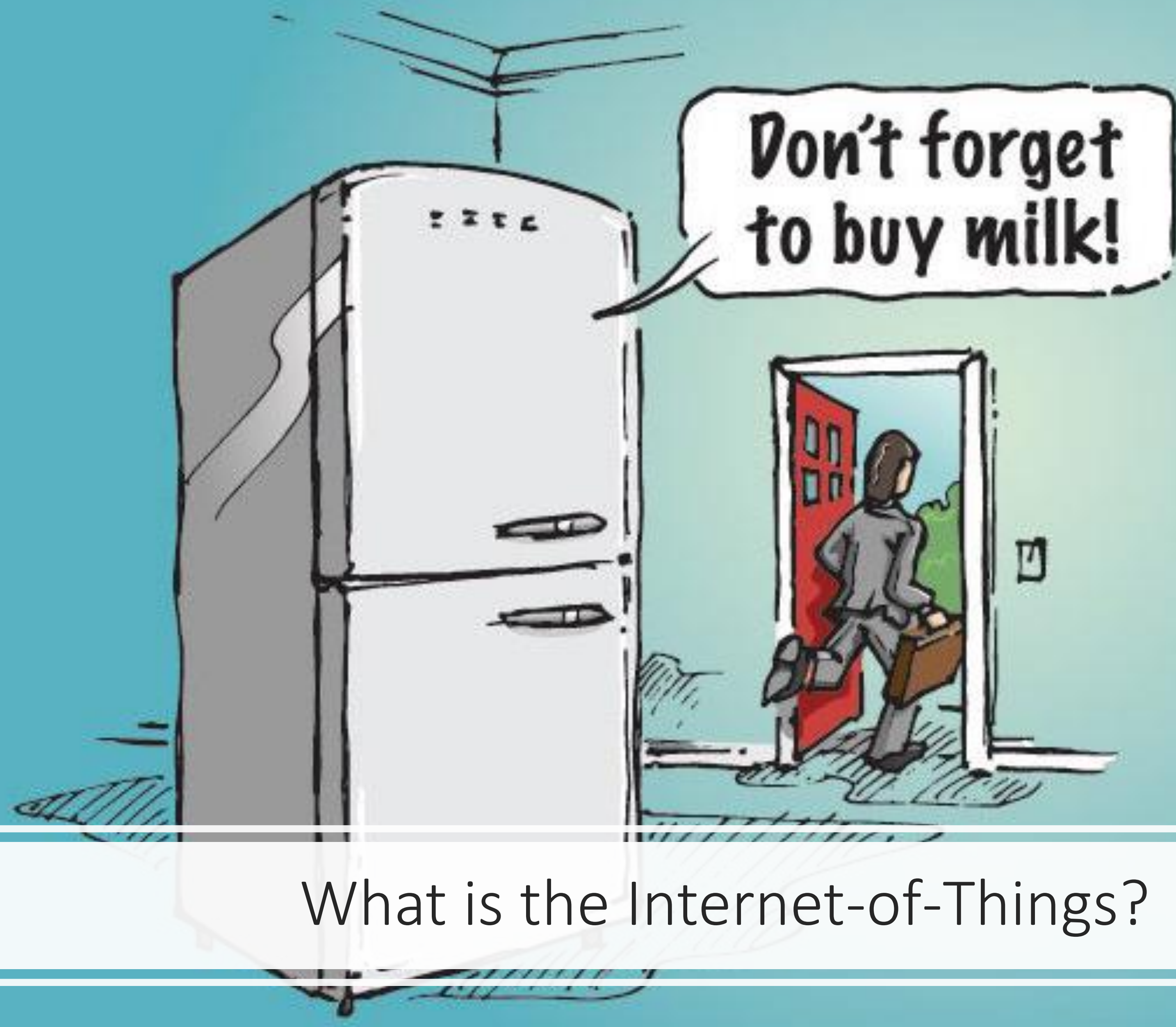
Internet-of-Things (IoT)

Introduction



What is the Internet-of-Things?

How Does My Fridge Do That?

- You are leaving the home (sense user)
- There’s no milk in fridge (sense object)
- Use this information to make a decision (process)
- Inform user of decision (communicate)

How Does My Fridge Do That?

You are leaving the home (sense user)

There's no milk in fridge (sense object)

- What type of sensor?
- Is milk needed?
- No milk or "little" milk? (prediction)

Use this information to make a decision (process)

Inform user of decision (notify)

You are leaving the home (sense user)

- What type of sensor?
- Distinguish between parent and child
- Identify reason for leaving home
- Identify other contexts (e.g., store hours)

There’s no milk in fridge (sense object)

Use this information to make a decision (process)

Inform user of decision (notify)

How Does My Fridge Do That?

You are leaving the home (sense user)

There’s no milk in fridge (sense object)

Use this information to make a decision (process)

- Where is processor?
- What are the rules?
- Fixed rules versus dynamic rules (learning)

Inform user of decision (notify)

How Does My Fridge Do That?

You are leaving the home (sense user)

There's no milk in fridge (sense object)

Use this information to make a decision (process)

Inform user of decision (notify)

- How?
- When?
- Privacy?
- Subtleness?
- Information overflow?

How Does My Fridge Do That?

Internet-of-Things (IoT)

Physical object (“thing”)

+

Controller (“brain”)

+

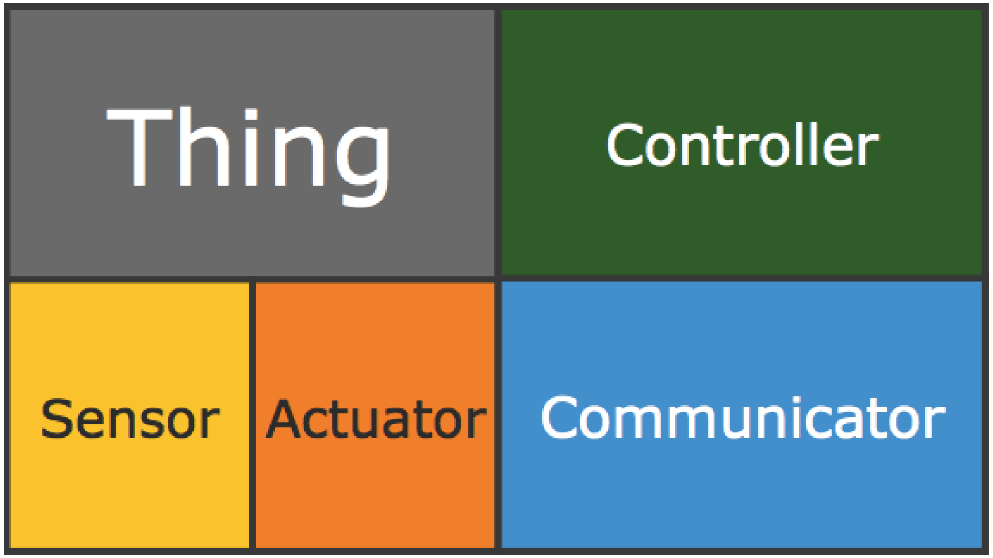
Sensors

+

Actuators

+

Networks (Internet)



Related Areas/Terminology

EMBEDDED SYSTEMS: NOT NECESSARILY CONNECTED

SENSOR NETWORKS: COLLECTION OF SENSOR DEVICES CONNECTED THROUGH WIRELESS CHANNELS

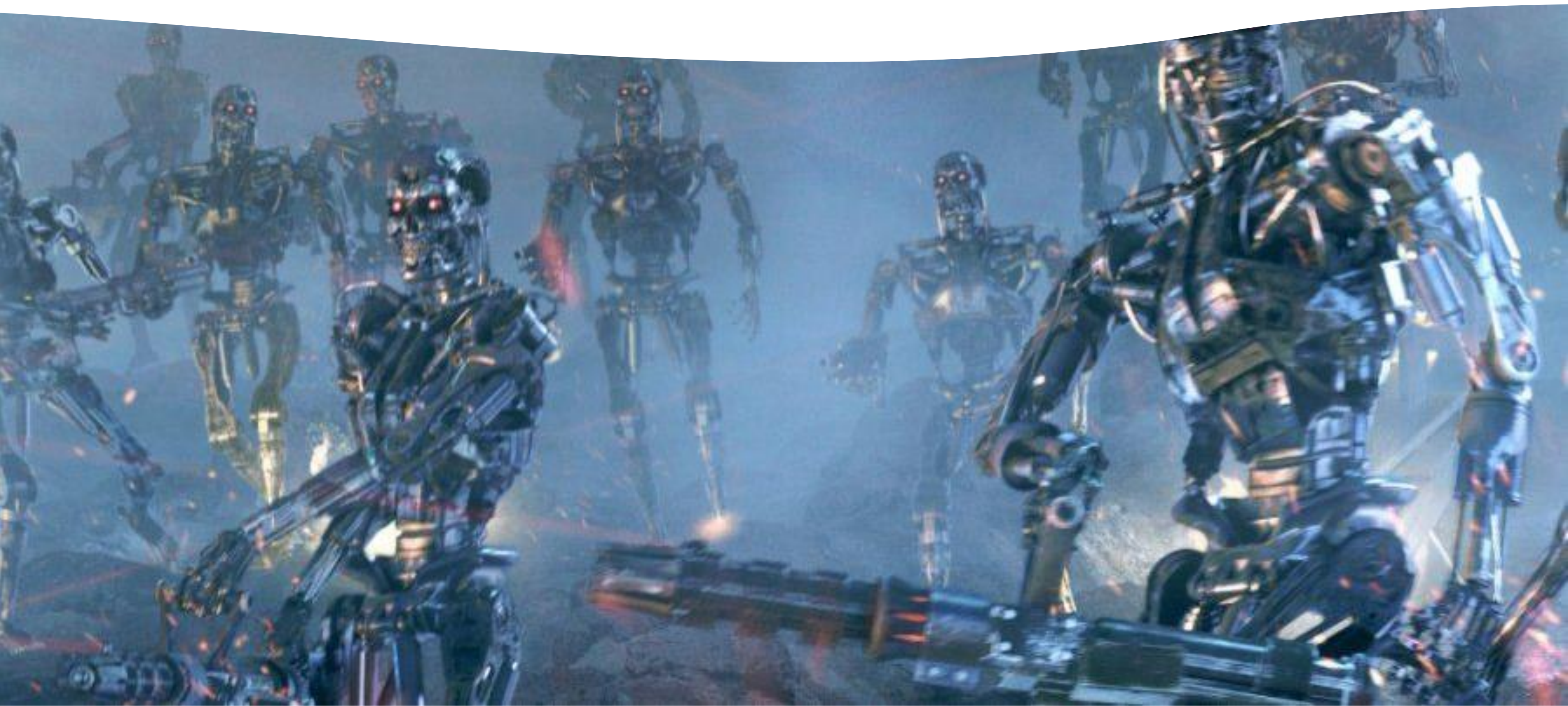
CYBER-PHYSICAL SYSTEMS: FOCUS ON INTERACTION BETWEEN PHYSICAL AND CYBER SYSTEMS

REAL-TIME SYSTEMS: FOCUS ON TIME CONSTRAINTS

PERVASIVE/UBIQUITOUS COMPUTING: FOCUS ON ANYTIME/ANYWHERE COMPUTING

Related Areas

- Machine-to-machine (M2M) communications
- Internet of Everything (Cisco Systems)
- “Skynet” (Terminator movie)



“Internet-of-Things”

Term coined by British entrepreneur Kevin Ashton, while working at MIT Auto-ID Labs

Referred to (and envisioning) a future global network of objects connected specifically by RFID (radio-frequency identification)

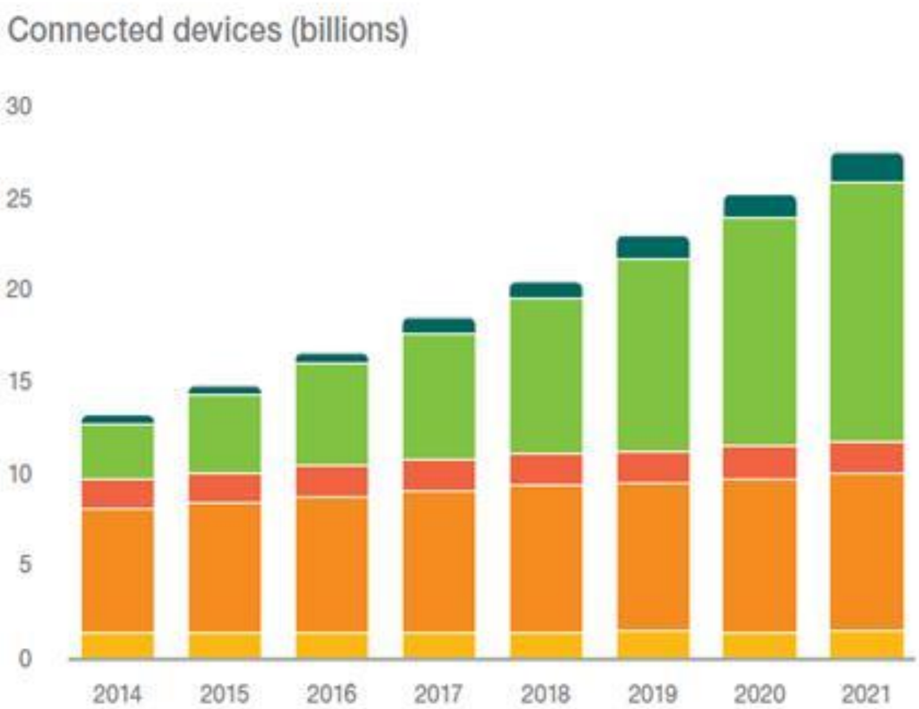
Complete automation of data collection

First article about IoT in 2004 from MIT; called it ‘Internet 0’.

*https://en.wikipedia.org/wiki/Internet_0

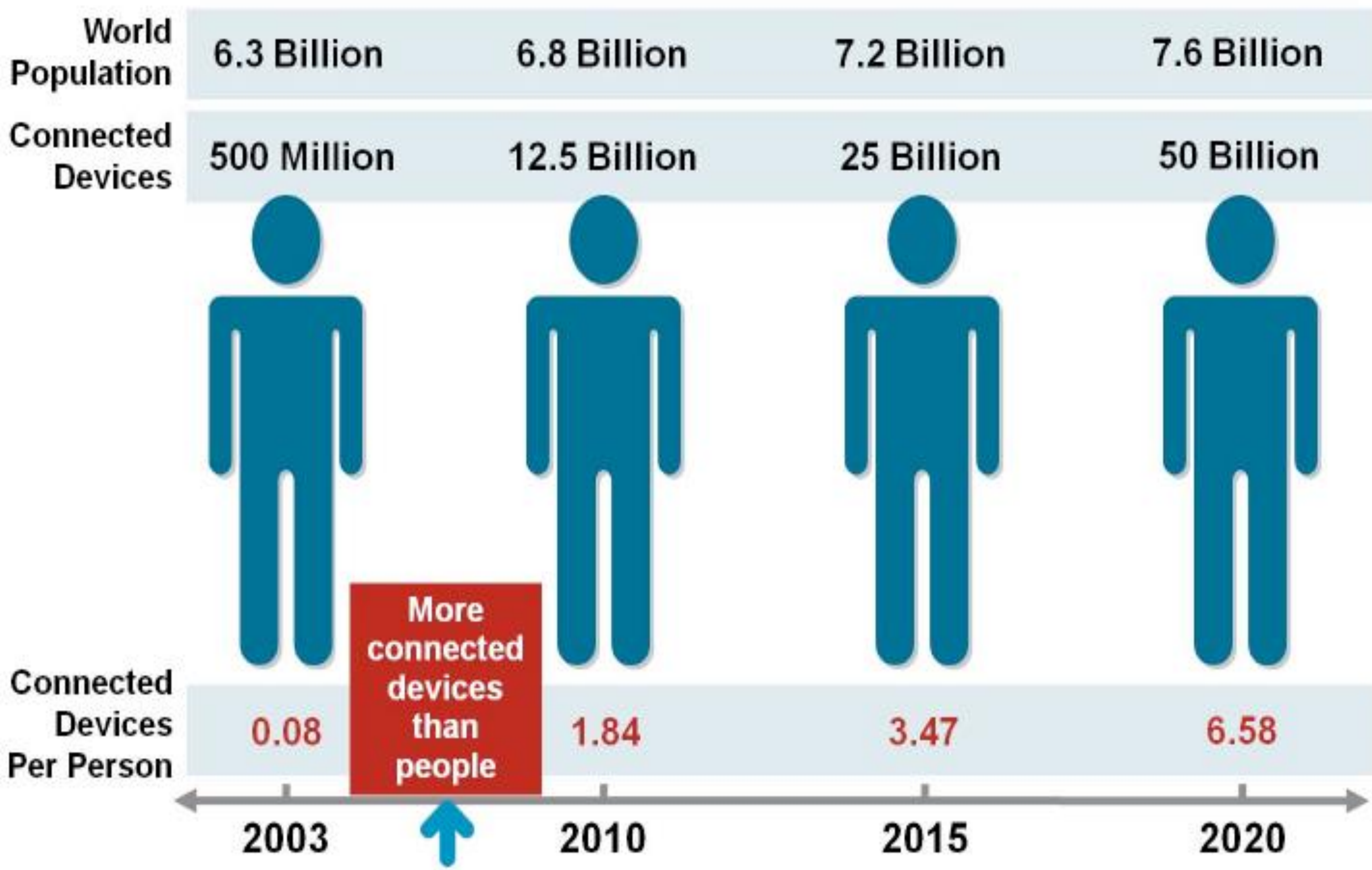
Internet-of-Things Vision & Growth

THE INTERNET OF THINGS



	15 billion (2015)	28 billion (2021)	CAGR 2015–2021
Cellular IoT	0.4	1.5	27%
Non-cellular IoT	4.2	14.2	22%
PC/laptop/tablet	1.7	1.8	1%
Mobile phones	7.1	8.6	3%
Fixed phones	1.3	1.4	0%

Internet-of-Things Vision & Growth



Source: Cisco IBSG, April 2011

What is IoT

- Internet of things (IoT) is the network of physical devices, vehicles, home appliances, and other items embedded with electronics, software, sensors, actuators, and connectivity which enables these things to connect, collect and exchange data[1].

- IoT refer to the connection of devices to the Internet.

[1]https://en.wikipedia.org/wiki/Internet_of_things

Internet-of-Things (IoT)

Introduction

What is IoT

- Internet of things (IoT) is the network of physical devices, vehicles, home appliances, and other items embedded with electronics, software, sensors, actuators, and connectivity which enables these things to connect, collect and exchange data[1].

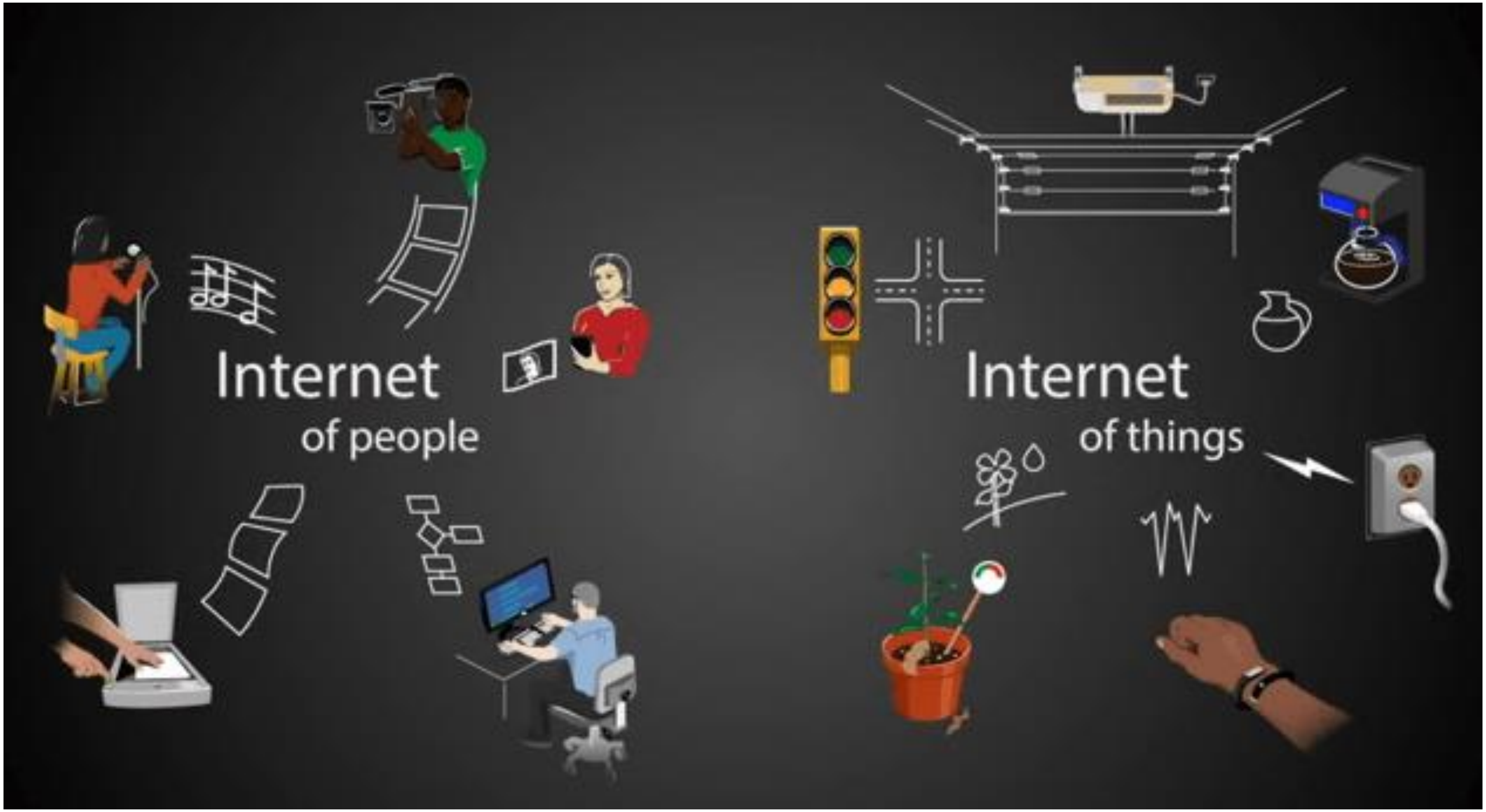
- IoT refer to the connection of devices to the Internet.

[1]https://en.wikipedia.org/wiki/Internet_of_things

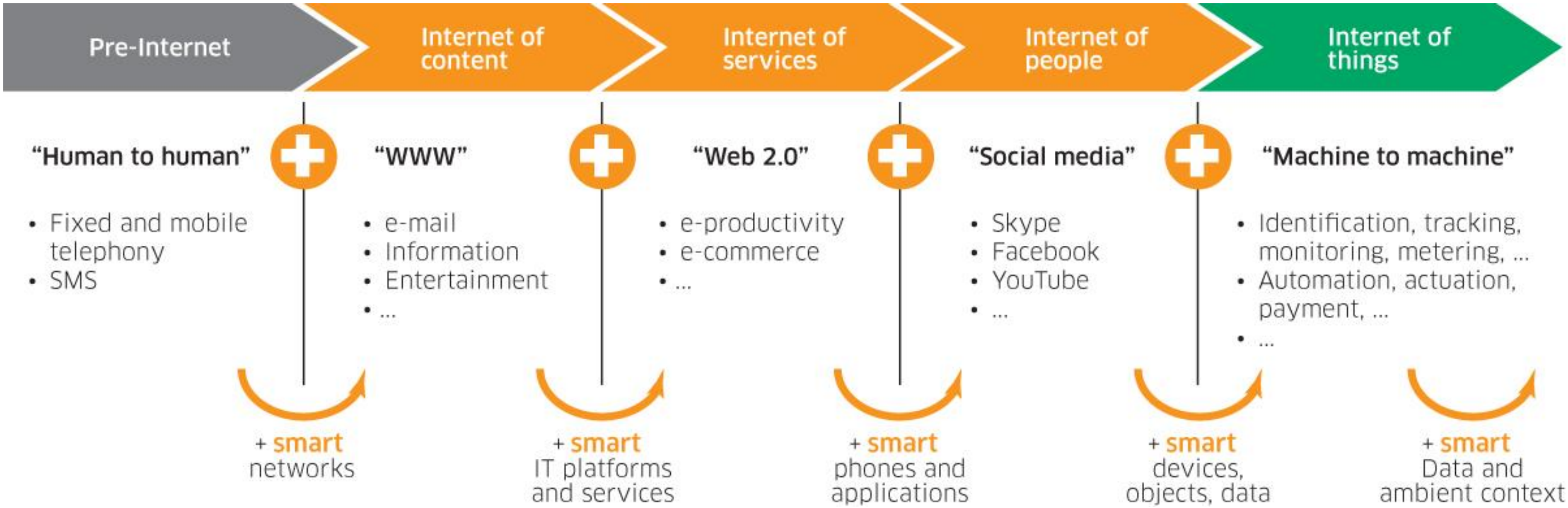
Internet of People (IOP)

- People are connected with the Internet.
- Internet is everywhere in the World.
- It is the primary connection between people.

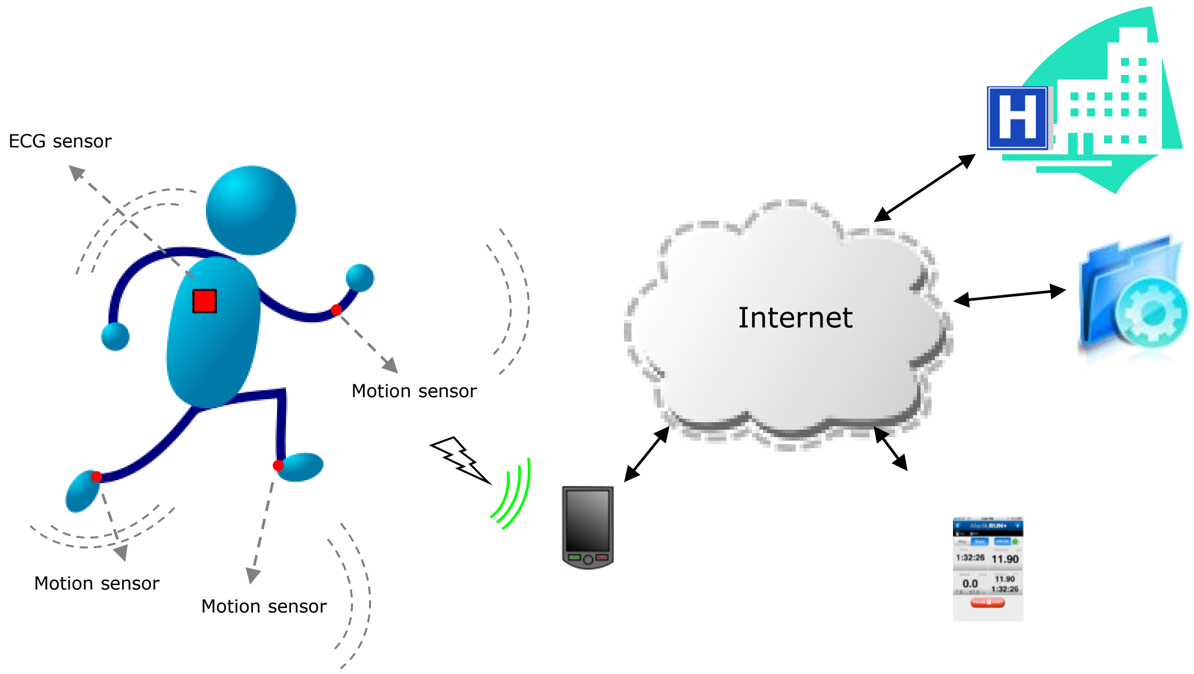
IOP IOT



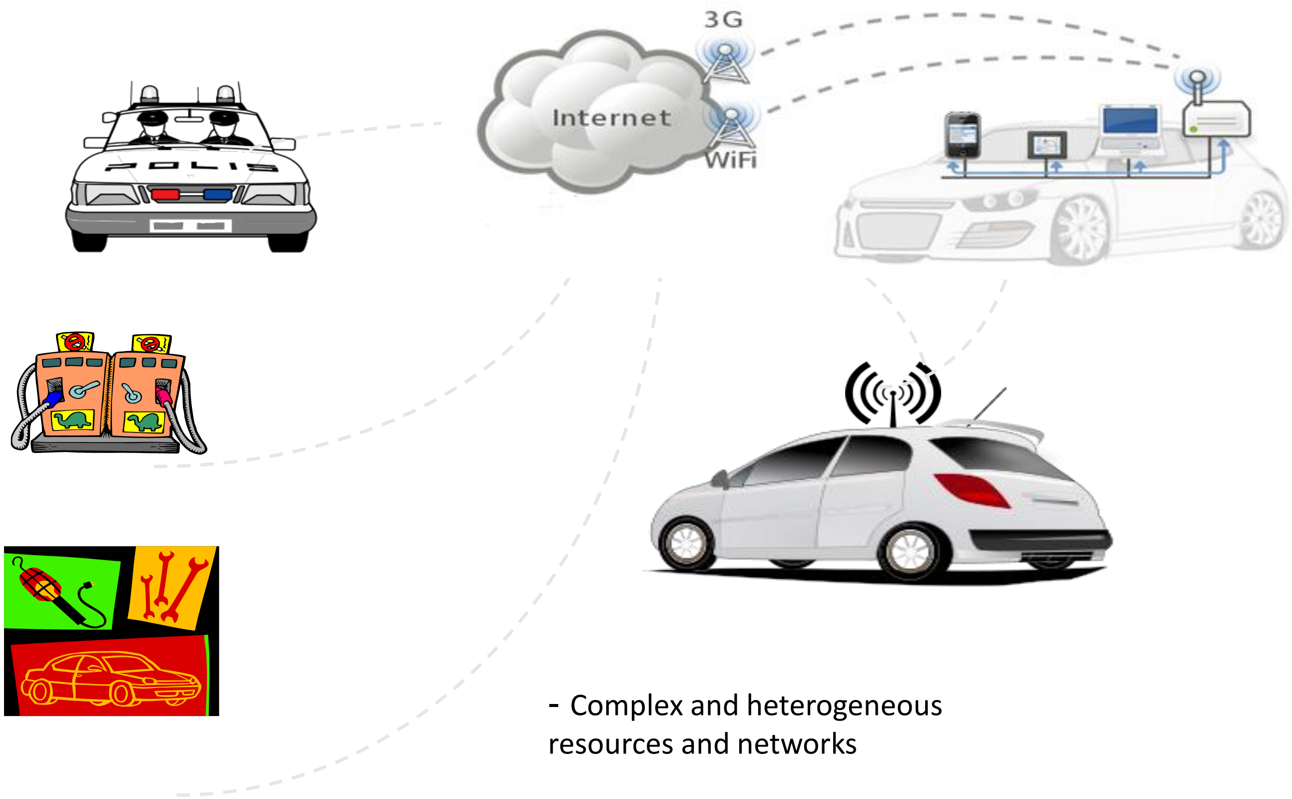
Internet-of-Things Evolution



People Connecting with Things



Things Connecting with Things



- Complex and heterogeneous resources and networks

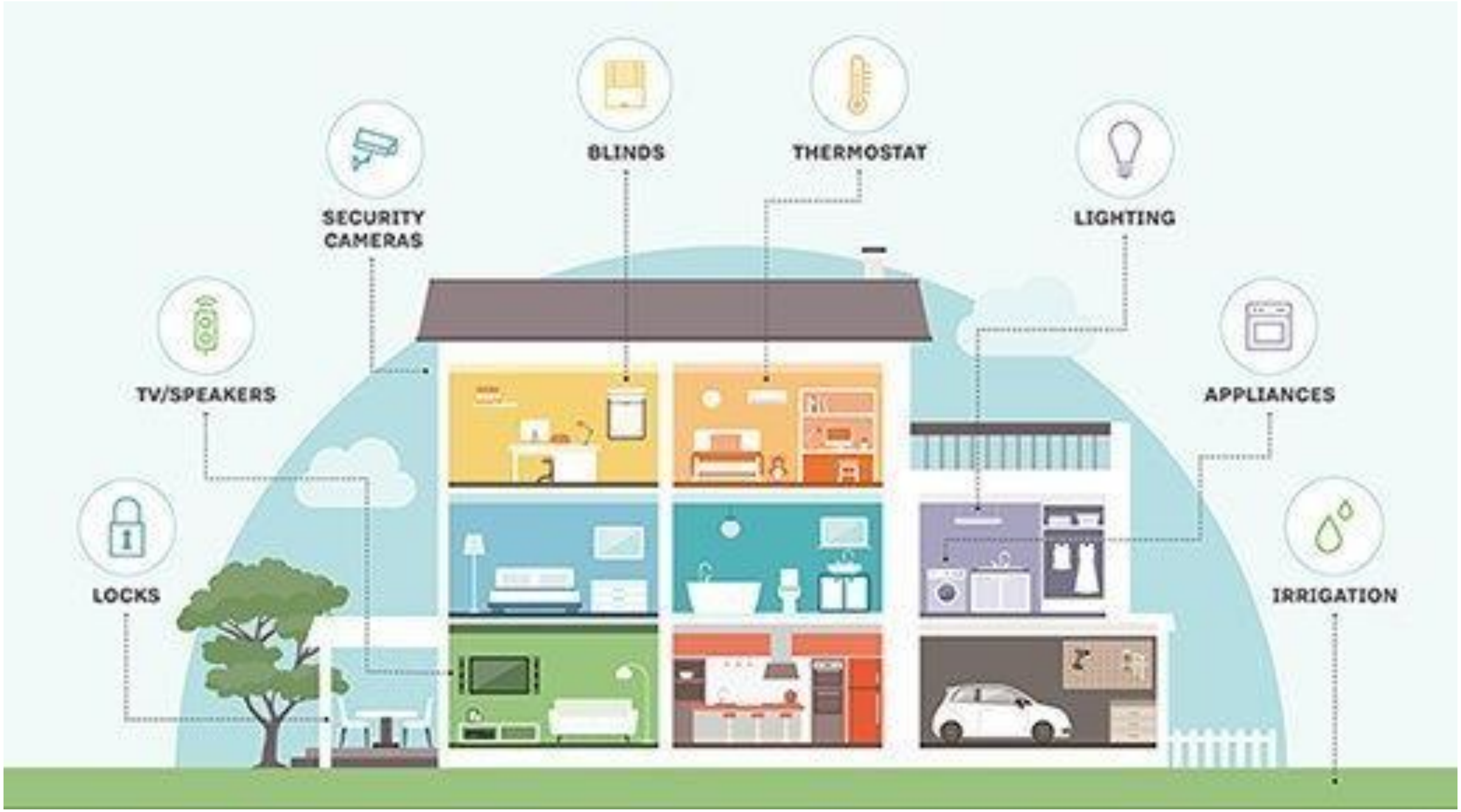
Where is IOT?

IOT is everywhere!


IoT

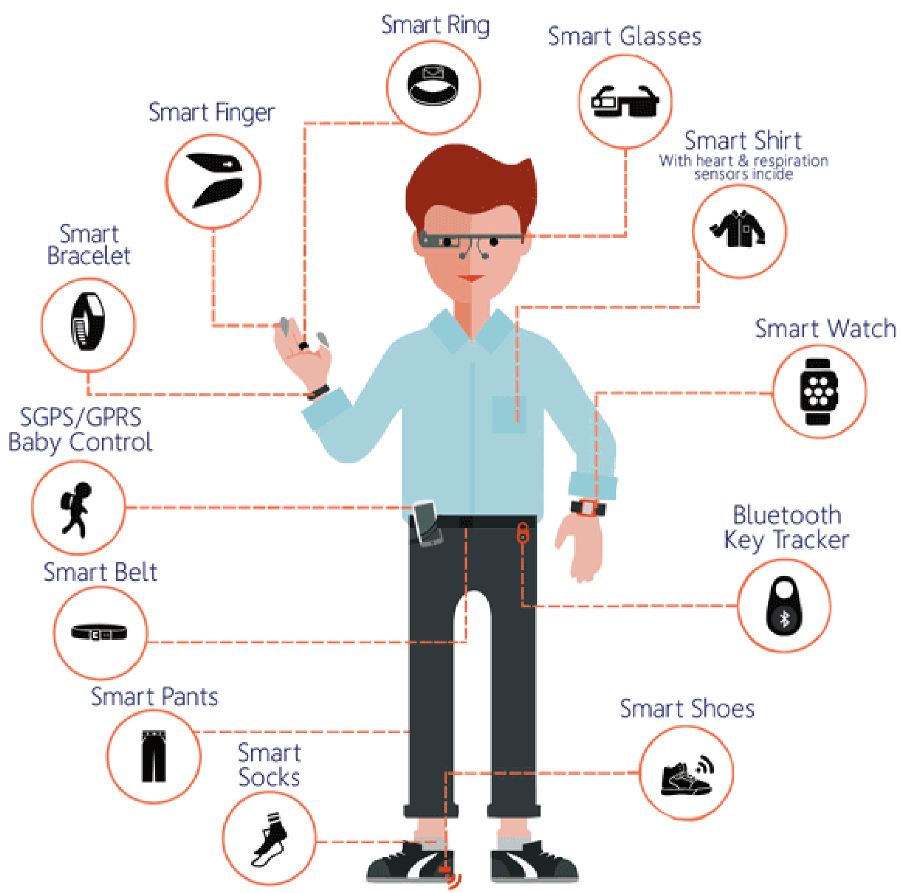
Applications

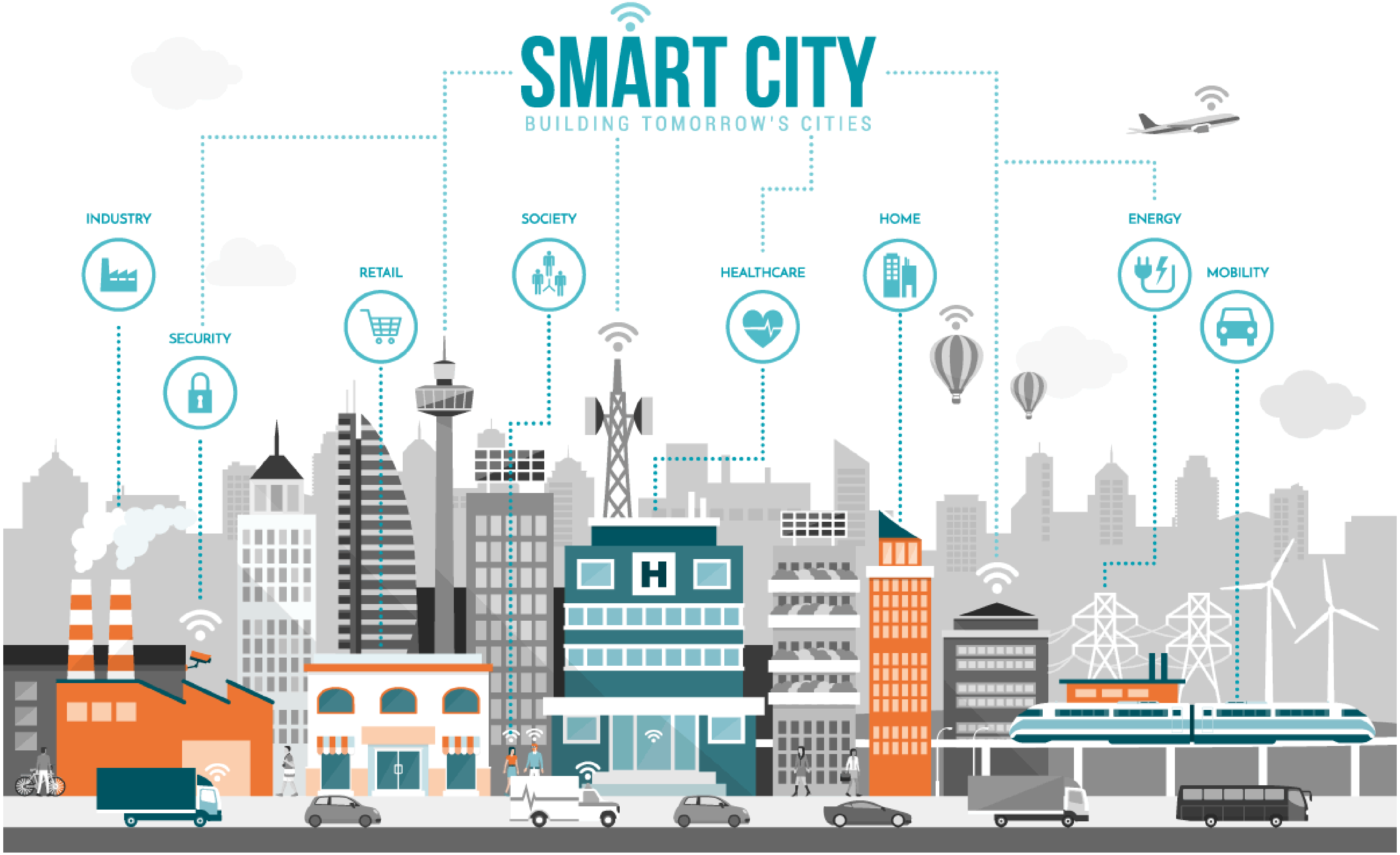
Smart Homes

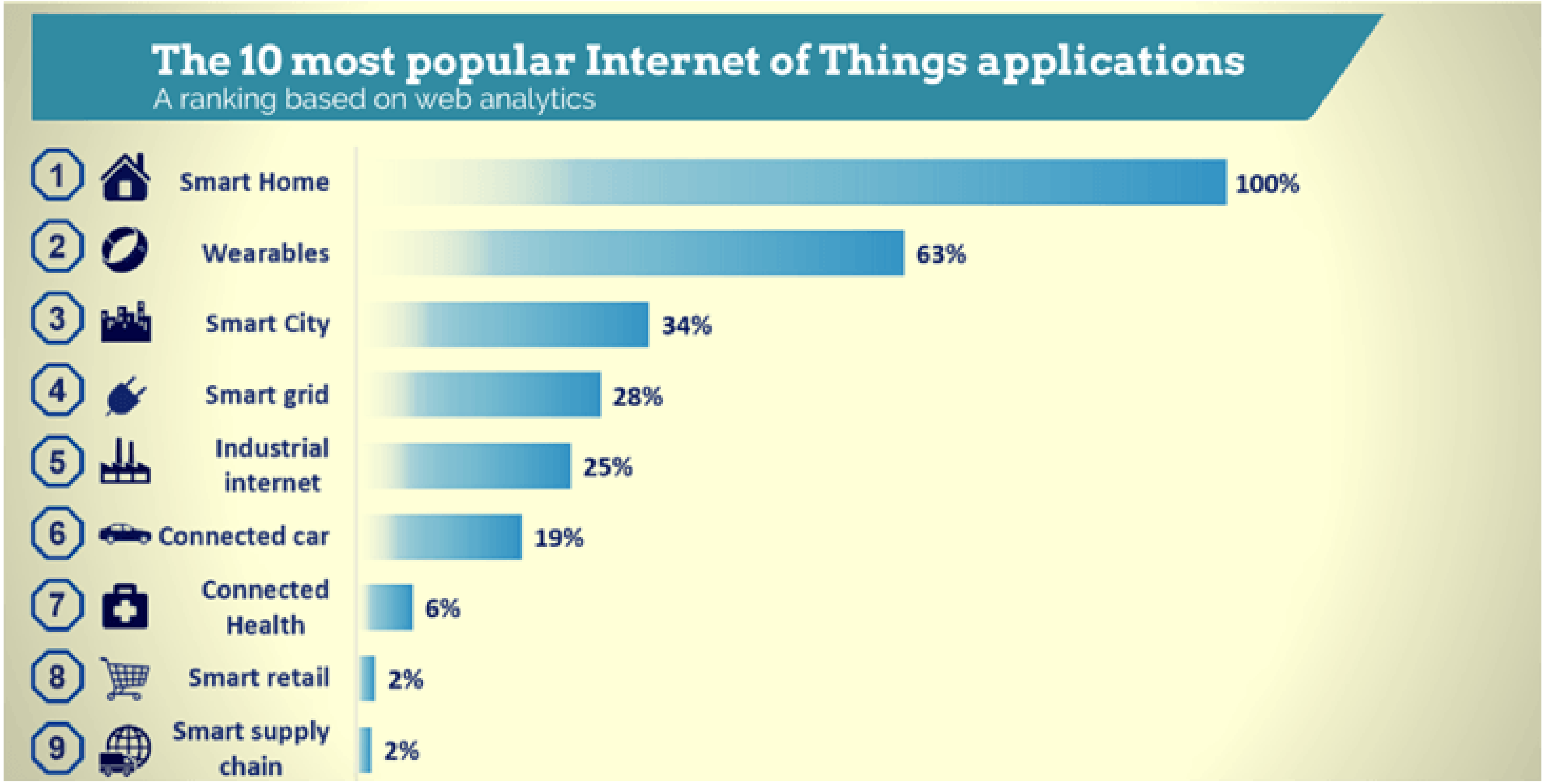


BLINDS
THERMOSTAT
SECURITY CAMERAS
LIGHTING
TV/SPEAKERS
APPLIANCES
LOCKS
IRRIGATION

Wearable


Smart Ring
Smart Glasses
Smart Finger
Smart Shirt
With heart & respiration sensors incide
Smart Bracelet
Smart Watch
SGPS/GPRS Baby Control
Bluetooth Key Tracker
Smart Belt
Smart Pants
Smart Shoes
Smart Socks


