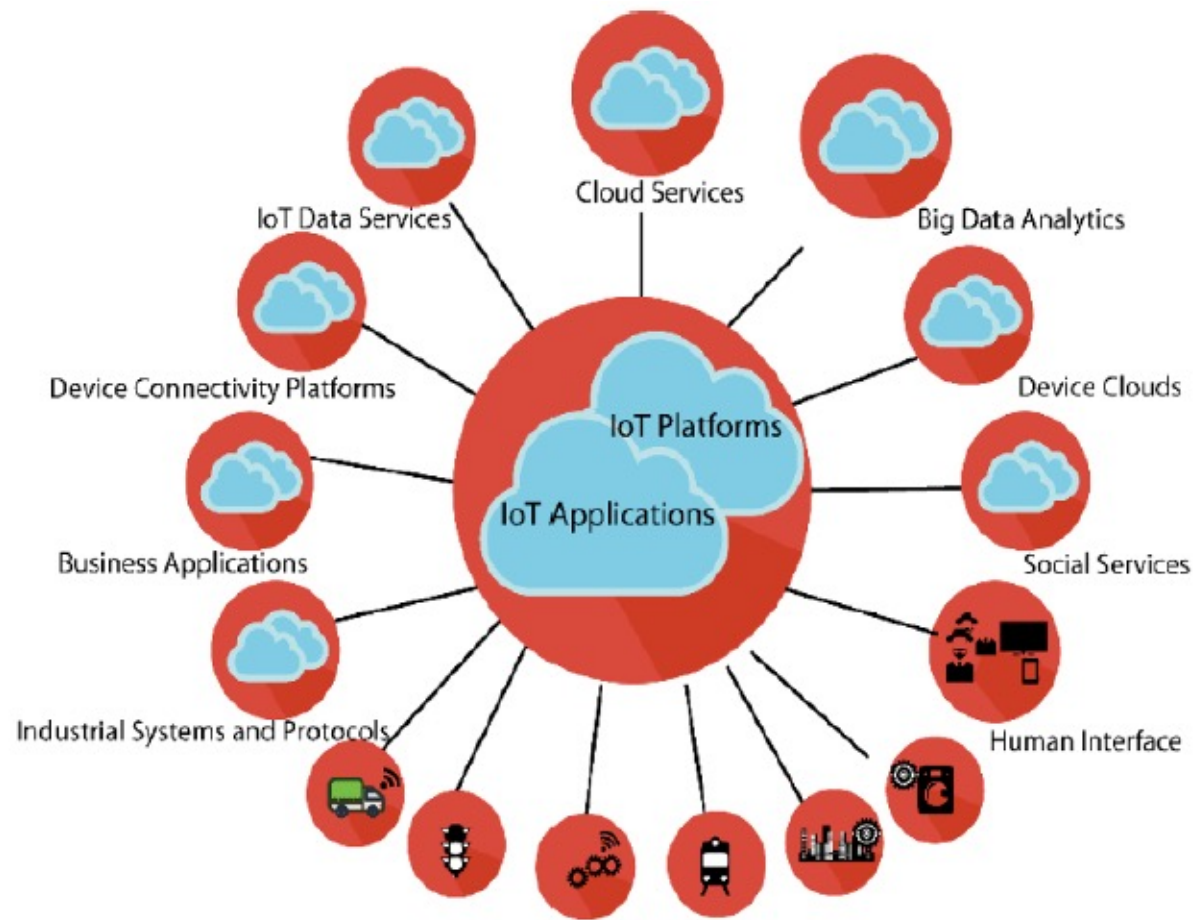


INTERNET OF THINGS

- IoT is a **network of networks** because it is based on the Internet and it makes use of the concept of Internetting
- This network is made up of objects linked to it by means of **wireless communications**
- Connected devices produce data and transfer it on the **network, cloud** platforms storing and then analyzing this data



APPLICATIONS



Home



Energy



Healthcare



Industry



Signage



Tourism



Security



Automotive



Transportation



Environment

APPLICATION EXAMPLES

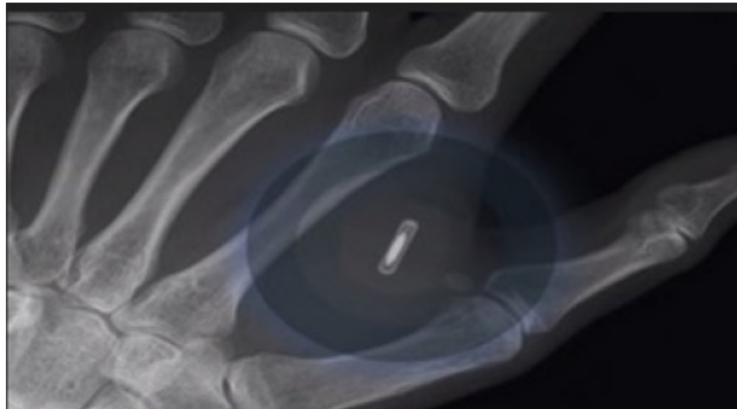
- Home automation: We have a connected and communicating habitat, if not to say intelligent, because several areas are involved in this revolution: **security, energy, lighting, health**, etc.
- Example:
 - controlling entrance to and exit from the house, without the need for door keys (e.g., the “Kevo” lock from Kwikset)



APPLICATION EXAMPLES

○ Payments:

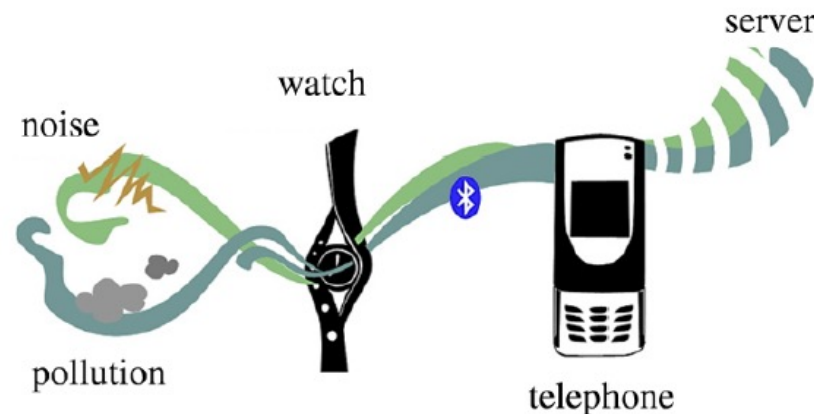
- Contactless payment by chips implanted under the skin of the hand
- To make a payment, you just have to present your hand at payment terminal



APPLICATION EXAMPLES

○ Green Watch Project

- A group of participatory citizen sensors is adopted to measure the levels of ozone and noise in an urban environment
- Geographic localization, necessary for getting the user's coordinates, is done with a GPS chip
- The data are communicated via a mobile terminal (a telephone) with a Bluetooth chip.