SMART CITY
BUILDING TOMORROW'S CITIES
INDUSTRY
SECURITY
RETAIL
SOCIETY
HEALTHCARE
HOME
ENERGY
MOBILITY
H



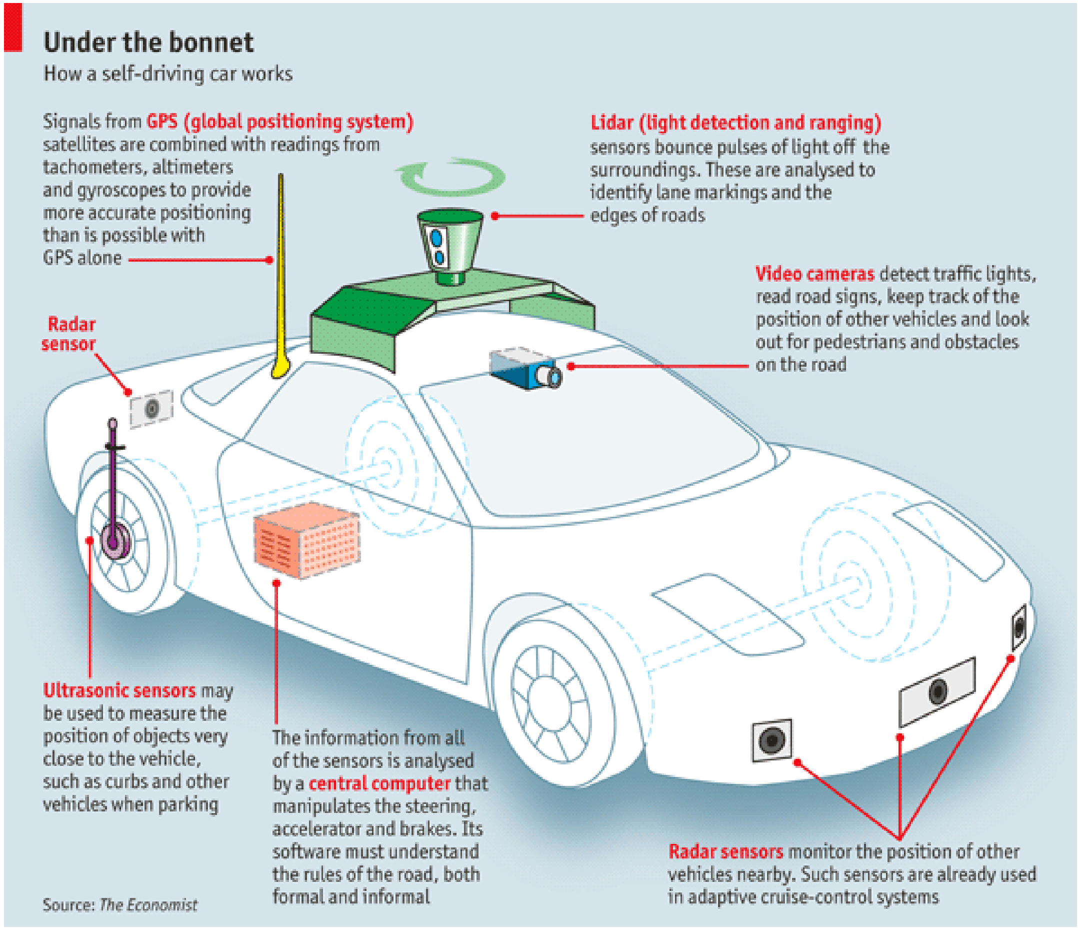
The 10 most popular Internet of Things applications
A ranking based on web analytics
1 Smart Home 100%
2 Wearables 63%
3 Smart City 34%
4 Smart grid 28%
5 Industrial internet 25%
6 Connected car 19%
7 Connected Health 6%
8 Smart retail 2%
9 Smart supply chain 2%

Augment Existing Things

Augmenting Life With New Things

- Smart City
- Smart Car
- Smart Me (healthcare, fitness, wellness)

Example: Connected Roadways



Example: Connected Roadways

[State of Self-Driving Car]


The Connected Factory in Action
INNOVATION
TAP COMMERCIAL INNOVATION
Mobilize employees and supervisors to move across the factory floor and access data wherever they are. The iPad and other like devices are making their way into industrial settings — along with an expectation that much of the commercial innovation it brings will also apply to industrial activities.
CONNECT ENGINEERS WITH MACHINES (M2M)
Apply predictive maintenance. Gain early warnings when production, machinery or network performance is about to degrade.
STATUS
"I see the fix!"
INDUSTRIAL ETHERNET
EFFICIENCY
LINK INFORMATION & OPERATIONAL TECHNOLOGY
Bridge the gap from data center to control room to collaborate and share best practices and common goals between manufacturing and IT.
OPTIMIZE ASSETS
Identify where your people, equipment, works in process and finished goods are in real-time. Adjust the schedule and inventory on the fly.
Systems
Processes
People
AGILITY
CONNECT & COLLABORATE EXTERNALLY
Extend visibility beyond your four walls. Link the extended supply chain and distribution to create dynamic workflows. Help and expertise are available in an instant.
EXPANDABLE INFRASTRUCTURE
Design and build an Industrial Ethernet infrastructure to minimize cost and effort to expand or improve processes. One infrastructure for safety, control, SCADA, Physical Security, and LAN.
SECURE
RISK
SECURE PHYSICAL & CYBER ASSETS
Traditional security devices, like keypad entry systems, call boxes and security cameras, need power from Industrial Ethernet cables, with secure networks, to protect your processes, people, and plans from cyber sabotage.
MAXIMIZE UPTIME
Design ruggedized industrial networking infrastructure that will endure in harsh environments with redundant communications, power and configuration backup — especially for business processes under extreme conditions.
BELDEN

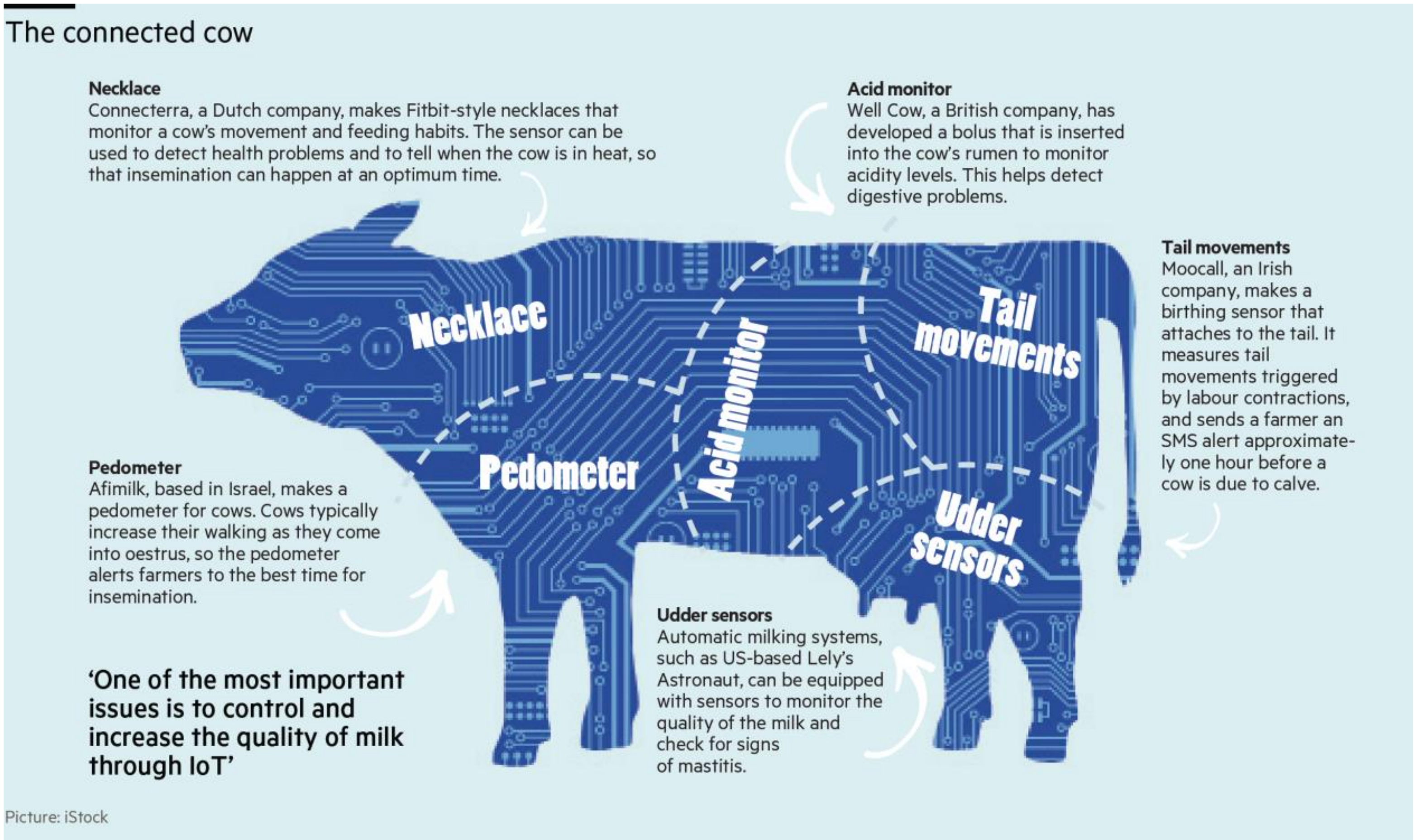
Example: Connected Factory

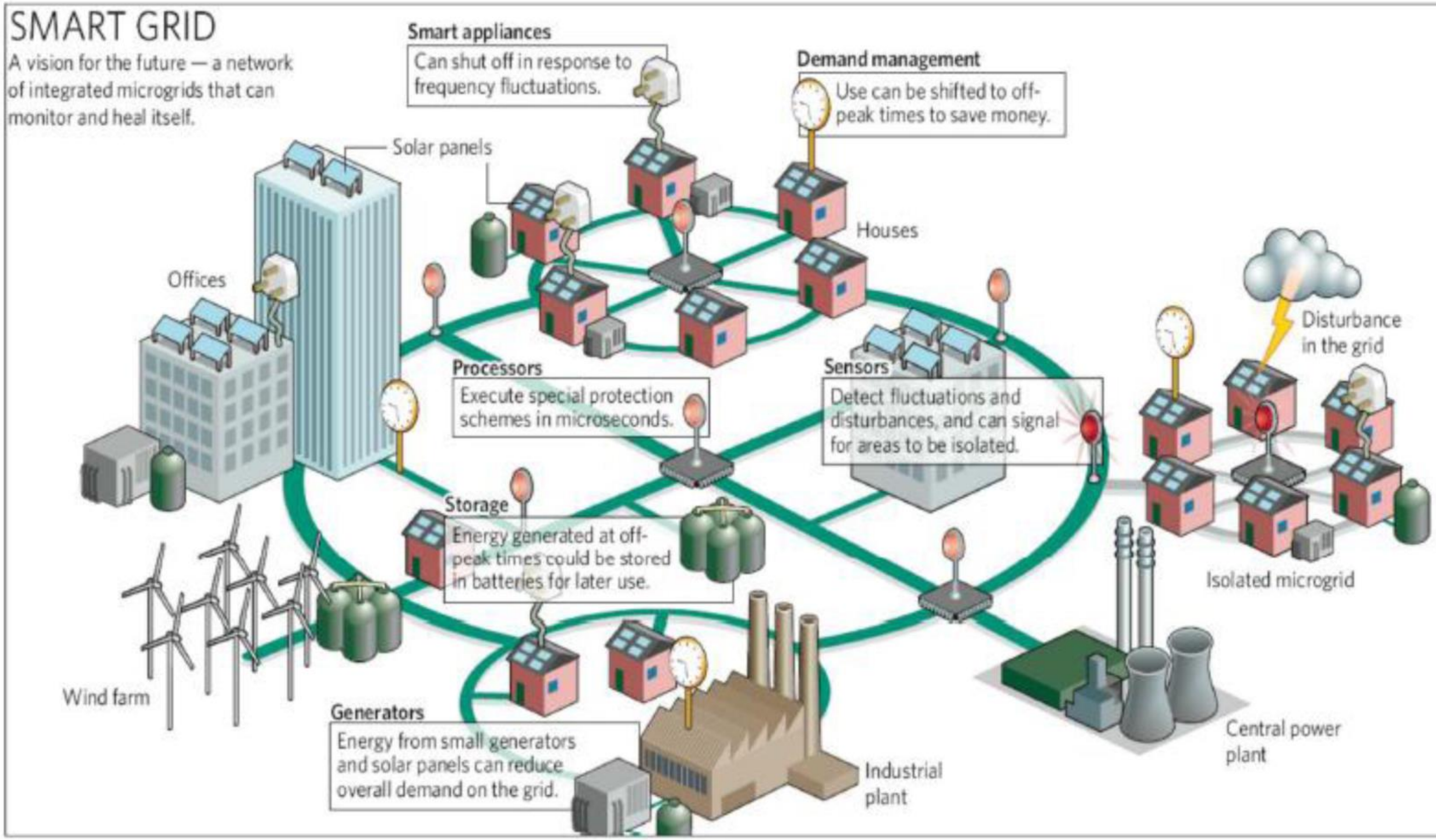
- New product and service introductions faster
- Increasing production, quality, uptime
- Mitigating unplanned downtime
- Protecting from cyber threats
- Worker productivity and safety

Example: Smart & Connected Buildings

- Energy management
- Lighting
- Safety
- HVAC
- Building automation
- Smart spaces

Example: Smart Creatures





Example: Smart Grid

Enablers: Portability

Reducing the size of hardware to enable the creation of computers that could be physically moved around relatively easily

Enablers: Miniaturization

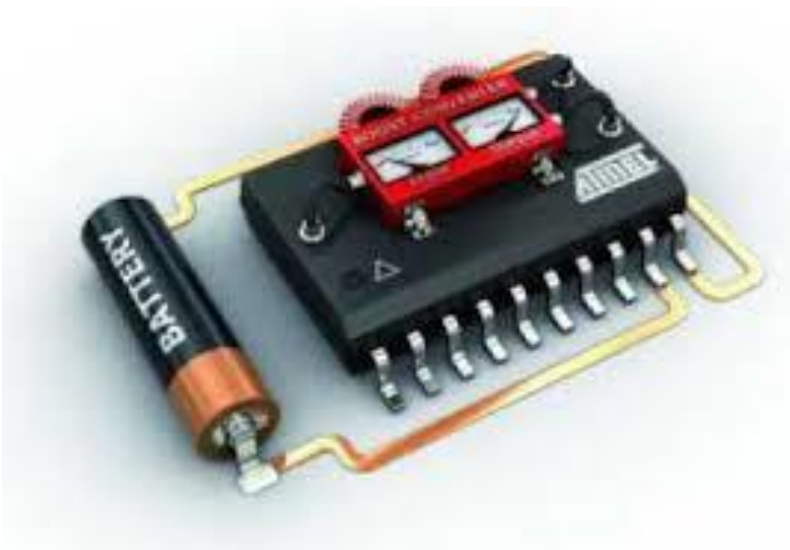
Creating new and significantly smaller mobile form factors that allowed the use of personal mobile devices while on the move

50mm x 50mm

35mm x 35mm

15mm x 15mm

Enablers: Low Power and Low Heat



- Low power architectures
- Low power radios
- Sleep modes
- Energy harvesting

Enablers: Connectivity

- Developing devices and applications that allowed users to be online and communicate via wireless data networks while on the move

Enablers: Convergence

Integrating emerging types of digital mobile devices, such as Personal Digital Assistants (PDAs), mobile phones, music players, cameras, games, etc., into hybrid devices.

Enablers: Divergence

Opposite approach to interaction design by promoting information appliances with specialized functionality rather than generalized ones

Enablers: Ecosystems

The emerging wave of *digital ecosystems* is about the larger wholes of pervasive and interrelated technologies that interactive mobile systems are increasingly becoming a part of.

Example: Smartphone

- Portability: carry it anywhere you want
- Miniaturization: make it possible to build device to fit in your pocket
- Connectivity: Wi-Fi, LTE/4G, cellular, Bluetooth
- Convergence: phone, camera, gaming device, movie streaming, music player, …
- Digital Ecosystem: cloud, social networks, software development kits, app stores, big data, standardization …

IoT Issues & Challenges

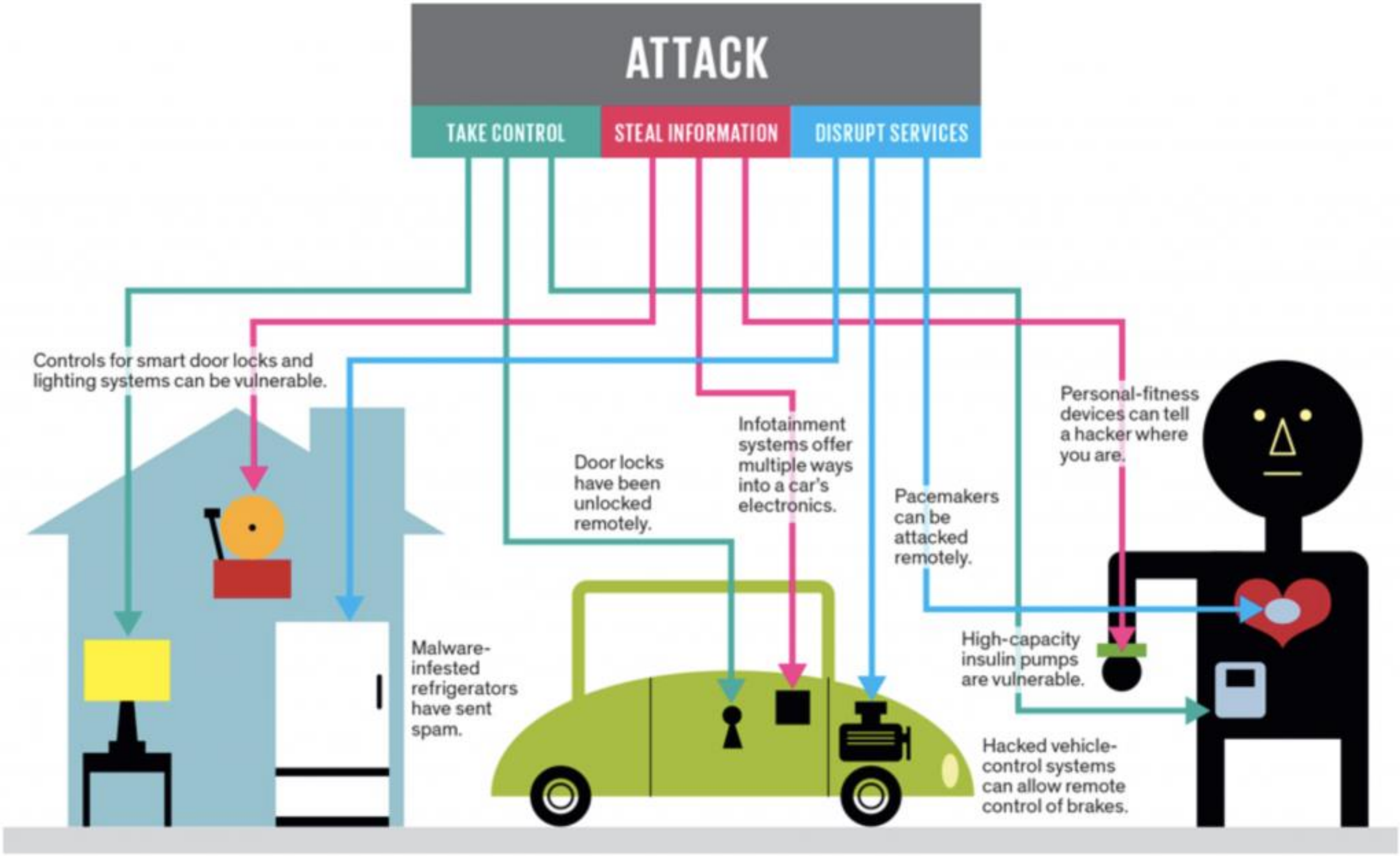


Illustration: J. D. King

BREAK

Last lecture
Application of IoTs

Components- Embedded Systems

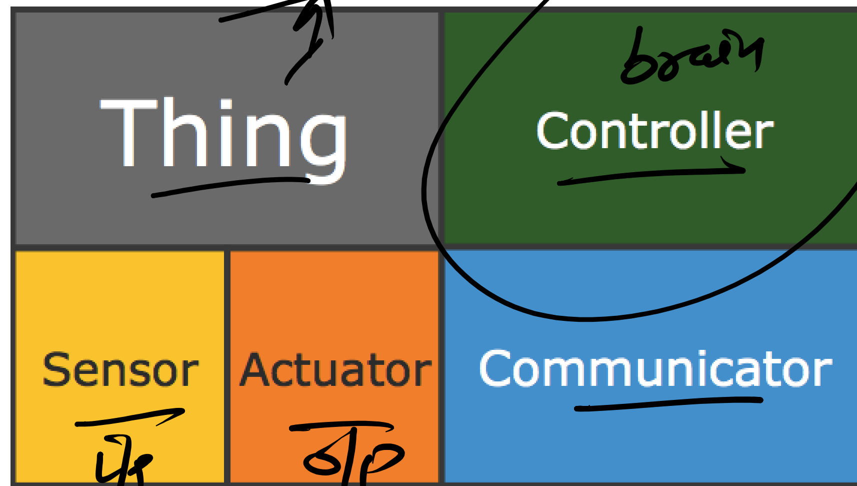
Internet-of-Things

Lecture 5

*University of Notre Dame and ARM

Bulb, chair, phone, A/c, washing m/c

device

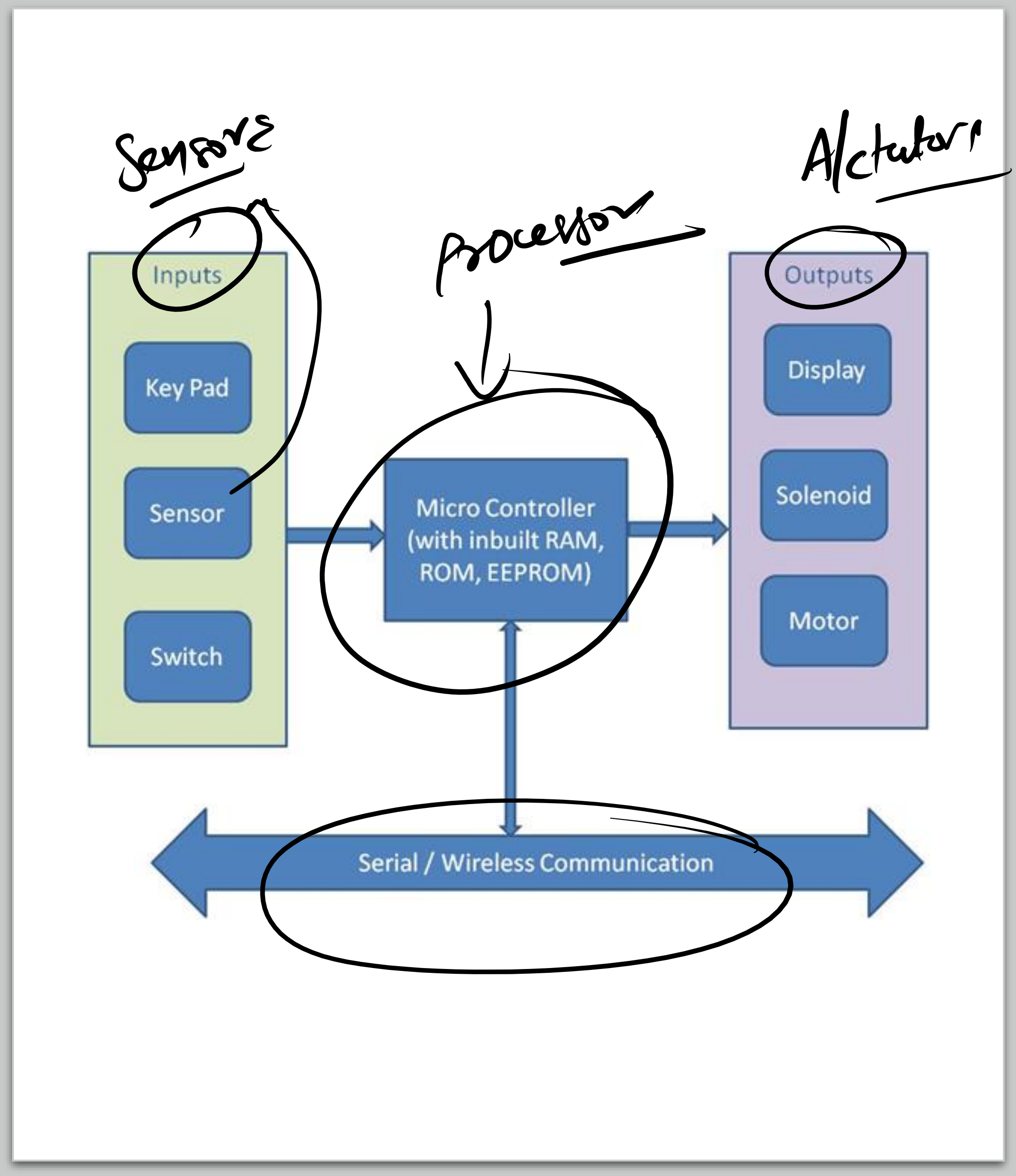


Components of an IoT Device

Embedded System/Computer

- "Any sort of device which includes a programmable computer but itself is not intended to be a general-purpose computer" -Wayne Wolf
  - General purpose
  - Dedicated

Specz to some applicatn



Embedded systems

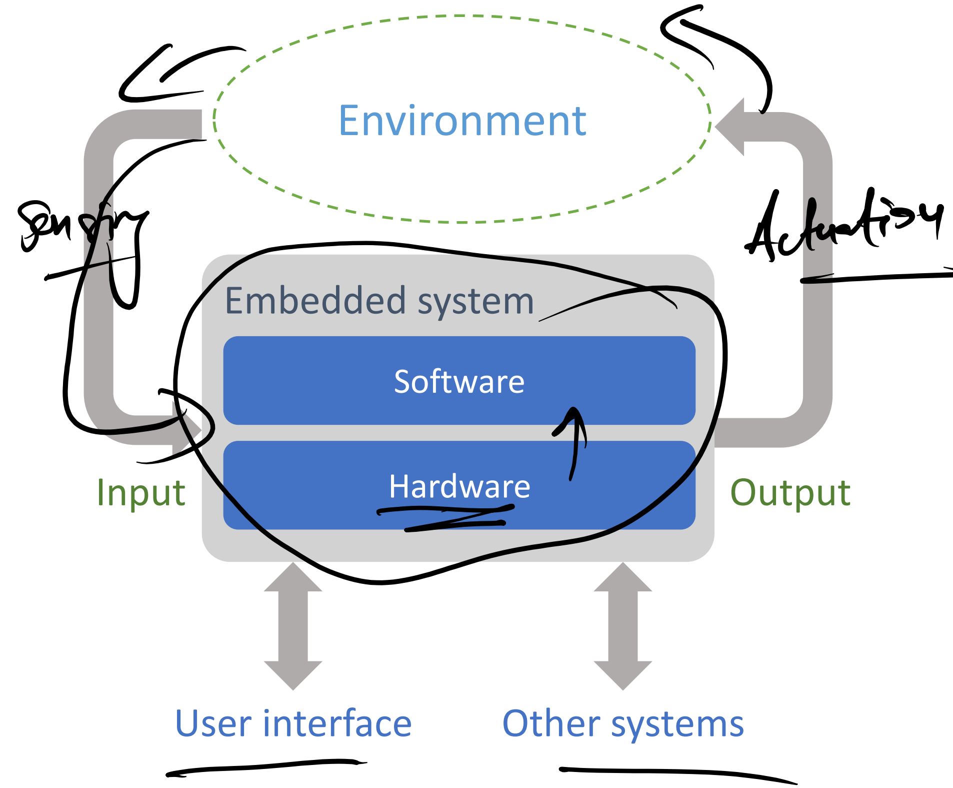
Internet of Things (IoT)

- Application-specific computer system
- Built into a larger system
- Often with real-time computing constraints

Adding embedded systems to larger systems

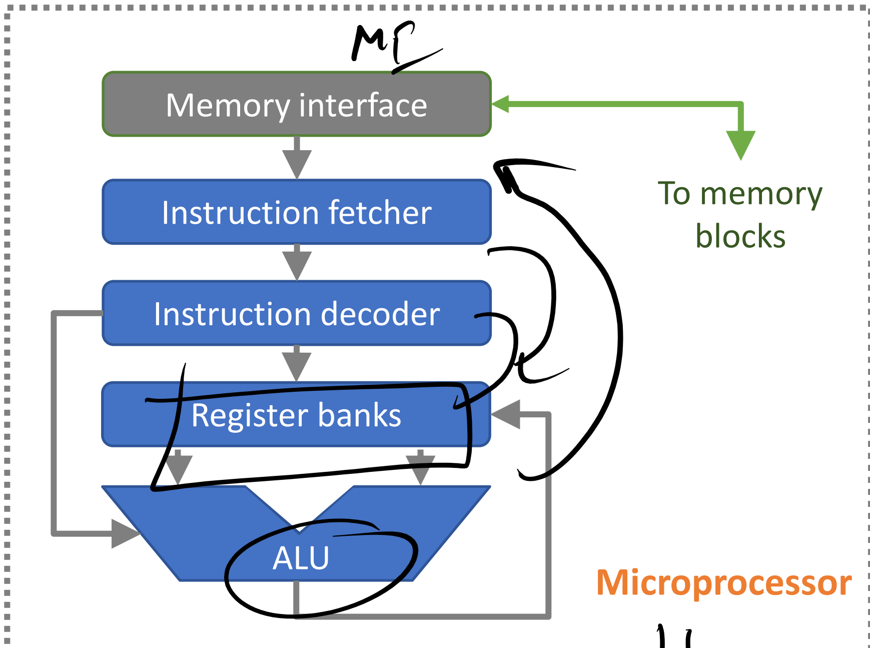
- Better performance
- More functions and features
- Lower cost, for e.g., through automation
- More dependability

Examples: smartphones, smart watches, printers, gaming consoles, wireless routers



CPUs → MCUs → Embedded Systems

Microprocessor or Central Processing unit (CPU)



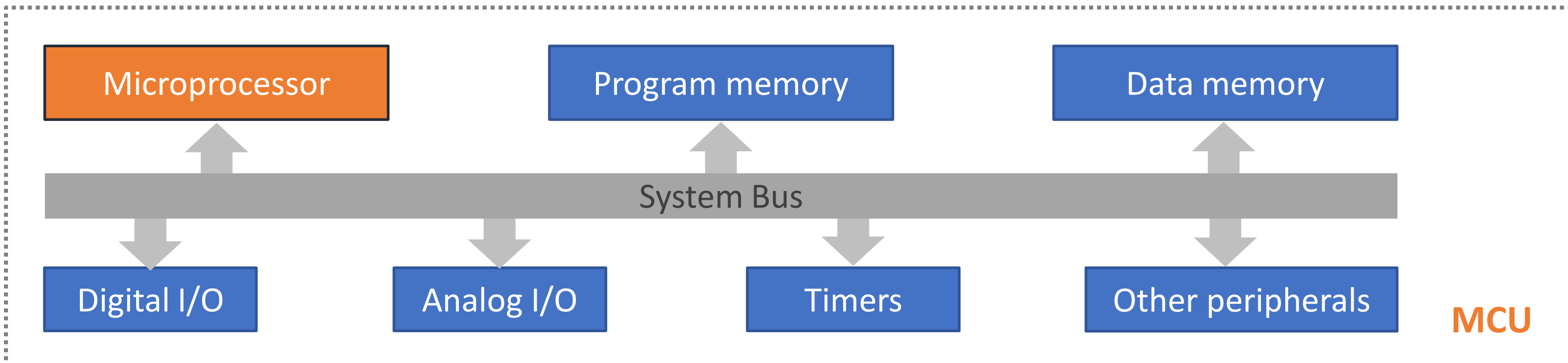
desktop | Computer

Defined typically as a single processor core that supports at least instruction fetching, decoding, and executing

Used for general purpose computing, but needs to be supported with memory and Input/Output (I/O) interfaces

CPUs → MCUs → Embedded Systems

Microcontroller Unit (MCU)



- Typically has a single processor core
- Has memory blocks, digital I/Os, analog I/Os, and other basic peripherals
- Used for basic control purposes, such as embedded applications

CPUs ⟶ MCUs ⟶ Embedded Systems

Embedded system

⇒ MCU's

washing m/c

- Typically implemented using MCUs
- Often integrated into a larger mechanical or electrical system
- Usually has real-time constraints

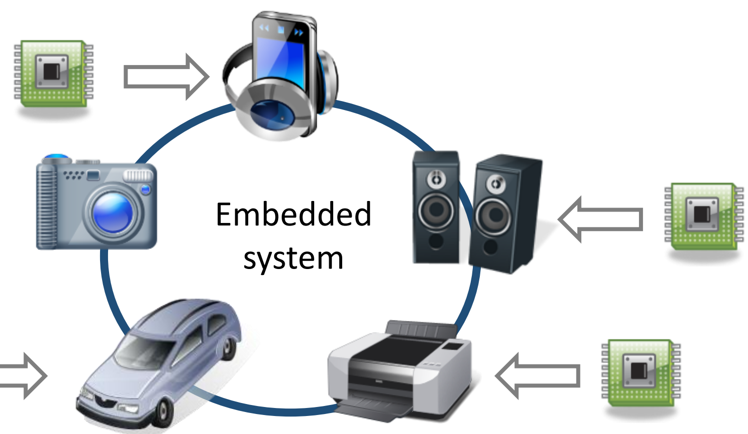
1st MCU

↳ 30
↳ 40
↳ 60
↳

spin the motor that much time.

CPU/up/GPU

MCU's



2nd MCU

Sensing

motor is getting hot

3rd MCU

Embedded system example: Bike computer

only one MCU

- Functions
  - Speed, cadence, distance, heart rate (HR) measurements
- Constraints
  - Size, weight, and cost; power and energy
- Inputs (Sensor)
  - Wheel rotation sensor and mode key
- Output
  - Liquid crystal display (LCD), BLE interface to smartphone
- Uses low performance microcontroller

Inputs:
Wheel rotation
Mode key

Outputs:
Display/transmit speed, HR, etc.

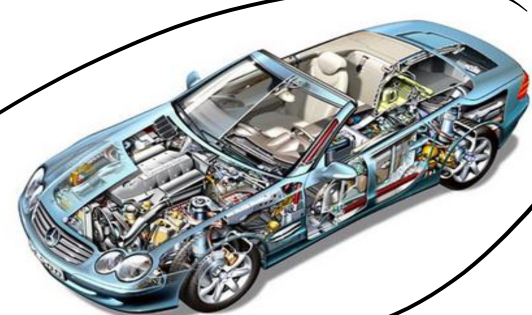
Embedded system example: Gasoline engine control unit

- Functions
  - Fuel injection
  - Air intake setting
  - Spark timing
  - Exhaust gas circulation
  - Electronic throttle control
  - Knock control

- Constraints
  - Reliability in a harsh environment
  - Cost
  - Weight

- Many inputs and outputs
  - Discrete sensors and actuators (MCU)
  - Network interface to rest of the car (MCU)

- Uses high performance microcontroller
  - E.g., 32-bit, 3MB flash memory, 150–300MHz



Automotive Embedded Systems

(Self drive car)

Adaptive cruise control systems detect if cars in front of you are too close and, if necessary, adjust the vehicle's throttle, may apply brakes, and/or sound an alarm.

Advanced airbag systems have crash-severity sensors that determine the appropriate level to inflate the airbag, reducing the chance of airbag injury in low-speed accidents.

Tire pressure monitoring systems send warning signals if tire pressure is insufficient.

Drive-by-wire systems sense pressure on the gas pedal and communicate electronically to the engine how much and how fast to accelerate.

Cars equipped with wireless communications capabilities, called *telematics*, include such features as navigation systems, remote diagnosis and alerts, and Internet access.

Cloud Computing → Fog Computing → Edge Computing

fog nodes

device itself

Automotive Embedded Systems

- Today's high-end automobile may have 100+ microprocessors:
  Seat belt; dashboard devices; engine control; ABS; automatic stability control; navigation system; infotainment system; collision avoidance system; tire pressure monitoring; lane warning; adaptive cruise control; climate control; airbag control unit; electric window and central locking; parking aid; automatic wiper control; alarm and immobilizer; power seat; electric power steering; electronic transmission; active suspension

Projects

pedestration
Person
Street li.
Red Lig
Zebra Cross
Other cars

Side mirror
Back mirror
A/C

Embedded Processor Market

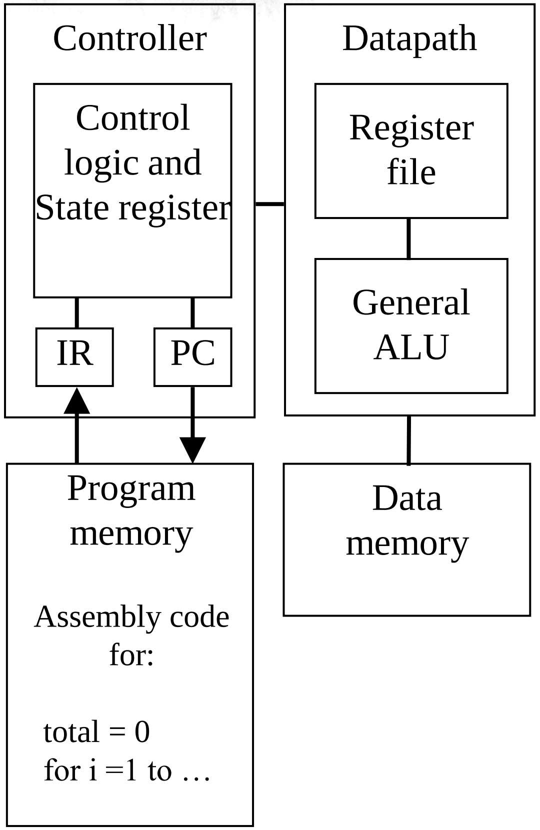
80 million PCs every year

3 billion embedded CPUs every year

Embedded systems market growing, while PC market mostly saturated

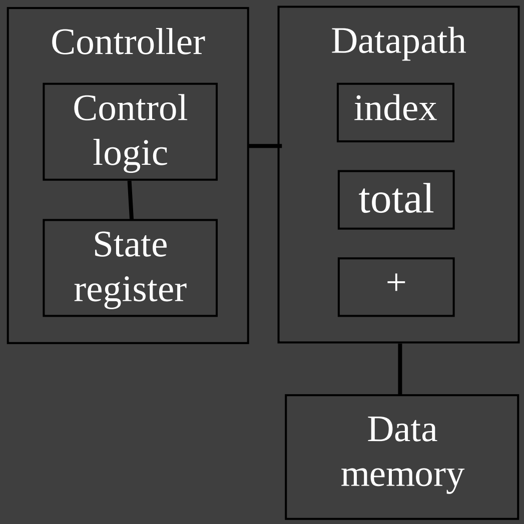
General-Purpose Processor

- Programmable device, “microprocessor”
- Features
  - Program memory
  - General data path with large register file and general ALU
- User benefits
  - Low time-to-market and NRE costs
  - High flexibility
- Examples: Intel Core i7, AMD Ryzen 5, etc.



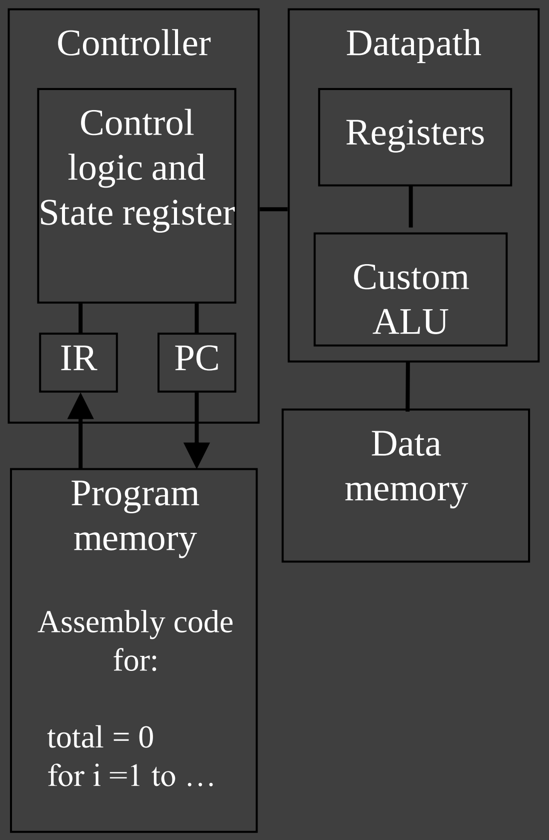
Dedicated Processor

- Digital circuit designed specifically for one purpose
- Features
  - Contains only the components needed to execute a single program
  - No program memory
- Benefits
  - Fast
  - Low power
  - Small size



Application-Specific Processor (ASIC)

- Programmable processor optimized for a particular class of applications that have common characteristics.
- Features
  - Program memory
  - Optimized data path
  - Special functional units
- Benefits
  - Some flexibility, good performance, size, and power, “reusable”



https://en.wikipedia.org/wiki/Application-specific_integrated_circuit

Characteristics of Embedded Systems

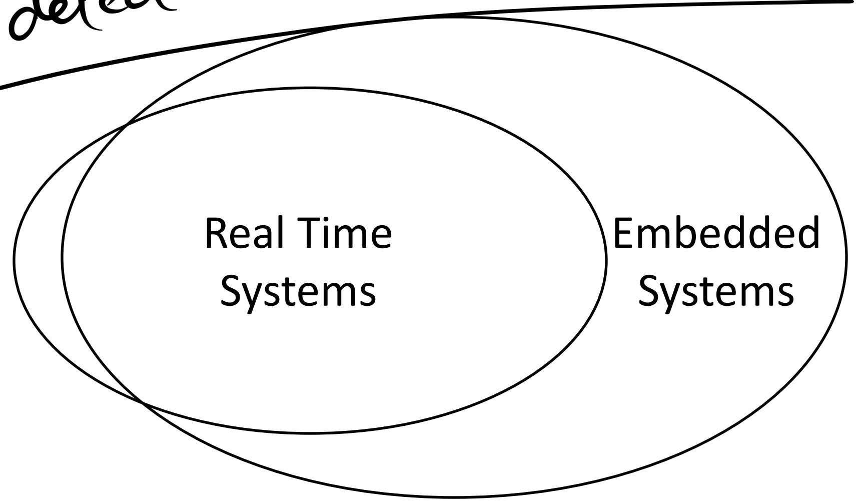
- Dedicated functionality
- Real-time operation
- Small size and low weight
- Low power
- Harsh environments
- Safety-critical operation
- Cost sensitive

Embedded vs. Real Time Systems

- Embedded system: is a computer system that performs a limited set of specific functions; it often interacts with its environment
- RTS: Correctness of the system depends not only on the logical results, but also on the **time** in which the results are produced

Tire pressure

pedestrian detection - in self drive car - RTS



Examples?

Examples

- Real Time Embedded:
  - Nuclear reactor control
  - Flight control
  - Basically, any safety critical system
  - GPS
  - MP3 player
  - Mobile phone
- Real Time, but not Embedded:
  - Stock trading system
  - Skype
  - Pandora, Netflix

  software
- Embedded, but not Real Time:
  - Home temperature control
  - Sprinkler system
  - Washing machine, refrigerator, etc.

Benefits of embedded systems

Greater performance and efficiency

More sophisticated control through software

Lower cost

Cheaper components

Reduced manufacturing costs

Reduced operating and maintenance costs

More features

Many not possible or impractical using other approaches

Better dependability

Adaptive systems that can compensate for failures

Better diagnostics to improve repair time

Constraints specific to embedded devices

Cost

Competitive markets penalize products that do not deliver adequate value for money

Size and weight limits

Mobile (aviation, automotive) and portable (e.g., handheld, wearable) systems

Power and energy limits

Battery capacity, cooling limits

Environment

Temperatures may range from -40 degrees C to 125 degrees C, or even more.

Functions of embedded systems

Closed-loop control system

Monitor a process, adjust an output to maintain the desired set point of operation (temperature, speed, direction, etc.)

Sequencing

Step through different stages based on environment and system conditions

Signal processing

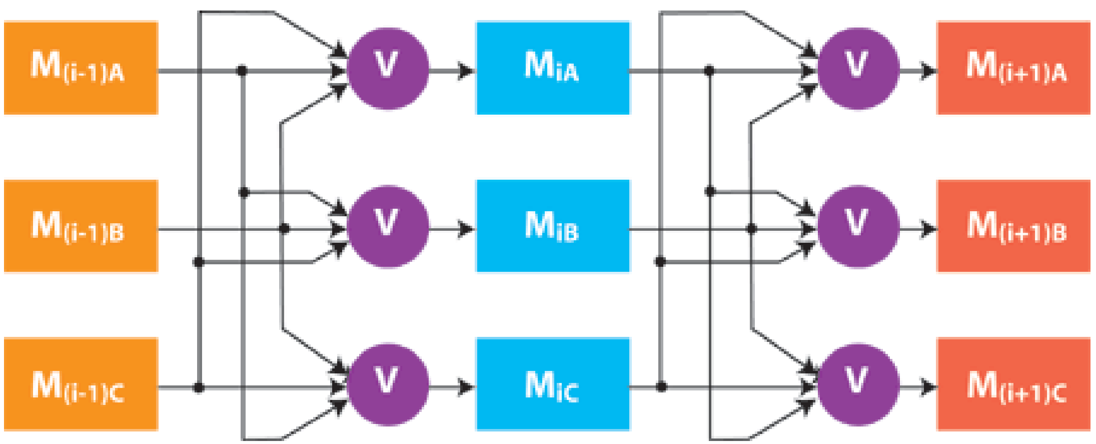
Remove noise, select desired signal features

Communications and networking

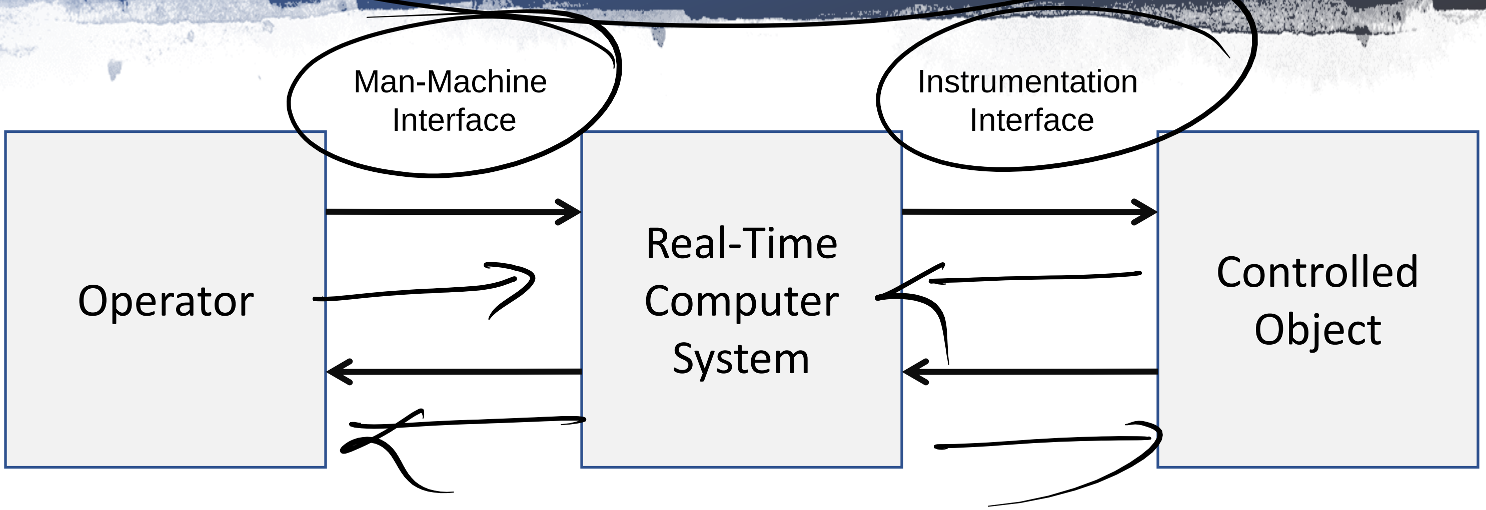
Exchange information reliably and quickly

Characteristics of RTS

- Event-driven (reactive) vs. time-driven
- Reliability/fault-tolerance requirements (example: triple modular redundancy)
- Predictability
- Priorities in multi-programmed systems



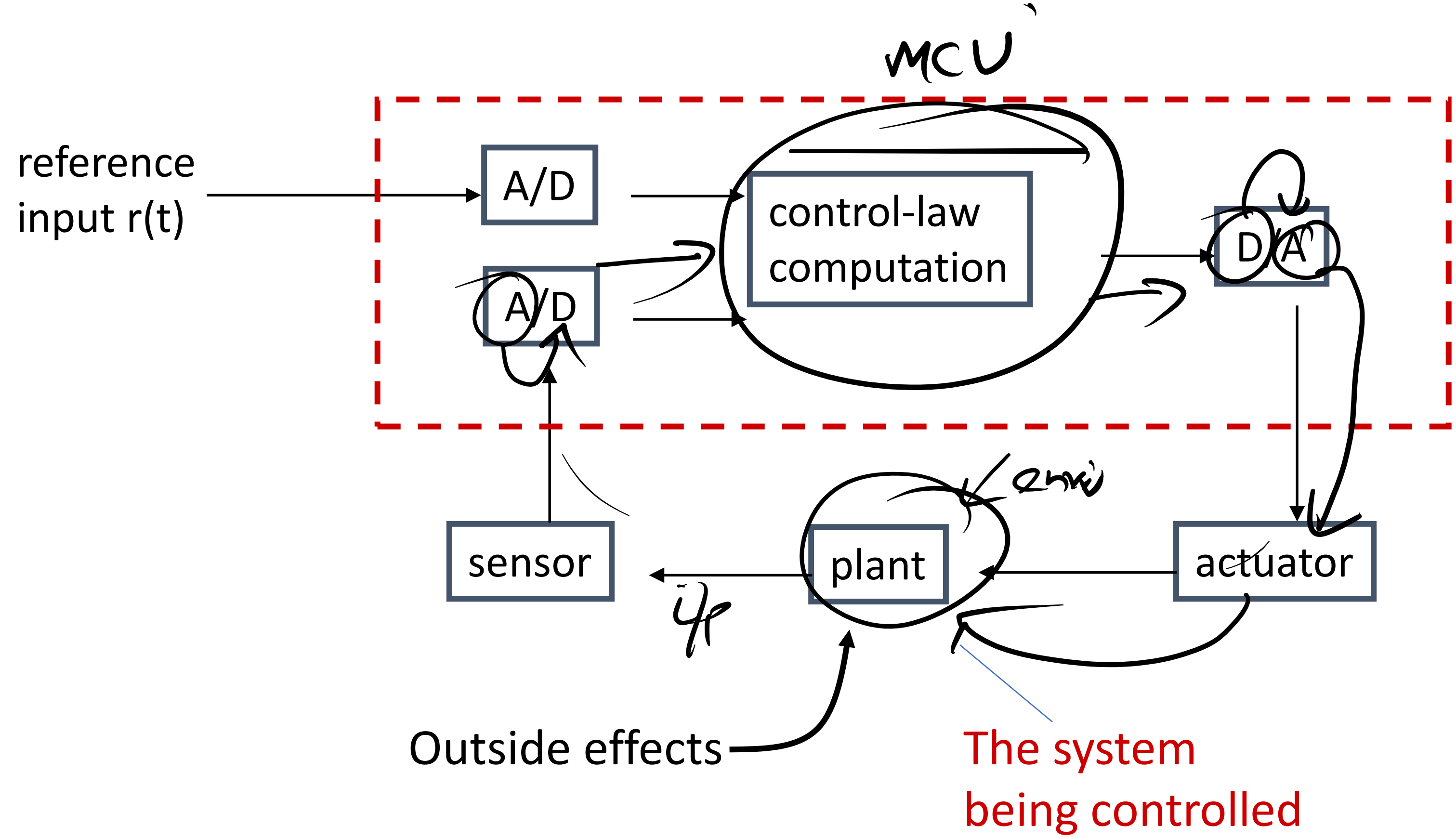
Control Systems



- **Man-machine interface:** input devices, e.g., keyboard and output devices, e.g., display
- **Instrumentation interface**: sensors and actuators that transform between physical signals and digital data.

Control System Example

Example: A simple one-sensor, one-actuator control system.



Control Systems Cont'd.

Pseudo-code for this system:

```
set timer to interrupt periodically with period T;
at each timer interrupt do
    do analog-to-digital conversion to get y;
    compute control output u;
    output u and do digital-to-analog conversion;
end do
```

process it

decode it

T is called the **sampling period**. *T* is a key design choice. *T*ypical range for *T*: seconds to milliseconds.

Sensors and Actuators

Sensors:

- They are mainly input components
- They sense and collect surrounding information

Actuators:

- They are mainly output components
- They alter the surrounding

Communications

- Connects devices with each other & the cloud
- Communication type:
  - Wireline (e.g., copper wires, optical fibers)
  - Wireless (e.g., RF, IR); RF-based communication is the most popular choice
- Popular RF-based communication solutions:
  - IEEE 802.15.4 (LR-WPANs) (Zigbee, 6LoWPAN) LoRaWAN
  - IEEE 802.11 (or Wifi)
  - Bluetooth (BLE)
  - Near Field Communication (NFC), e.g., RFID

Thank You!!

Next Session:

Human Computer Interaction

Contact me:

gauravsingal789@gmail.com

Gaurav.singal@nsut.ac.in

www.gauravsingal.in

LinkedIn:

https://www.linkedin.com/in/gauravsingal789/

Twitter: https://twitter.com/gaurav_singal

Human Computer Interaction

Internet-of-Things (IoT)

COCOS20

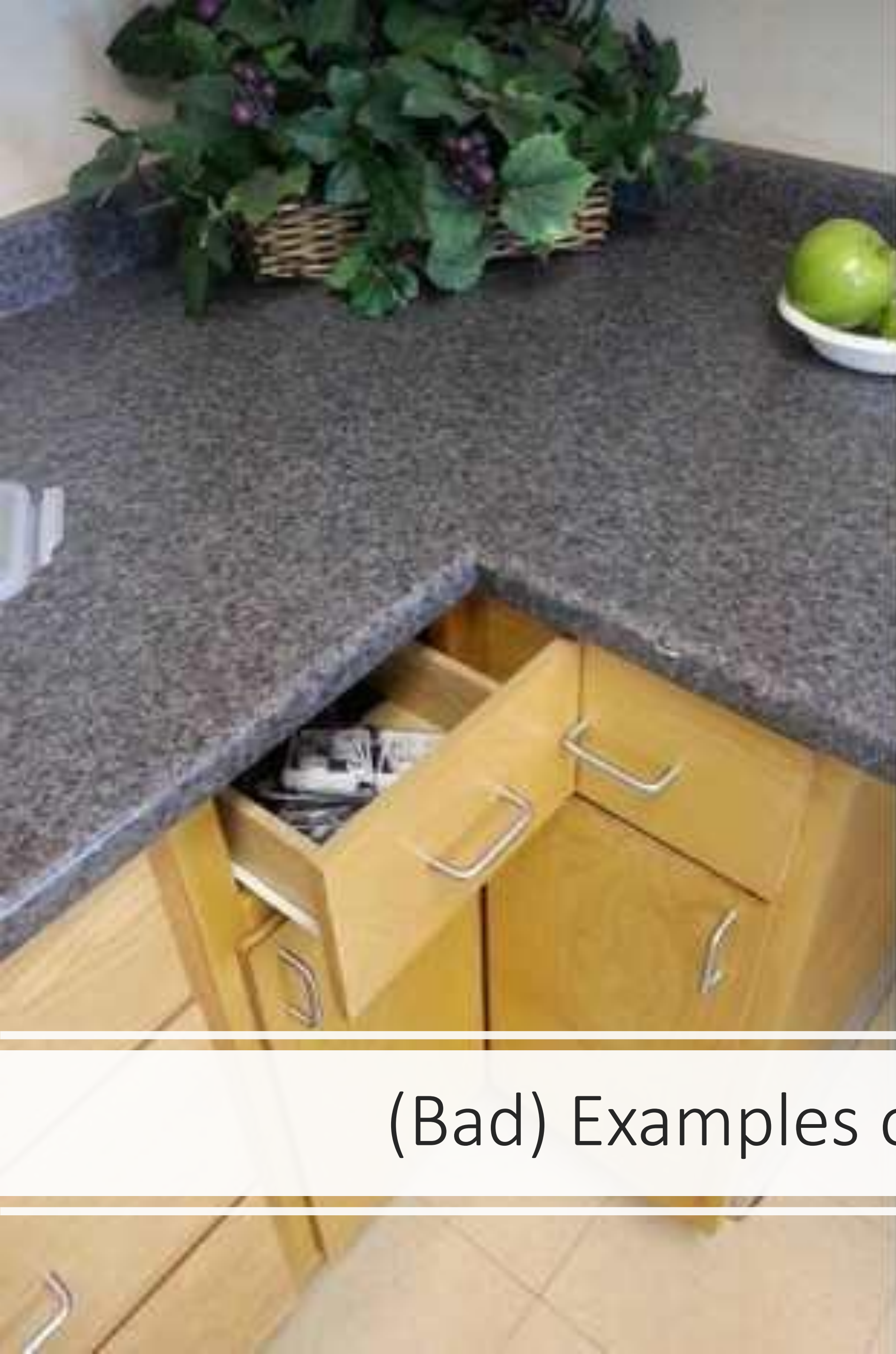
Smart Objects

Smart Refrigerator

- Objects that are able to **sense** the environment, **interpret** the environment, **self-configure**, **interact** with other objects and exchange information with people

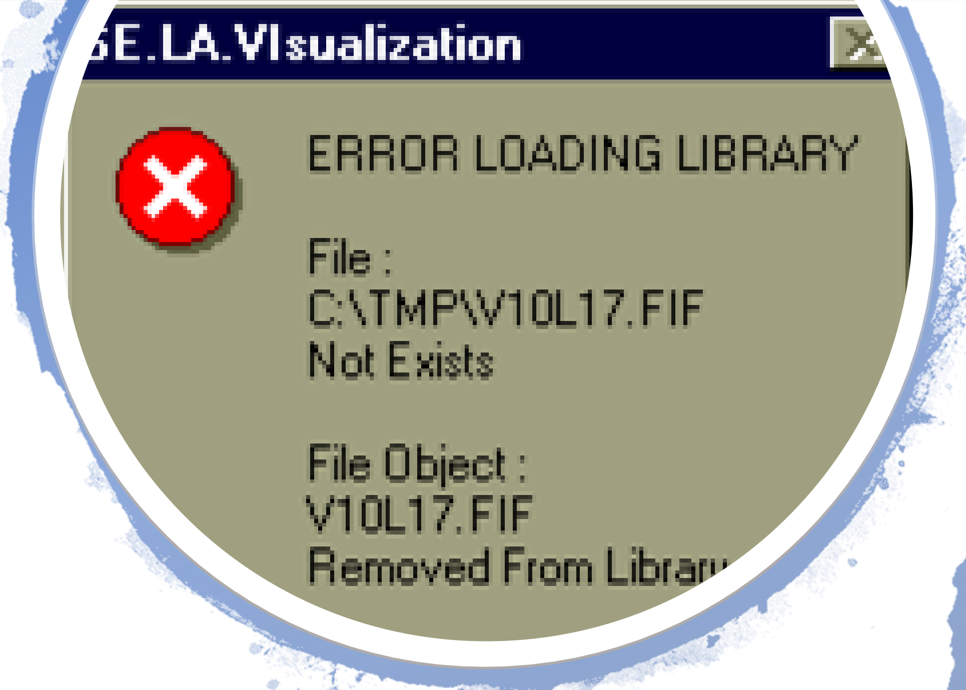
Traditional Computing System: HCI

"***Human-computer interaction*** *is a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them."* -- Association for Computing Machinery.

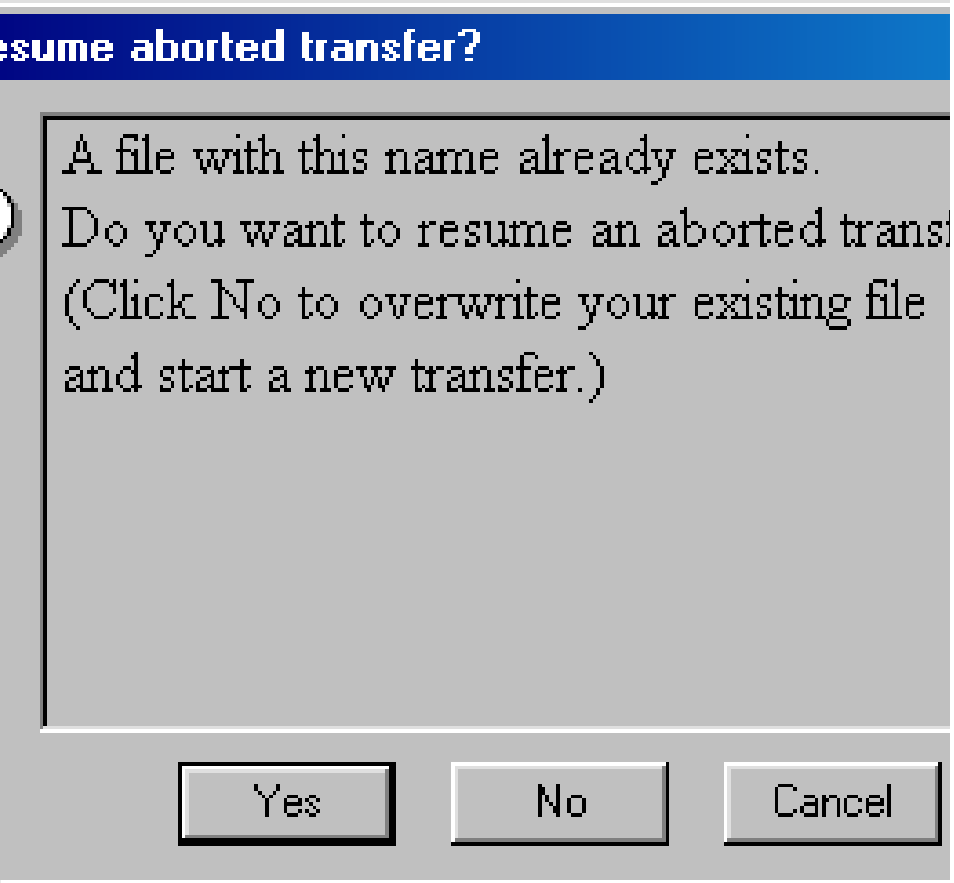


(Bad) Examples of User Interfaces

(Bad) Examples of User Interfaces



(Bad) Examples of User Interfaces

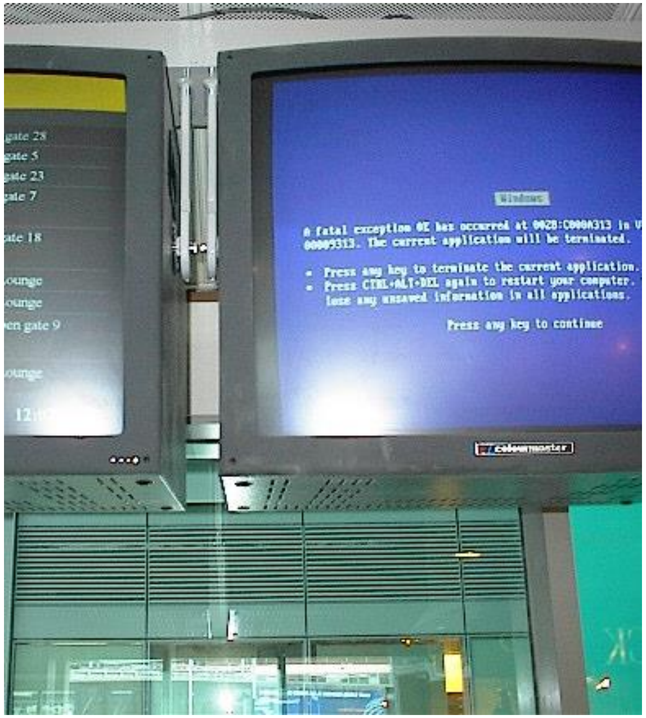


(Bad) Examples of User Interfaces

Why is HCI Important?

- It can affect
  - Effectiveness
  - Productivity
  - Morale
  - Safety

- Bad interfaces:
  - Confusing
  - Cumbersome
  - Time-consuming
  - Uninformative
  - Lead to errors
  - …

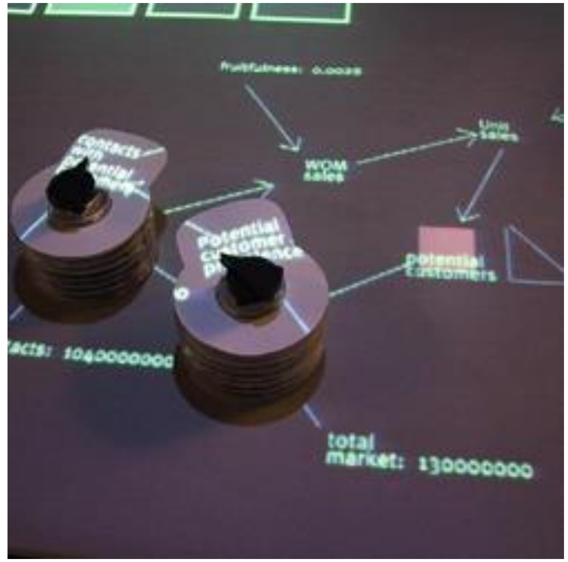


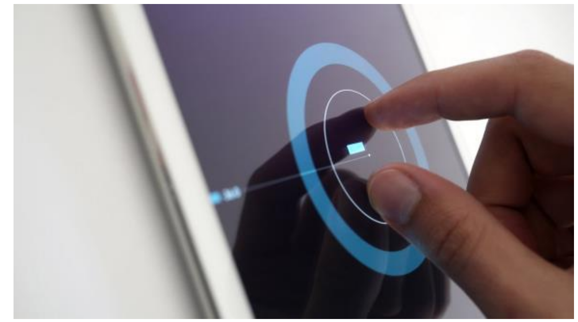
Interfaces

- **Keyboard/mouse/screen/speakers**
- Pen input
- Touch
- Speech/audio/sound
- Gesture, eye movement
- Tangible interfaces
- Virtual/augmented reality (VR, AR)
- Wearable computing
- **Multi-modal** interactive interfaces: more than just one input/output channel

Interface Discussion

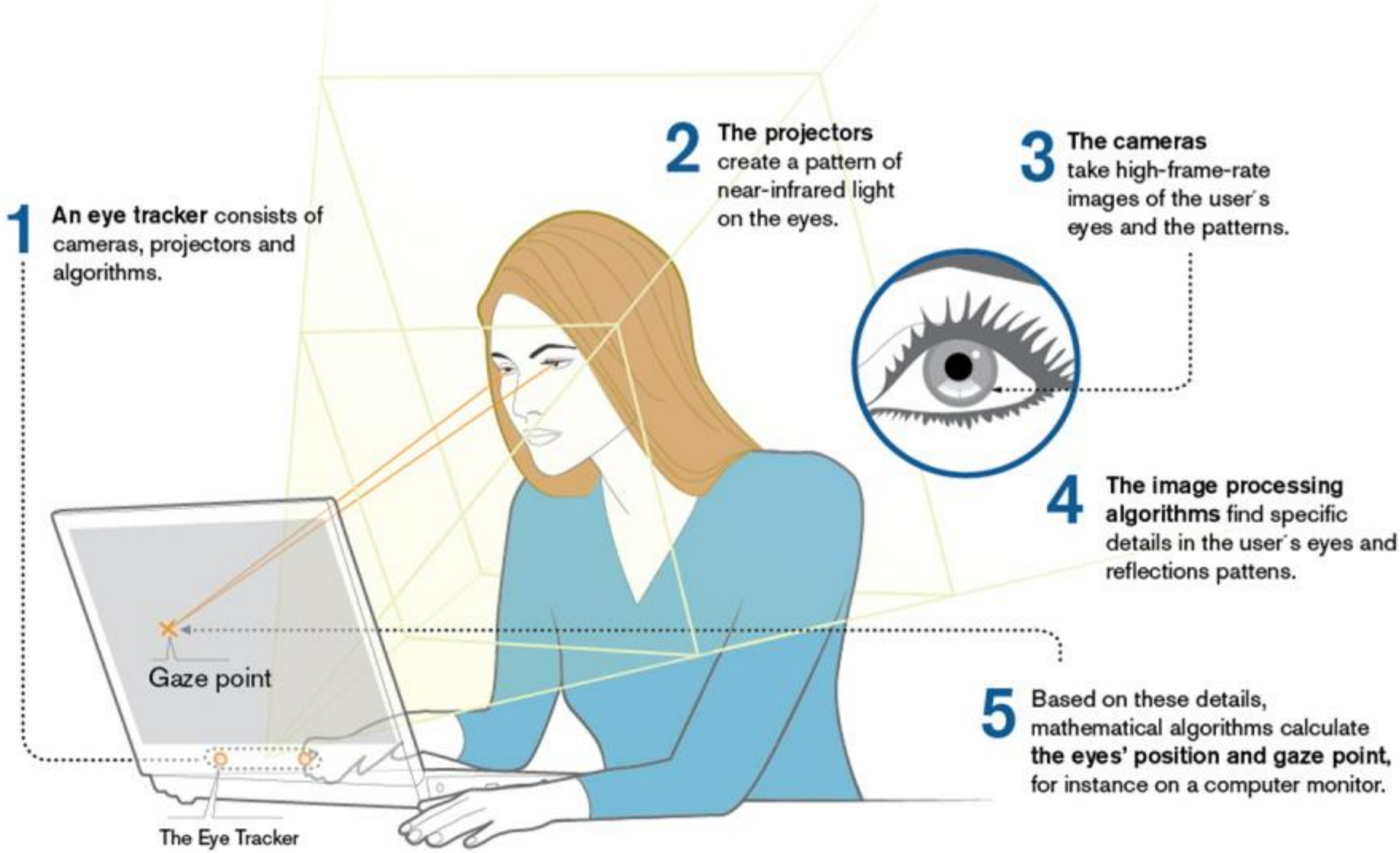
- **E**ase-of-Use?
- **F**lexibility?
- **A**ccuracy?
- **S**afety?
- **P**rivacy?





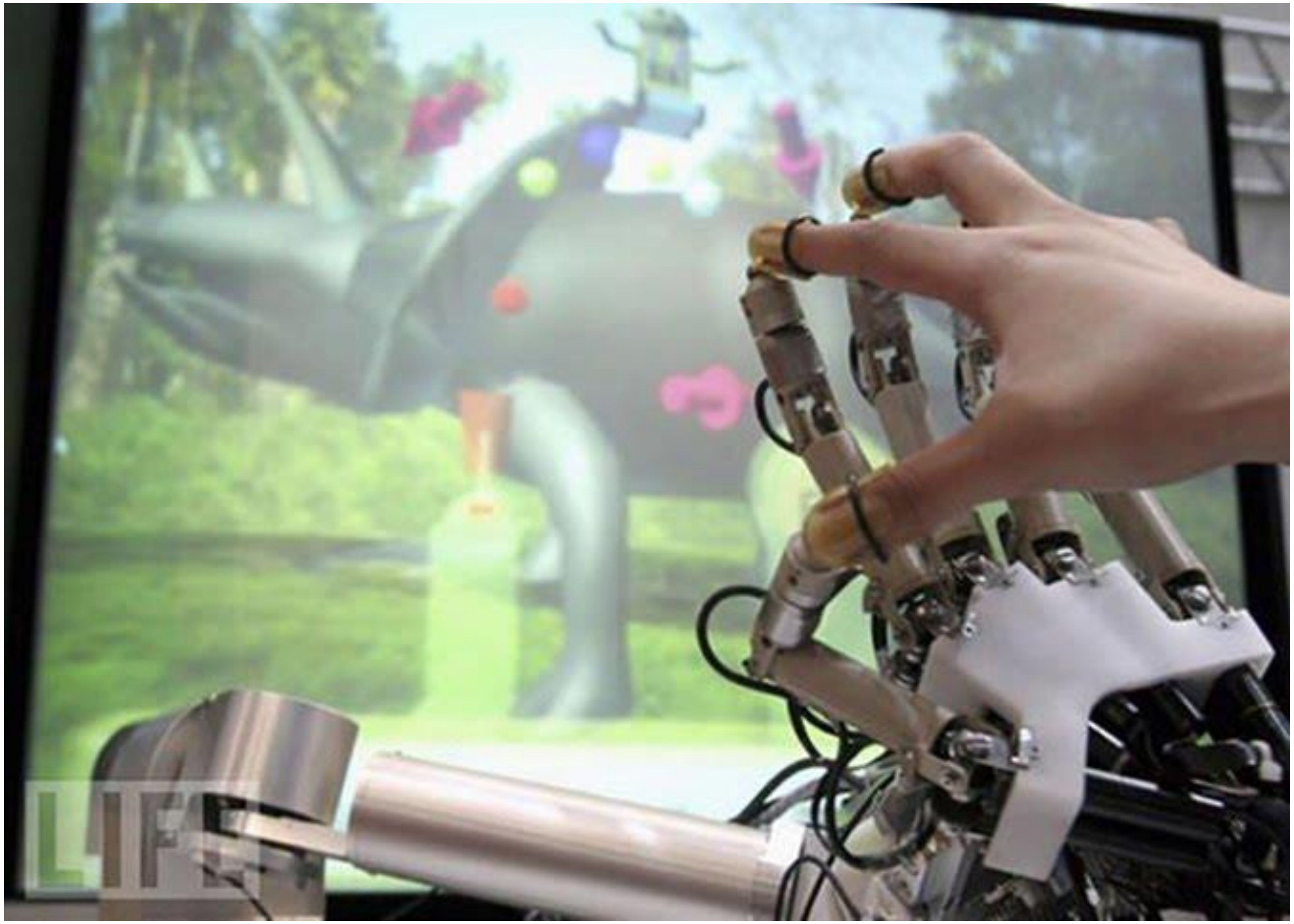
Touch as Input

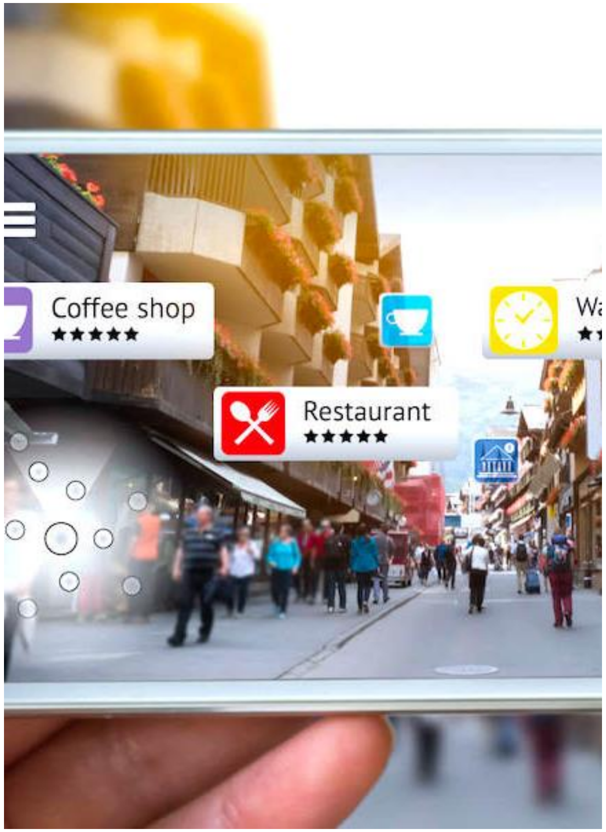
Gesture/Motion as Input

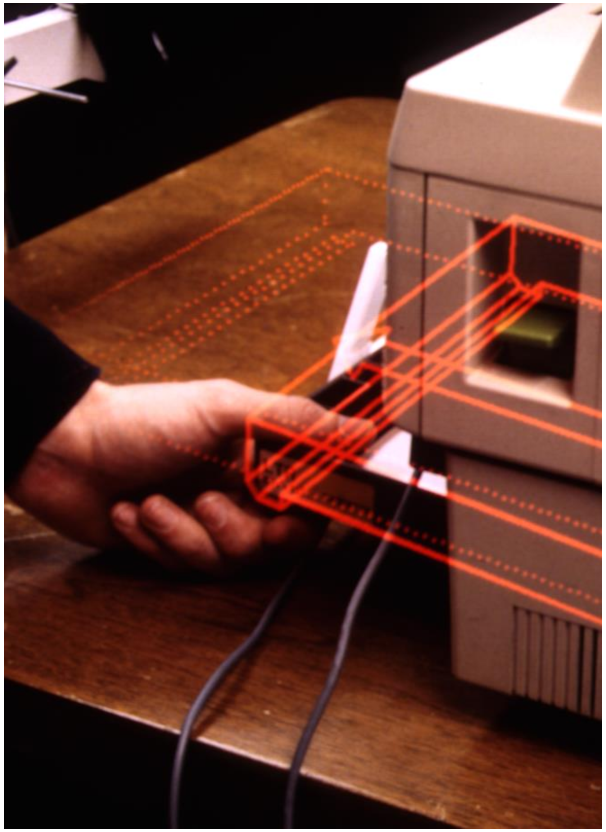


Eye Movement as Input

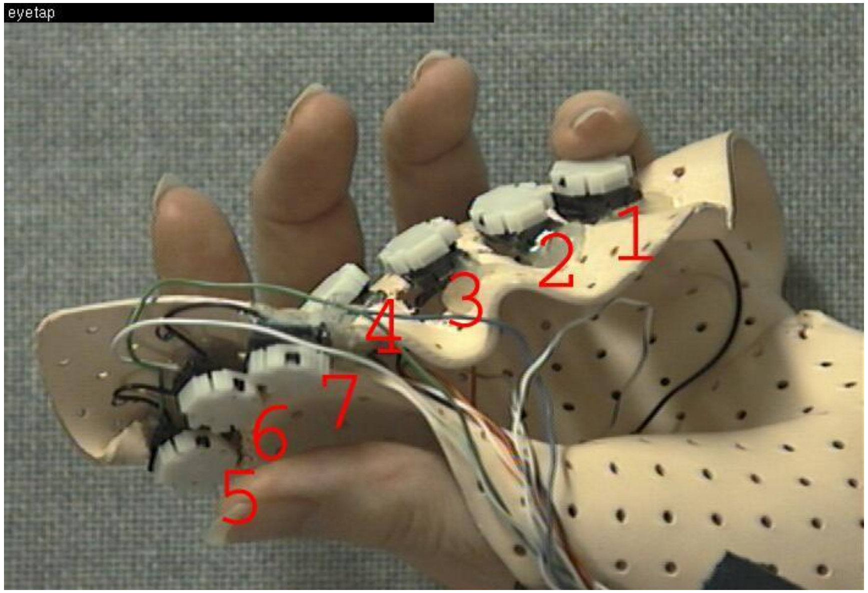
Haptic Interfaces







Augmented Reality

Wearable Computing

Computation devices accompany you, rather than you seeking them out

Speech Input

- Human beings have a great and natural mastery of speech
    - makes it difficult to appreciate the complexities
    - but it's an easy medium for communication

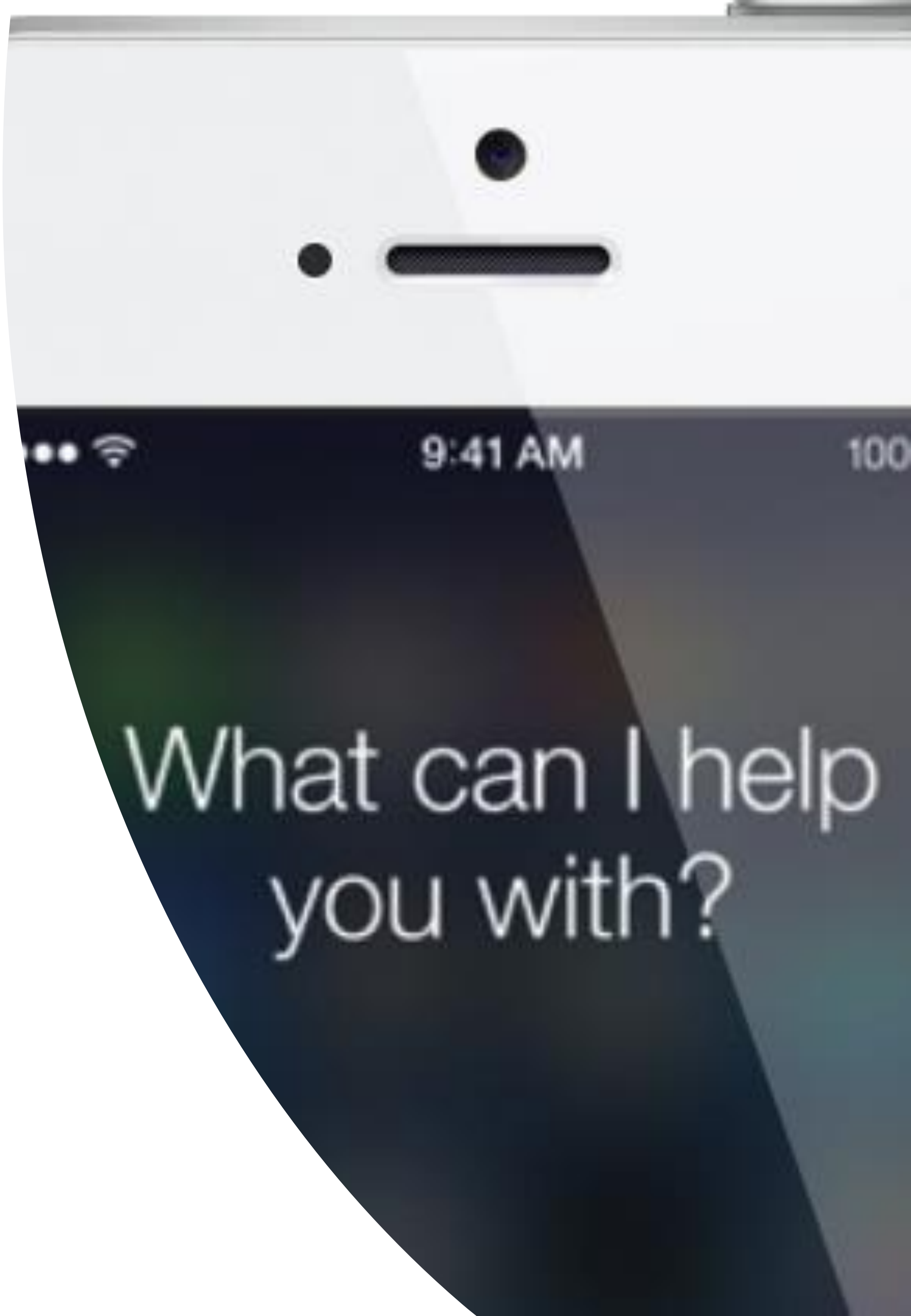
Windows Speech Recognition

- Supplied with every Windows machine
  - From ‘98 on
  - Almost no one used it
- What was the problem?
  - Need to “train” users to use early virtual assistants (VAs)
  - Microphone expense determines quality
  - No app buy-in.

And Then There Was Siri

A Technical Success

- Consistent microphone gives predictable quality
- Inclusion on every iPhone made it mainstream



And Then There Was Siri

- Misunderstandings
- Limited skills
- What Apple wants isn't always what users want
- No 3rd parties; limited innovation and evolution

Current Incarnations

- What these look like now
  - Specialized hardware
  - Domestic setting
  - Initially aimed at home automation
  - Mostly used for home entertainment
  - All open to 3rd parties

Voice “Explodes” into Mainstream

amazon
alexa

Google

IBM Watson™

Microsoft
Hi. I'm Cortana.
Ask me a question!

Seven Design Principles

1. **Equitable use**
   - same means for all users, do not segregate/stigmatize users, make design appealing

2. **Flexibility in use**
   - provide choice of methods & adapt to user's pace

3. **Simplicity and intuitiveness of use**
   - support user's expectations
   - accommodate different languages and literacy skills
   - provide prompting and feedback

Seven Design Principles

4. **Perceptible information**
    - redundancy of information: use different forms/modes
    - emphasize essential information

5. **Tolerance for error**
    - minimize impact caused by mistakes
    - remove potentially dangerous situations
    - hazards should be shielded by warnings

Seven Design Principles

6. Low physical effort

- comfort; minimize fatigue and effort
- repetitive or sustained actions should be avoided

7. Size and space for approach and use

- placement of system should be reachable by all users
- consider line of sight for standing and sitting user
- allow for variation in hand size
- provide room for assistive devices

Disabilities

- Federal law to ensure access to IT, including computers and web sites.

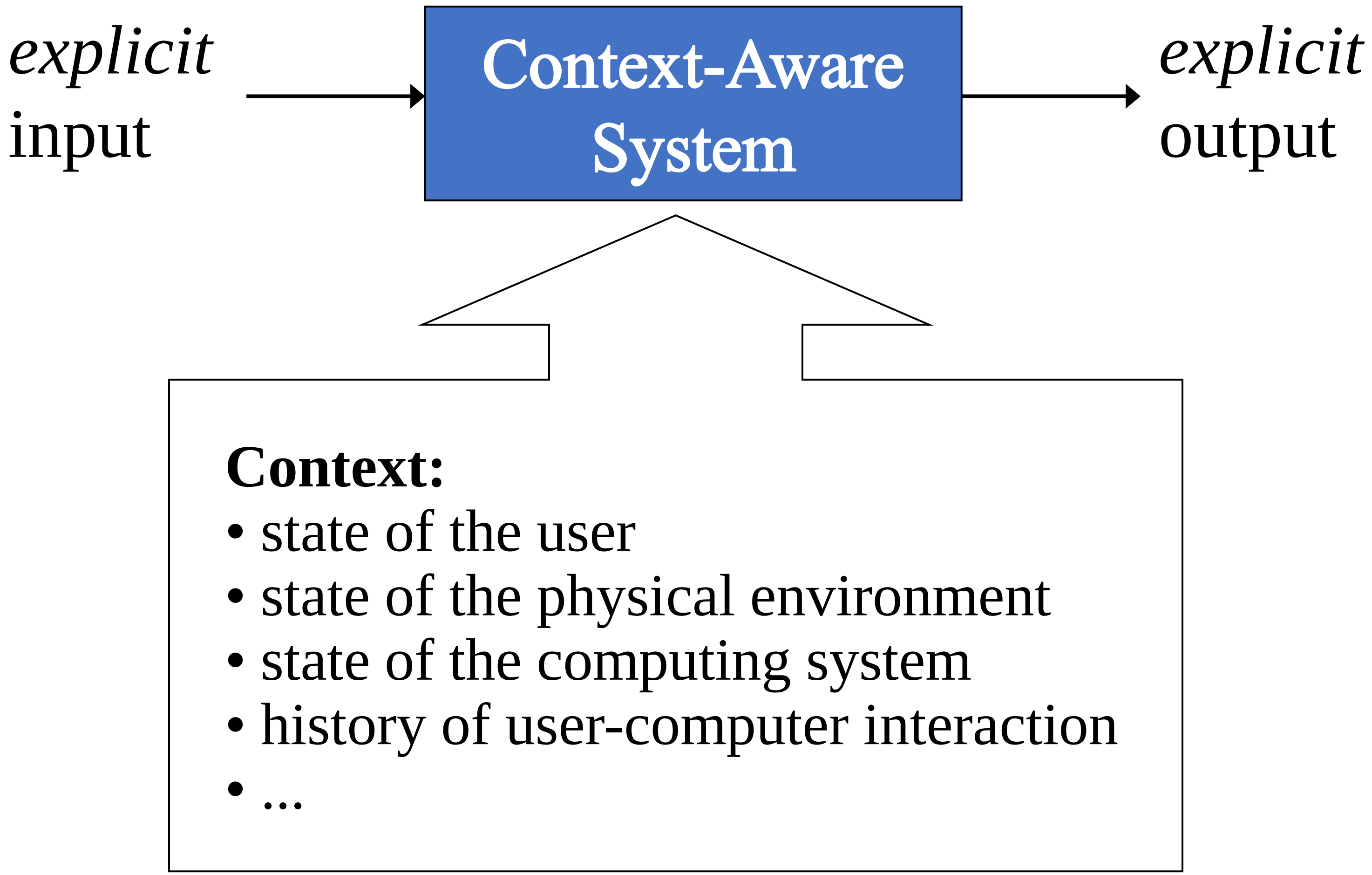
  - Vision (low vision, blind, color blind)
  - Hearing (deaf, limited hearing)
  - Mobility
  - Learning (dyslexia, attention deficit)

Disabilities

- Keyboard and mouse alternatives
- Color coding
- Font size
- Contrast
- Text descriptors for web images
- Magnification
- Text-to-speech; speech recognition
- Head-mounted optical mice
- Eye gaze control

System Structure

Context as **Implicit** Input



What is Context?