IT IS DIFFERENT OUT HERE

- The emergence of the IoT destroys every precedent and preconceived notion of network architecture
- To date, networks have been invented by engineers skilled in **protocols** and **routing** theory
- But the architecture of IoT will rely much more upon lessons derived from **nature** than traditional networking schemes



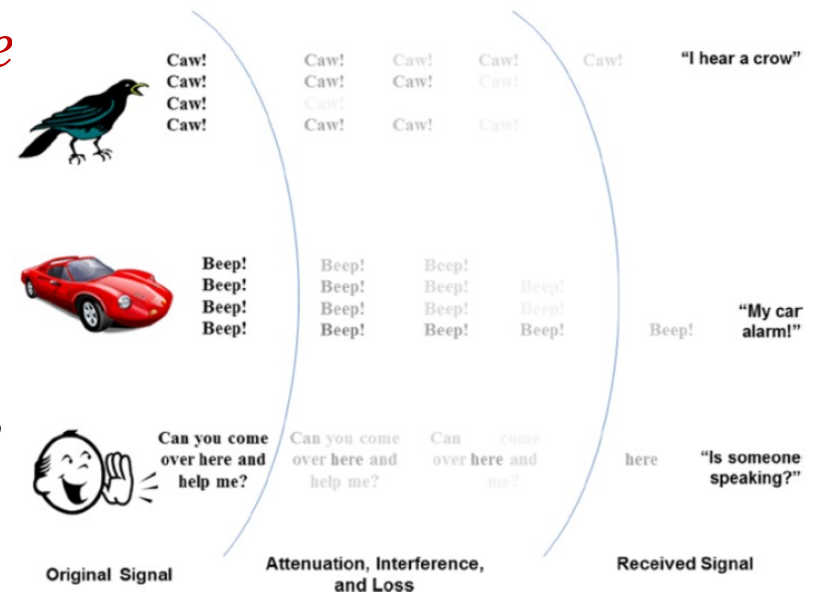
REQUIREMENT FOR NEW SOLUTIONS

- The coming explosion of much simpler devices creates tremendous challenges for networking paradigm in terms of:
 - **number** of devices,
 - Unprecedented **demands** for low-cost connectivity, and
 - impossibility of managing **diverse** equipment



REQUIREMENT FOR NEW SOLUTIONS

- The IoT architecture requires a much more **organic** approach
- **Reason:** *it represents an extreme frontier in communications*
- The **scope** and **breadth** of the devices to be connected are huge
- The connections to the edges of the network where these devices will be “low fidelity”:
 - **low-speed**,
 - **lossy** (where attenuation and interference may cause lost but generally insignificant data, as depicted and
 - **intermittent**



- Much of the communication will be:

- **Difference:** it is completely the *opposite* of networks such as the traditional Internet

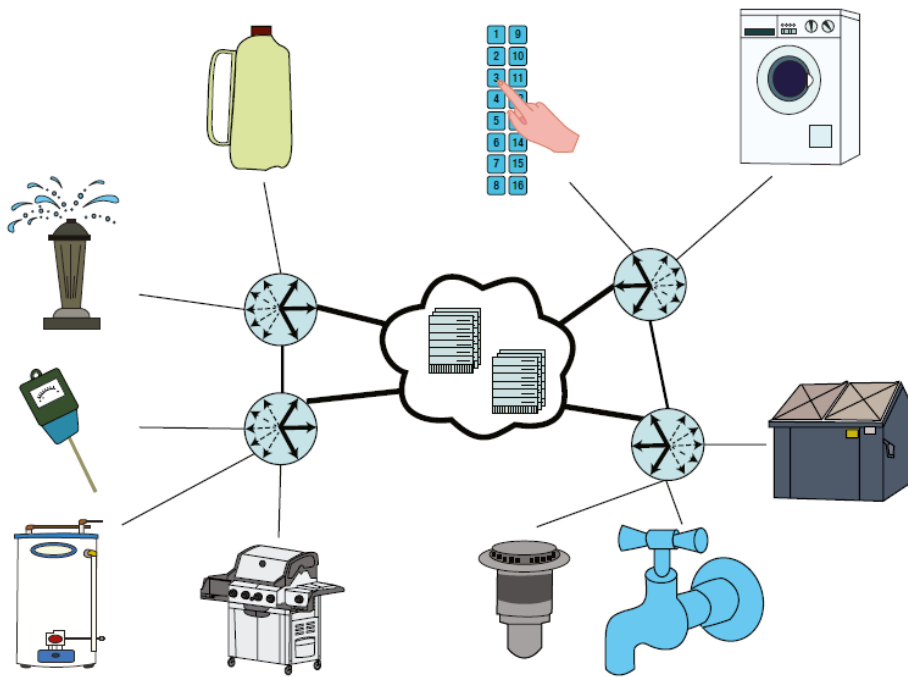


REQUIREMENT FOR NEW SOLUTIONS

- The vast majority of devices to be connected in the coming IoT are **very different**
- Examples:
 - sensors, valve controls, “smart dust,” parking meters, home appliances, etc
- Devices almost **never** contain the processors, memory, hard drives, and other features needed to run a protocol stack
- They are **not necessary** for the end devices’ prime function, and the **costs** of provisioning them with these features would be prohibitive, or at least high enough
- So these simpler devices are very much “**on their own**” at the frontier of the network



IoT SIZE



- The most important aspects of IoT is its incredible breadth and scope
- Devices on the IoT will vastly outnumber human beings on the planet
- The number of devices will continue to grow



IoT SIZE

- Some studies have focused only on the myriad addresses necessary for the devices and have pronounced IPv6 sufficient for the IoT
- But this mistakes *address* space for *addressability*
- *No central address repository or existing address translation scheme can possibly deal with the frontier aspects of the IoT*
- Nor can addresses alone create the costly needed networking “horsepower” within the appliances, sensors, and actuators



IoT SIZE

- Devices **from millions of manufacturers** based in hundreds of countries will appear on the IoT (and disappear) completely **unpredictably**
- This creates one of the greatest challenges of the IoT:

Management

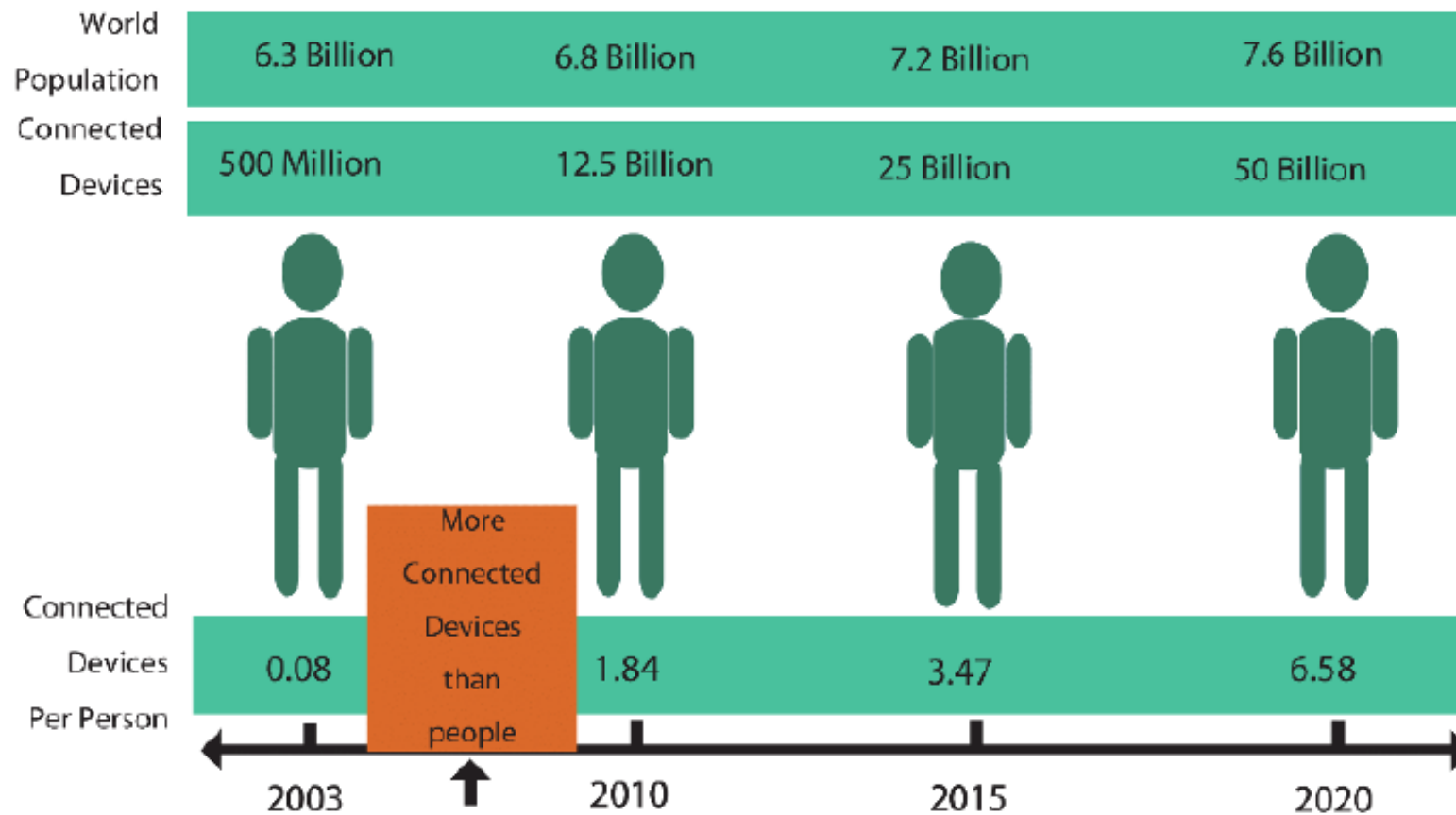


IoT SIZE

- This is a matter both of:
 - **scope** and
 - **device capabilities**
- Example:
 - Smartphones are expected to become the most common computing and communications platforms
 - Their number has recently been placed at 3.5 billion, or roughly one for every two persons on the planet