Examples of Context

- Identity (user, others, objects)
- Location
- Date/Time
- Environment
- Emotional state
- Focus of attention
- Orientation
- User preferences
- Calendar (events)
- Browsing history
- Behavioral patterns
- Relationships (phonebook, call history)
- … the elements of the user’s environment that the computer knows about…

Relevance of Context Information

- Trying to arrange lunch meeting
- Going to a job interview
- Going home after work and making evening plans
- Shopping
- Tourist
- ...

Definitions of Context

- “Context is **any information that can be used to characterize the situation of an entity**. An entity is a person, place, or object that is considered **relevant** to the interaction between a user and an application, including the user and applications themselves” [Dey et al. 2001]

Classification

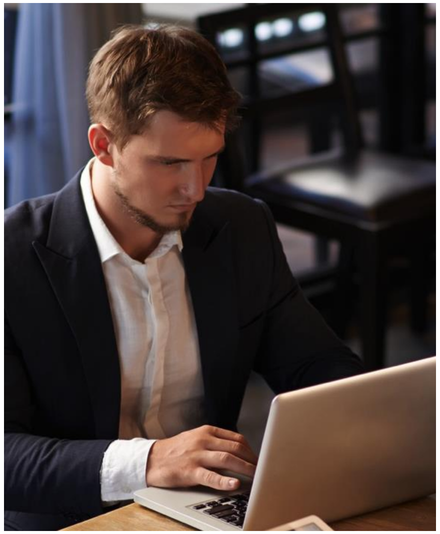
External (physical)

- Context that can be measured by hardware sensors
- Examples: location, light, sound, movement, touch, temperature, air pressure, etc.

Internal (logical)

- Mostly specified by the user or captured monitoring the user's interaction
- Examples: the user's goal, tasks, work context, business processes, the user's emotional state, etc.

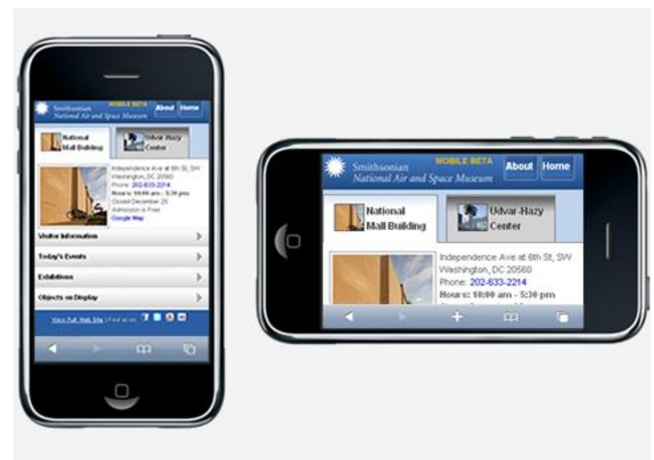
Context?



Context?

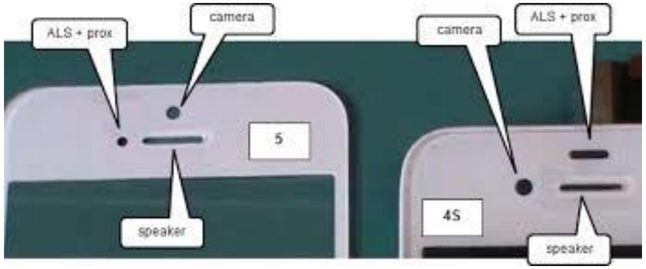
Simple Everyday Examples

- Smartphone adjusts the screen to the orientation of the device
- Apple Watch turns on display if arm lifted/rotated
- Orientation is determined by using both a gyroscope and an accelerometer



Simple Everyday Examples

- Phone display adjusts the brightness of the display based on the surrounding area
- Uses a light sensor



Simple Everyday Examples

- Device displays user's location, shows route to a desired destination, find nearby stores, geotag images on social media, etc.
- Uses location sensor

Simple Everyday Examples

- The time is displayed on the phone
    - Time zone change
    - Daylight savings time

Simple Everyday Examples

- Device disables touch screen when the user speaks on the phone
- Uses a proximity sensor (infrared signal travel time)

Challenges

- Lack of self-awareness
  - Knowing when to do or not to do something is hard
- Complexity
  - More rules do not necessarily yield more intelligence
  - But will become harder to maintain and understand
- Human-in-the-loop vs. automation
  - Loss of control vs. risk of human error
- Development
  - Sensing, aggregation, rules, etc., are complex issues
- Privacy
- User preferences
- Information overload

Contact me:

gauravsingal789@gmail.com

Gaurav.singal@nsut.ac.in

www.gauravsingal.in

LinkedIn:

https://www.linkedin.com/in/gauravsingal789/

Twitter: https://twitter.com/gaurav_singal

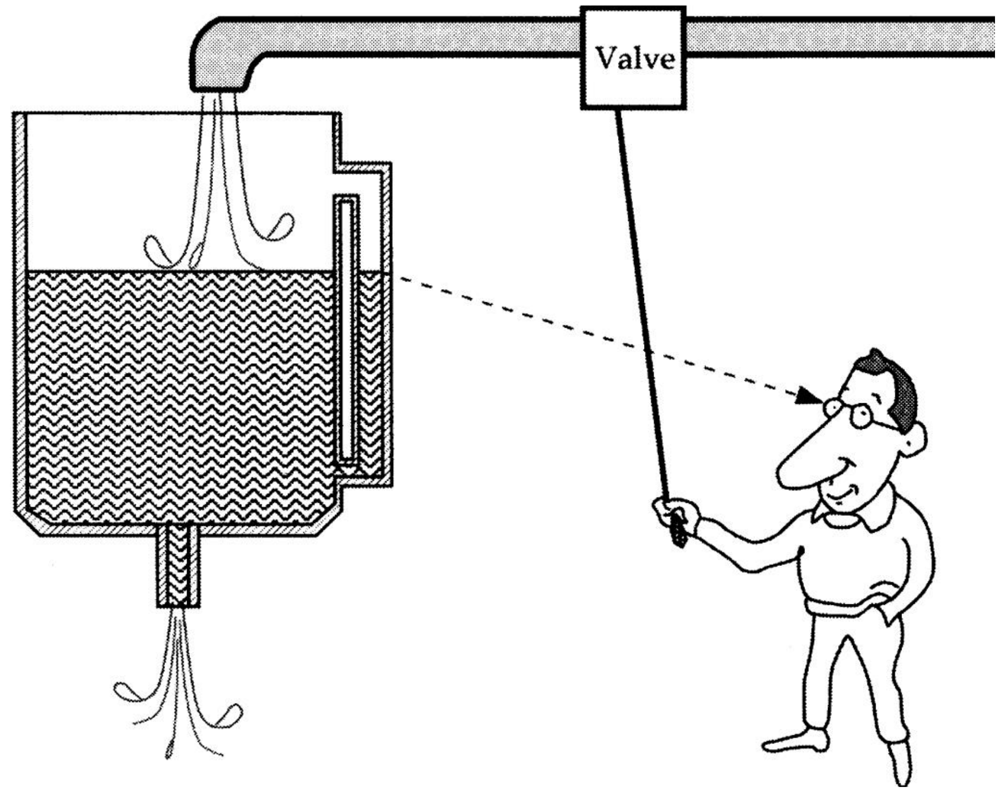
Sensors

Internet-of-Things (IoT)

COCOS20

Sensors

- A sensor is a device that receives a stimulus and responds with an electrical signal.



Sensors

Sensors are devices that measure a particular property of the environment and convert that into an electrical output

Typically, sensors have a linear function between the physical property measured and the electrical output

The sensitivity of a sensor indicates the minimum value of the measured input that can produce a certain output signal

An analog-to-digital converter employed to turn a sensor output into signals that can be processed by MCUs

Different physical phenomena underpin the operation of sensors, e.g., the piezoelectric effect (accumulation of electric charge when mechanical pressure is applied)

Computer-Process Interface

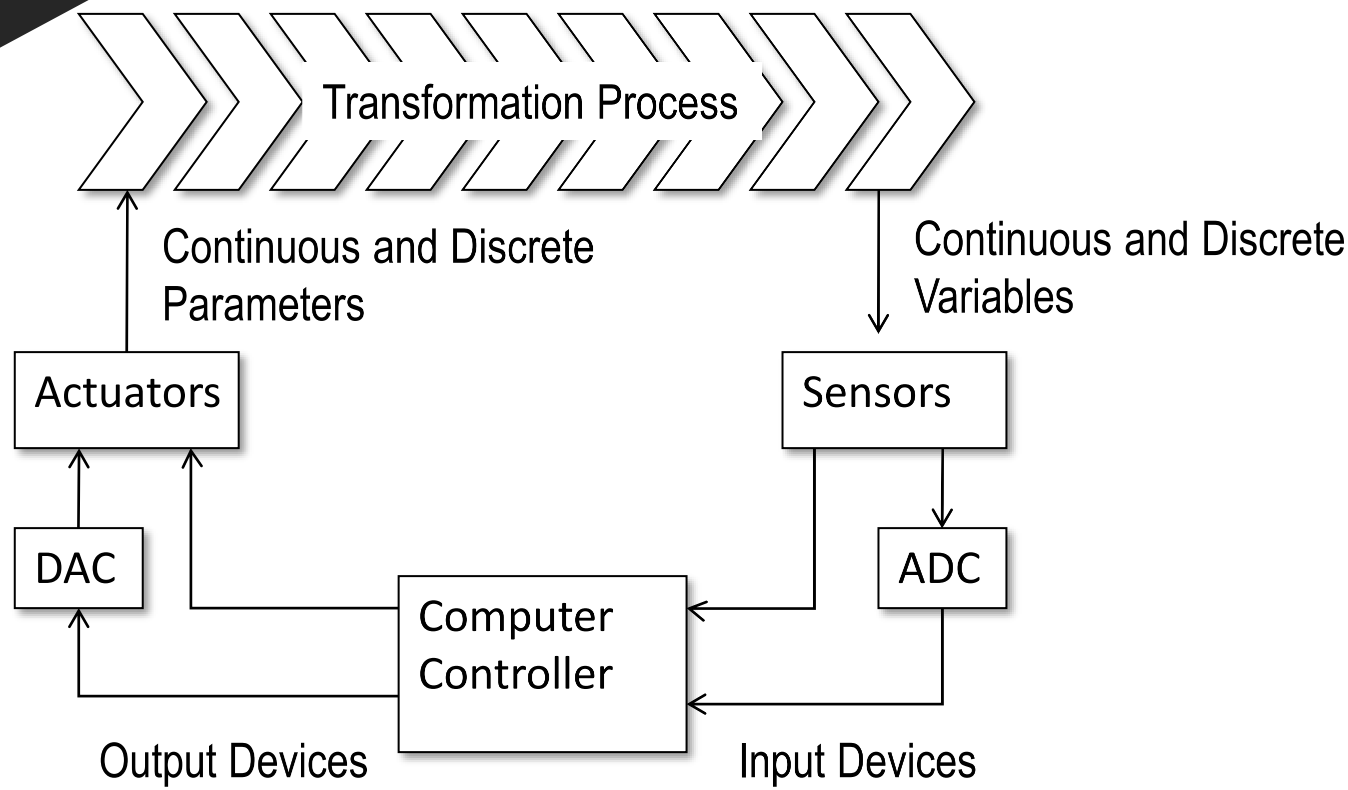
- To implement process control, the computer must collect data from and transmit signals to the production process
- Components required to implement the interface:
  - Sensors to measure continuous and discrete process variables
  - Actuators to drive continuous and discrete process parameters
  - Devices for ADC and DAC
  - I/O devices for discrete data

Need for Sensors

- Sensors are omnipresent. They embedded in our bodies, automobiles, airplanes, cellular telephones, radios, chemical plants, industrial plants and countless other applications.

- Without the use of sensors, there would be no automation!!

Computer Process Control System



What is a Stimulus?

- Motion, position, displacement
- Velocity and acceleration
- Force, strain
- Pressure
- Flow
- Sound
- Moisture
- Light
- Radiation
- Temperature
- Chemical presence

Visual Sensor

Ultrasound Sensor

Infrared Sensor

Types of sensors

- Push-button/switch

- Temperature

- Acceleration

- Pressure

- Optical/photodiode

- Humidity

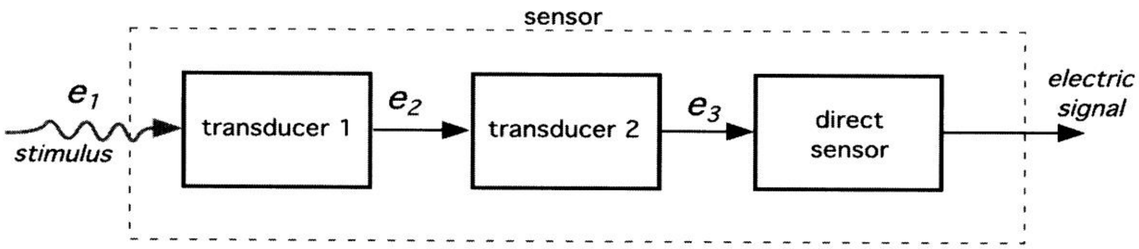
What is a Response?

When we say electrical, we mean a signal which can be channeled, amplified, and modified by electronic devices:

- **Voltage**
- **Current**
- **Charge**

Sensor as Energy Converter

- This conversion can be direct or it may require transducers



- Example:

  A chemical sensor may have a part which converts the energy of a chemical reaction into heat (transducer) and another part, a thermopile, which converts heat into an electrical signal.

  Microphone, Loud Speaker, Biological Senses (e.g. touch, sight,…, etc)

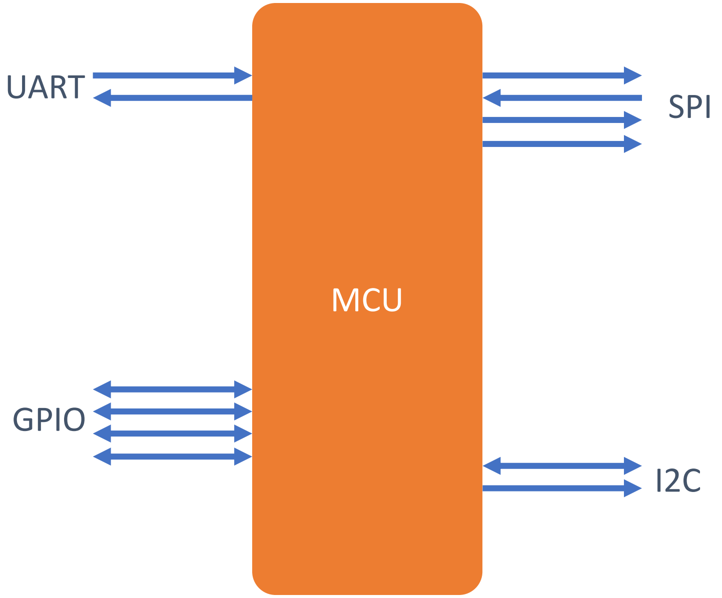
Physical Principles of Sensing

- Charges, fields, & potentials
- Capacitance
- Magnetism
- Induction
- Resistance
- Piezoelectric effect
- Seebeck and Peltier effects
- Thermal properties of materials
- Heat transfer
- Light

Input/Output

I/O defines how an MCU can interact with the environment

- Different I/O protocols are available
- Universal Asynchronous Receiver/Transmitter (UART) – two wires to send/receive data between devices
- Serial Peripheral Interface (SPI) – any number of bits can be sent/received without interruption
- Inter-Integrated Circuit (I^2C) – combines features of UART and SPI
- General purpose I/O (GPIO) – controllable by user at run time.



GPIO

- Can be set up to accept or source different logic voltage levels, through which MCU can control peripherals or receive external input/interrupts.
- If pins are configured for interrupts, they can be used to move wake-up from low-power/sleep modes.
- Can be grouped into a GPIO port and controlled as such.
- Pulse-width modulation (PWM) employed when the linear processes must be controlled (e.g., fans)
- Limited to low-current applications → transistors and relays used to help drive higher current loads.

Pulse-width modulation

Reducing average power output by switching on/off at high rates.

- Duty Cycle – the fraction of one period T during which a signal is active:

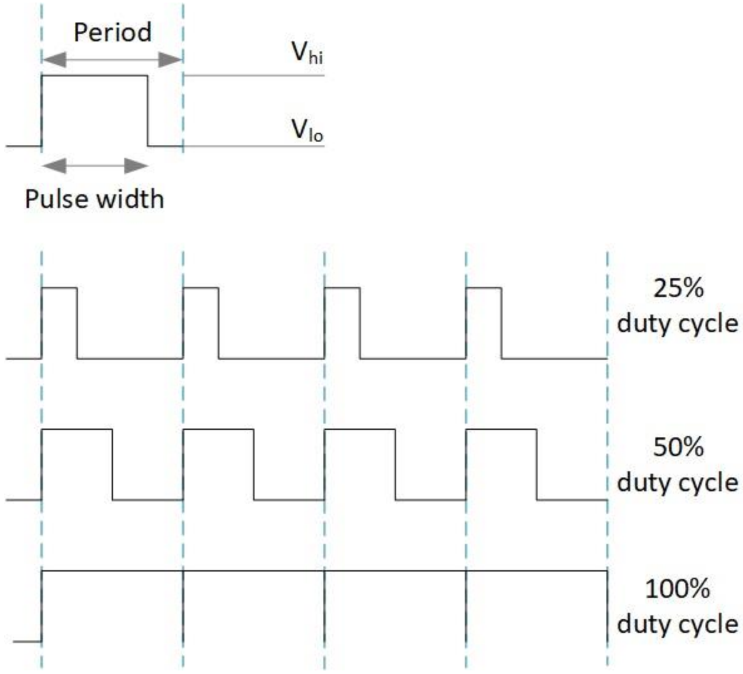
$$D = \frac{PW}{T} \times 100\ [\%]$$

- Frequency – rate of periods:

$$f = \frac{1}{T}\,[\mathrm{Hz}]$$

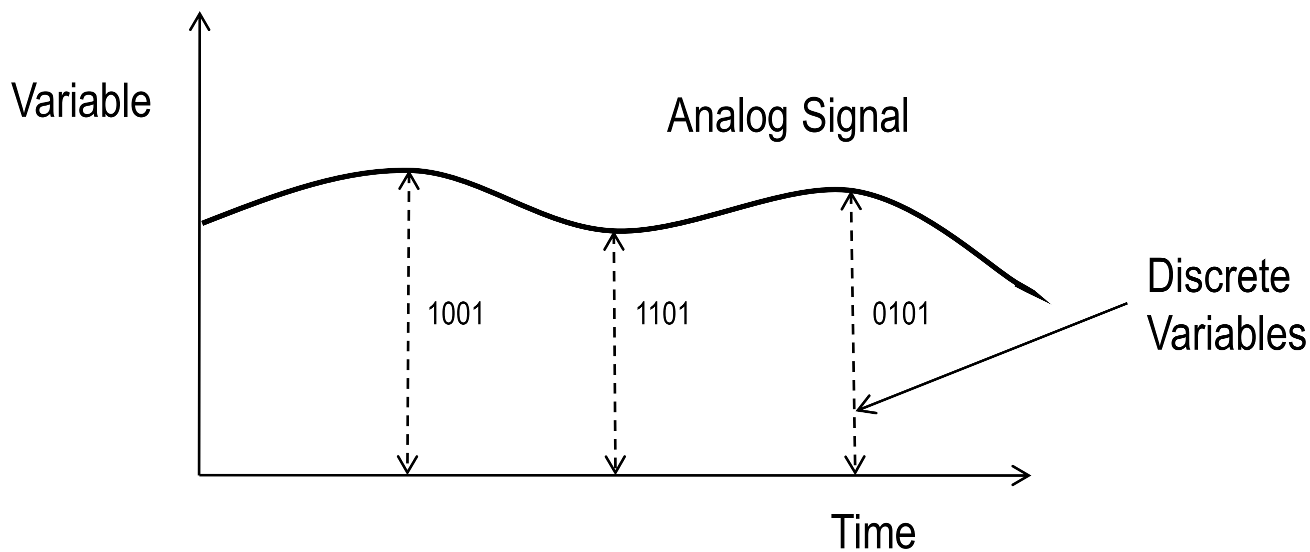
- Average voltage:

$$V_{avg} = V_{hi} \times \frac{D}{100}$$



Analog-to-Digital Conversion

- **Sampling** – converts the continuous signal into a series of discrete analog signals at periodic intervals
- **Quantization** – each discrete analog is converted into one of a finite number of (previously defined) discrete amplitude levels
- **Encoding** – discrete amplitude levels are converted into digital code



Features of an ADC

- **Sampling rate** – rate at which continuous analog signal is polled (e.g., 1000 samples/sec)
- **Quantization** – divide analog signal into discrete levels
- **Resolution** – depends on number of quantization levels
- **Conversion time** – how long it takes to convert the sampled signal to digital code
- **Conversion method** – means by which analog signal is encoded into digital equivalent
    - Example: Successive approximation method

Successive Approximation Method

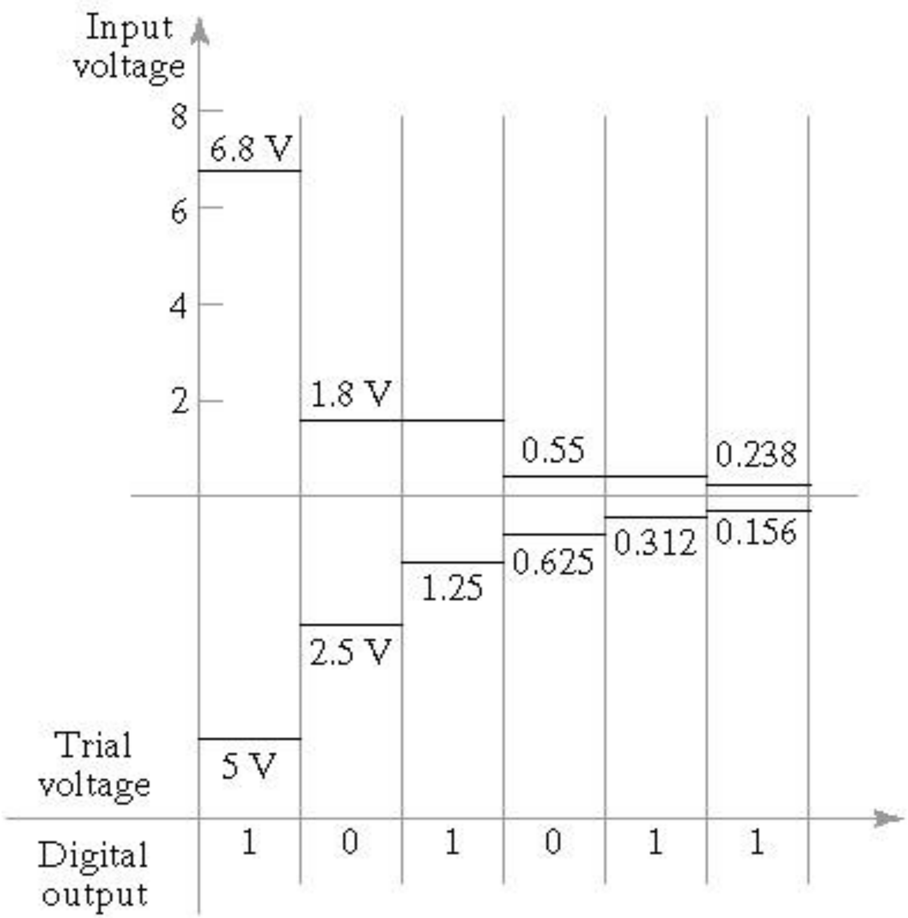
- A successive approximation ADC is a type of ADC that converts a continuous analog waveform into a discrete digital representation via a binary search through all possible quantization levels before finally converging upon a digital output for each conversion

Algorithm

- A series of trial voltages are successively compared to the input signal whose value is unknown
- Number of trial voltages = number of bits used to encode the signal
- First trial voltage is 1/2 the full scale range of the ADC
- If the remainder of the input voltage exceeds the trial voltage, then a bit value of 1 is entered, if less than trial voltage then a bit value of zero is entered
- The successive bit values, multiplied by their respective trial voltages and added, becomes the encoded value of the input signal

Example

- Analogue signal is 6.8 volts. Encode, using SAM, the signal for a 6 bit register with a full scale range of 10 volts.



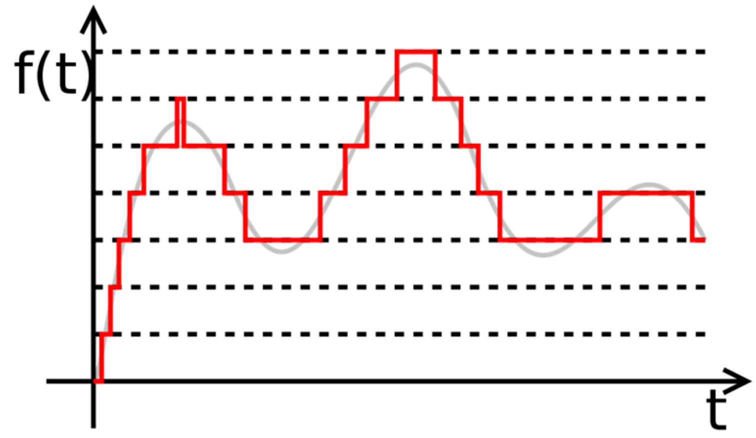
For six digit precision, the resulting binary digital value is 101011, which is interrupted as:

1×5.0 V
0×2.5 V
1×1.25 V
0×0.625 V
1×0.312 V
1×0.156 V

Total = 6.718 V

Example

$$Q = \frac{E_{FSR}}{2^M} = \frac{E_{FSR}}{N}$$



- Q = resolution in volts per step
- M = resolution in bits
- N = Number of intervals (steps, quantization levels)
- E_{FSR} = Full scale voltage range
- Quantization error = ½ of interval

- Voltage range 0 – 10V; M = 12 bits
- N = 4096 intervals (steps)
- Q = 2.44 mV/code

Example

- Using an analogue-to-digital converter, a continuous voltage signal is to be converted into its digital counterpart. The ADC has a 16-bit capacity, and full scale range of 60 V. Determine:
  - number of quantization levels
  - resolution
  - quantization error

Solution

(1) Number of quantization levels:

$= 2^{16} = 65,536$

(2) Resolution:

$= 60 / 65,536 = \sim 0.00092$ Volts

(3) Quantization error:

$= 0.00092/2 = 0.00046$ Volts

Digital-to-Analog Conversion

- Convert digital values into continuous analogue signal
  - Decoding digital value to an analogue value at discrete moments in time based on value within register

$$E_0 = E_{ref}\left\{0.5B_1 + 0.25B_2 + \cdots + \left(2^n\right)^{-1}B_n\right\}$$

Where E_0 is output voltage; E_{ref} is reference voltage; B_n is status of successive bits in the binary register

Example

- A DAC has a reference voltage of 100 V and has 6-bit precision. Three successive sampling instances 0.5 sec apart have the following data in the data register:

Instant	Binary Data
1	101000
2	101010
3	101101

- Output Values:

$$E_{01} = 100\{0.5(1)+0.25(0)+0.125(1)+0.0625(0)+0.03125(0)+0.015625(0)\}$$
$$E_{01} = 62.50V$$
$$E_{02} = 100\{0.5(1)+0.25(0)+0.125(1)+0.0625(0)+0.03125(0)+0.015625(0)\}$$
$$E_{02} = 65.63V$$
$$E_{03} = 100\{0.5(1)+0.25(0)+0.125(1)+0.0625(0)+0.03125(0)+0.015625(0)\}$$
$$E_{03} = 70.31V$$

Sensor Types: HW & SW

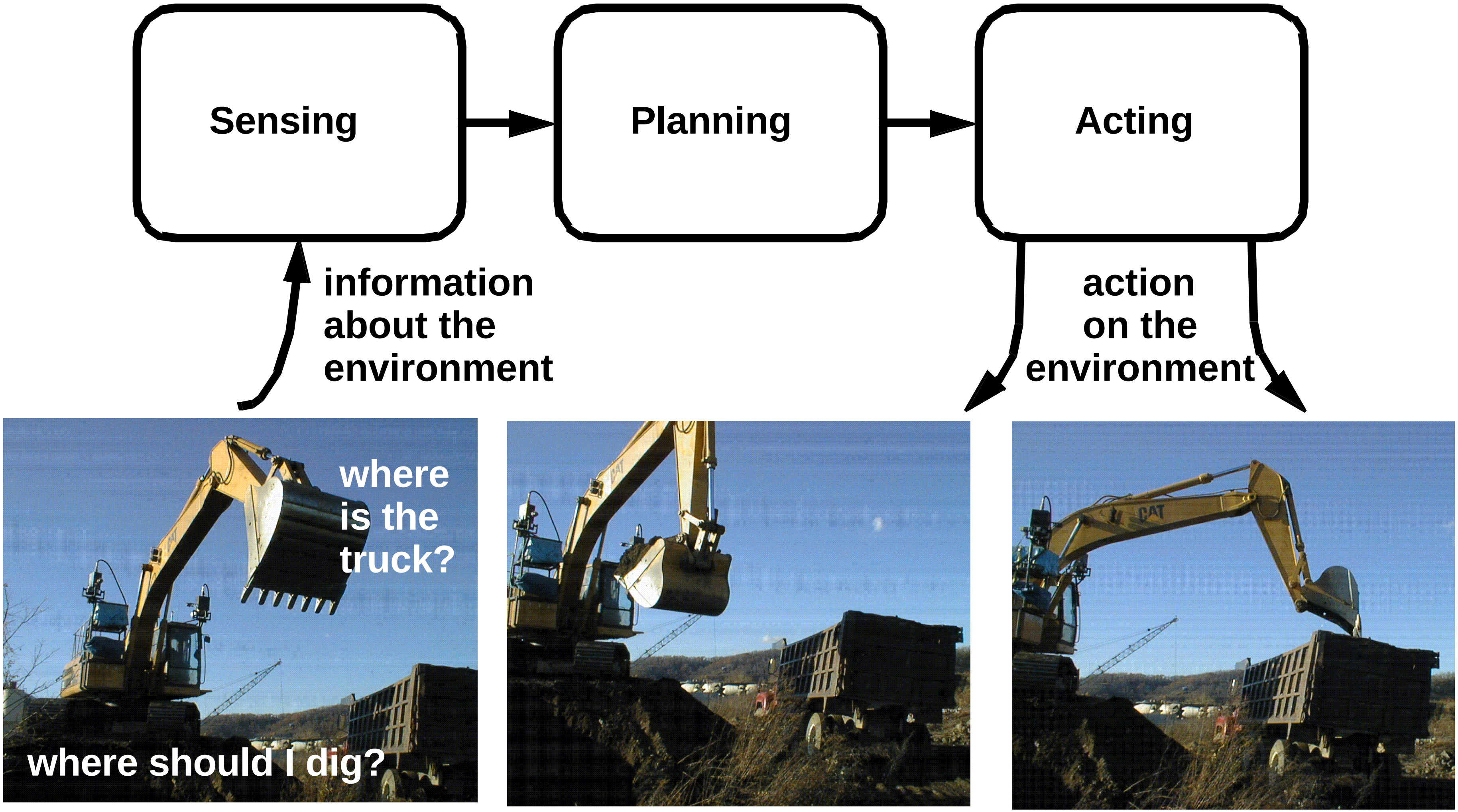
- **Hardware**-based sensors
  - Physical components built into a device
  - They derive their data by directly measuring specific environmental properties
- **Software**-based sensors
  - Not physical devices, although they mimic hardware-based sensors
  - They derive their data from one or more hardware-based sensors

Sensor List of Smartphone

Sensor	Function Type	Software-based or Hardware-based
Accelerometer	Motion Sensor	Hardware-based
Gyroscope	Motion Sensor	Hardware-based
Gravity	Motion Sensor	Software-based
Rotation Vector	Motion Sensor	Software-based
Magnetic Field	Position Sensor	Hardware-based
Proximity	Position Sensor	Hardware-based
GPS	Position Sensor	Hardware-based
Orientation	Position Sensor	Software-based
Light	Environmental Sensor	Hardware-based
Thermometer	Environmental Sensor	Hardware-based
Barometer	Environmental Sensor	Hardware-based
Humidity	Environmental Sensor	Hardware-based

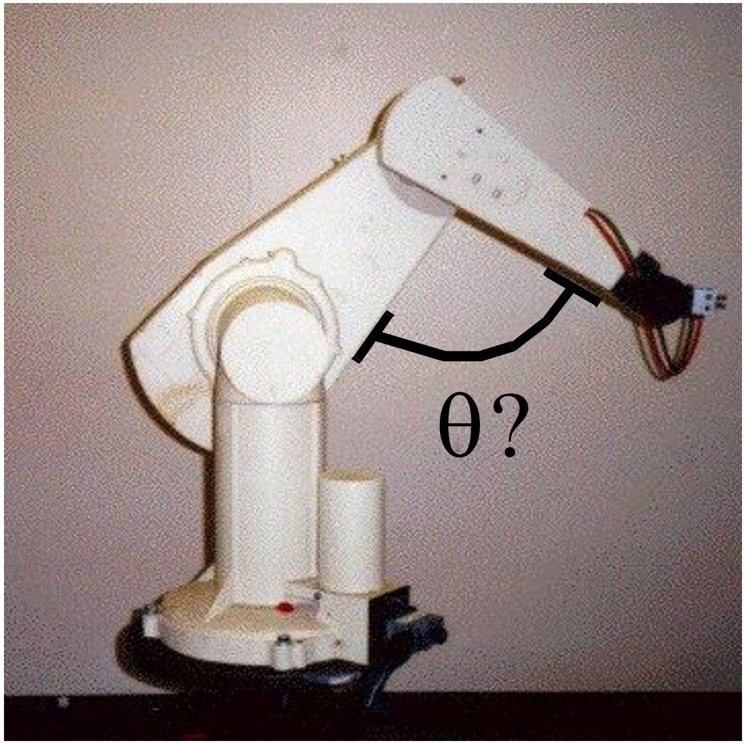
Overview of Our Sensors For Robotics

What makes a machine a robot?



Why do robots need sensors?

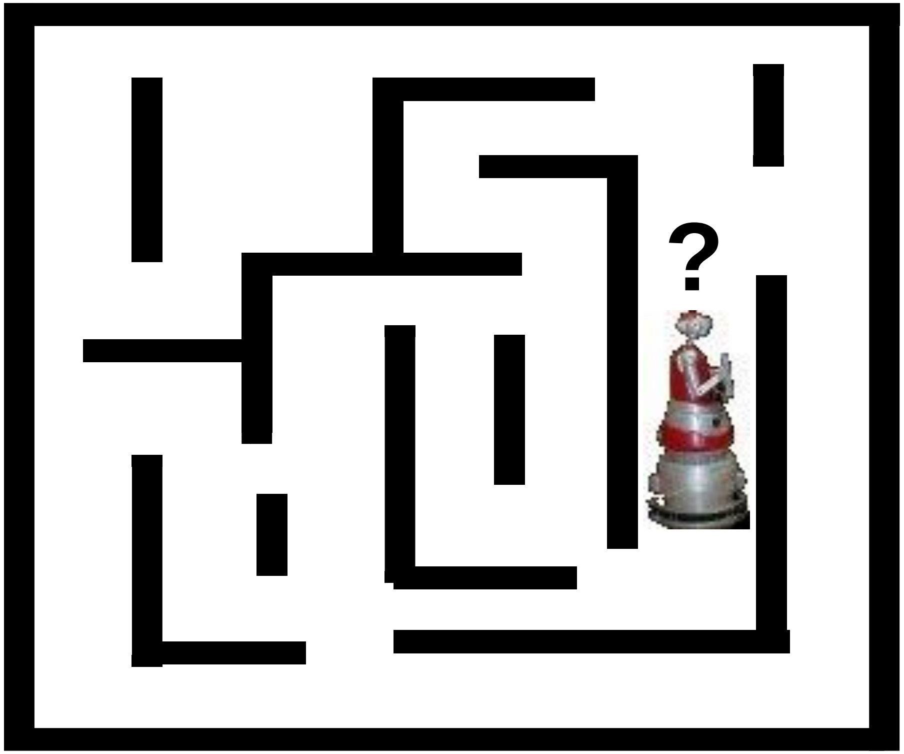
What is the angle of my arm?



internal information

Why do robots need sensors?

Where am I?



localization

Why do robots need sensors?

Will I hit anything?

obstacle detection

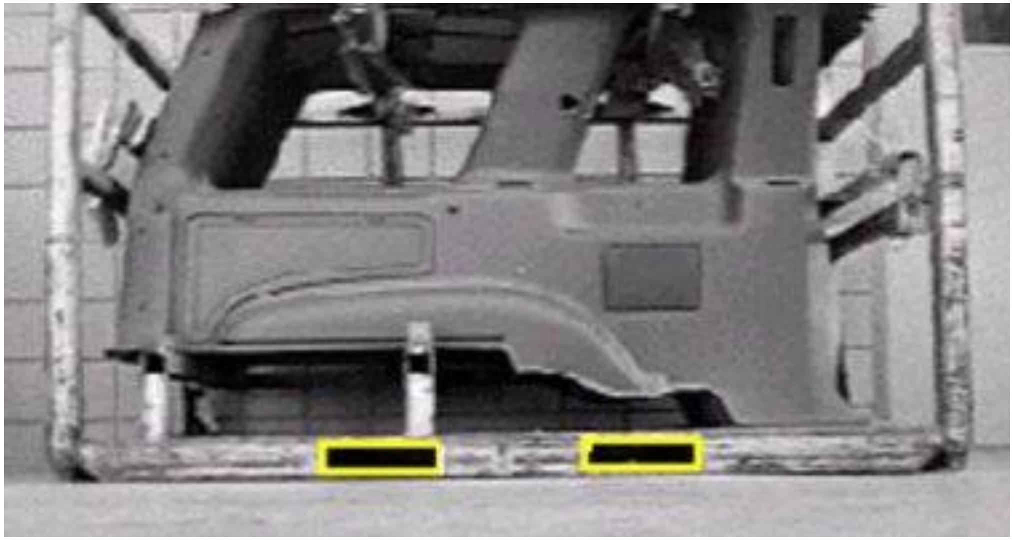
Sensing for specific tasks

Where is the cropline?

Autonomous harvesting

Sensing for specific tasks

Where are the forkholes?



Autonomous material handling

Sensing for specific tasks

Where is the face?

Face detection & tracking

Types of Sensors

- Active
  - send signal into environment and measure interaction of signal w/ environment
  - e.g. radar, sonar
- Passive
  - record signals already present in environment
  - e.g. video cameras

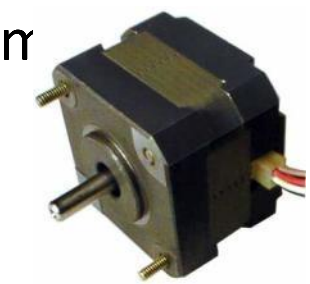
Actuators

- Hardware devices that convert a controller command signal into a change in a physical parameter
  - The change is usually mechanical (e.g., position or velocity)
  - An actuator is also a transducer because it changes one type of physical quantity into some alternative form
  - An actuator is usually activated by a low-level command signal, so an amplifier may be required to provide sufficient power to drive the actuator

Types of Actuators

1. Electrical actuators
   - Electric motors
     - DC servomotors
     - AC motors
     - Stepper motors
   - Solenoids

2. Hydraulic actuators
   - Use hydraulic fluid to amplify the controller comm
     signal



3. Pneumatic actuators
   - Use compressed air as the driving force

Contact me:

gauravsingal789@gmail.com

Gaurav.singal@nsut.ac.in

www.gauravsingal.in

LinkedIn:

https://www.linkedin.com/in/gauravsingal789/

Twitter: https://twitter.com/gaurav_singal

IoT System Architectures and Standards

COCSC20

Syllabus

This module will cover the following aspects

- Key considerations for IoT architectures
- Cloud, fog, and edge paradigms
- The role of gateways in IoT
- IoT internetworking approaches
- Standards that enable practical IoT deployment and interoperability

The IoT architectural landscape

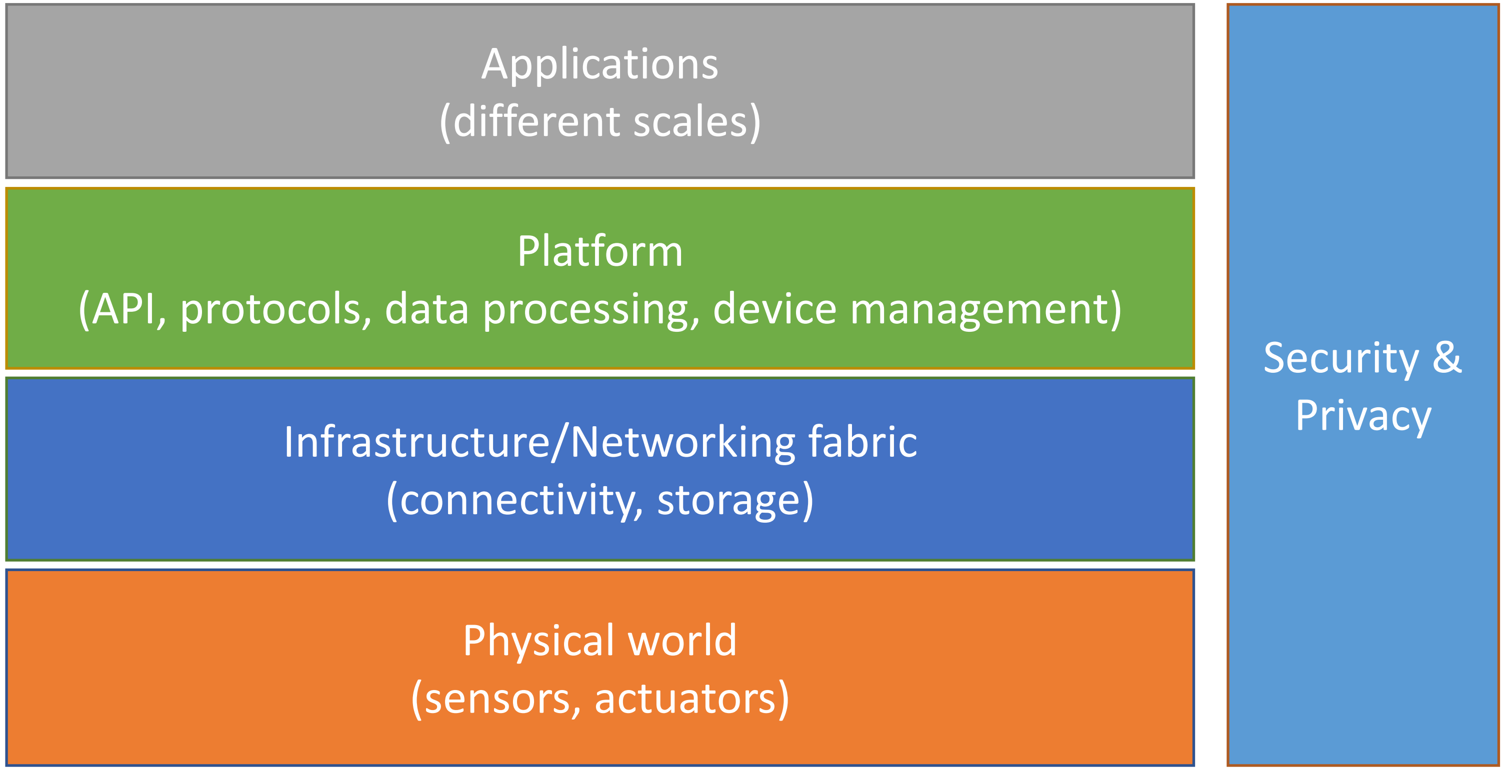
- Thousands of new applications exist, spanning countless domains (verticals).
- Each application has its unique requirements → combining these leads to systems that are complex, difficult to manage, and often proprietary.
- Defining a unified architecture is challenging and interoperability problematic, if there are too many standards to choose from.
- Efforts by multiple entities to define common frameworks, including international standardization bodies, multi-national collaborative research projects, industry consortiums, and large commercial actors.
- Device/protocol documentation is scattered and often difficult to navigate.
- We will focus on the key principles different architectural patterns share and examine some examples.

Key considerations for IoT architectures

- What application domains should be covered?
- Where to place the “intelligence”?
- What networking structure should be employed?
- How to modularize systems, so as to manage complexity and enable programmability?
- What are the cost and scalability implications?

A layered view to IoT architecture

At a high level, stakeholders may converge to a shared vision



This approach enables to break up complexity, share resources more easily, and promote interoperability

Advantages of the layer approach

- Allows IoT device manufacturers to focus strictly on improving their performance, power consumption, etc. – expose only well-defined interfaces to software platforms.