IoT SIZE



- The estimated **700 billion IoT devices** cannot be individually managed
- They can only be accommodated
- It will simply not be possible to administer the addressing of this huge population of devices through traditional means such as IPv6
- *Nor will it be necessary to do so*
- ***Solution***: Self-addressing and self-classification



EXCHANGE OF DATA

- Much of today's Internet traffic is primarily **human-to-machine** oriented
- Examples: e-mail, web browsing, and video streaming, etc
- **Large chunks** of data generated by machines and consumed by humans
- They tend to be **asymmetrical** and **bursty** in data flows
- A large amount of data exchanged in each “session” or “conversation”



EXCHANGE OF DATA

- The typical IoT data flow will be nearly diametrically opposed to the legacy model
- **Machine-to-machine** communications require minimal **packaging** and **presentation overhead**
- Example: a sensor in a farmer's field may have only a single value to send



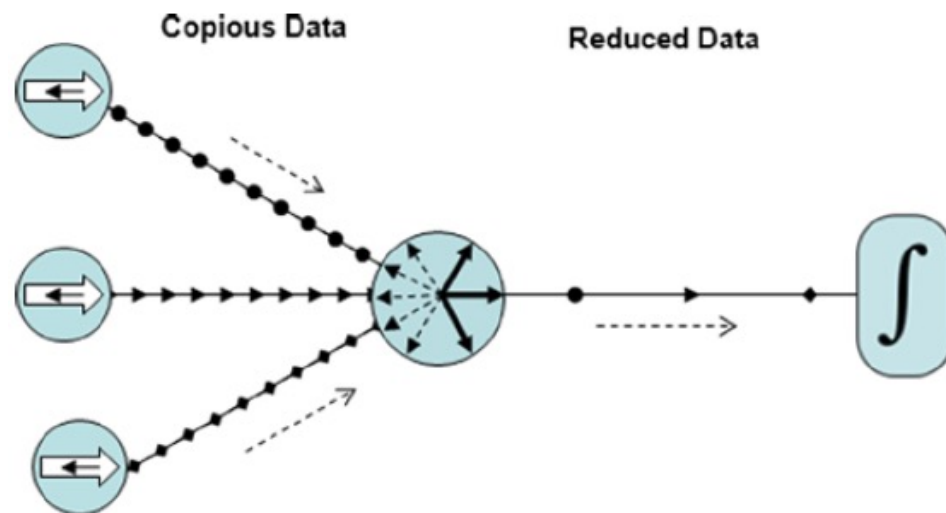
EXCHANGE OF DATA

- Sensors value might change slowly throughout the day, but the frequency of *meaningful* updates will be low
- Many of these IoT devices may be simplex or nearly simplex in data flows
- Example: **broadcasting** a state or **reading** over and over while switched on without even the capacity to “listen” for a reply



EXCHANGE OF DATA

- Another aspect is raised for IoT messages:
 - it's individually *unimportant*
- For simple sensors and state machines, the variations in conditions over time may be small
- Any individual transmission is likely completely **uncritical**
- IoT messages are collected and interpreted elsewhere, and a gap in data will simply be ignored or extrapolated



DEALING WITH LOSS

- Today's traditional Internet is extremely reliable, even if labeled “best effort”
 - **Reason:** Overprovisioning of bandwidth and backbone routing diversity
 - Edge of the Network: the *vast statistical majority of the IoT* connections may often be intermittent and inconsistent in quality
 - Examples:
 - Devices may be switched off at times or powered by solar cells with limited battery back-up
 - Wireless connections may be of low bandwidth or shared among multiple devices
- 

DEALING WITH LOSS

- Traditional protocols such as TCP/IP are designed to deal with **lossy** and **inconsistent** connections by *resending data*
- Even though the data flowing to or from any IoT device may be exceedingly small, it will grow quite large in **aggregate IoT traffic**
- **Redundancy**: The inefficiencies of resending vast quantities of *mostly individually unimportant* data



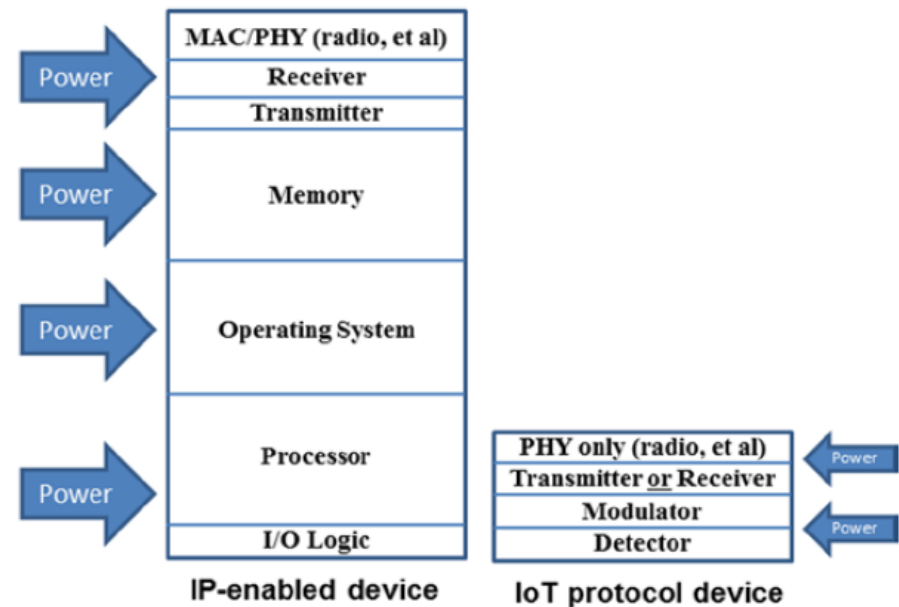
DEALING WITH LOSS

- A key reason why robust protocols aren't needed (or possible) for IoT is:
 - *the required overhead and the minimal processing, memory, and communications capabilities*
- If IoT is considered from the *edge* of the network, the application of legacy protocols becomes impractical
- It makes sense to provide a new solution that can run side by side with existing IP



RISKS

- Devices may be a subject for **hacking** or **inadvertent misconfiguration**
- The operating systems and protocol stacks also require updating and management
- Providing **security** and **upgrades** on the scale of the IoT for a massive number of devices is simply a very difficult task



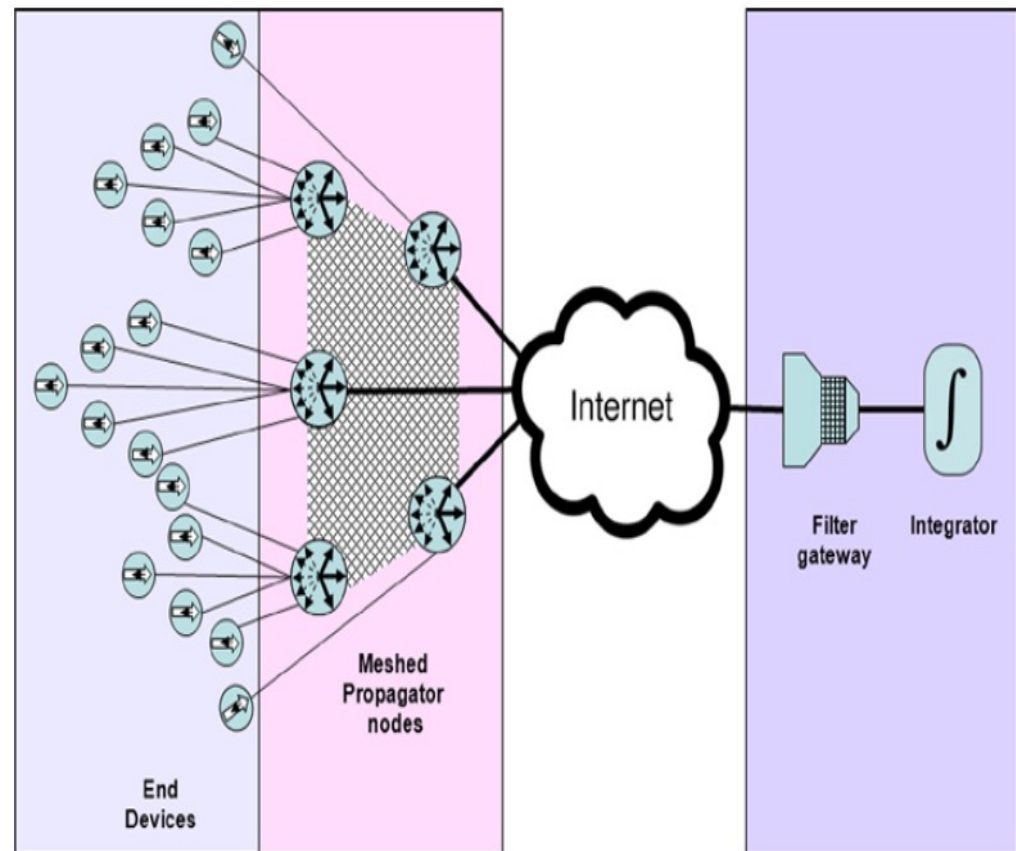
FUNCTIONAL LEVELS

- The network is divided into three functional classes
- It allows deployment of networking functionality (and cost and complexity) only where and when needed
- These three classes are:
 - The *end devices*
 - *Propagator nodes* providing transport and gateways to the traditional Internet
 - *Integrator functions* offering analysis, control, and human interfaces to the IoT



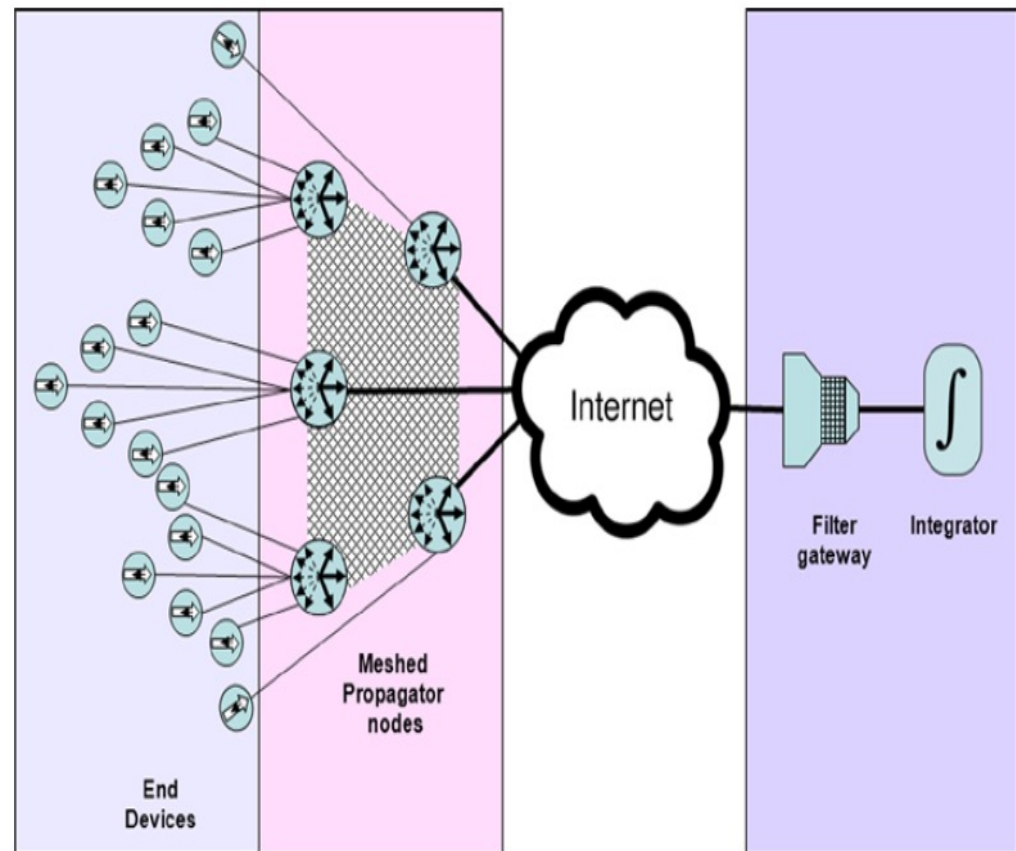
FUNCTIONAL LEVELS

- At the edge of the network are the simple end devices
- They transmit or receive their small amounts of data in a variety of ways:
 - **wirelessly,**
 - via **power line networking,**
 - being **directly connected** to a higher-level device



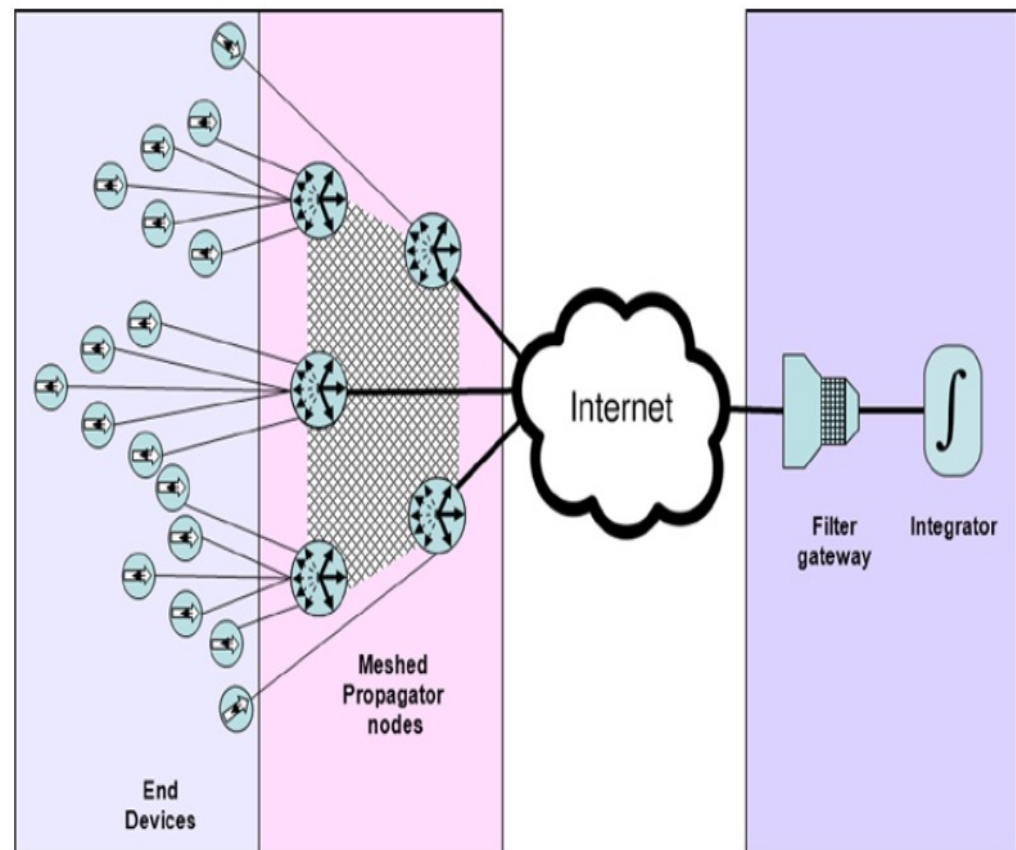
FUNCTIONAL LEVELS

- IoT architecture involves *no error-checking, routing, higher-level addressing*, etc
- Edge devices (Level I) are fairly mindless “worker bees” existing on a minimum of data flow