- Easier to share and partition strictly the network and computing resources (slicing); reducing the burden on service providers to build and manage networks – Infrastructure/Network as a Service (IaaS/NaaS)

- Enables software/app developers to build applications without having to understand the specifics of a device – Platform as a Service (PaaS)

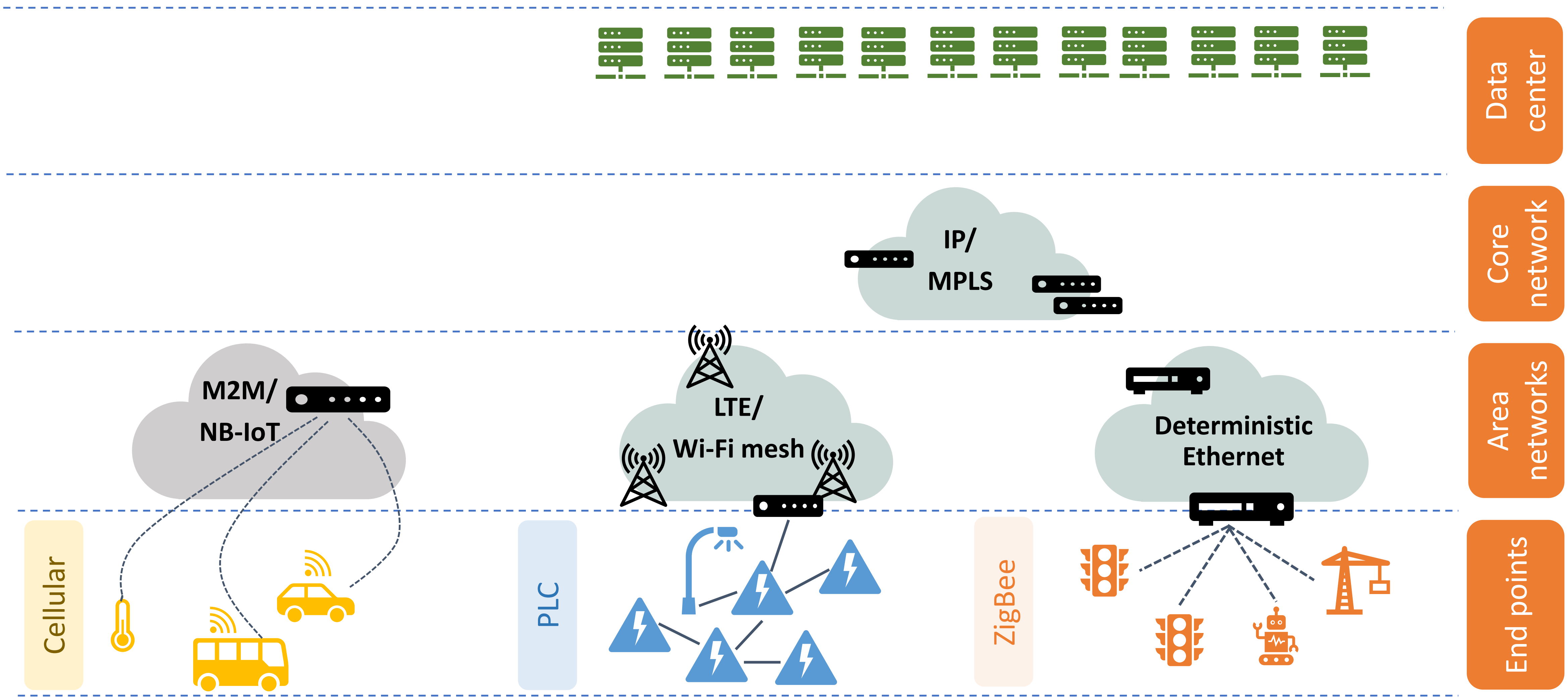
Security challenges

How to secure the entire ecosystems, from hardware to application?

- Hardware isolation (Arm TrustZone)
- Middleware (Speculative Store Bypass Barrier – SSBB)
- Network isolation (Software-defined Networking – SDN)
- Data confidentiality in transit (Transport Layer Security – TLS)
- Software isolation (containers)

End-to-end security not straightforward

A practical network-centric view



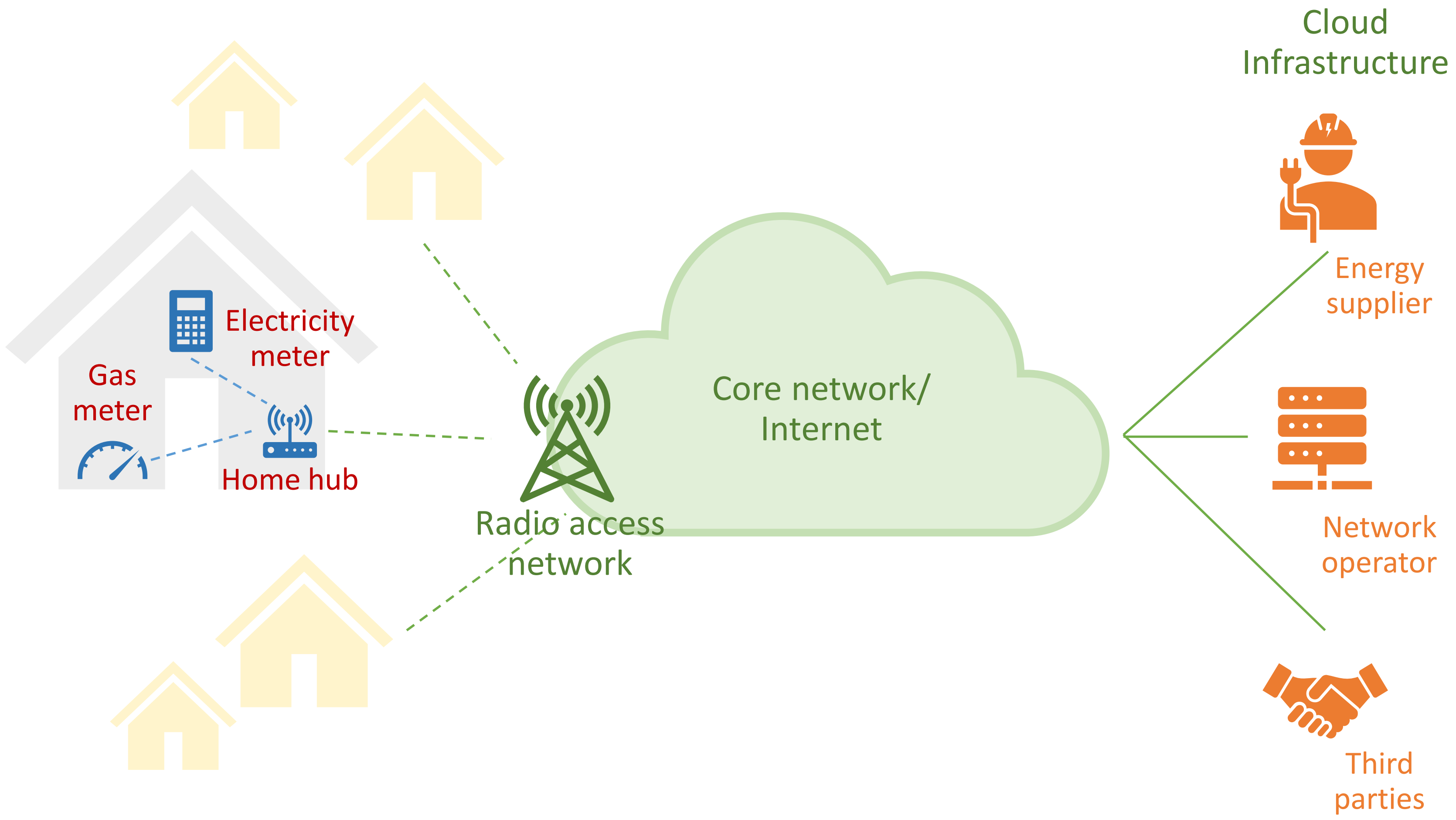
Cloud vs. Fog vs. Edge

The information processing view

Cloud computing

- Cloud dominated the networked systems landscape until recently
- All intelligence on powerful servers, including relational databases, control functions, data analytics engines, web interfaces, etc.
- End devices merely information gatherers
- Might not scale as the number of IoT devices grows, and applications continue to diversify and generate more data

Example: Smart metering



- Sensing performed by simple sensors
- Information relayed by home hub over cellular network
- Data processed in the cloud by different stakeholders

Cloud vs. Fog vs. Edge

The information processing view

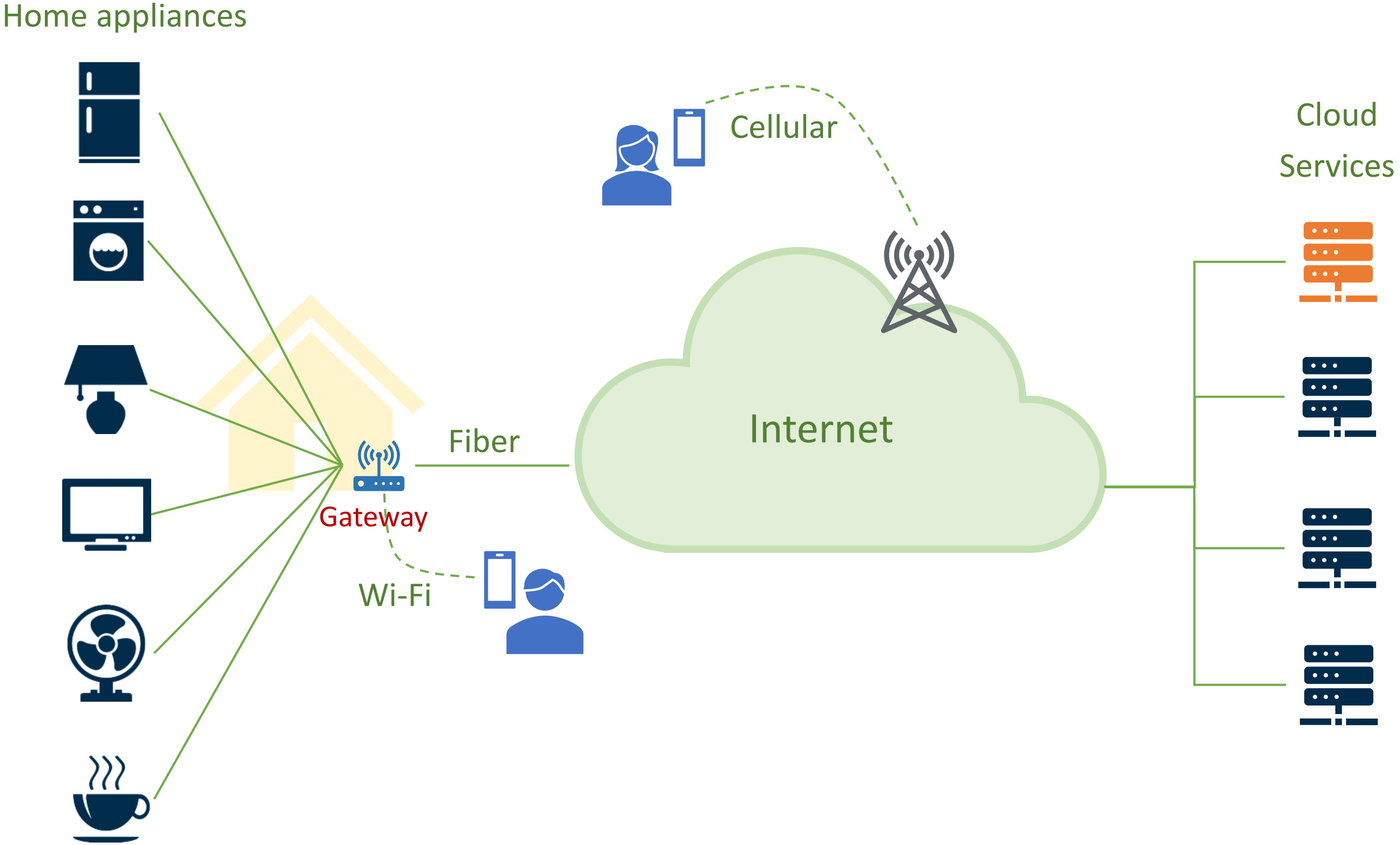
- Pushing some of the intelligence closer to the device, for e.g., to access networks or gateways
- This includes data aggregation, compression, (partial) processing, making localized decisions
- IoT devices kept simple, no direct communication with end servers, still battery powered
- Resource management implemented across different network layers – management could be regarded as an application

Fog computing

The role of gateways in fog architectures

- Data filtering and processing (for e.g., aggregation of summaries, compression, etc.)
- Protocol translation and interfacing between different connectivity technologies
- Data flow multiplexing, packet routing
- Security (for e.g., data encryption, firewalling)
- Scalability problem: as the number of devices grows, so will the number of gateways that are required

Example: Home automation



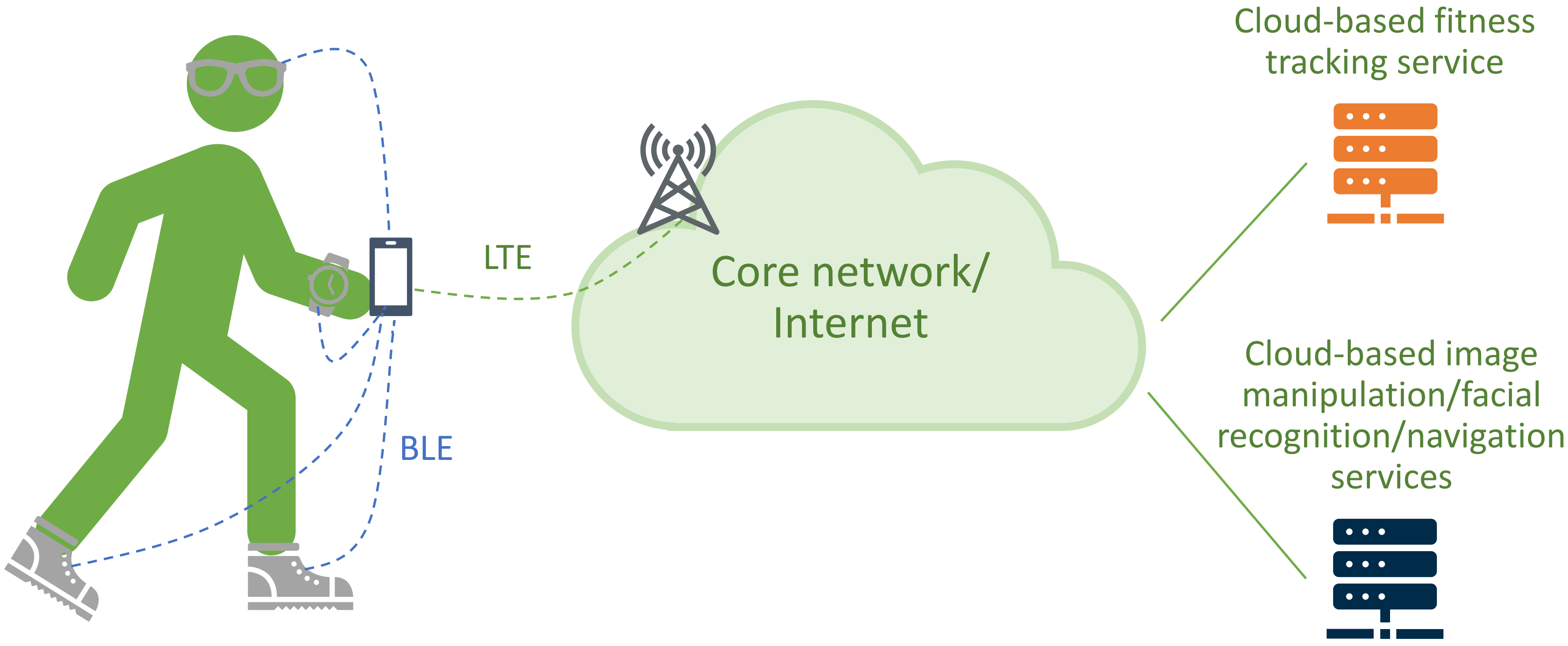
- Gateway performs protocol translation
- Incorporates basic network intrusion detection system
- Cloud services continue to perform analytics

Smartphones as gateways

The fitness and healthcare domain

- Embed multiple networking technologies (Wi-Fi, 3G/4G, Bluetooth/BLE, NFC, etc.)
- Run full TCP/IP stacks, thus maintain end-to-end connectivity with cloud
- Can connect to multiple devices within close proximity simultaneously
- Ability to enforce secure transport (e.g., TLS/HTTPS)
- Sufficient computing power to pre-process/augment collected data

Example: Wearables



- Smartphone communicates over BLE with wearable devices
- Performs minimal information pre-processing
- Relays data to cloud-based services

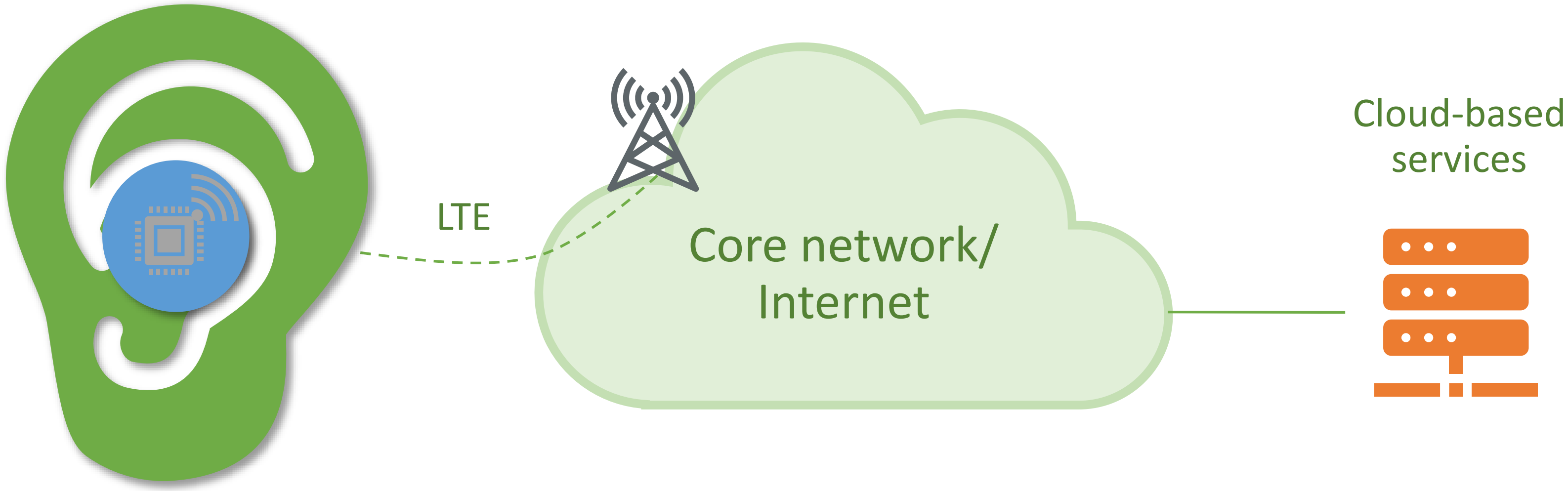
Cloud vs. Fog vs. Edge

The information processing view

Edge computing

- Pushing compute power, communication capabilities, intelligence down at device level
- Processing as much as possible where data is collected
- Transmitting only key information or summaries
- Enabling new applications: automotive IoT, virtual/augmented reality, in-ear computing

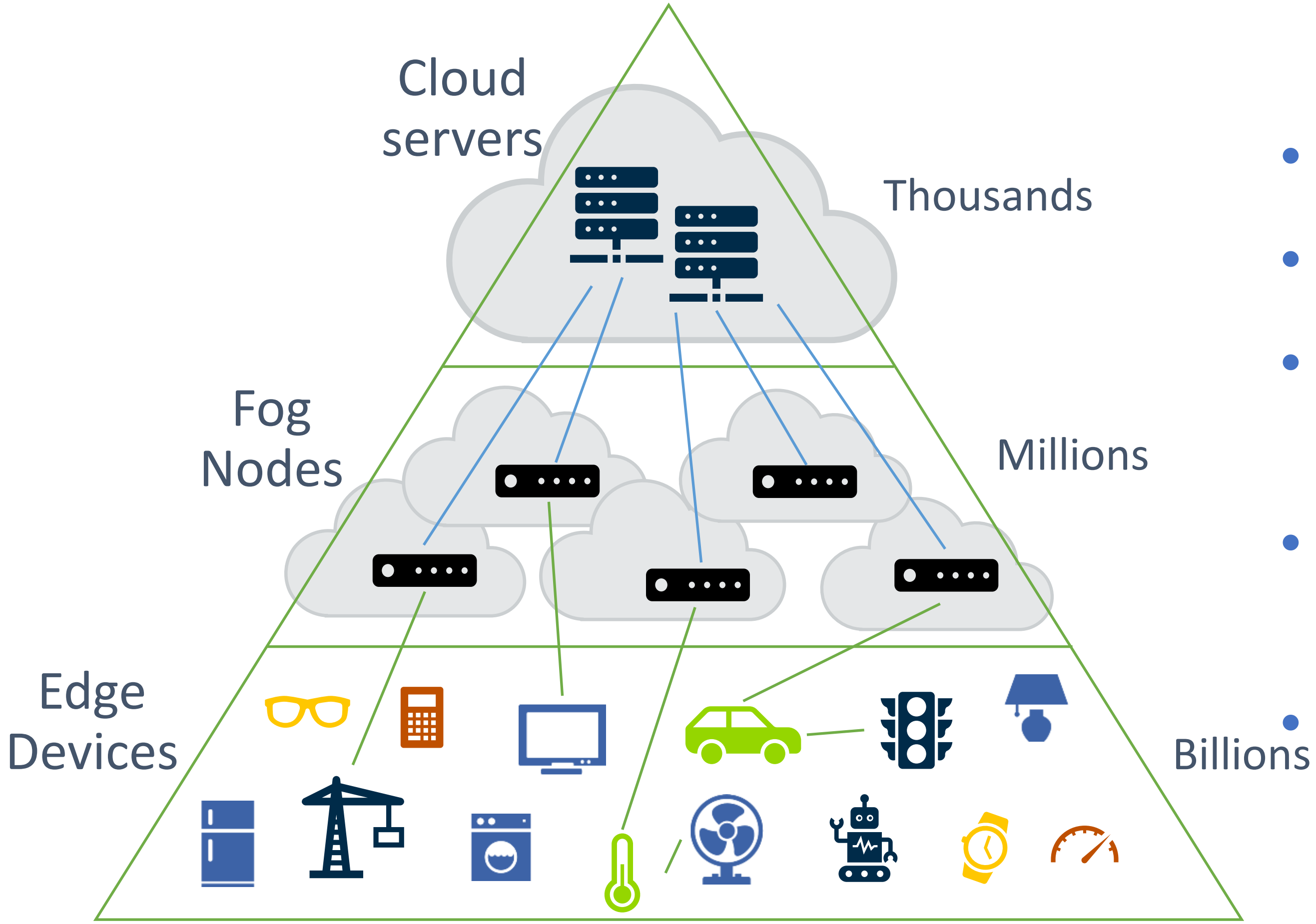
Example: Hearables



- **Hardware:** Low-power chips specialized in computationally intensive tasks (Arm Ethos)
- **Software:** AI libraries optimized for constrained devices (uTensor)
- **Neural networks:** compressed/pruned models

Choosing the right IoT architecture

Performance and cost remain the dominant architectural drivers



- What are the application requirements?
- What data needs to be acted on locally?
- Where is most of the computing power?
- How much networking infrastructure should be deployed/used?
- Where are the trust boundaries?

Standards for IoT

Multiple regulation bodies and industry alliances are standardizing the means by which devices can interact with each another and with gateways or cloud services.

The Institute of Electrical and Electronics Engineers (IEEE)

- Primarily dealing with defining protocols for (wireless) access networks
- Targeting the Industrial, Scientific, and Medical (ISM) bands (e.g., 2.4GHz, 5GHz, 900MHz in some regions, etc.)
- From an IoT perspective, the most relevant technologies include
  - IEEE 802.15.4, on which ZigBee builds, and
  - IEEE 802.11ah (HaLow) that is an amendment to the IEEE 802.11 specification (typically used for Wi-Fi) that enables low-power wide-area networking

Standards for IoT

Multiple regulation bodies and industry alliances are standardizing the means by which devices can interact with each another and with gateways or cloud services.

The 3rd Generation Partnership Project (3GPP)

- Focuses on specifying cellular network architectures and protocols (e.g., GSM, 3G, 4G-LTE, etc.)
- Developing standards for cellular communications tailored to IoT applications
  - LTE-M – compatible with existing LTE networks, easy to roll out, limited to 1Mb/s speeds
  - NB-IoT – deployed in same or different frequency bands, lower capacity (200Kb/s), different modulation and coding schemes, and does not require gateways.

Standards for IoT

The Internet Engineering Task Force (IETF)

- Focuses on specifying protocols that are used across the Internet; these standards are known as Requests for Comments (RFCs)

- IoT relevant standards include
  - Addressing/internetworking for low power devices (IPv6 over Low-Power Wireless Personal Area Networks – 6LoWPAN)
  - Routing (Routing Over Low-power and Lossy networks – ROLL)
  - End-to-end communications (Constrained Application Protocol – CoAP)
  - Security (Datagram Transport Layer Security – DTLS)
  - Software updating (Software Updates for Internet of Things – SUIT)

- Also offers experience-based guidance
  - Example: The JavaScript Object Notation (JSON) Data Interchange Format – RFC 8259

Standards for IoT

Industry alliances

- Bluetooth – wireless personal area networks (WPANs); defines application profiles
- ZigBee – WPANs building on IEEE 802.15.4; inexpensive consumer/industrial applications
- LoRaWAN – LPWAN based on chirp spread spectrum technology

Collaborative associations

- The Alliance for IoT Innovation (AIoTI) – European Commission framework supporting interaction between IoT players to drive innovation, standardization, and policy.
- Open Connectivity Foundation (OCF) – Industry-led framework aiming to develop IoT standards, interoperability guidelines, and provide a device certification program.

Standards for IoT

Other standardization bodies relevant to IoT

- National Institute of Standards and Technology (NIST) – works on a range of science, technology, and engineering topics
  - Example: Advanced Encryption Standard (AES)

- International Organization for Standardization (ISO) – promotes a broad range of proprietary, industrial, and commercial standards
  - Example: Internet of Things (IoT) – Reference Architecture (ISO/IEC 30141:2018)

- International Telecommunication Union (ITU) – recommendations, reference models
  - Example: ITU-T Y.4000/Y.2060 - Overview of the Internet of things

Choosing among IoT standards is not straightforward

Source: AIOTI WG3 (IoT Standardisation) – Release 2.8

Thank You

Contact me:

gauravsingal789@gmail.com

Gaurav.singal@nsut.ac.in

www.gauravsingal.in

LinkedIn: https://www.linkedin.com/in/gauravsingal789/

Twitter: https://twitter.com/gaurav_singal

Computer Network Recap

Internet-of-Things (IoT)

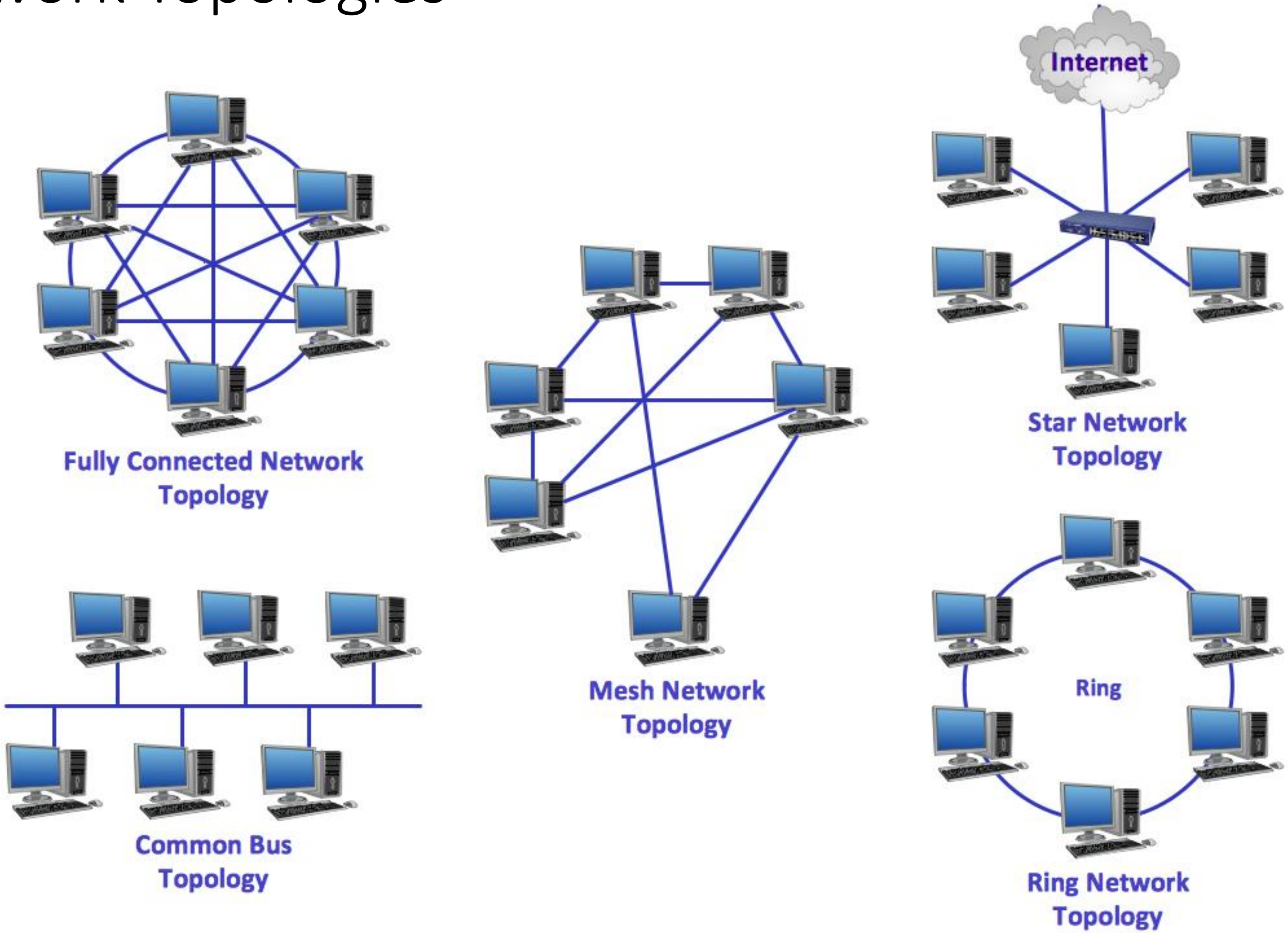
COCOS20

University of Notre Dame

Computer Network Terminology

- **Network:** group of computers and associated devices that are connected by communication facilities
- **Wide Area Network (WAN)**: world-wide (Internet)
- **Metropolitan Area Network (MAN):** city-scale.
- **Local Area Network (LAN):** laboratory/office-scale (Ethernet).
  - **WLAN:** wireless LAN (Wi-Fi).
  - **WPAN:** wireless personal area network (Bluetooth).
  - **WBAN:** wireless body area network.

Network Topologies



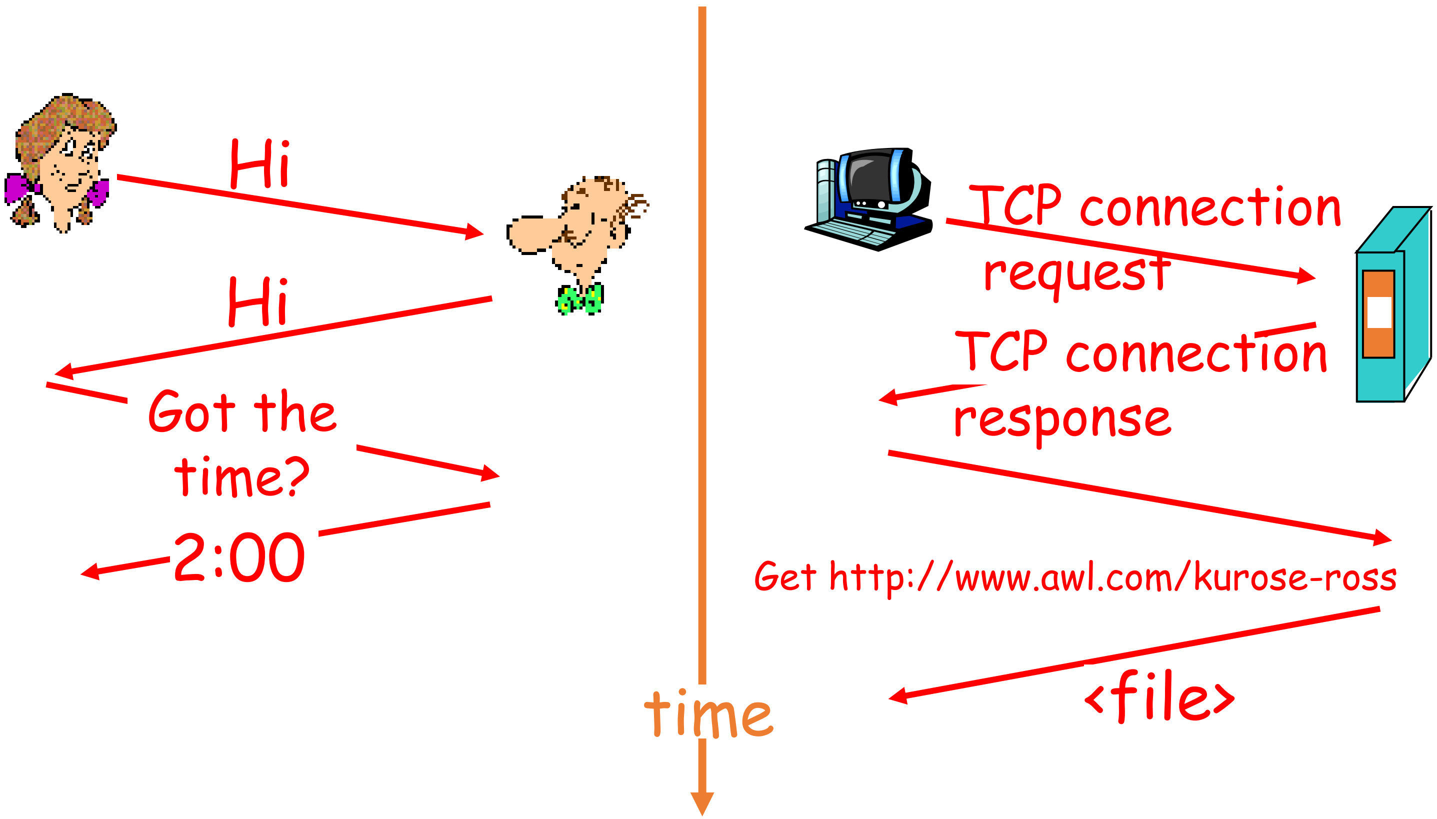
Network Protocols

- Protocols are the **building blocks** of a network architecture.
- Formal standards and policies enabling communication.
- IEEE (Institute of Electrical and Electronics Engineers): standardization
  - Example: Project 802
    - 802.3: Ethernet
    - 802.11: WLAN
    - 802.15: WPAN

Communication

- Who initiates communication?
- Order of communication?
- How long can I talk?
- How loud can I speak?
- Do I have to say something specific at beginning or end?
- Do I have to add meta information?
- What do I do if I get interrupted?
- What do I do if I was not understood?

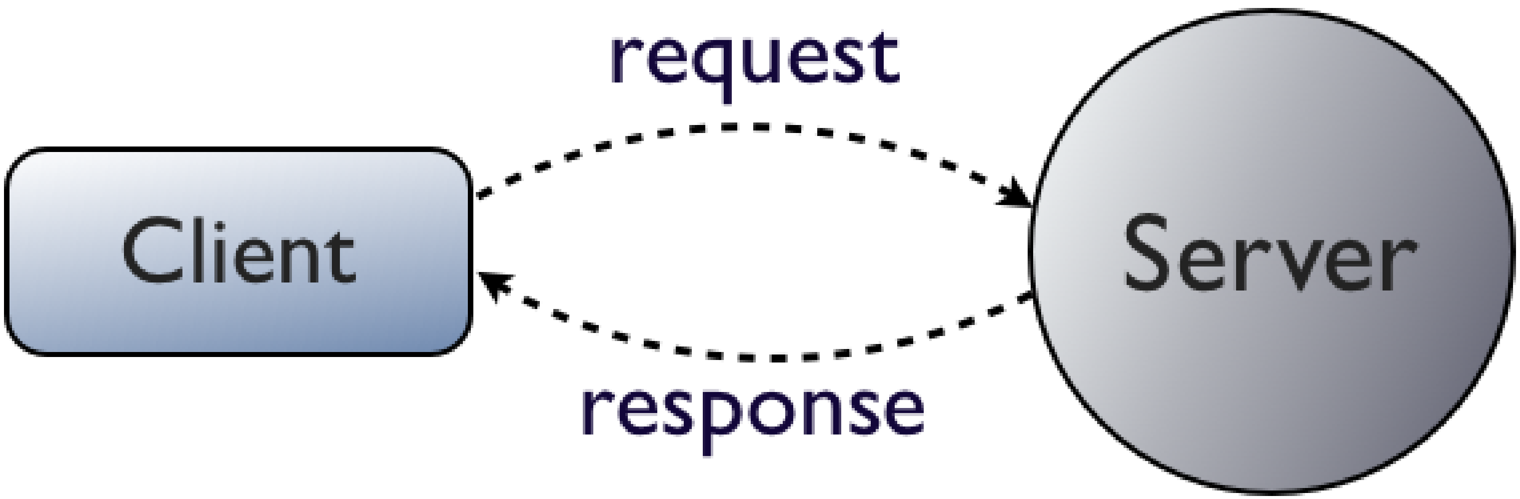
Protocols



Hi
Hi
Got the time?
2:00
TCP connection request
TCP connection response
Get http://www.awl.com/kurose-ross
<file>
time

Client/Server Model

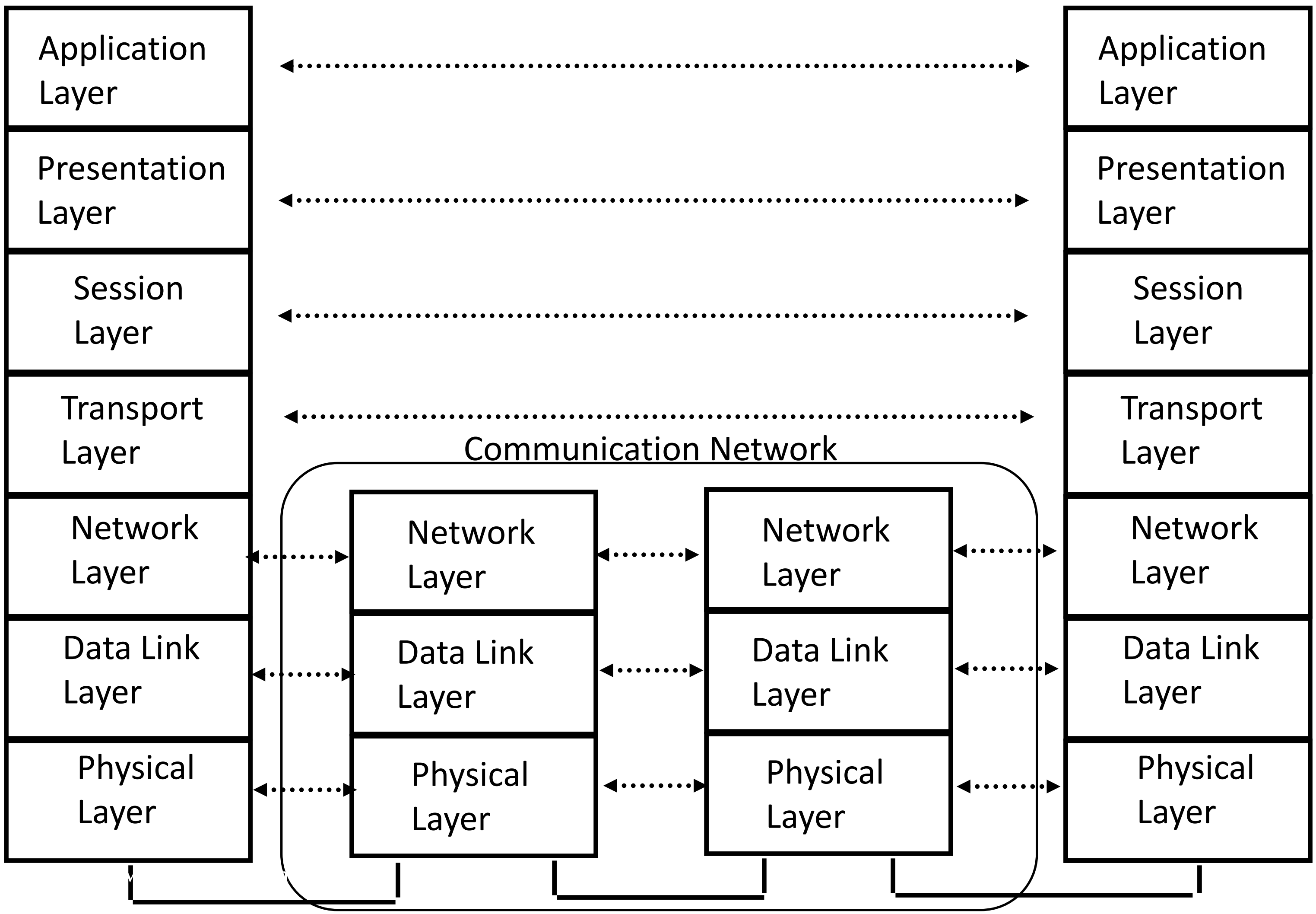
- Client: “active” (initiates communication)
- Server: “passive” (listens and responds)



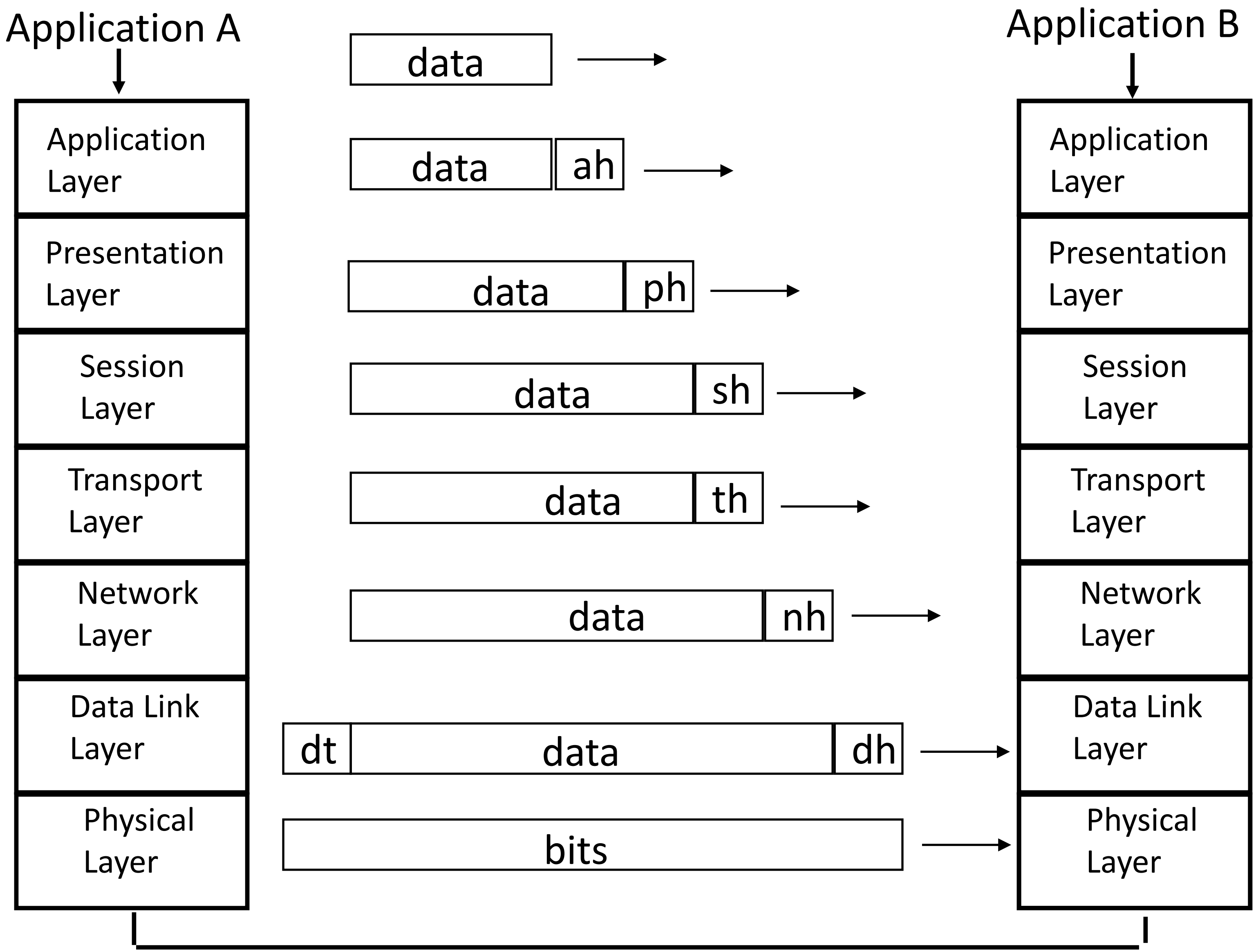
Client/Server Model Examples

- HTTP (Hypertext Transfer Protocol)
- SMTP (Simple Mail Transfer Protocol)
- SSH (Secure Shell)
- DNS (Domain Name System)
- NFS/AFS (Network/Andrew File System)

Network Protocols ("Protocol Stack")



Network Protocols (Headers/Trailers)



Why a Layered Design?