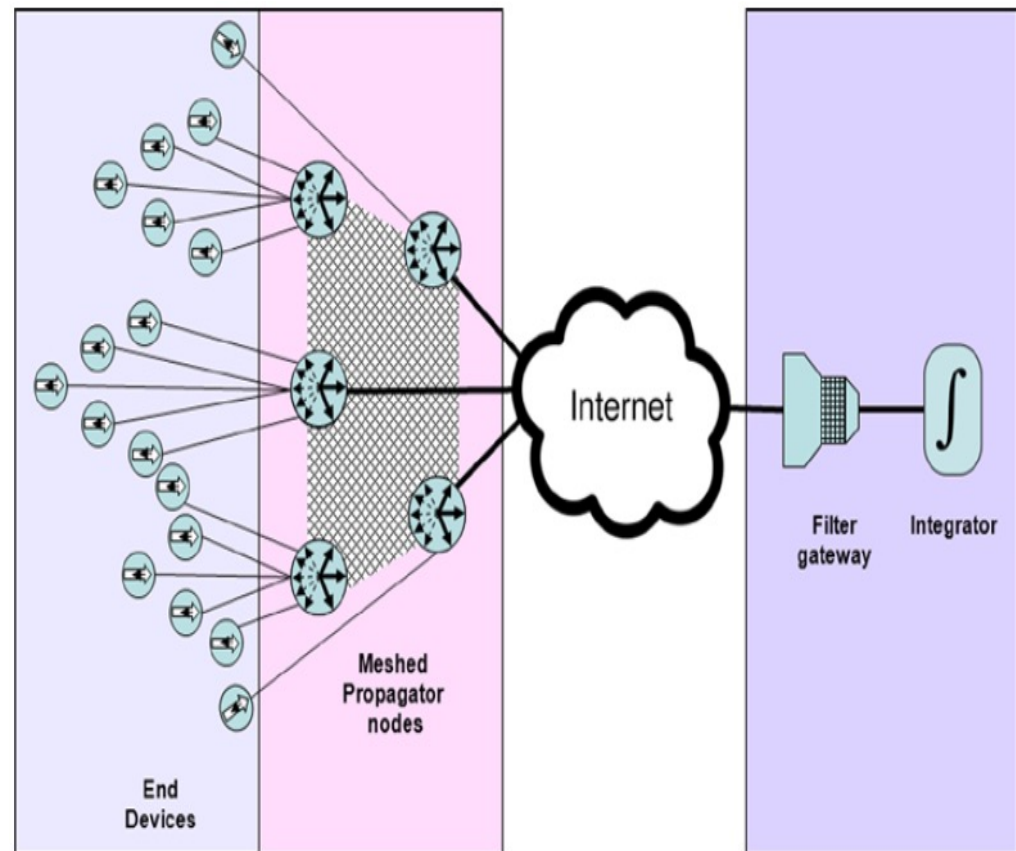
PROPAGATORS

- The protocol intelligence resides elsewhere in the IoT network: within the *Level II propagator nodes*
- They are like familiar traditional networking equipment, e.g., routers
- Based on a *simple set of rules* propagators decide how to broadcast these transmissions to other propagator nodes or to the higher-level *integrator devices*



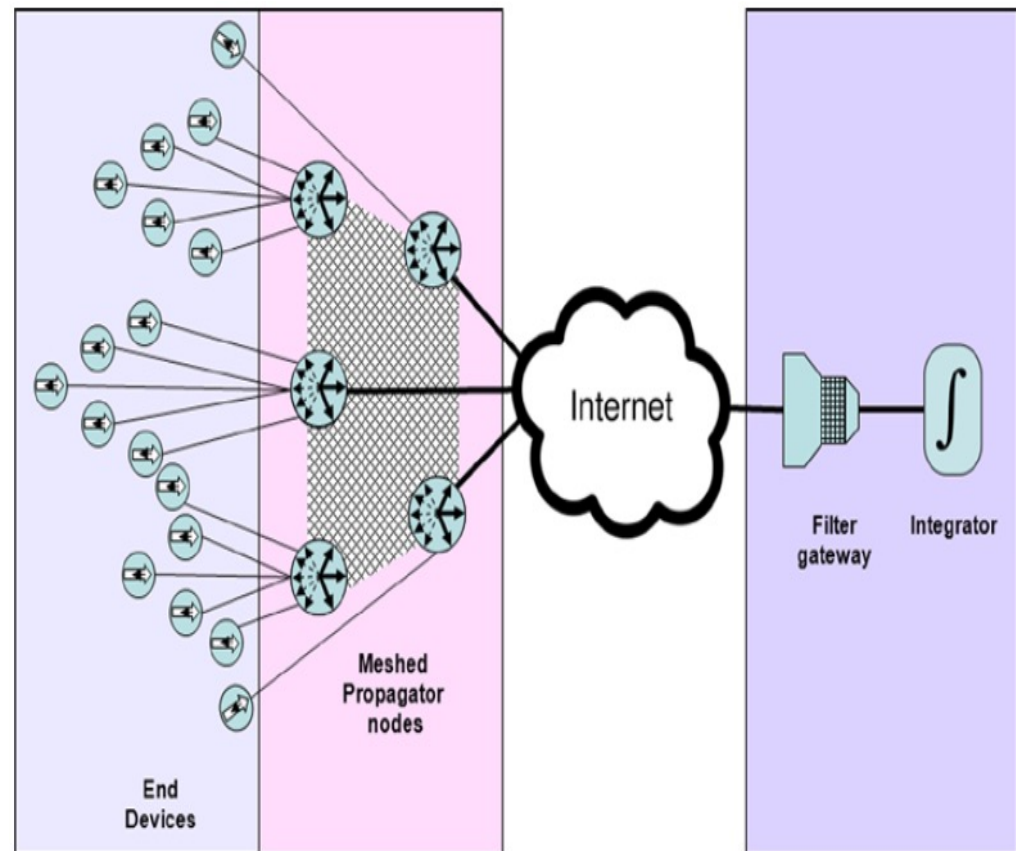
PROPAGATORS

- Scalability:
 - Propagators must be capable of a great deal of **discovery** and **self-organization**
- They will:
 - **recognize** other propagator nodes,
 - **set up** simple routing tables of adjacencies, and
 - **discover** likely paths to the appropriate integrators



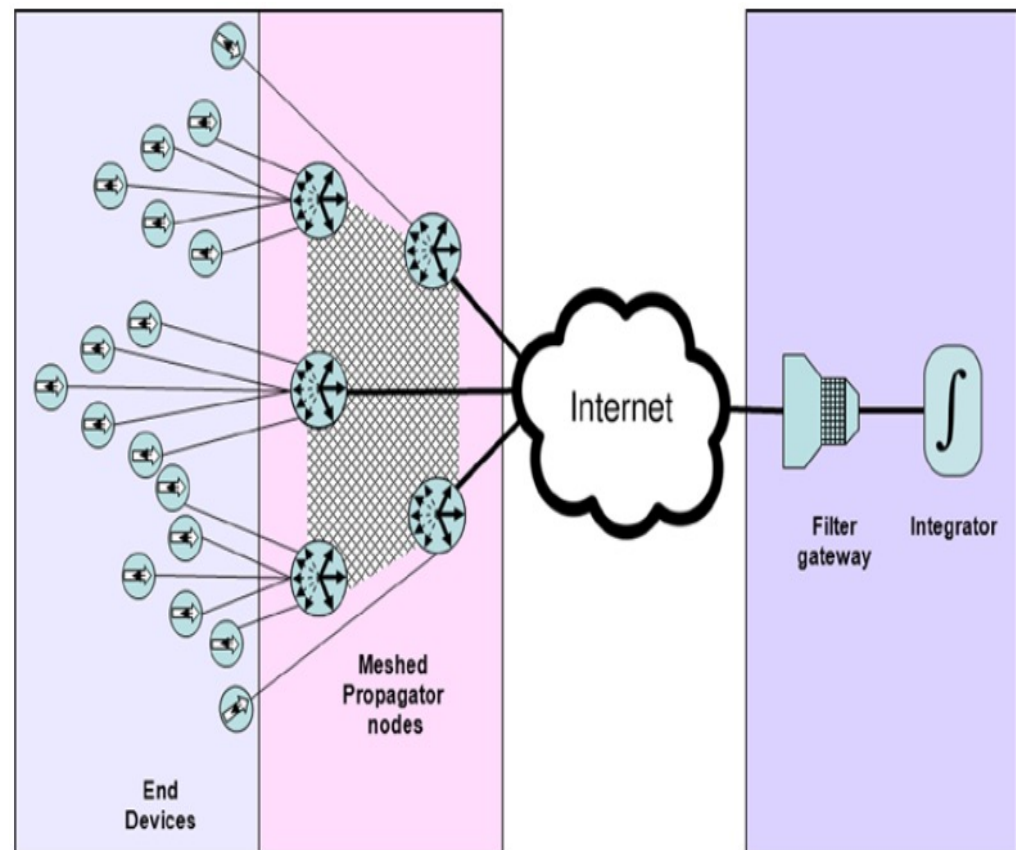
PROPAGATORS

- Capabilities:
 - able to prune and
 - optimize broadcasts
- Data passing from and to end devices may be **combined** with other traffic and forwarded in the general direction of their transmission “arrow”
- Any of the standard networking protocols can be used
- Propagators will perform **translation** between different networks (power line or Bluetooth to ZigBee or Wi-Fi, for example).



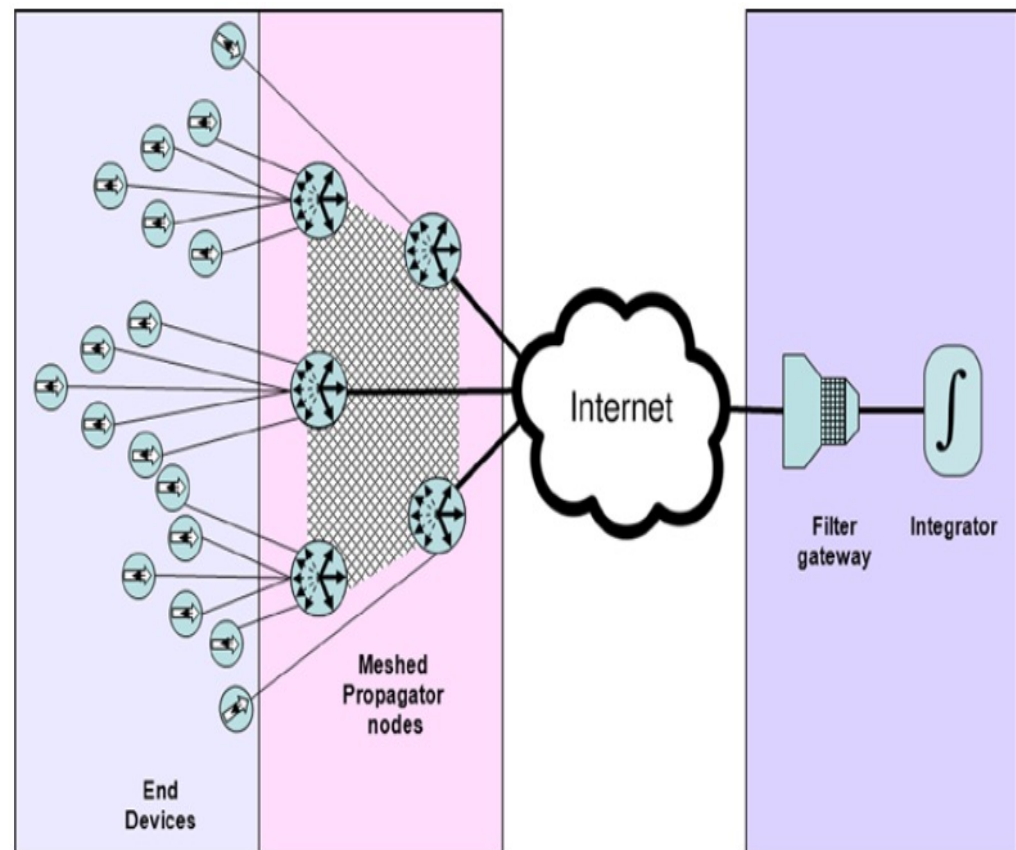
PROPAGATORS

- Additional capability:
 - the capacity to be managed and “tuned” by integrator functions across the network
- This will take the form of a software *publishing agent*
- This publishing agent will become part of the information “neighborhood” created by one or more integrator functions



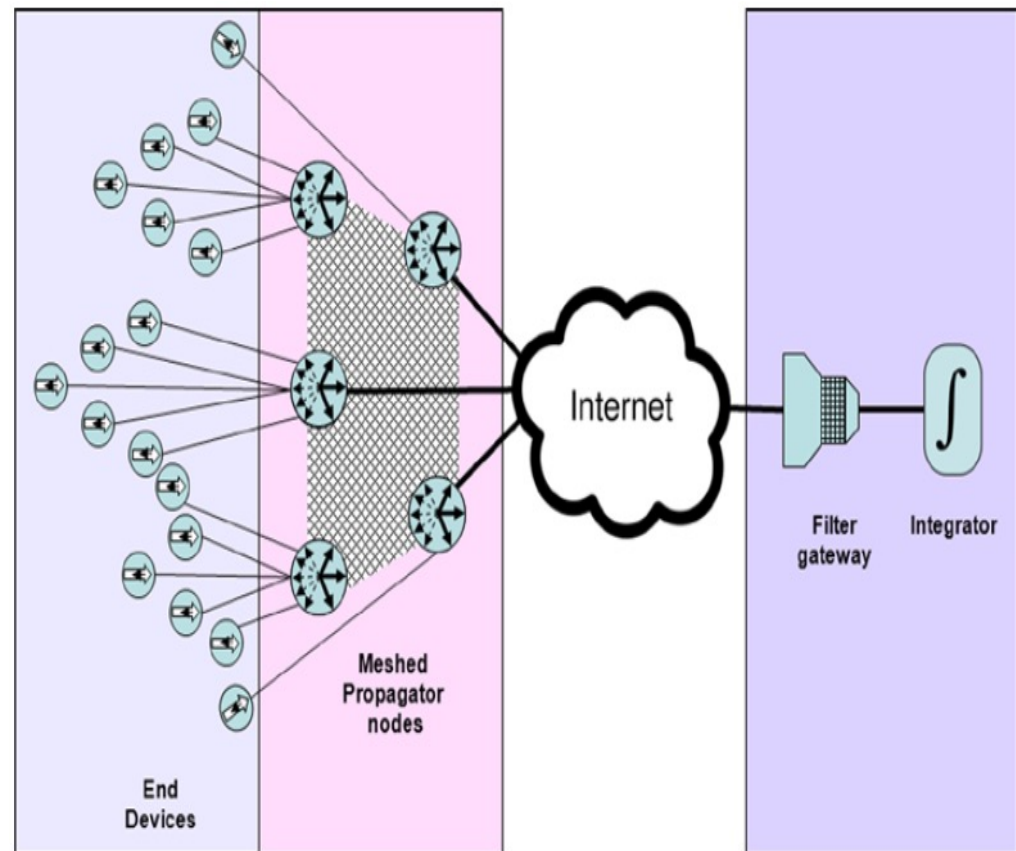
PROPAGATORS

- Similar to **Software Defined Networking**, the integrator function will apply **higher-level management to particular propagator nodes**
- Examples: frequency of data transmission, network topology, and other networking functionality