- An explicit structure for dealing with a complex system
- Simplifies the design process
- Modularity of layers eases maintenance and updating of system components
- Accommodates incremental changes

Open System Interconnection (OSI)

UPPER LAYERS	7	**Application Layer**	✓ Message format, Human-Machine Interfaces
	6	**Presentation Layer**	✓ Coding into 1s and 0s; encryption, compression
	5	**Session Layer**	✓ Authentication, permissions, session restoration
TRANSPORT SERVICE	4	**Transport Layer**	✓ End-to-end error control
	3	**Network Layer**	✓ Network addressing; routing or switching
	2	**Data Link Layer**	✓ Error detection, flow control on physical link
	1	**Physical Layer**	✓ Bit stream: physical medium, method of representing bits

Physical Layer (Layer 1)

- **Physical/electrical characteristics**
- Cable type, length, connectors, voltage levels, signal durations, ...
- Binary data (bits) as electrical or optical signals
- Frequencies (wireless)

Wireless Characteristics

- VLF = Very Low Frequency
- LF = Low Frequency
- MF = Medium Frequency
- HF = High Frequency
- VHF = Very High Frequenc

UHF = Ultra High Frequency

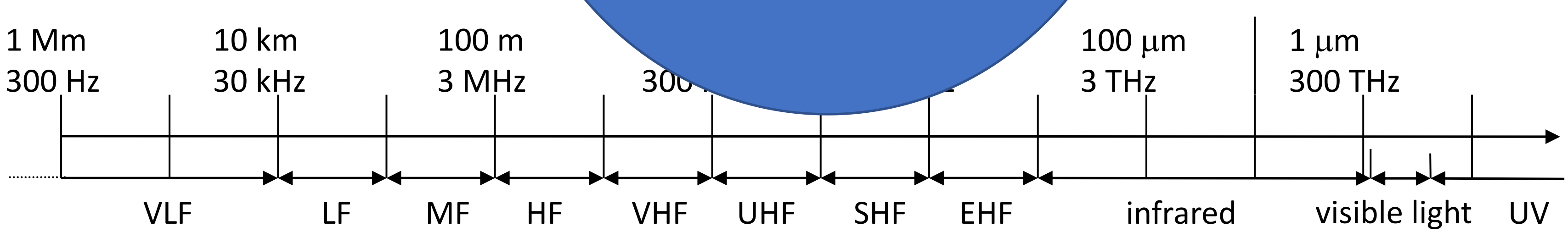
r High Frequency

h Frequency

ight

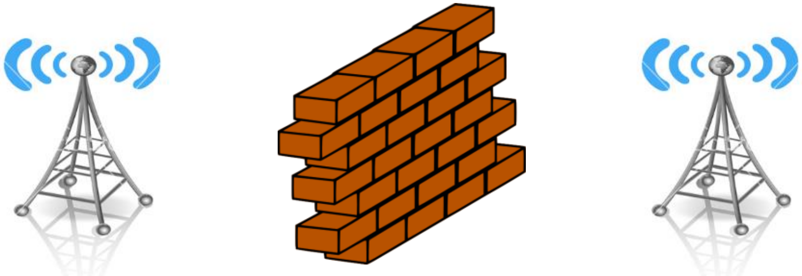
What is Frequency?

- Frequency and wave le
  - $\lambda = c/f$
  - wave length λ, speed of



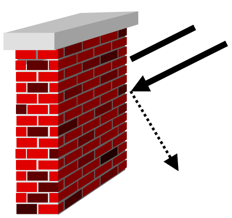
Frequencies for Mobile Communication

- Low Frequencies:
  - low data rates
  - travel long distances
  - follow Earth's surface
  - penetrate objects and water (submarine communication)
- High Frequencies:
  - high data rates
  - short distances
  - straight lines
  - cannot penetrate objects ("**Line of Sight**" or **LOS**)

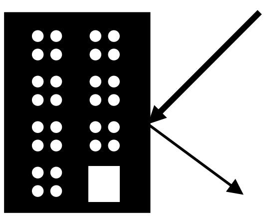


Other Propagation Effects

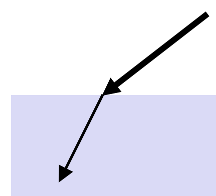
- **Shadowing**
- **Reflection** at large obstacles
- **Refraction** depending on the density of a medium
- **Scattering** at small obstacles
- **Diffraction** at edges



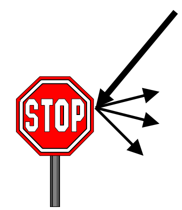
shadowing



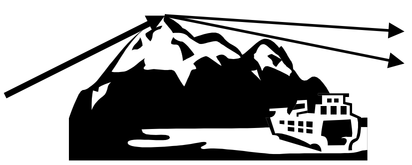
reflection



refraction



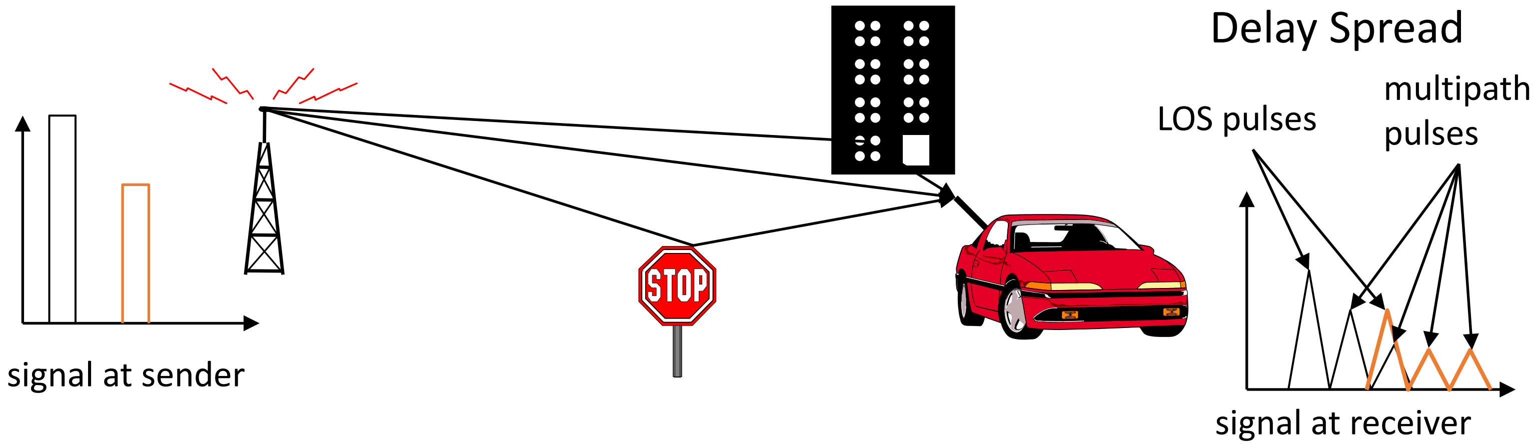
scattering



diffraction

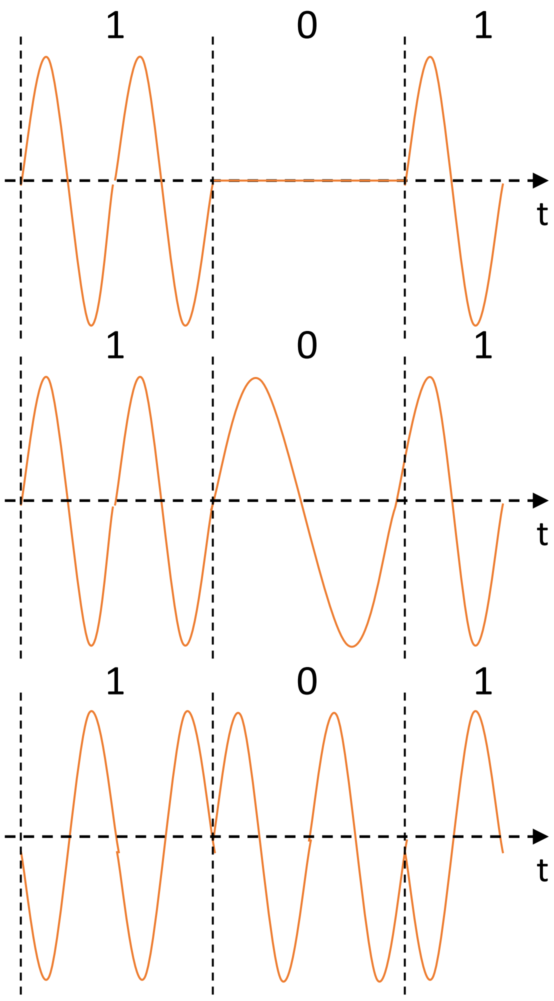
Multipath Propagation

- Signal can take **many different paths** between sender and receiver due to reflection, scattering, diffraction



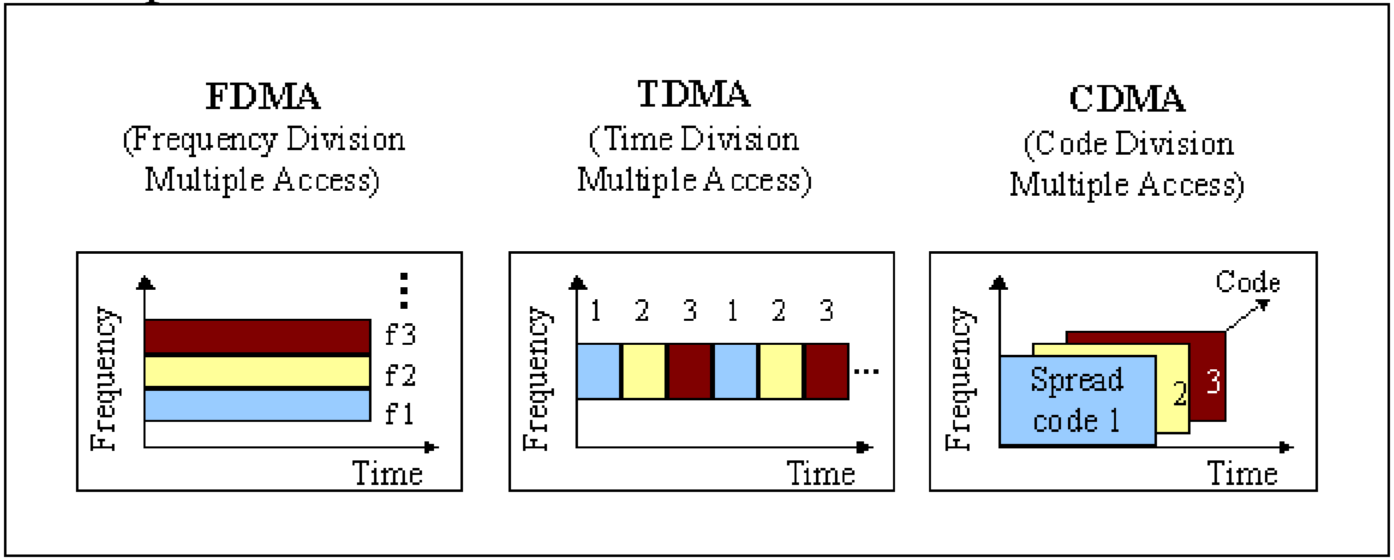
Digital Modulation

- **Amplitude Shift Keying (ASK)**
- **Frequency Shift Keying (FSK)**
- **Phase Shift Keying (PSK)**



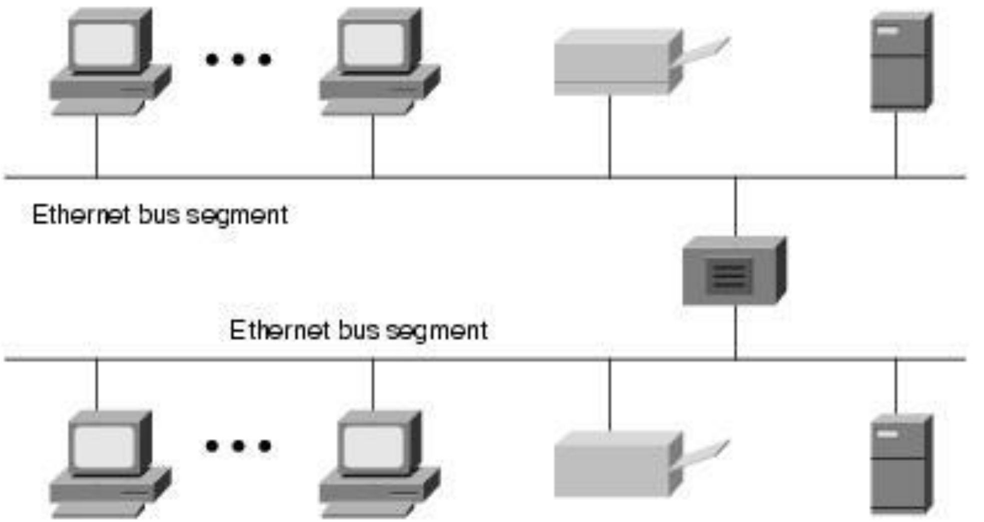
Data Link Layer (Layer 2)

- **Defines when/how medium will be accessed for transmission**
- Units typically called “frames”; error detection/correction; divided into sublayers, including: **MAC = Medium Access Control** (MAC address 6f:00:2b:23:1f:32)
- Cell phone example:



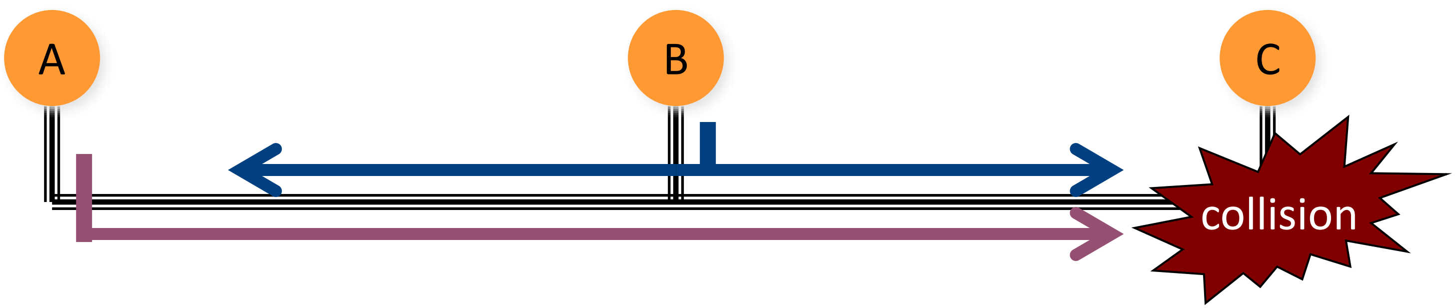
Example: Ethernet (802.3)

- Most popular LAN technology, uses bus architecture
- Easy to install, inexpensive
- Data is broken into **packets**

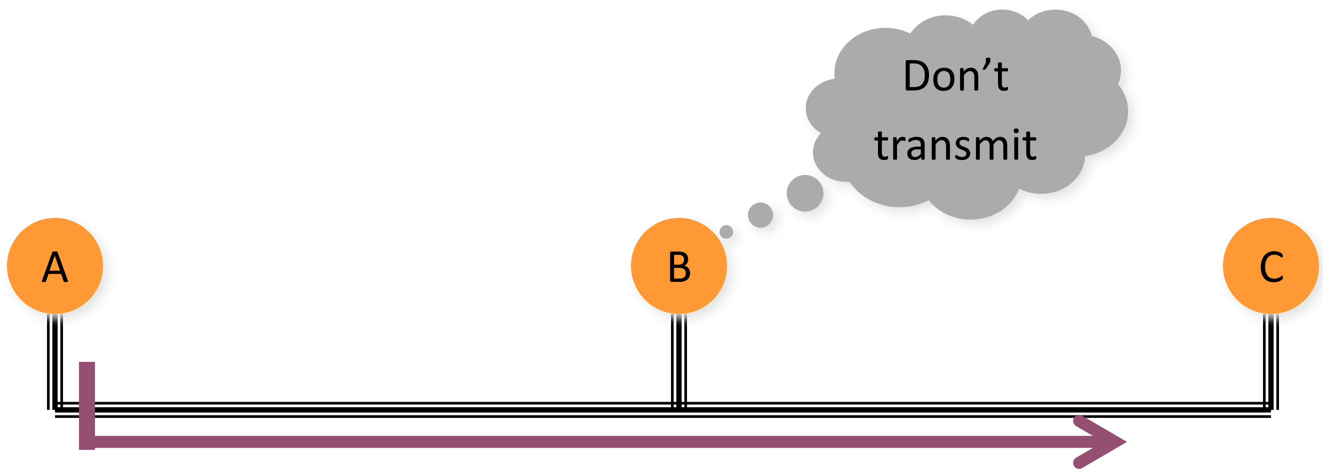


Example: Ethernet

- Medium Access Control (MAC) protocol
- **CSMA/CD** Protocol
  - **C**arrier **S**ense
  - **M**ultiple **A**ccess
  - **C**ollision **D**etection



Example: Ethernet

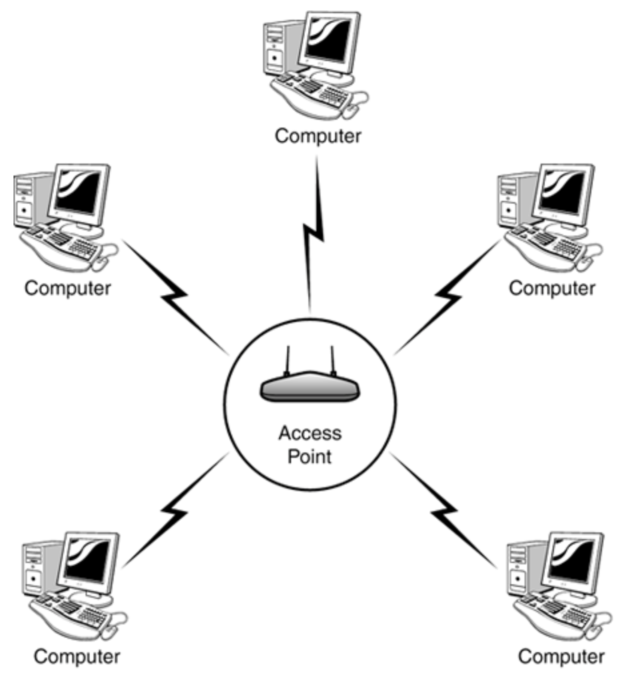


Can collisions still occur?

- "Sense" (listen) carrier ("is anyone else talking right now?")
- If "busy": wait; if "idle": transmit
- CD: Keep listening while transmitting
  - If collision detected: retry at a later time

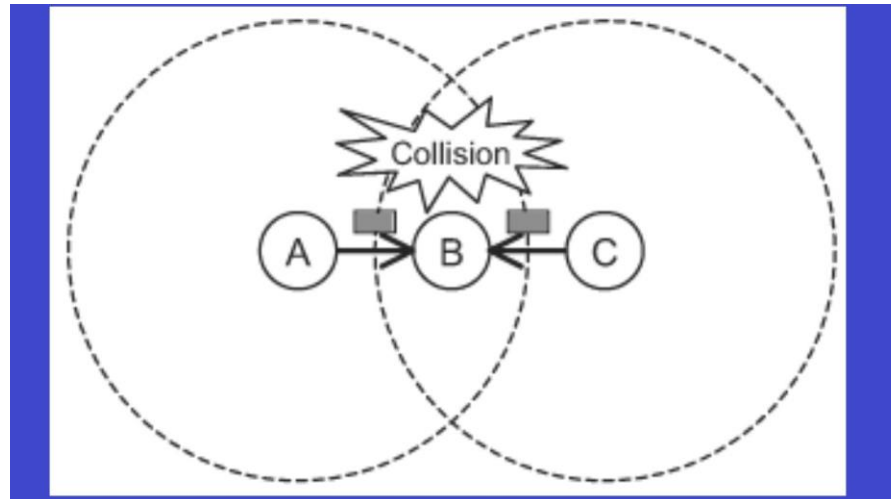
Example: Wi-Fi (802.11)

- Most popular wireless LAN architecture



Access point
Wi-Fi router
Base station
Hotspot

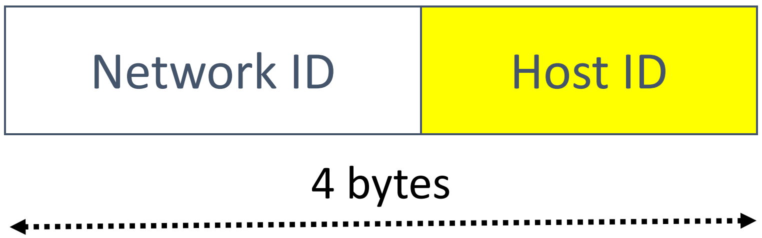
Example: Wi-Fi (802.11)



- **CSMA/CA** Protocol
  - **C**arrier **S**ense
  - **M**ultiple **A**ccess
  - **C**ollision **A**voidance
    - Channel reservations:
      - Transmitter sends request-to-send (RTS)
      - Receiver sends clear-to-send (CTS)
    - Advantages:
      - Nodes hearing RTS and/or CTS keep quiet
      - If collision, only small RTS or CTS packets are lost.

Network Layer (Layer 3)

- **Dominant protocol: IP = Internet Protocol**
- Addressing and routing (sender & receiver IP address)
- Uses 32-bit **hierarchical address space** with location information embedded in the structure



- IPv4 address is usually expressed in dotted-decimal notation, e.g.:

128.100.11.56

IPv4

Class A
Subnet Mask

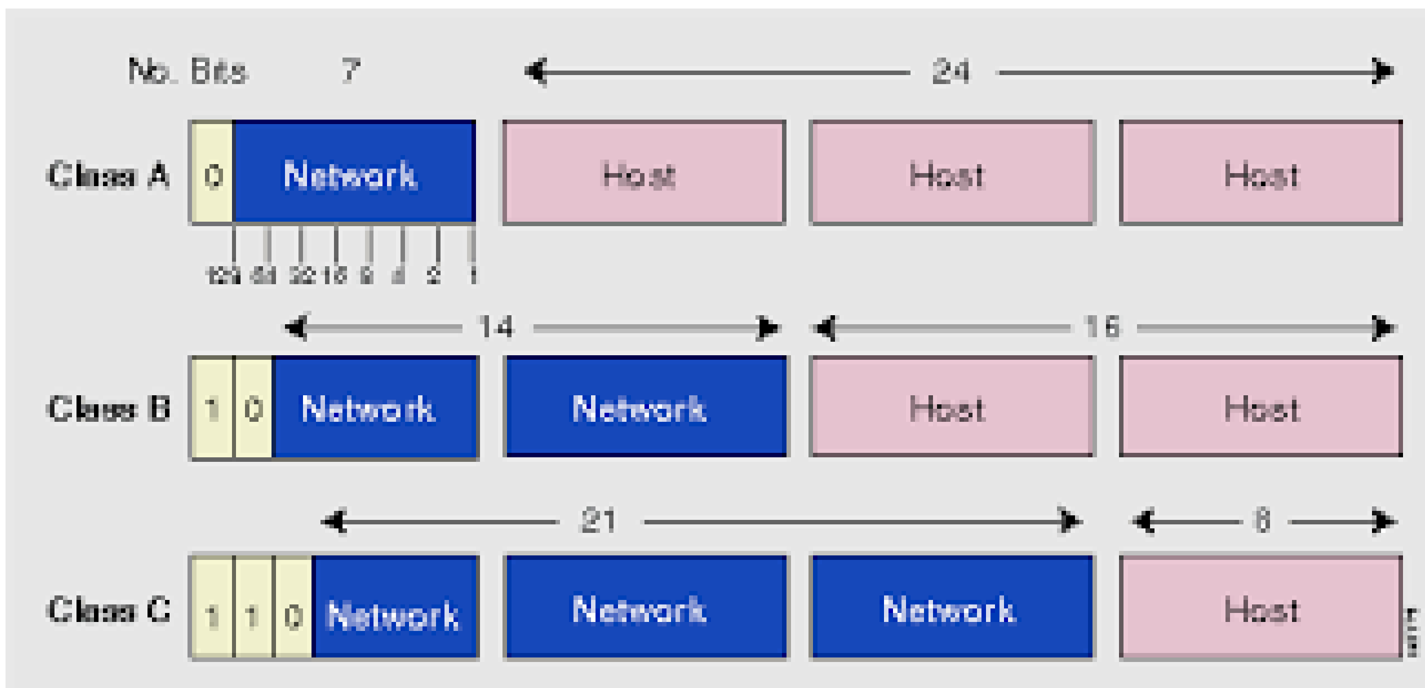
Netwok	Host	Host	Host
255	0	0	0

Class B
Subnet Mask

Netwok	Network	Host	Host
255	255	0	0

Class C
Subnet Mask

Netwok	Network	Network	Host
255	255	255	0



IPv6

- IPv6 addresses are 128 bits long
- 16 bytes of IPv6 address are represented as a group of hexadecimal digits, separated by colons, e.g.:

  2000:fdb8:0000:0000:0001:00ab:853c:39a1
- Shorthand – leave out groups of zeros and leading zeros:

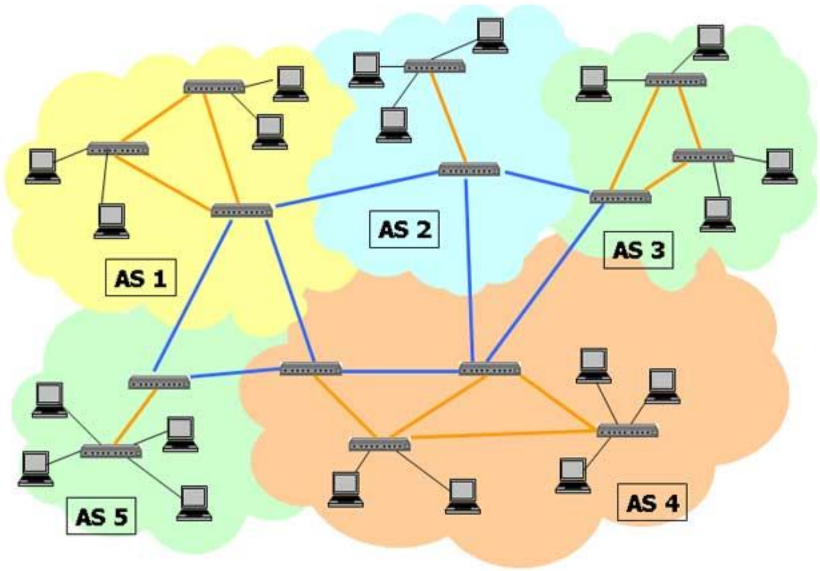
  2000:fdb8:::1:ab:853c:39a1

- IPv4 Address space: 4,294,967,296 Addresses
- IPv6 Address space: ($3.4 * 10^{38}$)

340,282,366,920,938,463,463,374,607,431,768,211,456

Routers

- Form backbone of the Internet
- Use IP layer to identify source and destination of packets
- Look up **routing tables** that determines **“next hop”**



Destination	Next Hop
147.39.21.X	131.19.18.121
89.44.X.X	131.19.22.119
203.21.X.X	137.18.47.48

Transport Layer (Layer 4)

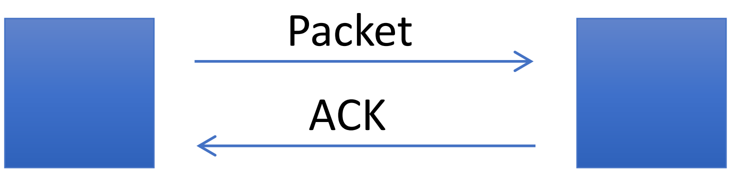
- **UDP** (User Datagram Protocol)

UDP Header	Data

- Adds more addressing: "**ports**"
  - IP address tell you which computer
  - Ports tell you which application on that computer
  - Example: a web server "listens" to requests on port 80
  - Web browser: http://www.google.com:80 = http://216.58.216.100:80
    - ":80": optional
  - **Unreliable!**
    - Packets can get lost; packets can arrive out of order

Transport Layer

- **TCP** (Transmission Control Protocol)
- **Reliable** protocol!
- Adds ports (just like UDP), but also provides:
  - In-order delivery of packets (using sequence numbers)
  - Reliable delivery: using acknowledgment (ACK) packets



  - **Flow control & congestion control:**
    - Allows receiver to slow down sender
    - Allows “network” to slow down sender

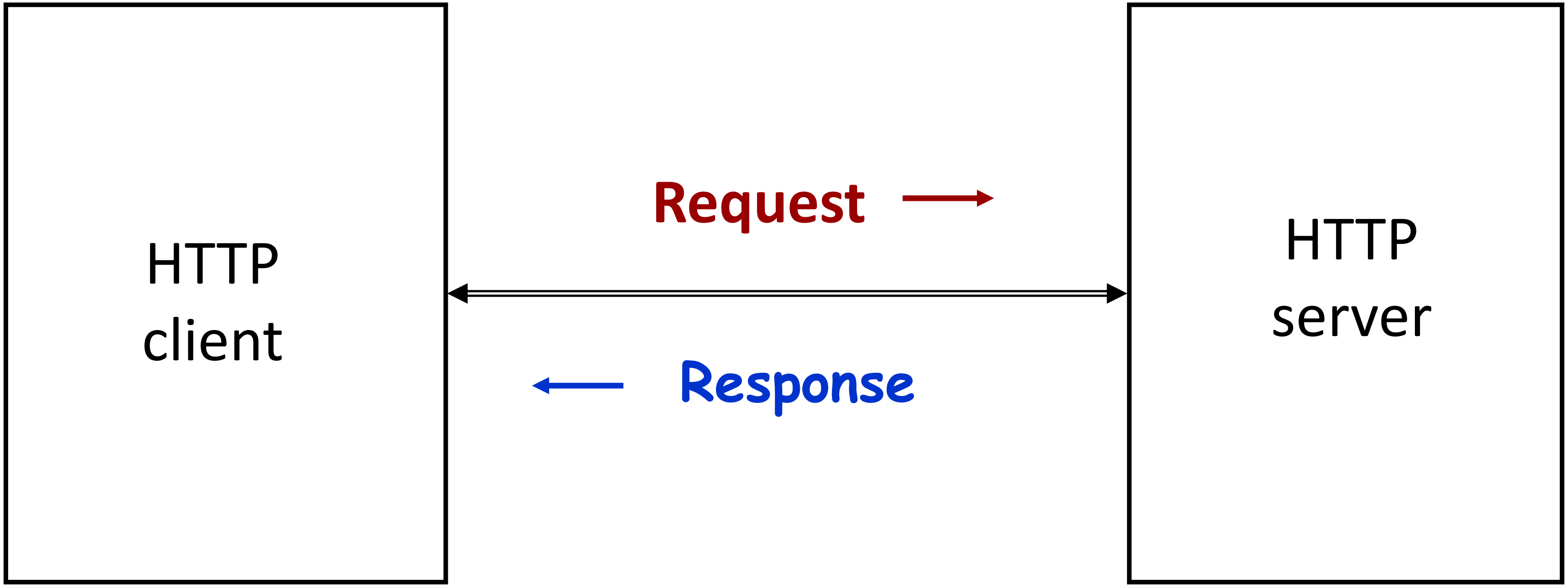
UDP vs TCP

- TCP:
  - typical choice of most applications
  - do not want to lose data, out-of-order arrival, etc.
  - email, web traffic, financial transactions, etc.
- UDP:
  - can be “faster”
    - no flow/congestion control “slowing down” traffic
    - no retransmissions
    - good for “real-time” traffic
  - out-of-order arrival: can also “reorder” at application level
  - loss of data: can be acceptable
    - missing frames in video/audio stream

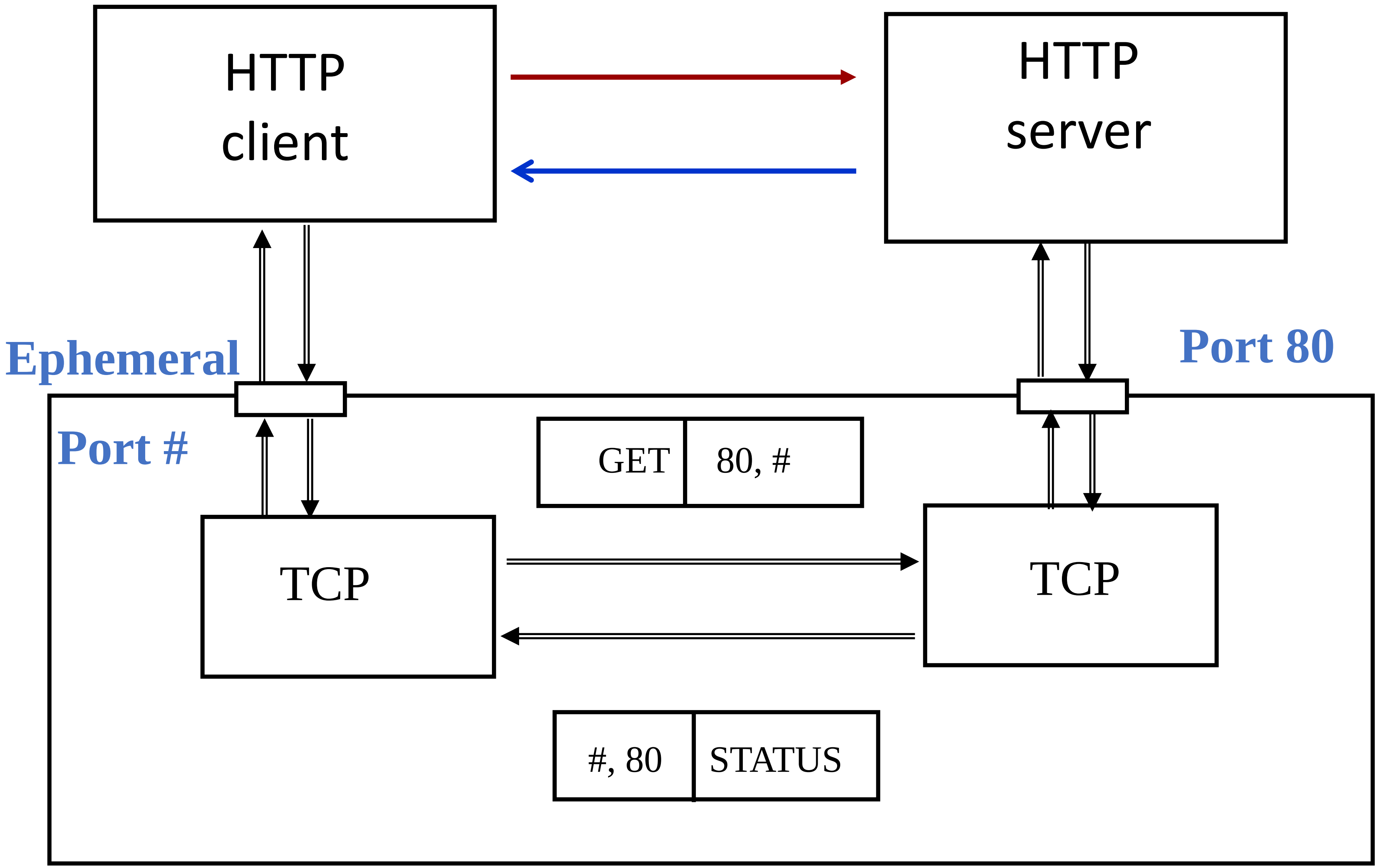
Upper Layers (Layers 5-7)

- Session Layer
  - Management of “sessions”
- Presentation Layer
  - Data translation, formatting, encryption, compression
- Application Layer
  - Interface between user applications and lower network services

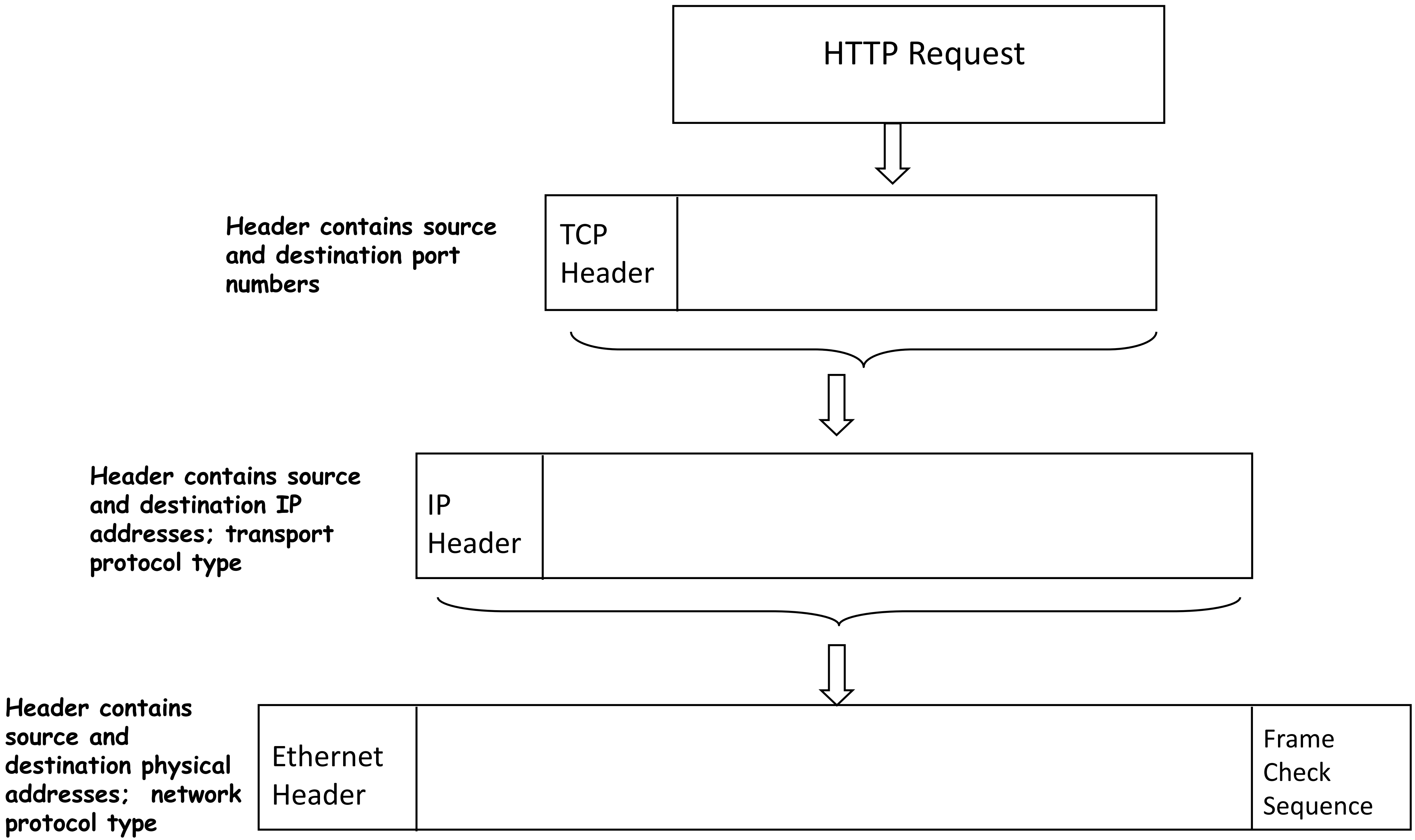
Example: Web Servers



Example: Web Servers



Example: Web Servers



Thank You

Contact me:

gauravsingal789@gmail.com

Gaurav.singal@nsut.ac.in

www.gauravsingal.in

LinkedIn: https://www.linkedin.com/in/gauravsingal789/

Twitter: https://twitter.com/gaurav_singal

IoT Architectural View

COCSC20

Basic Premises

Devices

send and receive data interacting with the

Network

where the data is transmitted, normalized, and filtered using

Edge Computing

before landing in

Data storage / Databases

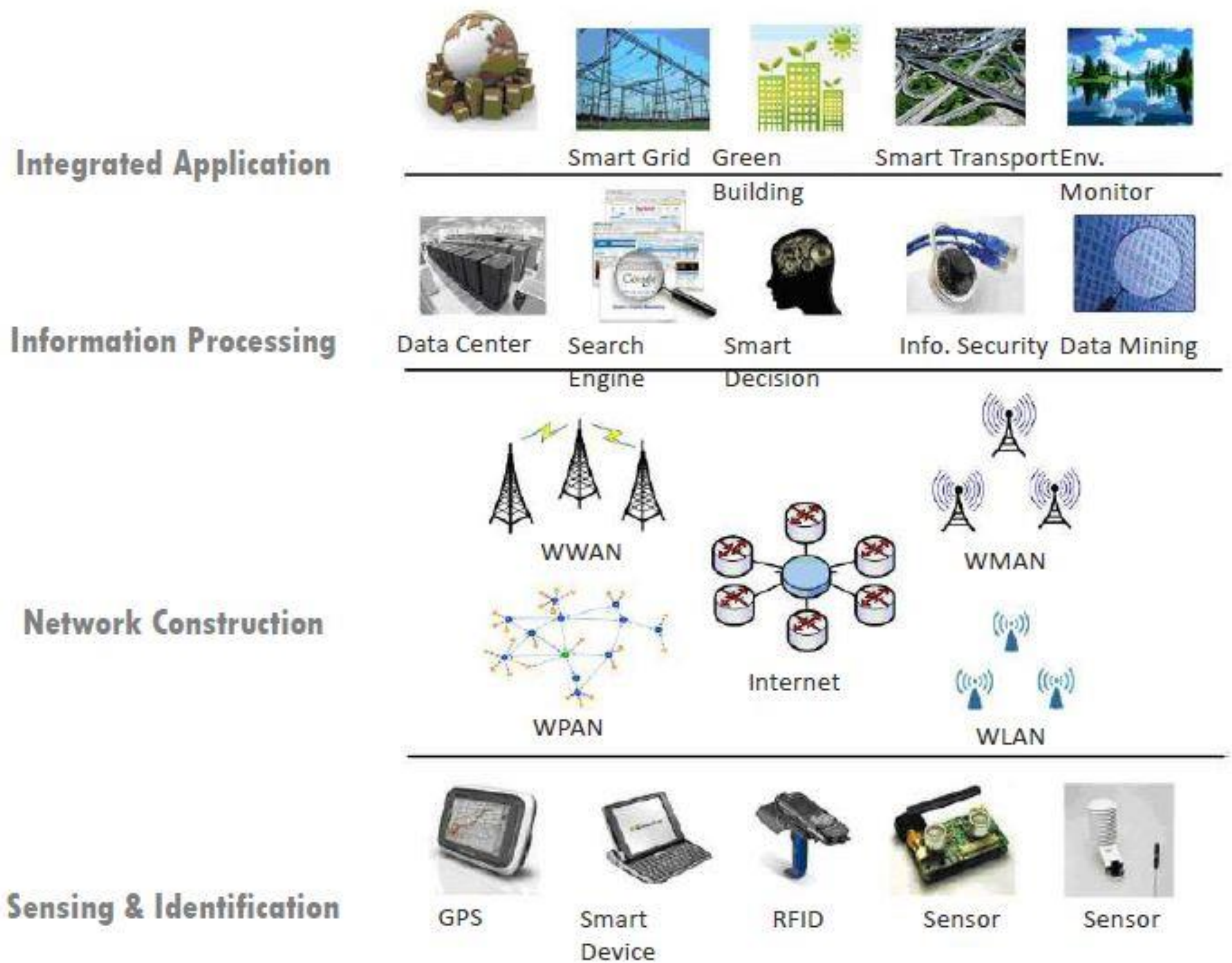
accessible by

Applications

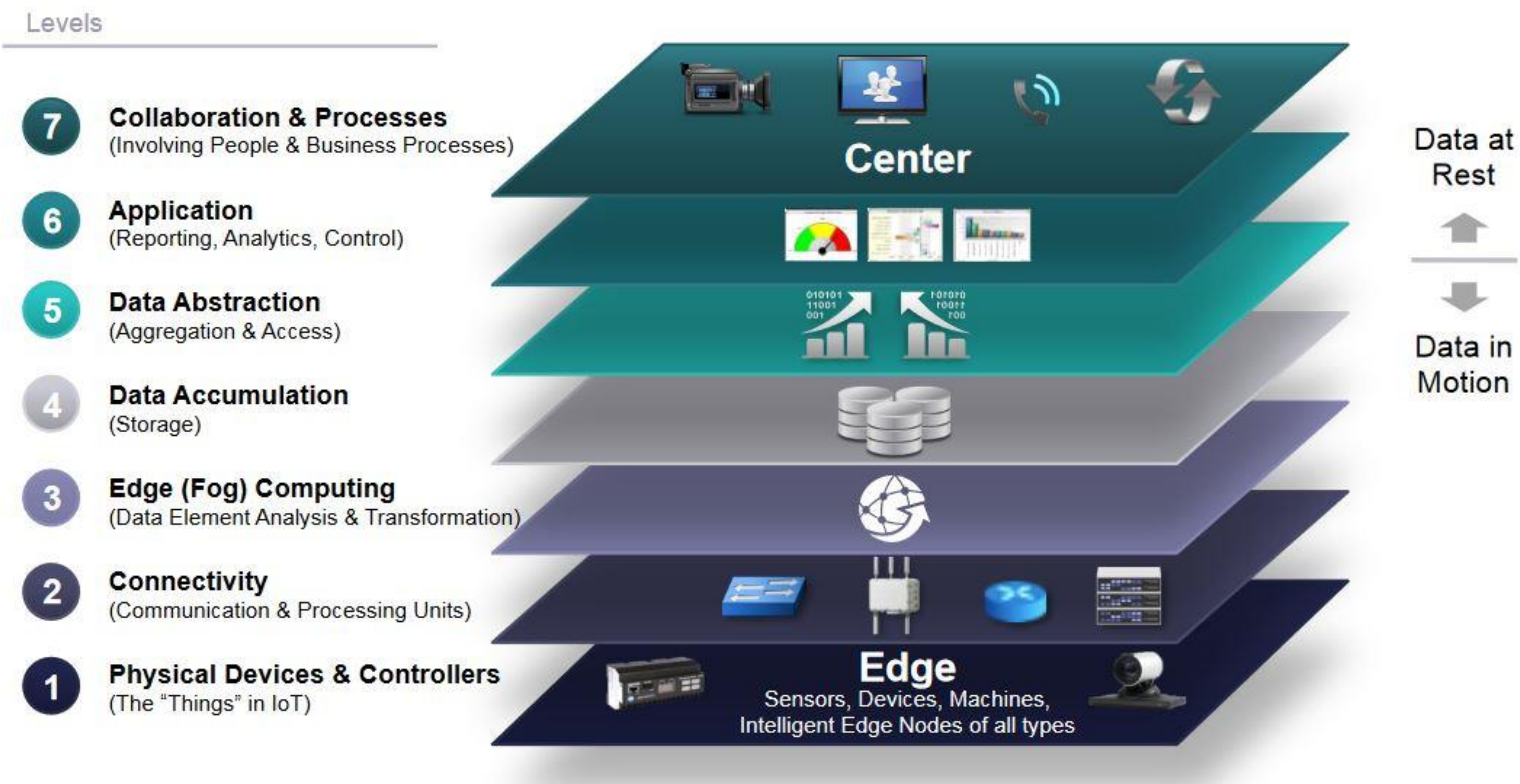
which process it and provide it to people who will