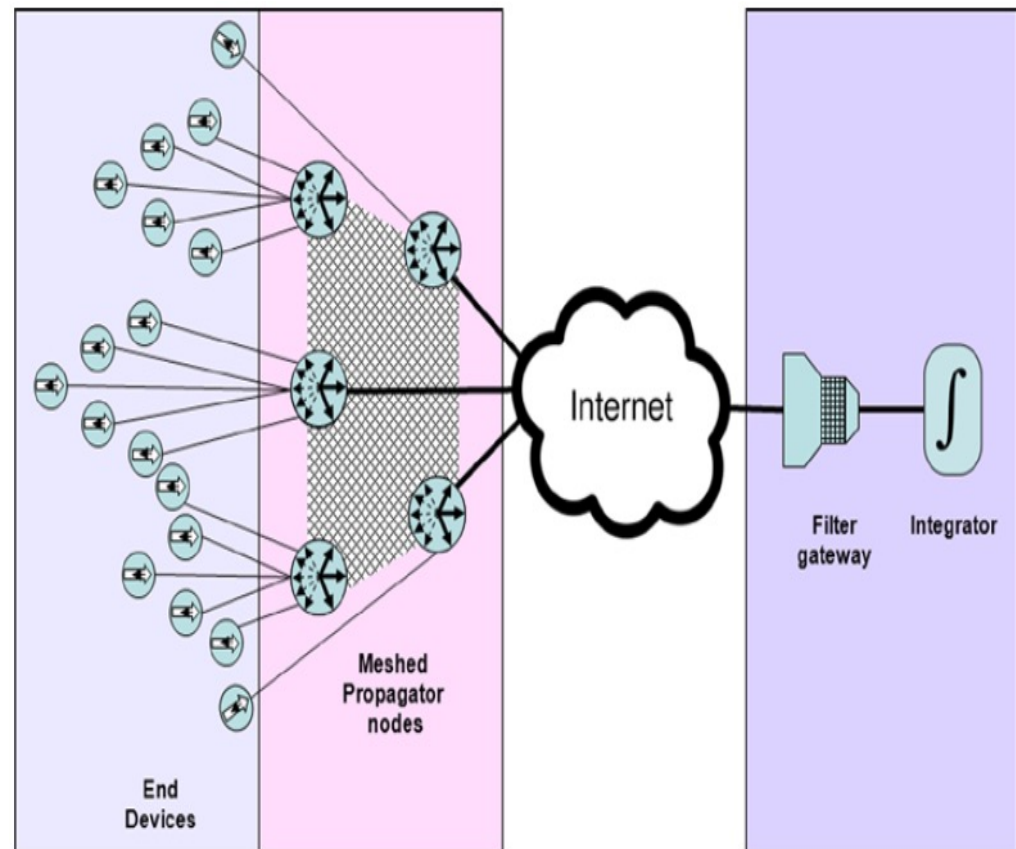
COLLECTION, INTEGRATION, ACTION

- **Integrator functions** are where the data streams are analyzed and acted upon
- Integrator functions:
 - *send their own transmissions* to get information or set values at devices
 - *may incorporate a variety of inputs*, from big data to social networking trends, etc
- Integrator functions are the *human* interface to the IoT
- They should be built to *reduce the large amounts of collected data* to a simple set of alarms, exceptions, and other reports for consumption by humans



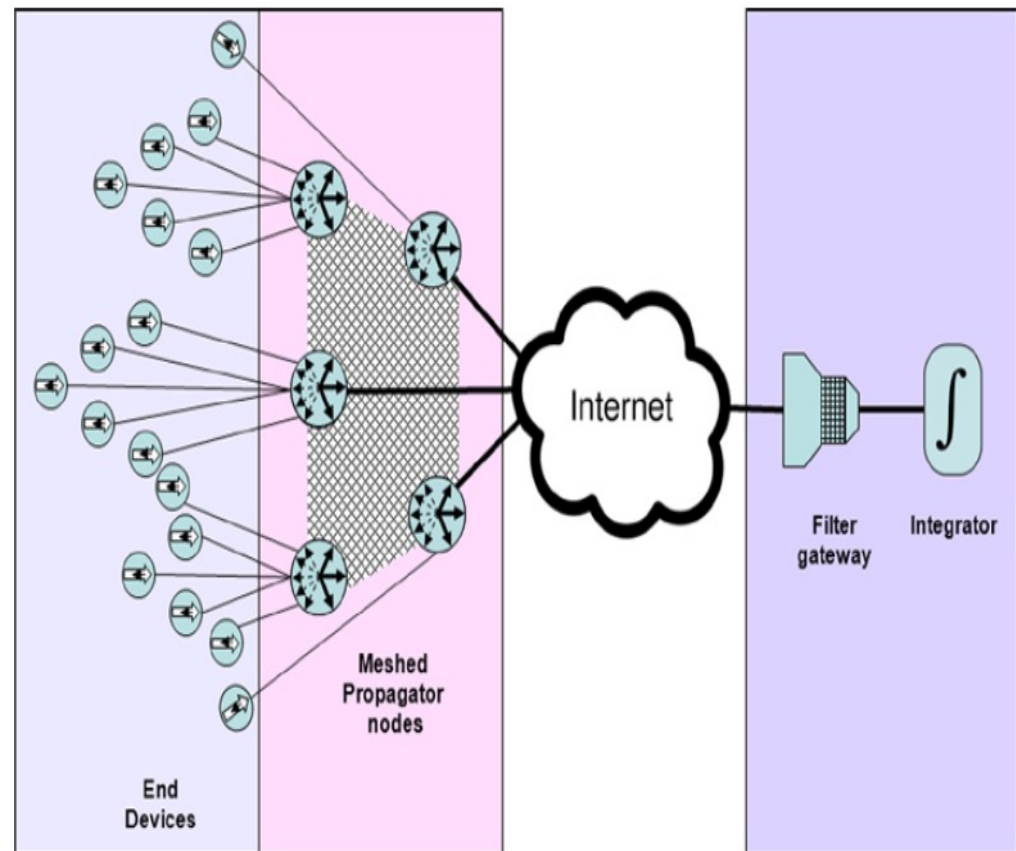
COLLECTION, INTEGRATION, ACTION

- Integrated scheduling and decision-making in integrator functions allow to operate *transparently* and *without human intervention*
- One integrator function might be needed for an average household operating on a smartphone, computer, or home entertainment device
- Or the integrator function could be scaled up to a huge global enterprise
- Example: Tracking and managing energy usage across a corporation



FILTERING

- An additional device is the *filter gateway*
- Filter gateways are two-armed routers, with a connection to the Internet and a connection to the integrator function
- The filter gateway ensures that *only meaningful data is forwarded to the integrator function*
- Filter gateways may use a simple set of rules (set by the attached integrator function) to filter the traffic
- Neighborhoods can be *geographic, functional, time-based*, or some combination of many other factors



THE ECOSYSTEM OF THE INTERNET OF THINGS



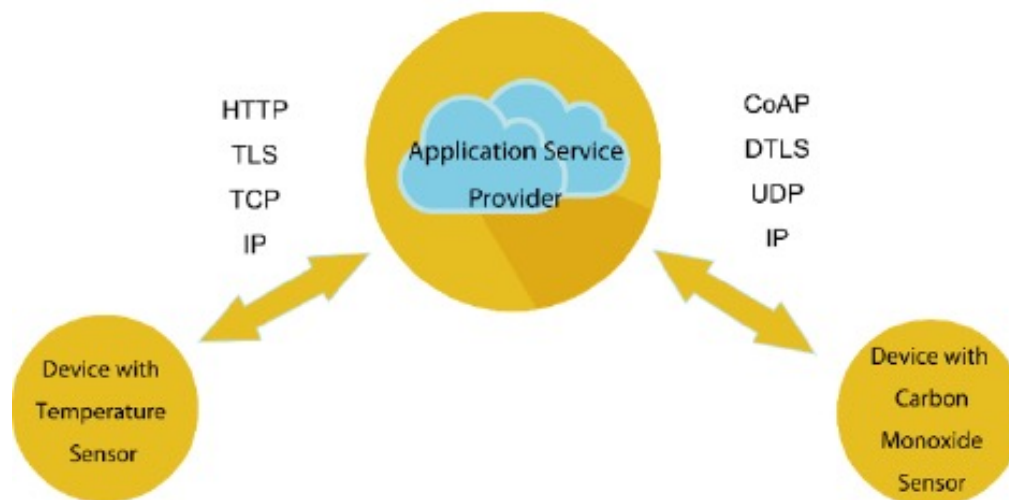
COMMUNICATION BETWEEN OBJECTS

- Communication between two products from different manufacturers
- It is possible thanks to the integration of a wireless technology e.g., Bluetooth or ZigBee
- Example: homes automation
- The devices communicate most often based on IP (Internet Protocols) and on the Internet



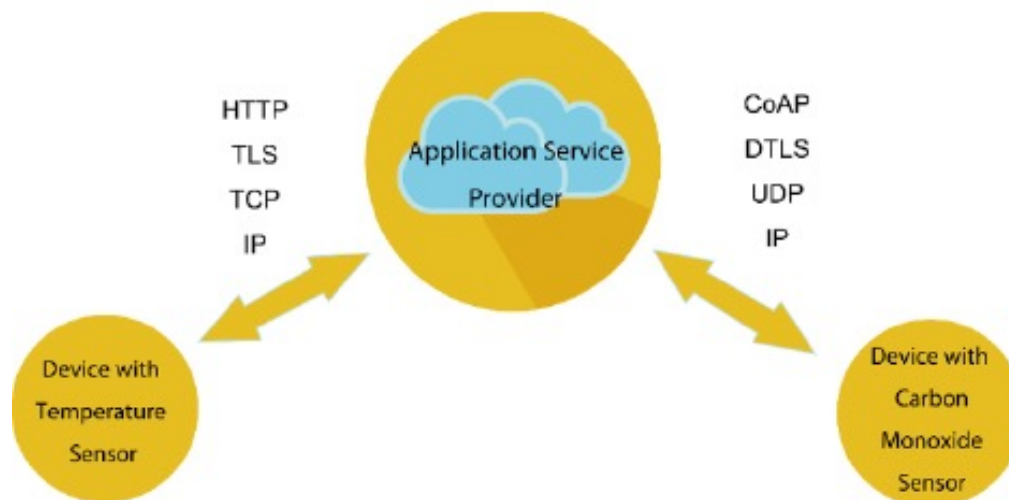
COMMUNICATION FROM OBJECTS TO CLOUD

- Data collected by sensors travels to service platforms via a network
- Real time collection to a specific platform located in the Cloud
- Generally, the platform belongs to the manufacturer of the sensor
- Interactions only involve a single service provider
- Consequently, there is no need to ensure interoperability



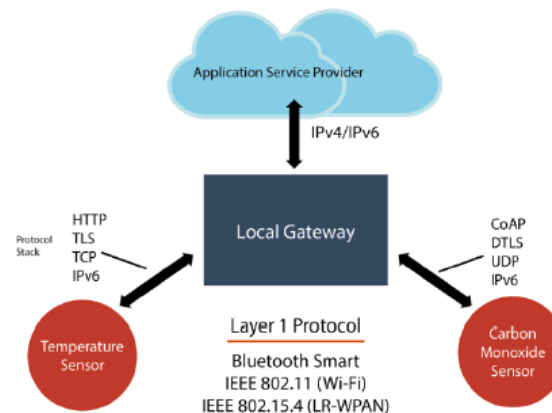
COMMUNICATION FROM OBJECTS TO CLOUD

- Transmissions are carried out on a network that supports IP to assure interoperability
- Manufacturers authorize the development of third-party applications that can exploit their products
- There are also various protocols and standards based on IP dedicated to communication between devices and service platforms
- Examples: *CoAP* (Constrained Application Protocol, RFC 7252); *UDP* (User Datagram Protocol); *REST* (Representational State Transfer) and *HTTP* (HyperText Transfer Protocol)



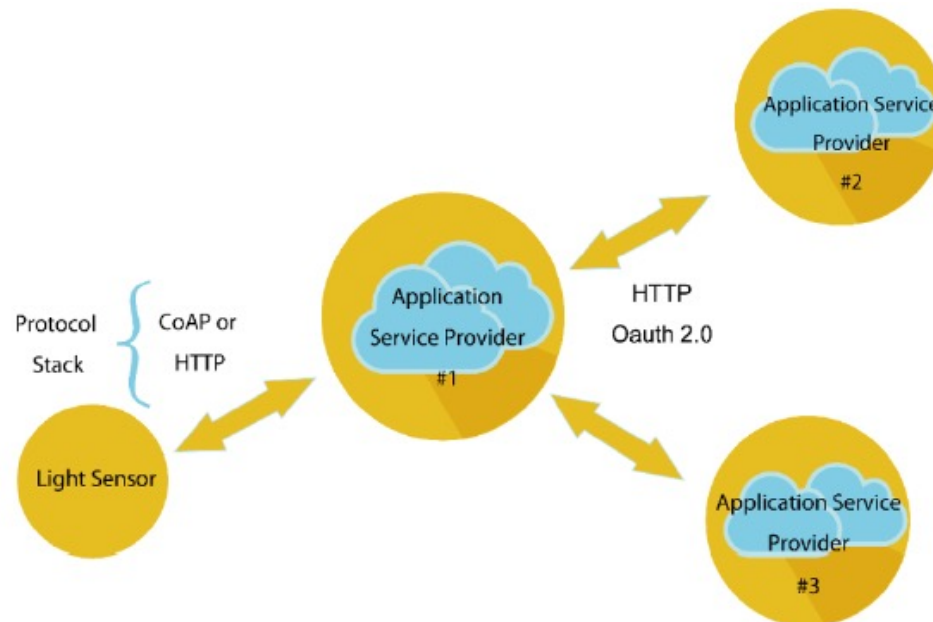
COMMUNICATION FROM OBJECTS TO A GATEWAY

- An intermediary is necessary to make the connection between the sensors and the Cloud
- Manufacturers use a “gateway” that retrieves information collected and then route it to the service platform
- Example: smartphones is a gateway between the smart wristband and the online application
- **Advantage:** it allows the addition of devices not compatible with protocols favoring interoperability such as IP, in a system adapted to this type of technology



COMMUNICATION FROM OBJECTS TO BACK END

- The model address the problem of data sharing between service providers
- The platforms set up **APIs**, most often a REST web service, which allows access to all, or a part, of the information collected by the service
- Example: Data collected by an activity tracker could be used by a service specializing in behavioral analysis



LAYERED ARCHITECTURE

Levels

- 7 Collaboration & Processes
(Involving People & Business Processes)
- 6 Application
(Reporting, Analytics, Control)
- 5 Data Abstraction
(Aggregation & Access)
- 4 Data Accumulation
(Storage)
- 3 Edge Computing
(Data Element Analysis & Transformation)
- 2 Connectivity
(Communication & Processing Units)
- 1 Physical Devices & Controllers
(The "Things" in IoT)



(from Cisco)



STEPS

- The problem that arises is the following: how is this going to work?

The setting up of the IoT goes through several other steps:

- **Identification.** Knowing how to precisely determine *which* object is connected to *what*, in *what way* and in *which location*
- **Capture.** Sensors multiplication, miniaturization and integration into the environment has proven to be necessary
- **Connection.** Linking objects with each other so that they can exchange data in a more autonomous manner



STEPS

Identify	Capture	Connect	Integrate	Network
Making possible the identification of each connected element.	Implementation of devices bringing the real and virtual worlds closer. The objects basic functions (the temperature sensor for the thermometer for example).	Establish a connection between the objects so they can communicate and exchange data.	Use a communication means connecting objects to the virtual world.	Linking objects and their data to the computing world via a network (the Internet, for example).
IPv4, IPv6, 6LoWPAN	MEMS, RF MEMS, NEMS	SigFox, LoRa	RFID, NFC, Bluetooth, Bluetooth LE, ZigBee, Wi-Fi, cellular networks	CoAP, MQTT, AllJoyn, REST HTTP