Act and Collaborate

IoT 4 Layers model



Reference Model



CISCO IoT Reference Model

1

Physical Devices & Device Controllers

(The “Things” in IoT)

IoT “devices” are capable of:

- Analog to digital conversion, as required
- Generating data
- Being queried / controlled over-the-net

2 Connectivity
(Communication & Processing Units)

Level 2 functionality focuses on East-West communications

Connectivity includes:

- Communicating with and between the Level 1 devices
- Reliable delivery across the network(s)
- Implementation of various protocols
- Switching and routing
- Translation between protocols
- Security at the network level
- (Self Learning) Networking Analytics

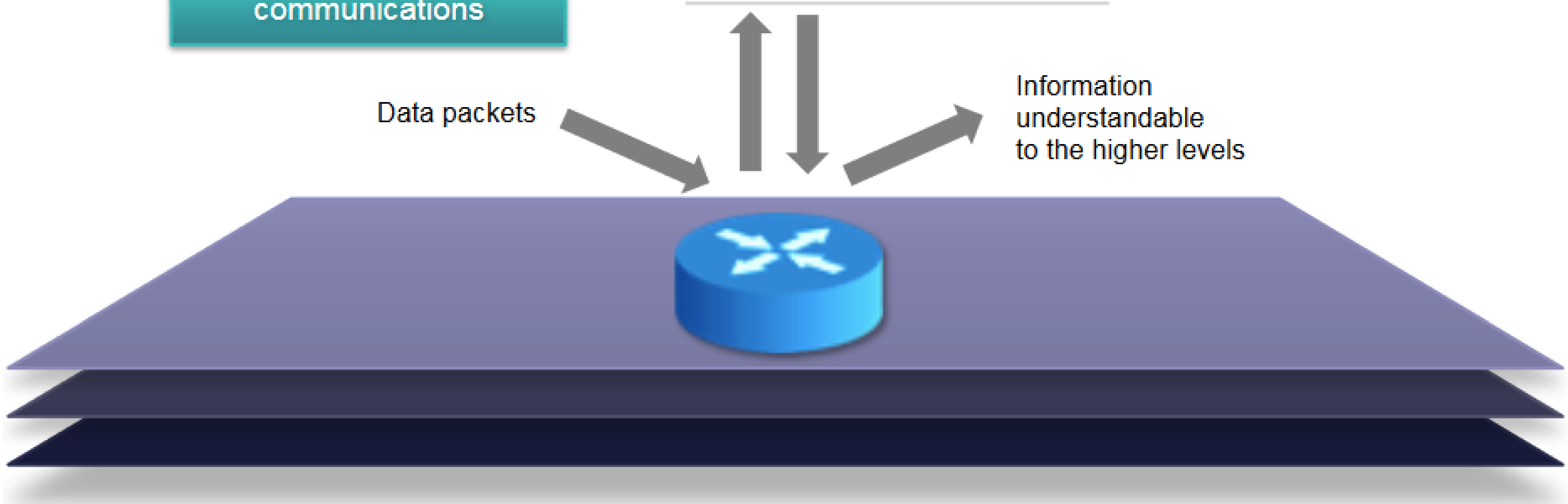
3

Edge (Fog) Computing
(Data Element Analysis
& Transformation)

Level 3 functionality focuses on North-South communications

Include;

- Data filtering, cleanup, aggregation
- Packet content inspection
- Combination of network and data level analytics
- Thresholding
- Event generation



4 Data Accumulation (Storage)

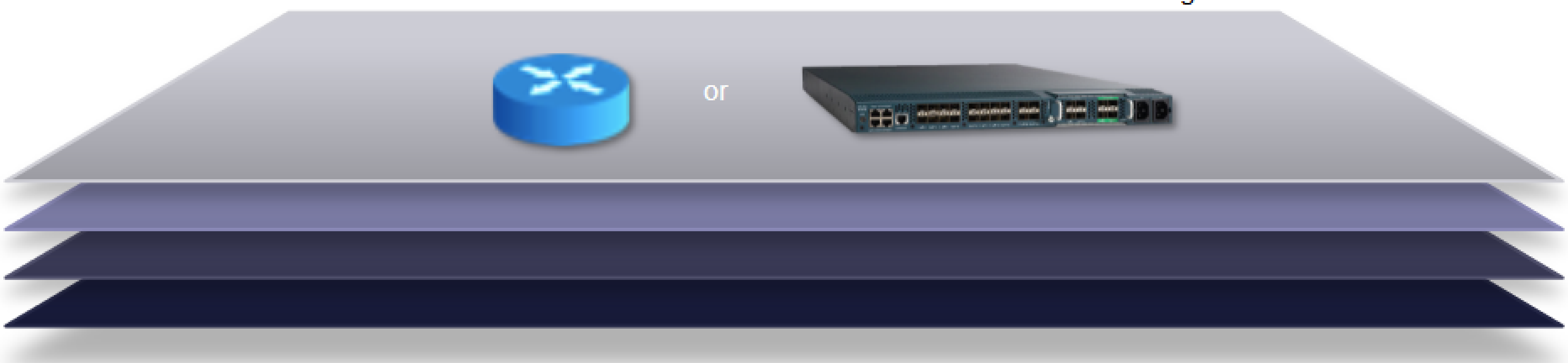
- Event filtering/sampling
- Event comparison
- Event joining for CEP
- Event based rule evaluation
- Event aggregation
- Northbound/southbound alerting
- Event persistence in storage

Query Based Data Consumption

Event Based Data Generation

Making network data usable by applications

1. Converts data-in-motion to data-at-rest
2. Converts format from network packets to database relational tables
3. Achieves transition from 'Event based' to 'Query based' computing
4. Dramatically reduces data through filtering and selective storing



5 Data Abstraction
(Aggregation & Access)

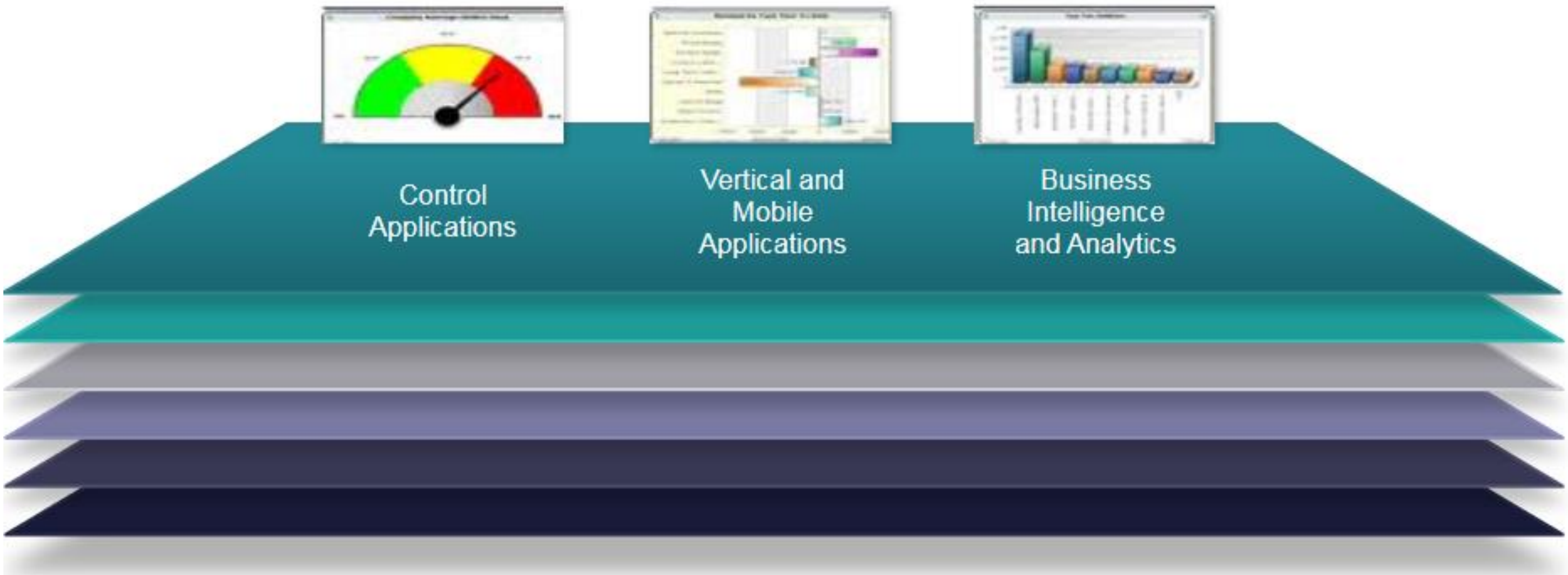
Abstracting the data interface for applications

Information Integration

1. Creates schemas and views of data in the manner that applications want
2. Combines data from multiple sources, simplifying the application
3. Filtering, selecting, projecting, and reformatting the data to serve the client applications
4. Reconciles differences in data shape, format, semantics, access protocol, and security

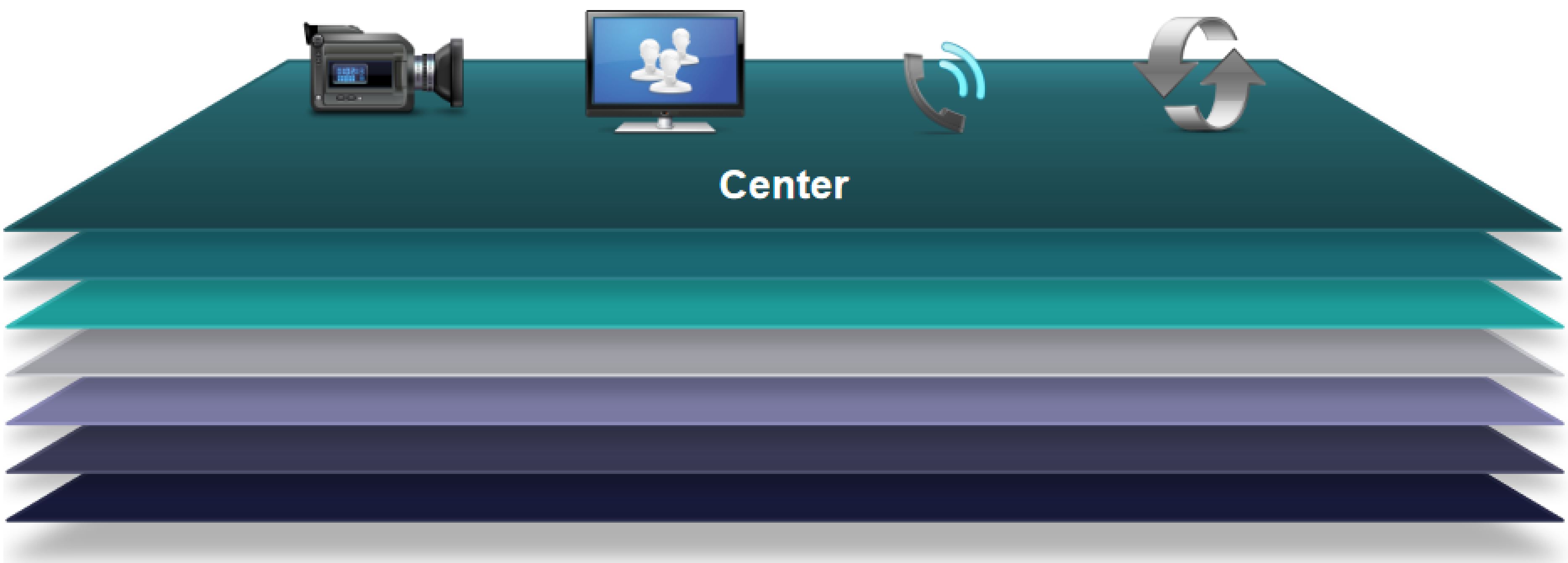
6

Application
(Reporting, Analytics, Control)



7 Collaboration & Processes

(Involving people and business processes)



How Many Layers in OSI model?

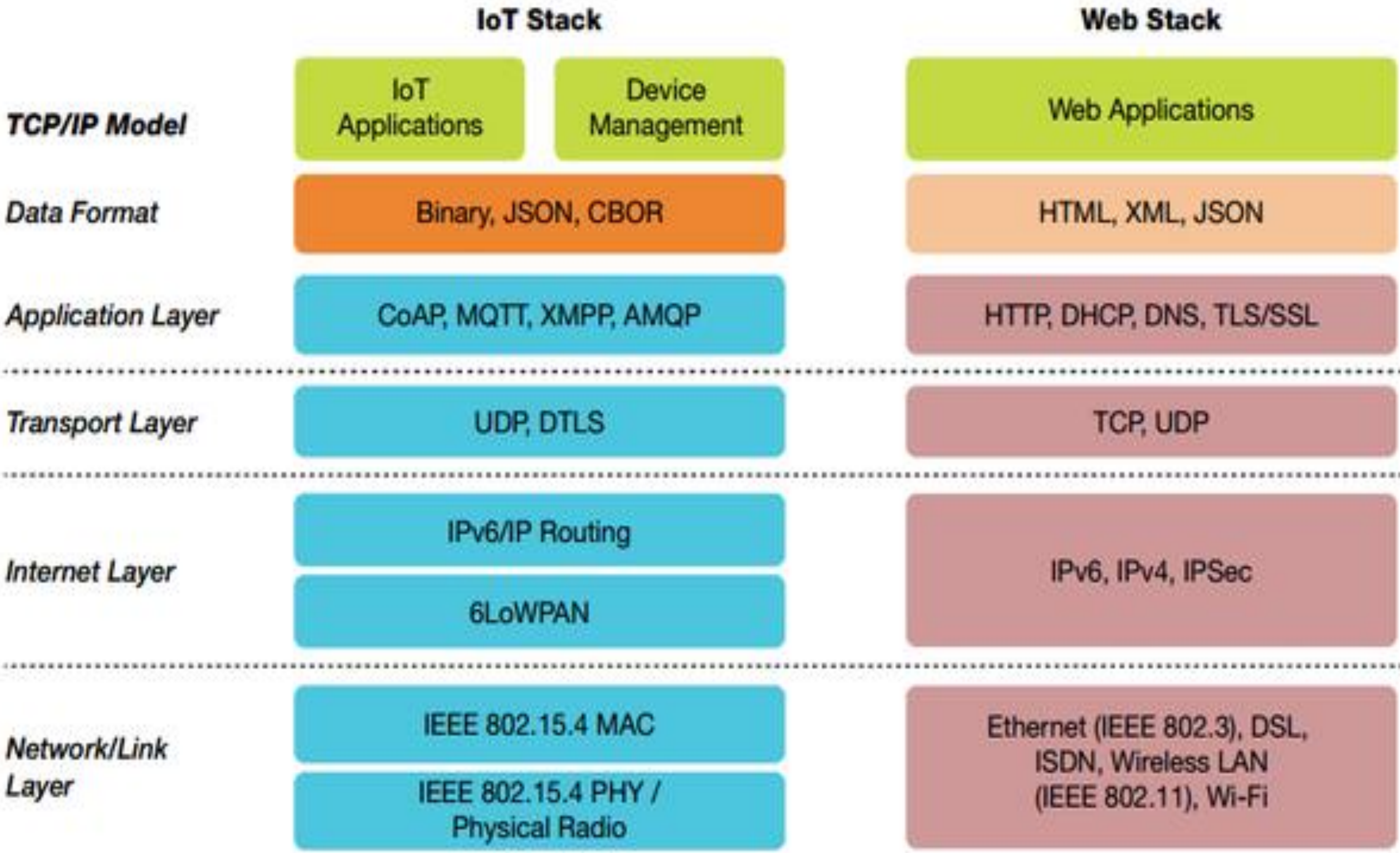
A. Four
B. Five
C. Six
D. Seven
E. None of the above.

TCP/IP stands for?

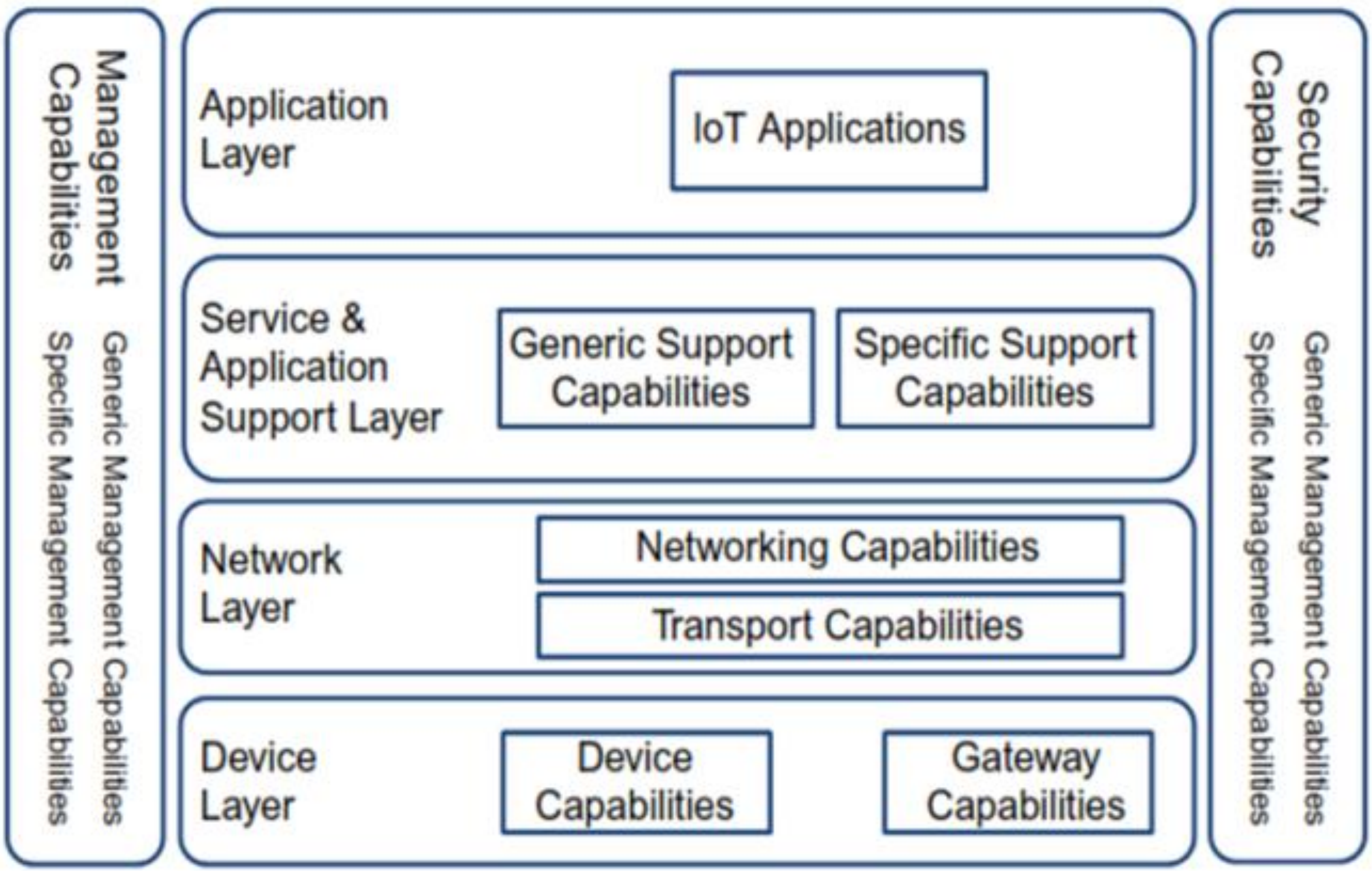
Transmission Control Protocol/Internet Networking Protocol have

A. Four
B. Five
C. Six
D. Seven
E. None of the above.

OSI MODEL	TCP/IP MODEL
Application Layer	Application Layer
Presentation Layer	
Session Layer	
Transport Layer	Transport Layer
Network Layer	Internet Layer
Data Link Layer	Network Access Layer
Physical Layer	


IoT Stack
Web Stack
TCP/IP Model
IoT Applications
Device Management
Web Applications
Data Format
Binary, JSON, CBOR
HTML, XML, JSON
Application Layer
CoAP, MQTT, XMPP, AMQP
HTTP, DHCP, DNS, TLS/SSL
Transport Layer
UDP, DTLS
TCP, UDP
Internet Layer
IPv6/IP Routing
6LoWPAN
IPv6, IPv4, IPSec
Network/Link Layer
IEEE 802.15.4 MAC
IEEE 802.15.4 PHY / Physical Radio
Ethernet (IEEE 802.3), DSL, ISDN, Wireless LAN (IEEE 802.11), Wi-Fi

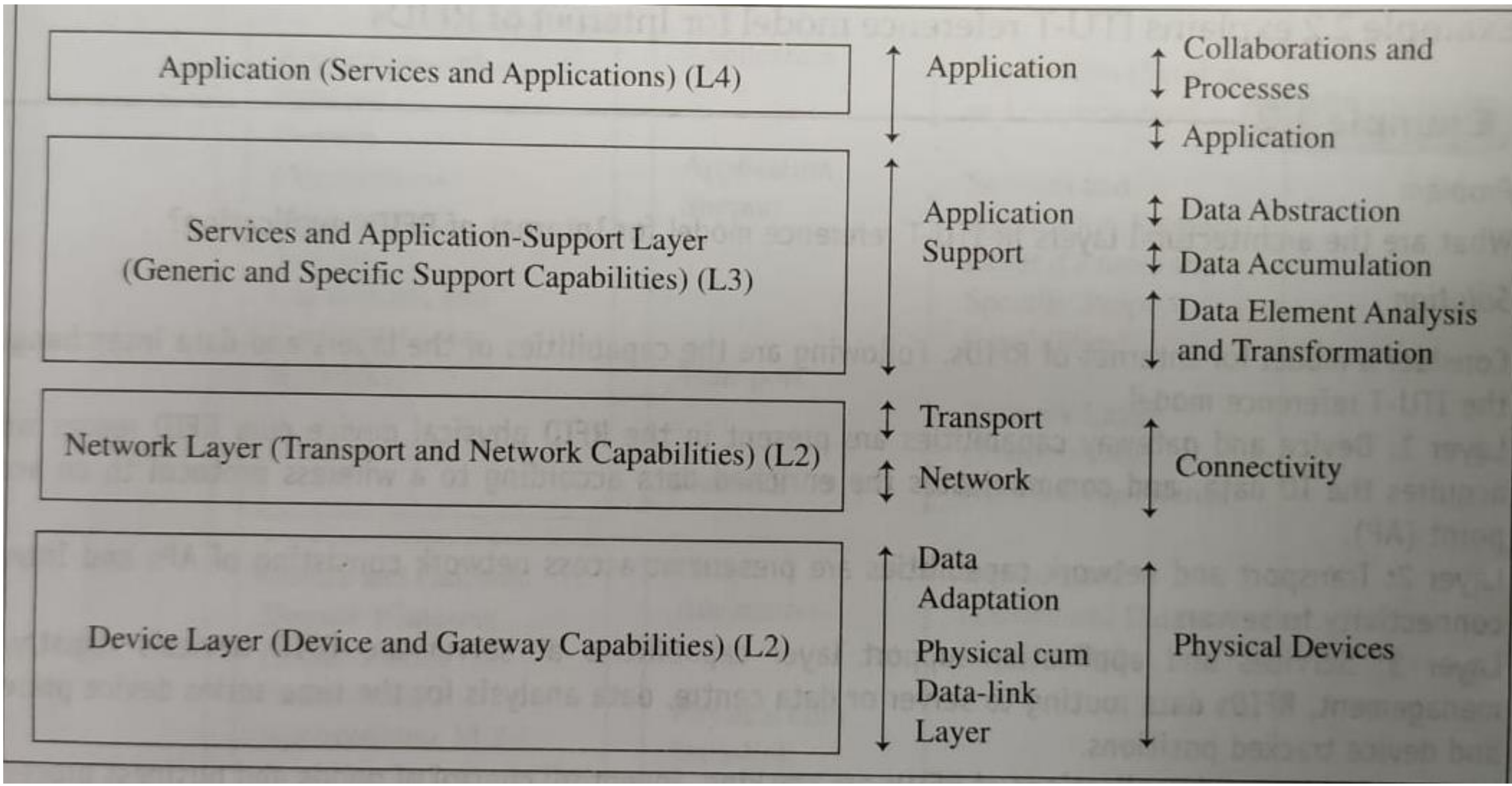
ITU-T IoT Reference Model



ICMP stands for

A. Internet Connect Message Protocol
B. Internet Control Message Protocol
C. International Connect Message Protocol
D. International Control Message Protocol

Comparison



Application (Services and Applications) (L4)
Services and Application-Support Layer
(Generic and Specific Support Capabilities) (L3)
Network Layer (Transport and Network Capabilities) (L2)
Device Layer (Device and Gateway Capabilities) (L2)
Application
Application Support
Transport
Network
Data Adaptation
Physical cum Data-link Layer
Collaborations and Processes
Application
Data Abstraction
Data Accumulation
Data Element Analysis and Transformation
Connectivity
Physical Devices

Thank You

Contact me:

gauravsingal789@gmail.com

Gaurav.singal@nsut.ac.in

www.gauravsingal.in

LinkedIn: https://www.linkedin.com/in/gauravsingal789/

Twitter: https://twitter.com/gaurav_singal

Mobile Adhoc Networks

COCSC20

Wi-Fi

- Wi-Fi:
  - name is NOT an abbreviation
- **Wireless Local Area Network (WLAN)** technology.
- WLAN and Wi-Fi often used synonymous.
- Typically **in 2.4 and 5 GHz bands.**
- Based on **IEEE 802.11** family of standards.

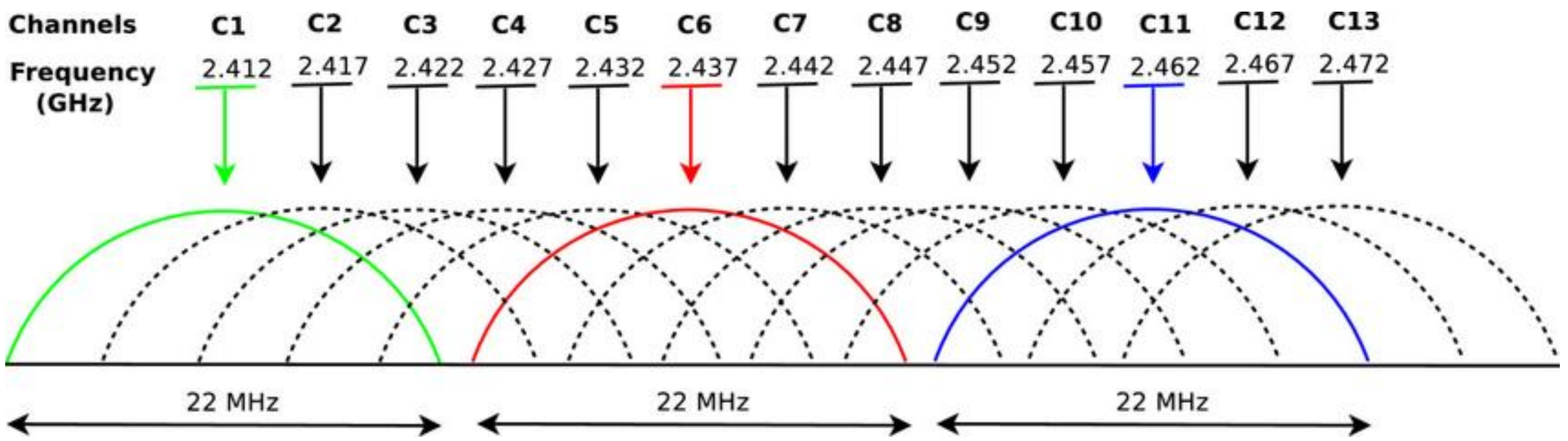
IEEE

- IEEE (Institute of Electrical and Electronics Engineers) established the 802.11 Group in 1990. Specifications for standard ratified in 1997.
- Initial speeds were 1 and 2 Mbps.
- IEEE modified the standard in 1999 to include:
  - 802.11b
  - 802.11a
  - 802.11g
  - 802.11n
  - **802.11ac**

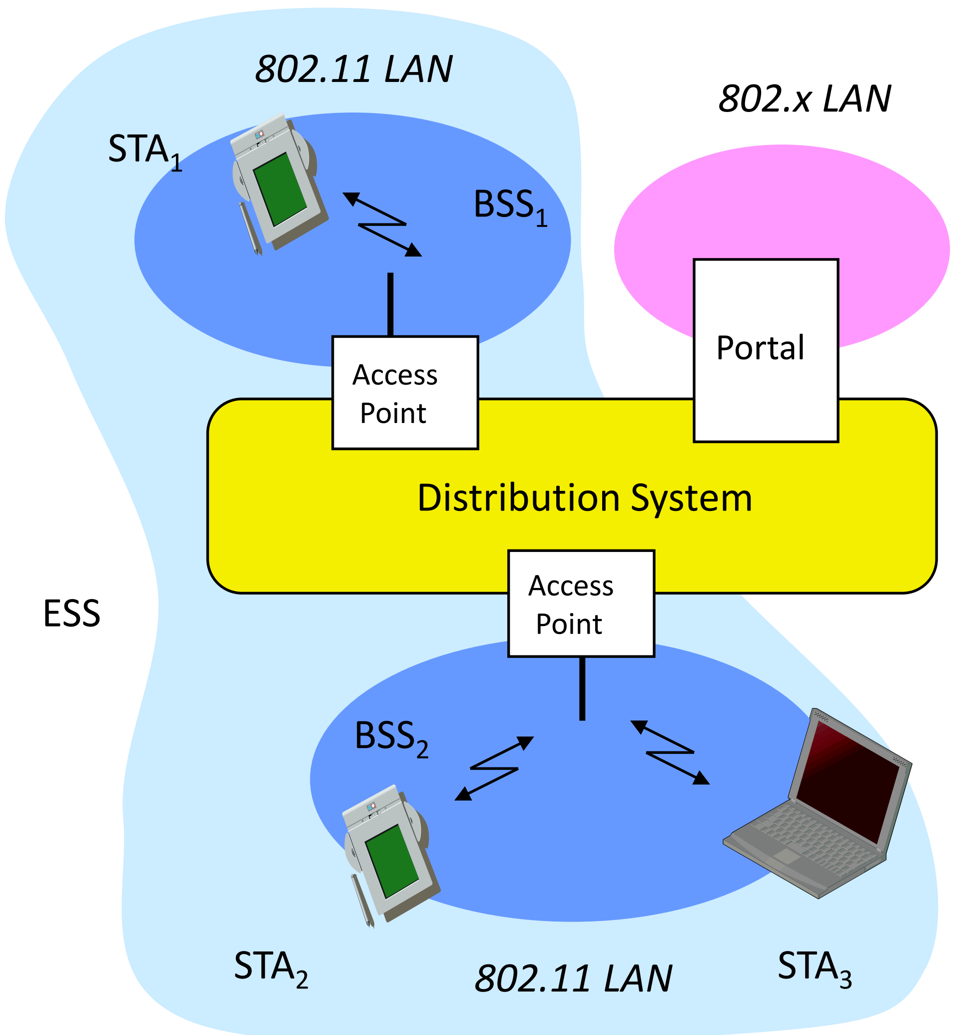
WLAN (Wi-Fi)

802.11 Wireless Standards					
IEEE Standard	802.11a	802.11b	802.11g	802.11n	802.11ac
Year Adopted	1999	1999	2003	2009	2014
Frequency	5 GHz	2.4 GHz	2.4 GHz	2.4/5 GHz	5 GHz
Max. Data Rate	54 Mbps	11 Mbps	54 Mbps	600 Mbps	1 Gbps
Typical Range Indoors*	100 ft.	100 ft.	125 ft.	225 ft.	90 ft.
Typical Range Outdoors*	400 ft.	450 ft.	450 ft.	825 ft.	1,000 ft.

Wi-Fi Channels

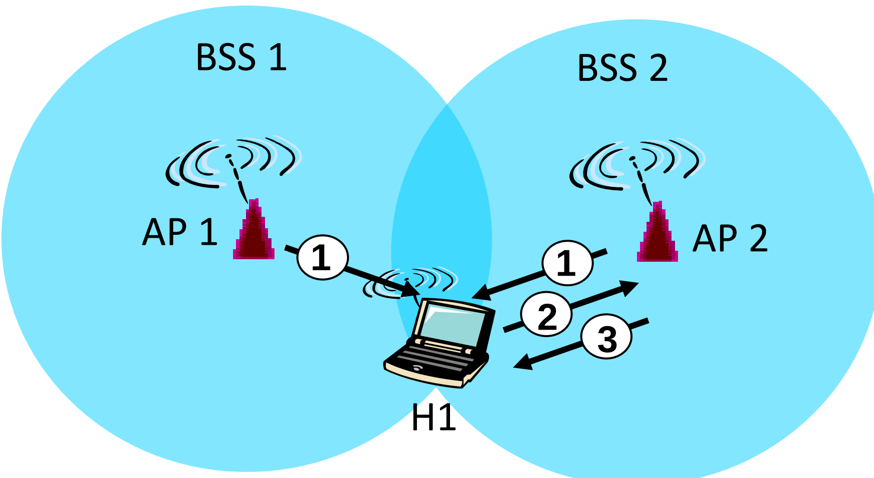


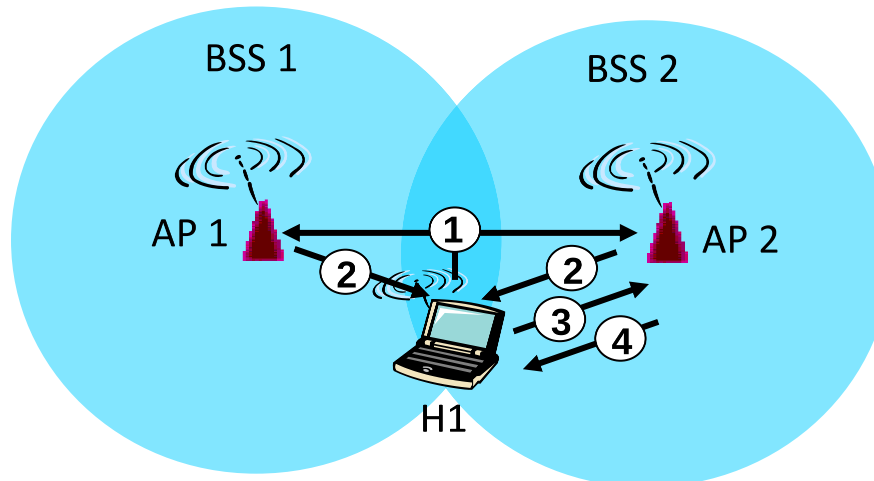
802.11 - Architecture of an Infrastructure Network



- Station (STA)
  - terminal with access mechanisms to the wireless medium and radio contact to the access point
- Basic Service Set (BSS)
  - group of stations using the same radio frequency
- Access Point
  - station integrated into the wireless LAN and the distribution system
- Portal
  - bridge to other (wired) networks
- Distribution System
  - interconnection network to form one logical network (ESS: Extended Service Set) based on several BSS

Wi-Fi (802.11) Scanning





Passive Scanning

(1) Beacons sent from APs

(2) Association Request sent from H1 to selected AP

(3) Association Response sent from AP to H1

Active Scanning

(1) Probe Request (broadcast) sent from H1

(2) Probe Response sent from APs

(3) Association Request sent from H1 to selected AP

(4) Association Response sent from AP to H1

Wi-Fi Alliance Mission Statement

- Non-profit organization
- Certify the interoperability of products and services based on IEEE 802.11 technology
- Grow the global market for **Wi-Fi® CERTIFIED** products and services across all market segments, platforms, and applications
- Rigorous interoperability testing requirements

Certificate & Logo

Wi-Fi® Interoperability Certificate **Certification ID:** 24567832AP

This certificate represents the capabilities and features that have passed the interoperability testing governed by the Wi-Fi Alliance.

Detailed descriptions of these features can be found at www.wi-fi.org/certificate

Certification Date:	February 14, 2004
Category:	Access Point
Company:	Name of Company
Product:	Wireless LAN Access Point/Router Model#EX1010
Model/SKU #:	EX1010

This product has passed Wi-Fi certification testing for the following standards:

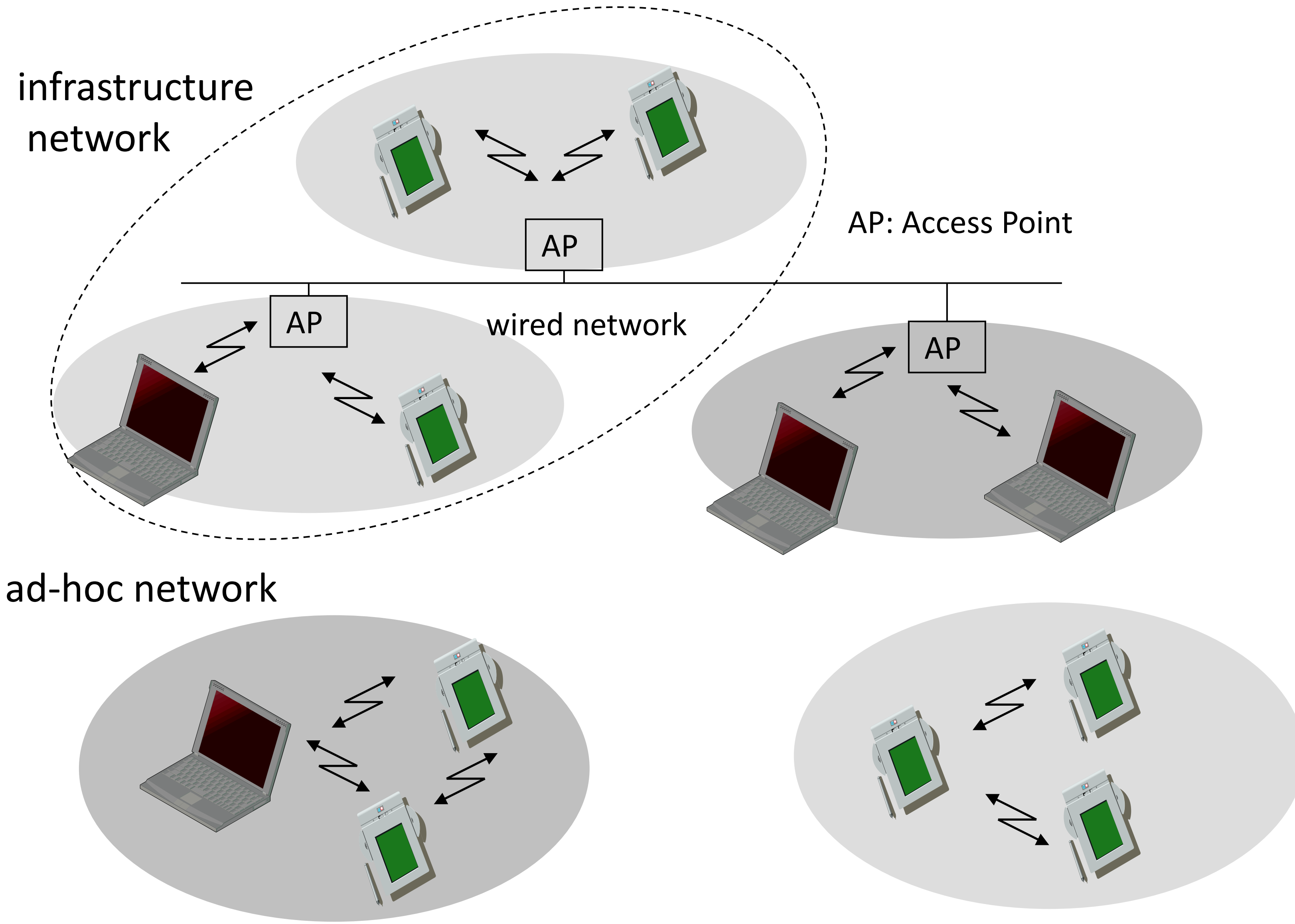
IEEE Standard	Security	Quality of Service	Public Access
802.11a 802.11b 802.11g 802.11n	WPA - Personal WPA - Enterprise WPA2 - Personal (802.11i) WPA2 - Enterprise (802.11i)	WME (802.11e EDCA profile) WSM (802.11e HCCA profile)	
Regulatory 802.11d 802.11h	**Supplicant** EAP-TLS EAP-TTLS/MSCHAPv2 EAP-TTLS/PAP PEAPv0/EAP-MSCHAPv2 PEAPv1/EAP-GTC PEAPv1/EAP-MD5 EAP-SIM		
	Authentication Server EAP-TLS EAP-TTLS/MSCHAPv2 EAP-TTLS/PAP PEAPv0/EAP-MSCHAPv2 PEAPv1/EAP-GTC PEAPv1/EAP-MD5		

For more information: www.wi-fi.org/certified_products

Certificate inside packaging (optional)

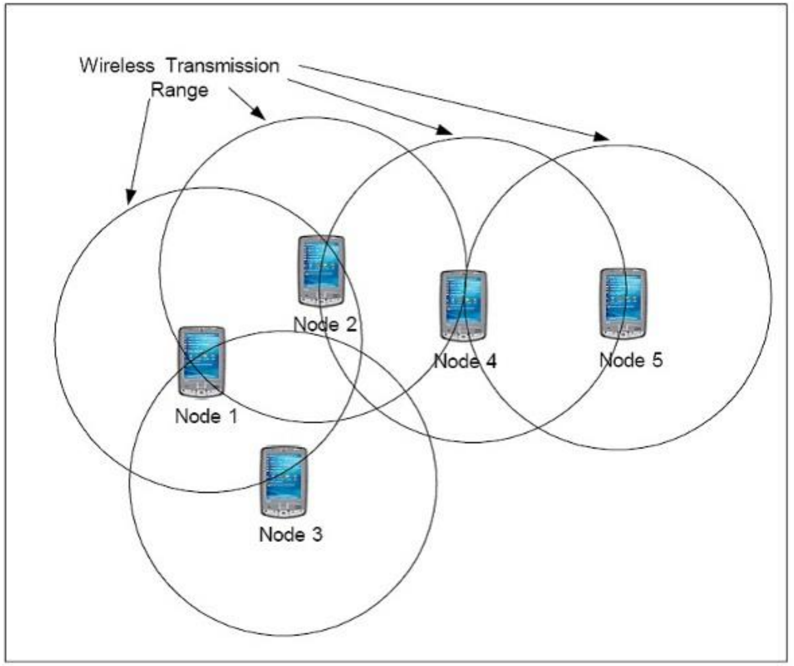
- Logo on product packaging (mandatory)
- Helps retailers and consumers

Infrastructure vs. Ad-Hoc Networks



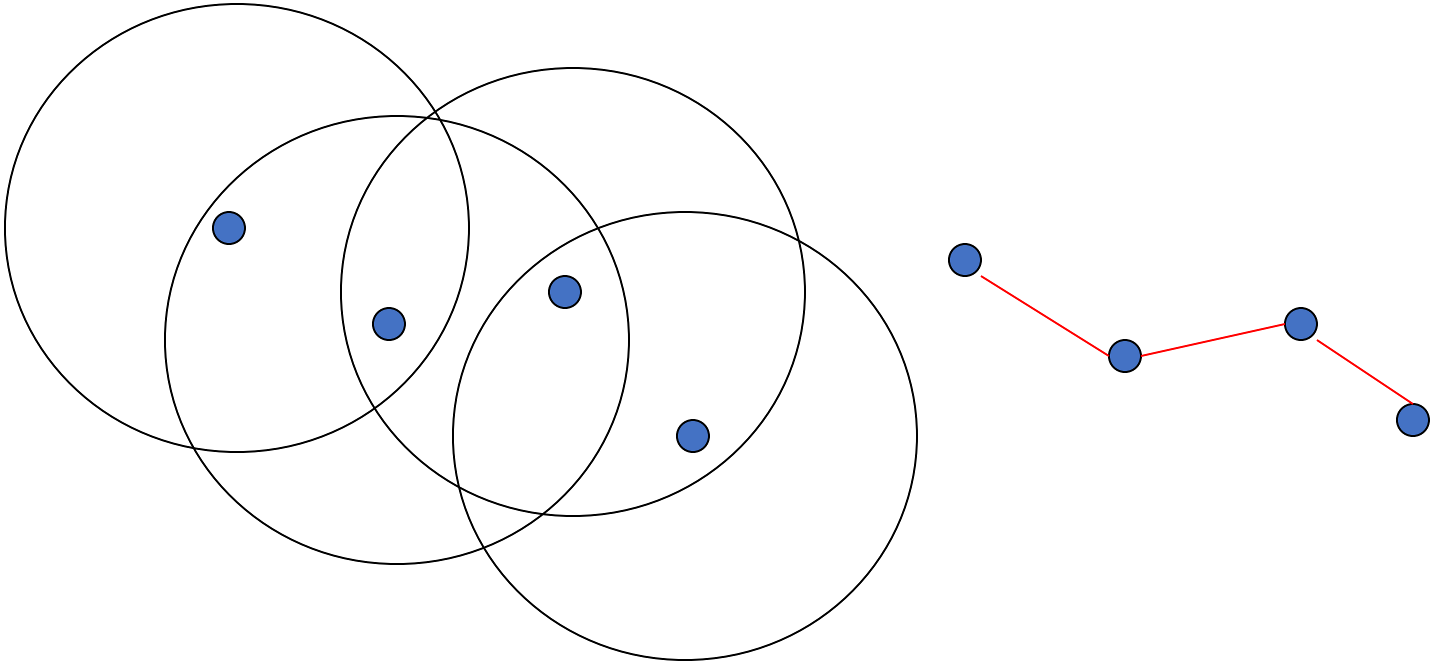
Infrastructure-Less (Ad-Hoc)

- Ad-hoc means *'for this purpose'*
- No need for infrastructure (like routers, cell towers, etc.)
- MANET: **M**obile **A**d-Hoc **Net**work



Routing

- Packets may need to traverse multiple links to reach destination
- Mobility causes route changes



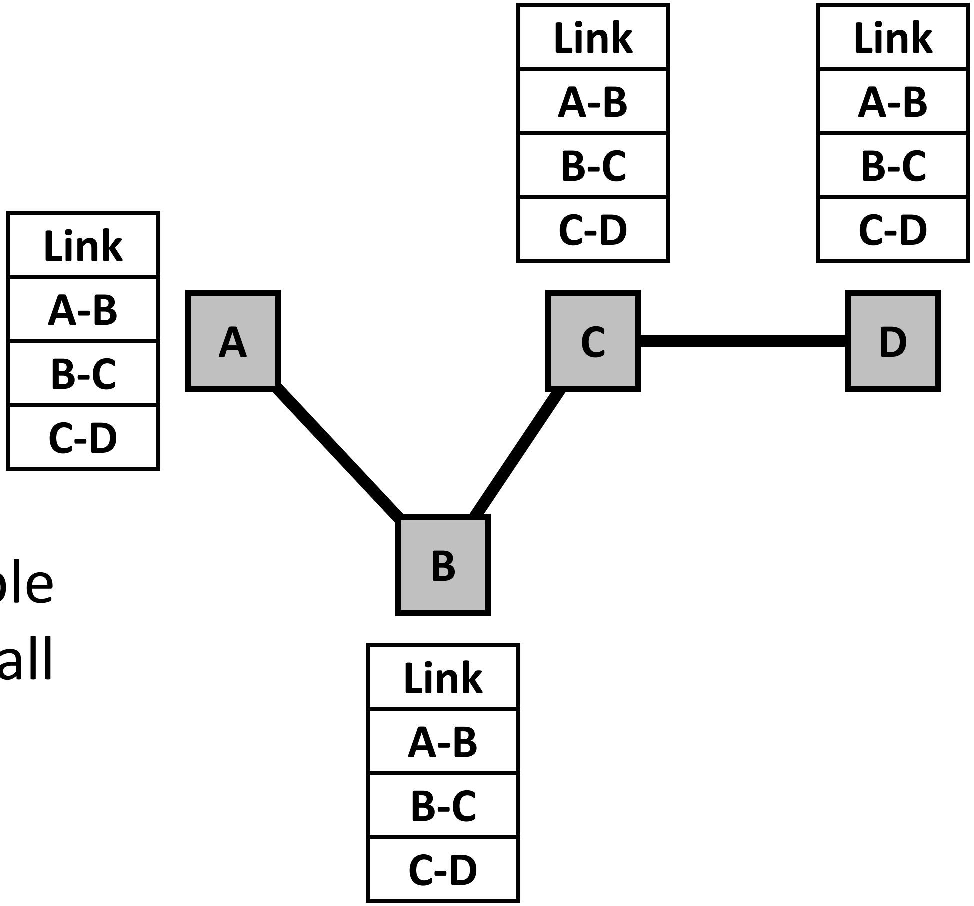
Ad-Hoc Routing Protocol

- An ad-hoc routing protocol is a convention that controls how nodes decide which way to route packets between computing devices in a mobile ad-hoc network
- Foundation in most protocols: **neighbor discovery**
  - Nodes send periodic announcements as broadcast packets (beacon messages, alive messages, …)
  - Can embed "neighbor table" into such messages; allows nodes to learn "2-hop neighborhood"
- Popular types of routing protocols:
  - **Proactive**
  - **Reactive**
  - **Geographic**

Proactive: "Link-State" Algorithms

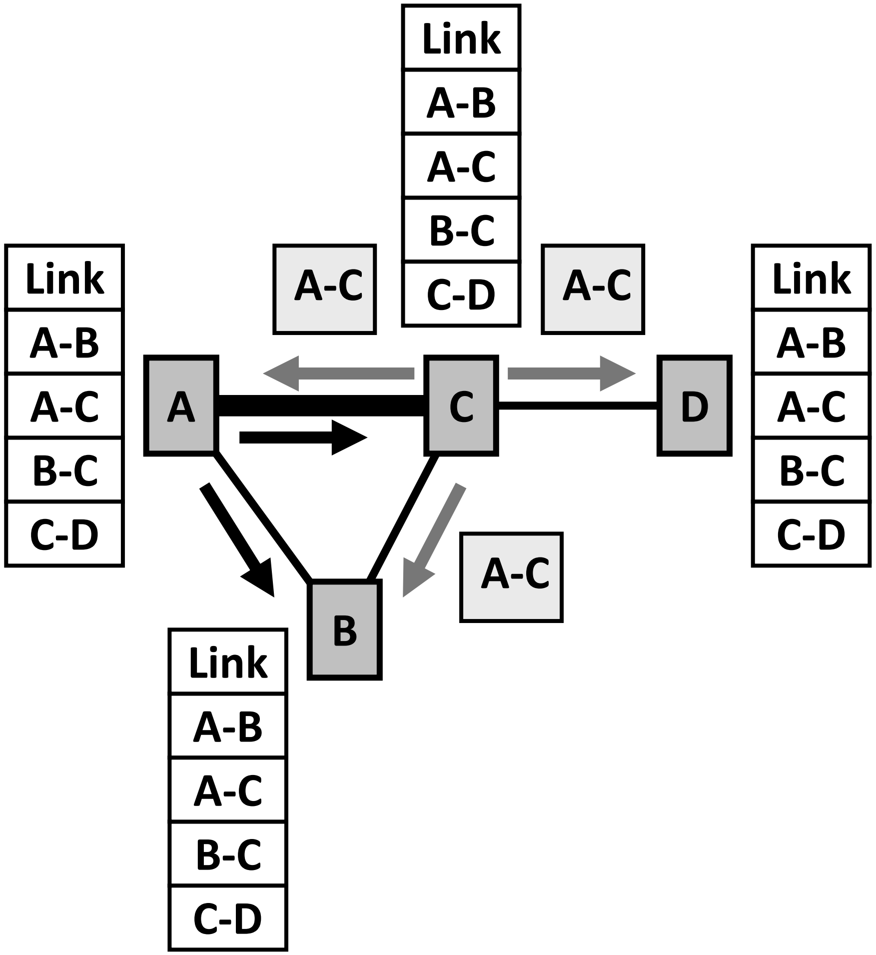
- Each node shares its link information so that all nodes can build a map of the full network topology

- Assuming the topology is stable for a sufficiently long period, all nodes will have the same topology information



Proactive: "Link-State" Algorithms

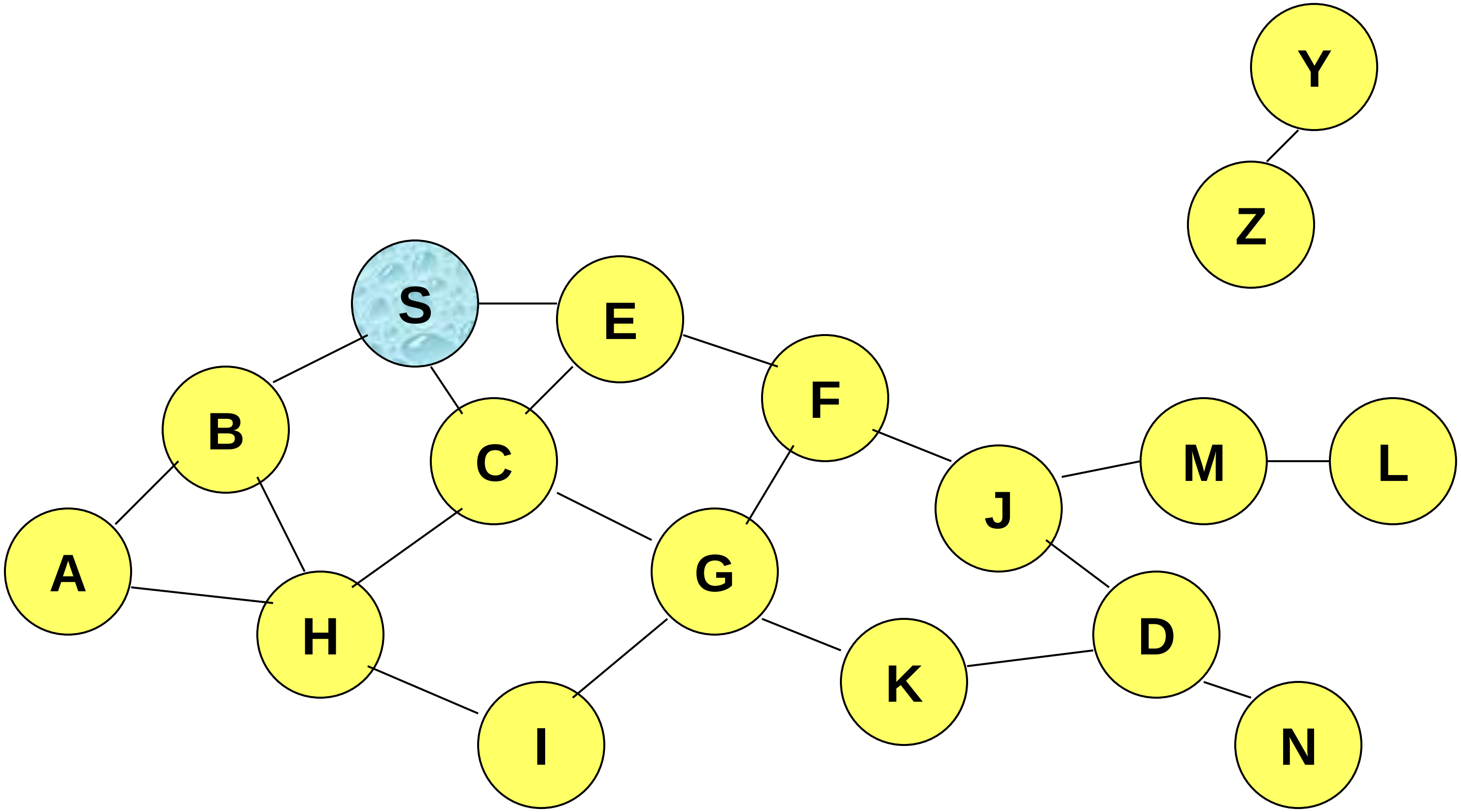
- Link information is updated when a link changes state (goes up or down)
  - by sending small "hello" packets to neighbors
- Nodes A and C propagate the existence of link A-C to their neighbors and, eventually, to the entire network



Reactive: DSR

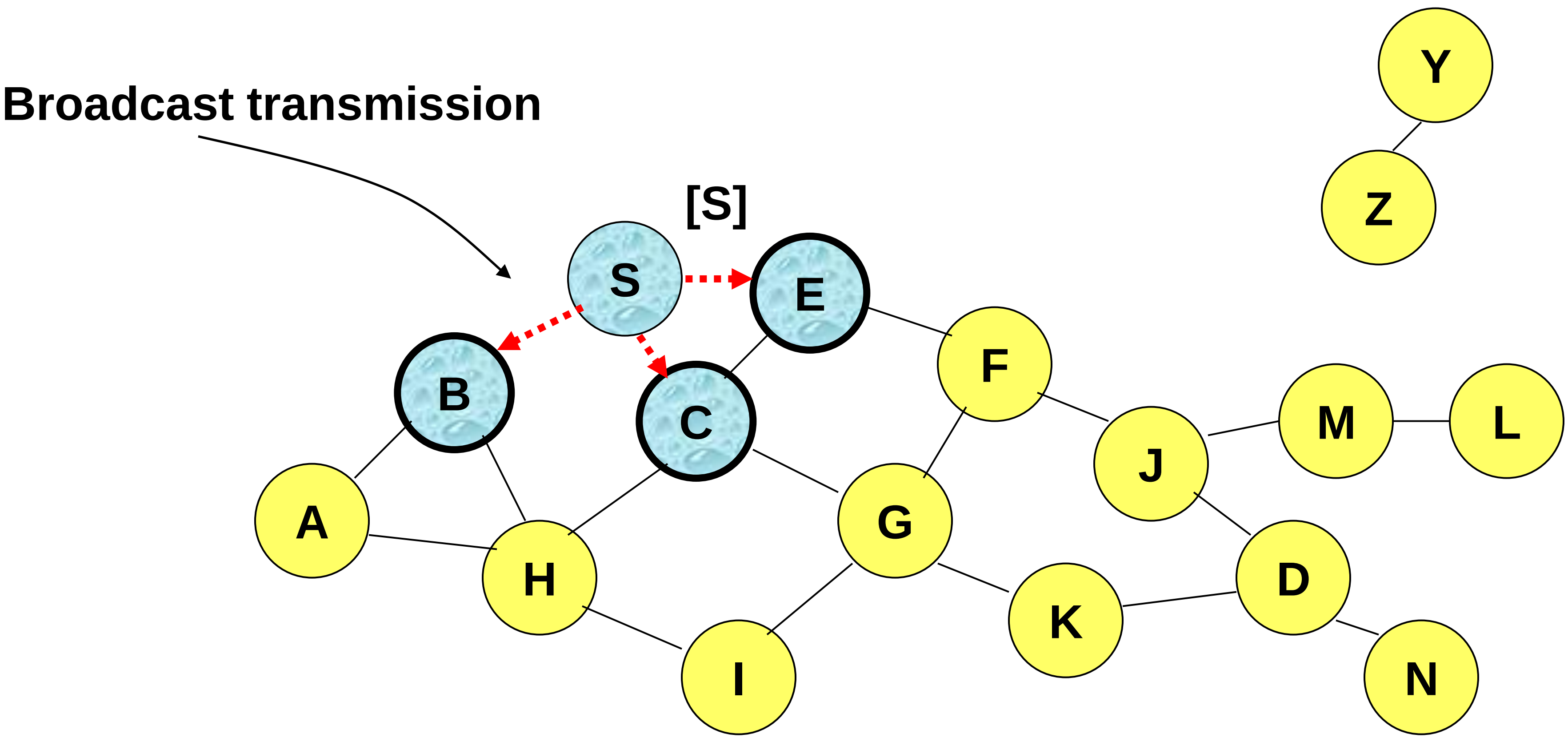
- **D**ynamic **S**ource **R**outing
- Search for route when needed only
  - Search using **Route Request (RREQ)** broadcasts
  - Response using **Route Reply (RREP)** message
- Every message along route contains entire path to help intermediate nodes to decide what to do with message

Route Discovery in DSR



Represents a node that has received RREQ for D from S

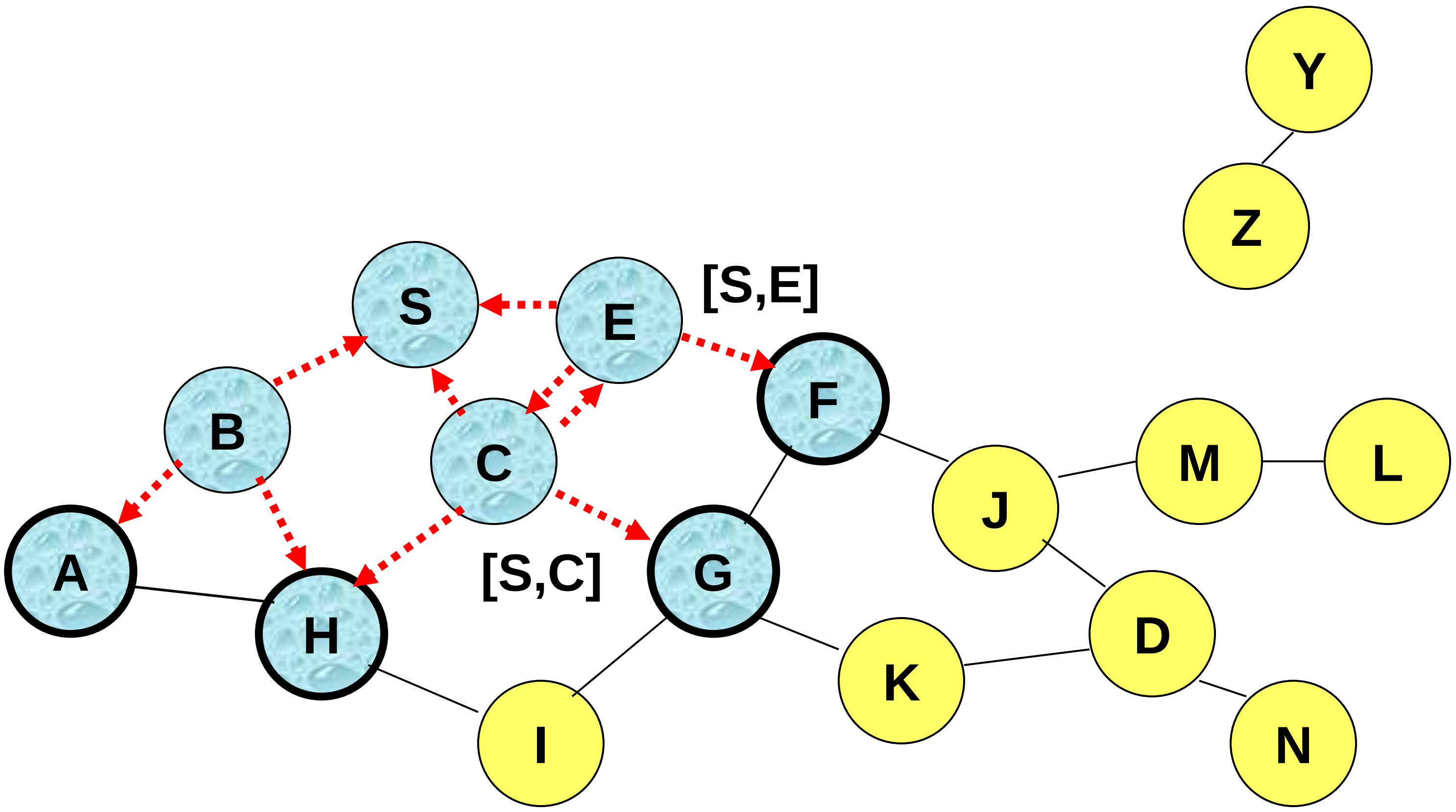
Route Discovery in DSR



Represents transmission of RREQ

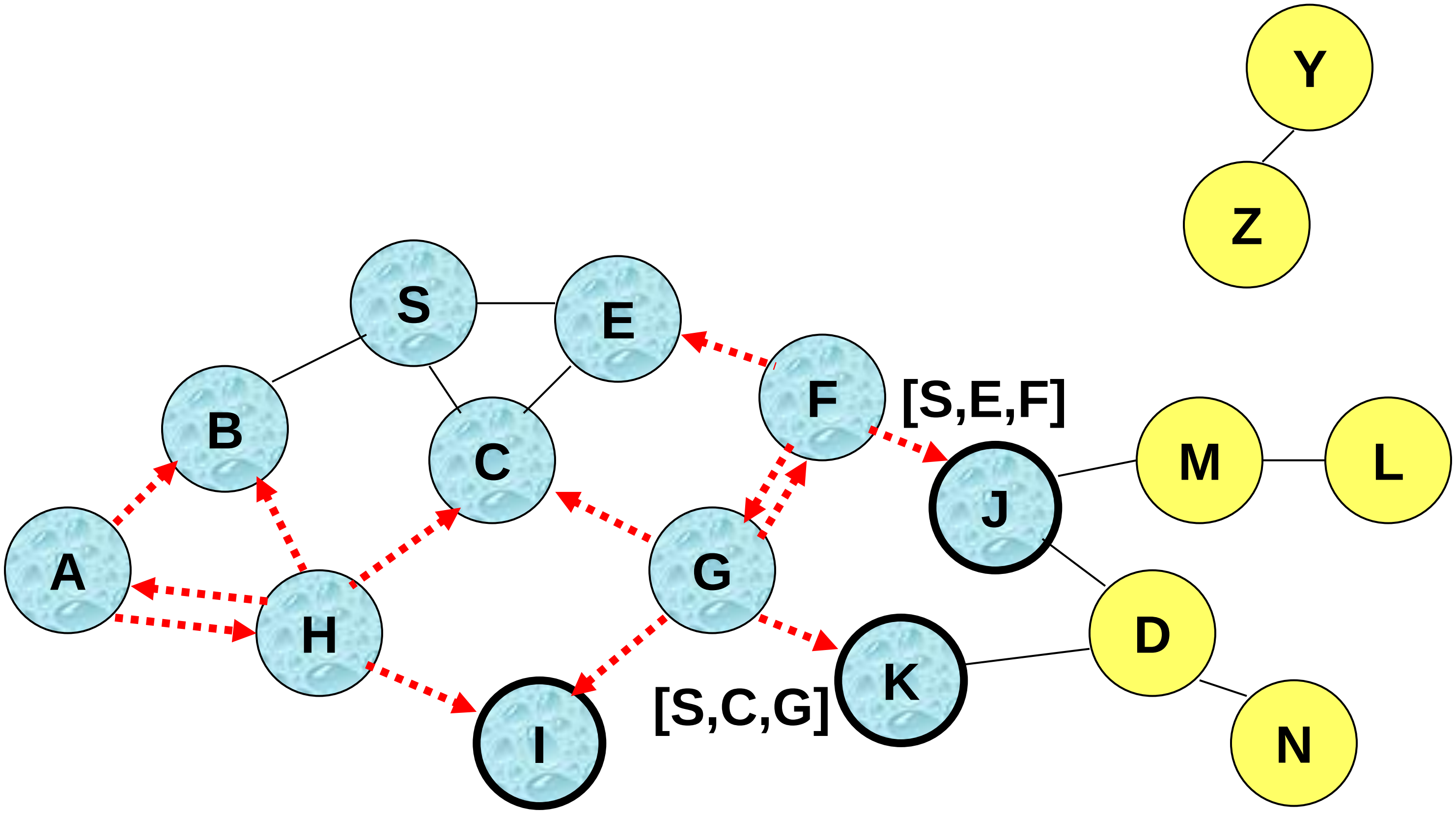
[X,Y] Represents list of identifiers appended to RREQ

Route Discovery in DSR



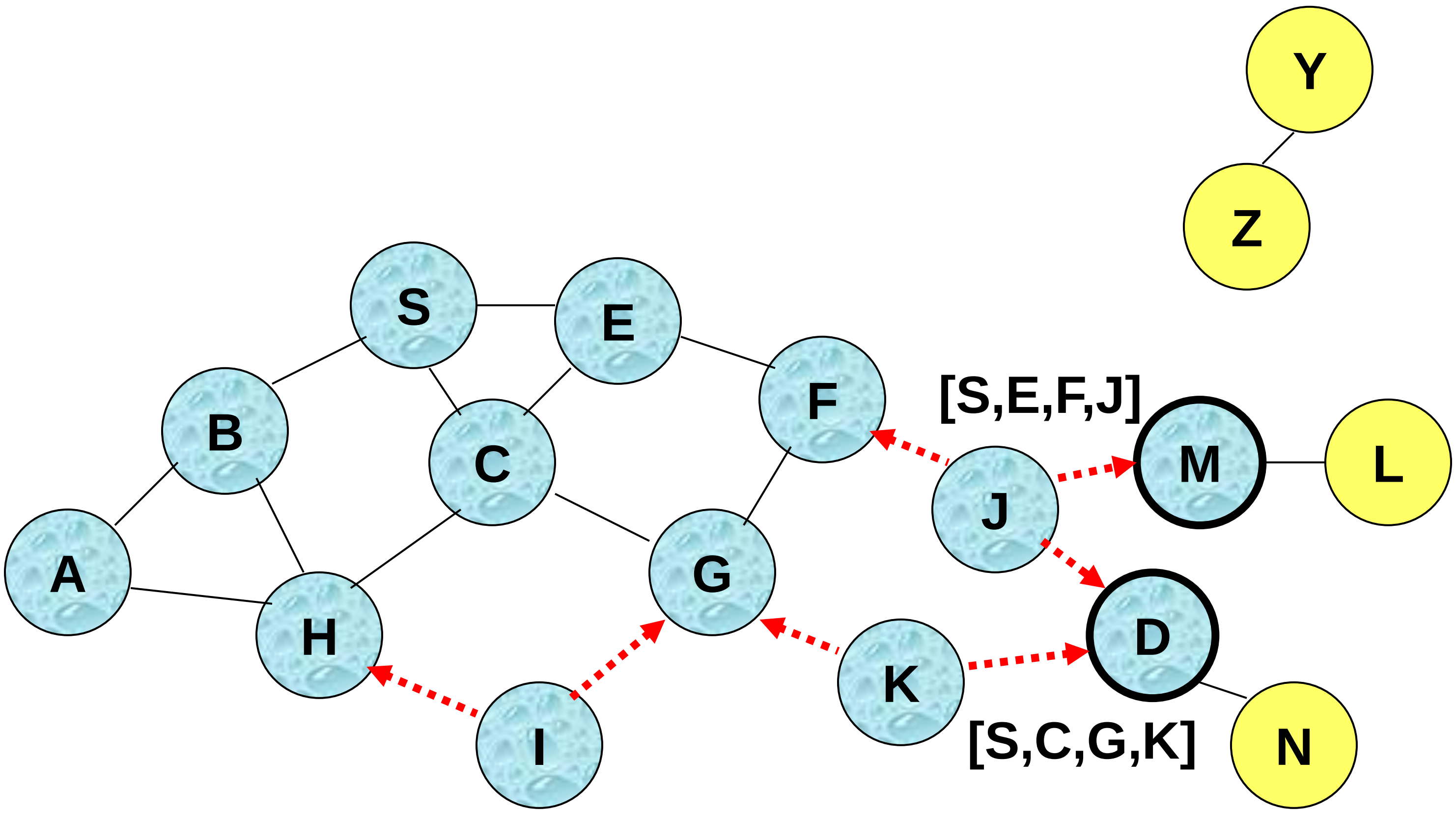
- **Node H receives packet RREQ from two neighbors: potential for collision**

Route Discovery in DSR



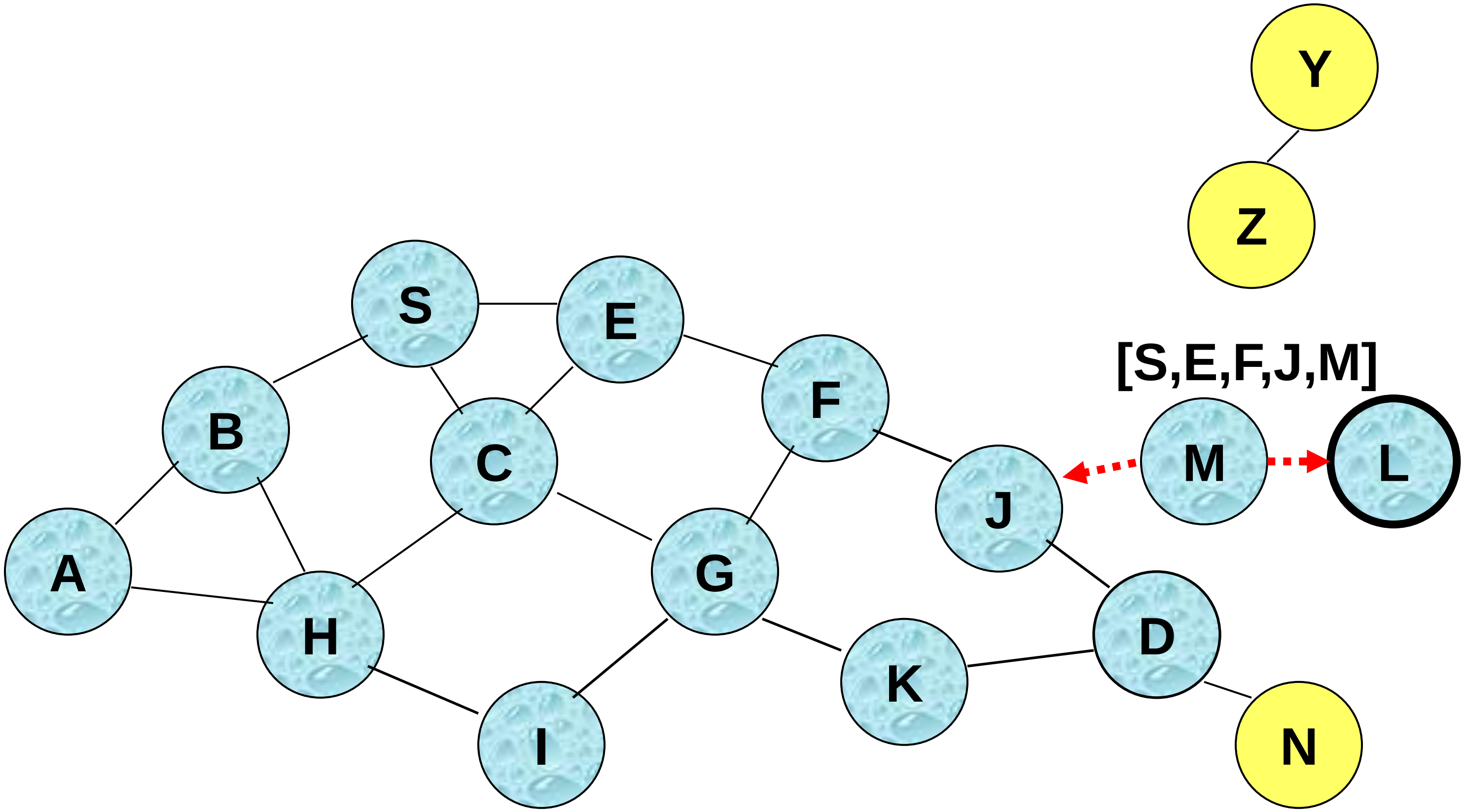
- **Node C receives RREQ from G and H, but does not forward it again, because node C has already forwarded RREQ once**

Route Discovery in DSR



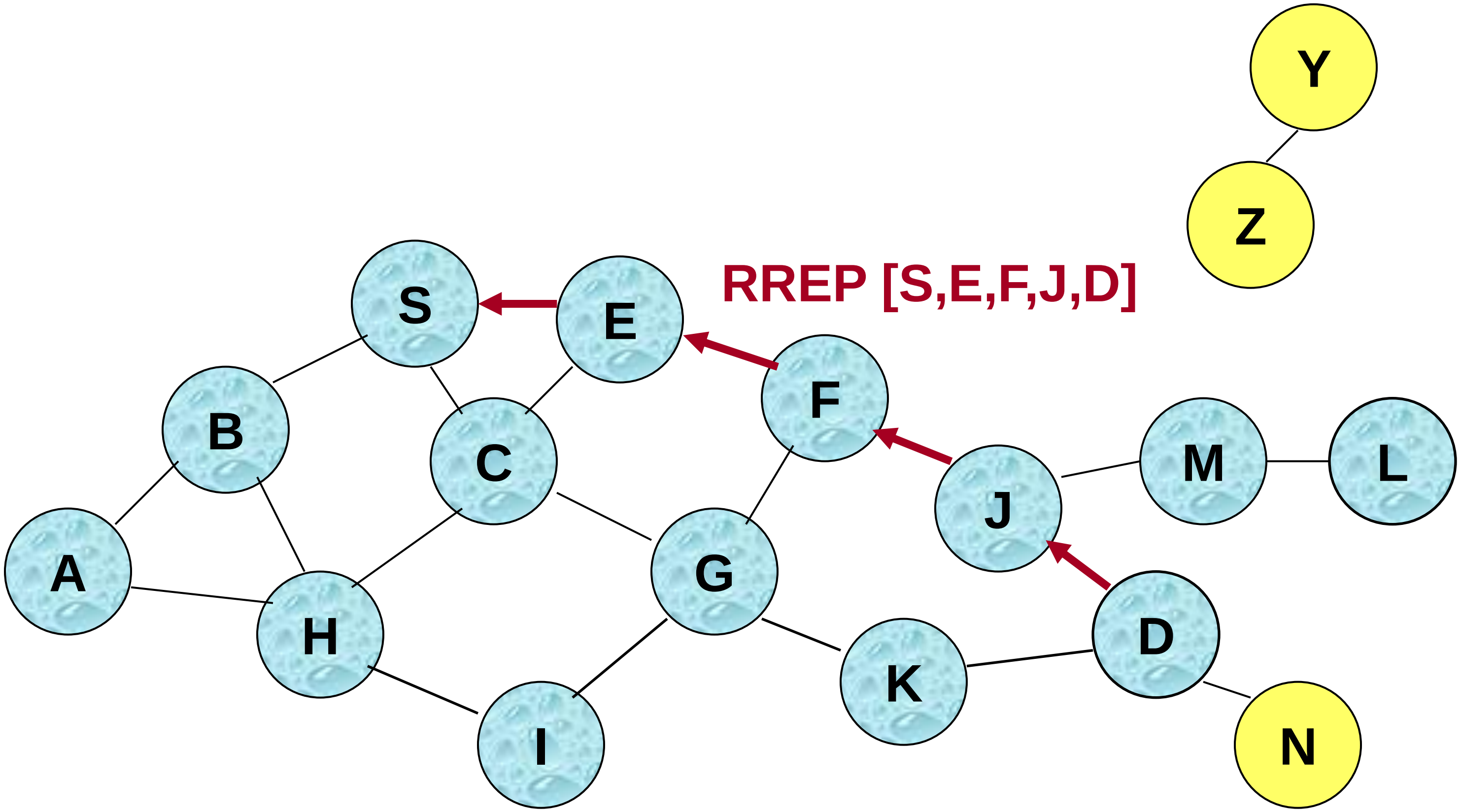
- Nodes J and K both broadcast RREQ to node D
- Since nodes J and K are hidden from each other, their transmissions may collide

Route Discovery in DSR



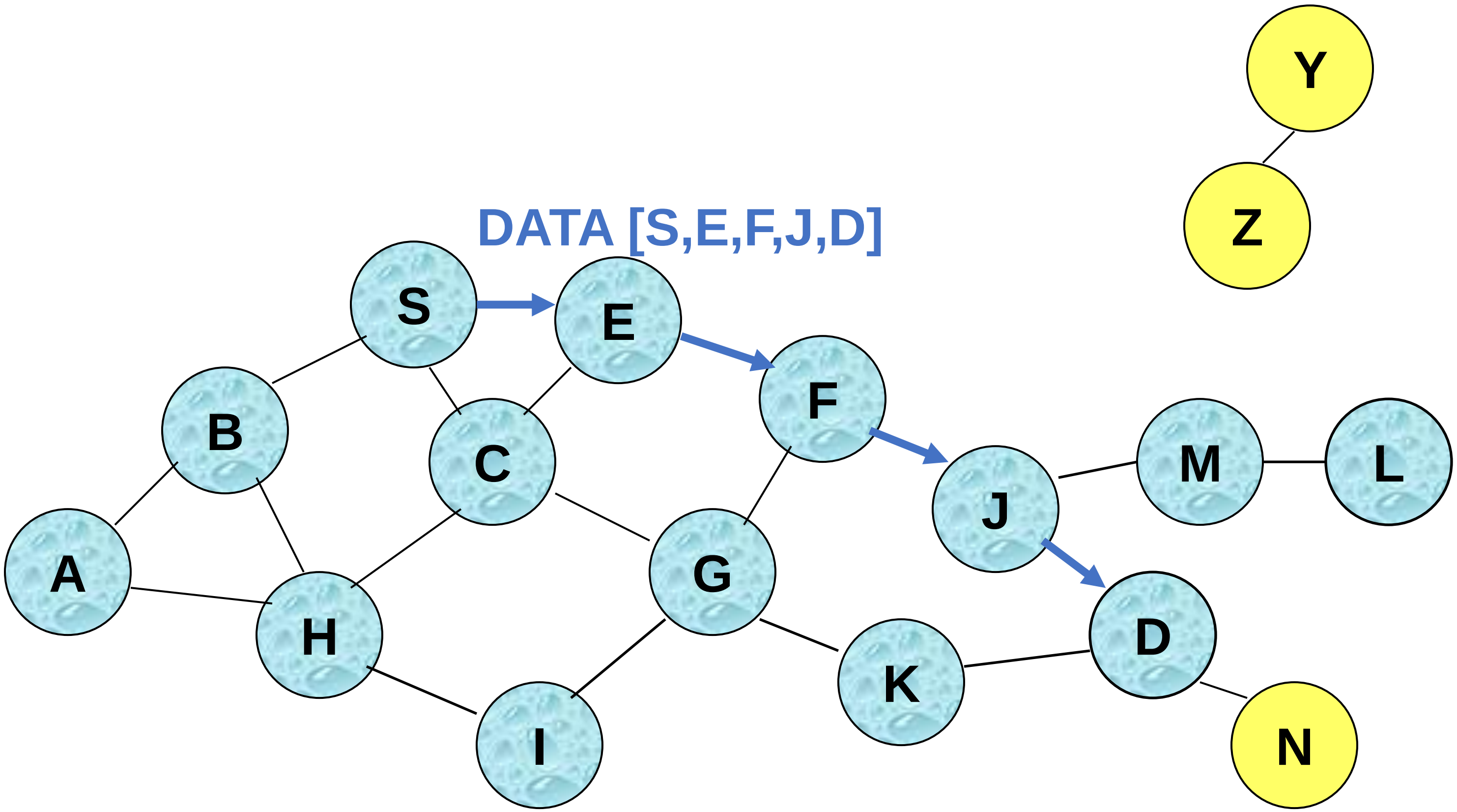
- **Node D does not forward RREQ, because node D is the intended target of the route discovery**

Route Reply in DSR



Represents RREP control message

Data Delivery in DSR



Packet header size grows with route length

Proactive vs Reactive

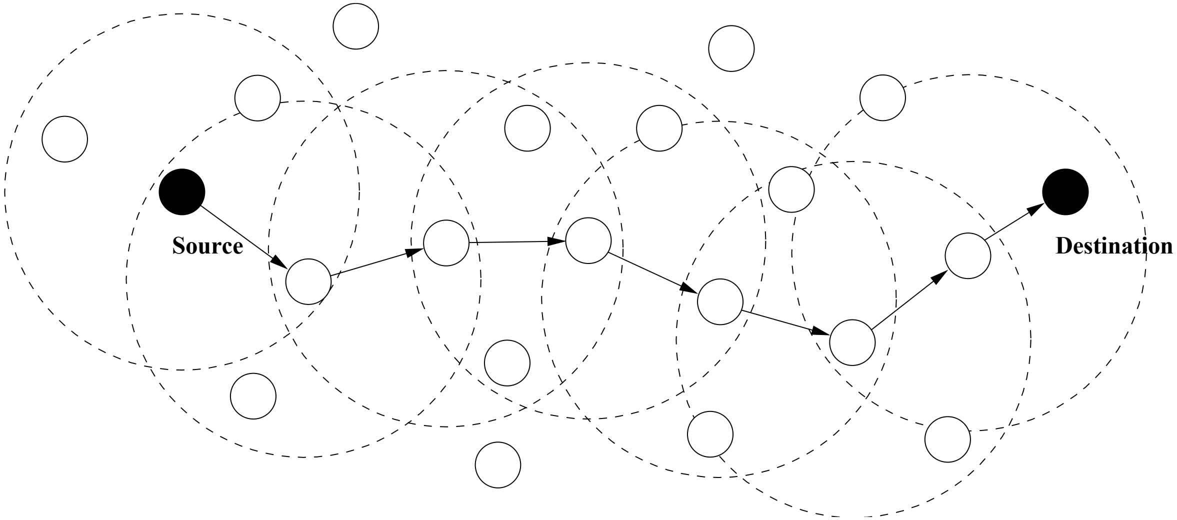
- Reactive:
  - Only establish/maintain routes between nodes needed them (in contrast: tables store ALL routes)
  - Store entire route in each message; message size grows with route length
  - Route requests cause “flooding”
- Proactive:
  - Route information always available; no need to search for route (but route information can be outdated)
  - Continuous exchange of route change updates

Geographic Routing

- Nodes use location information to make routing decisions
  - sender must know the locations of itself, the destination, and its neighbors
  - location information can be queried or obtained from a **location broker**
  - location information can come from GPS (Global Positioning System) or some other form of positioning technology.

Unicast Location-Based Routing

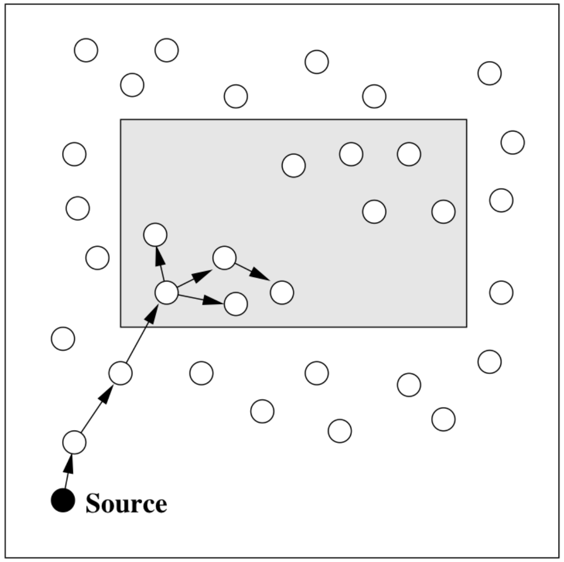
- One single destination
- Each forwarding node makes localized decision based on the location of the destination and the node's neighbors (**greedy forwarding**)
- Challenge: packet may arrive at a node without neighbors that could bring packet closer to the destination (**voids** or **holes**)



Geocasting

- Packet is sent to all or some nodes within specific geographic region
- Example: query sent to all sensors within geographic area of interest

- Routing challenge:
  - propagate a packet near the target region (similar to unicast routing)
  - distribute packet within the target region (similar to flooding)



Thank You

Contact me:

gauravsingal789@gmail.com

Gaurav.singal@nsut.ac.in

www.gauravsingal.in

LinkedIn: https://www.linkedin.com/in/gauravsingal789/

Twitter: https://twitter.com/gaurav_singal