IoT – design aspects

MOBILE AND CYBER-PHYSICAL SYSTEMS

STEFANO CHESSA

Learning objectives

IoT devices characteristics

- Each device is usually:
  - Low power, low cost system
  - Small
  - Autonomous

:) put them there, forget them respect on what's happening around

equipped with: Full microsys

1) Processor
2) Memory
3) Radio Transceiver
4) Sensing devices
   - Acceleration, pressure, humidity, light, acoustic, temperature, GPS, magnetic, …
5) Actuators – depending on the case
6) Battery, solar cells, .. wind turbines

freedom on deployment everywhere

Issues in IoT design

Energy efficiency
- Sensors are battery-powered or use energy harvesting => lifetime will be limited
- Need for HW/SW energy efficient solutions

Adaptatability to changing conditions
- Need for dynamic network management & programming

Low-complexity, low overhead protocols
- Need at any level of the protocol stack due to limitation of nodes' resources

Security
- At all layers ot the stack

Multihop communications
- Need for protocol stacks & routing protocols

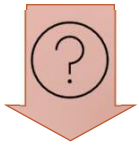
Mobility
- Need for dynamic routing protocols

Data storage & (pre-)processing

…

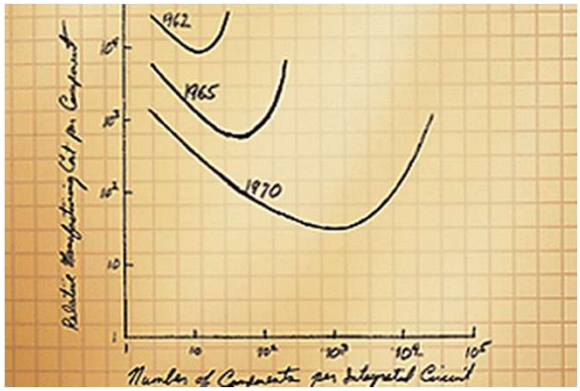
IoT design

Many limitations in the WSN design are related to processing, memory, battery and communication constraints



The evolution of HW technologies will overcome these constraints?

5

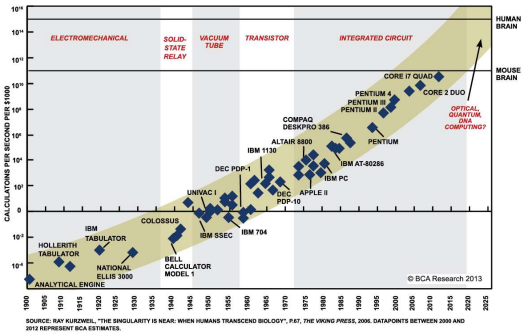


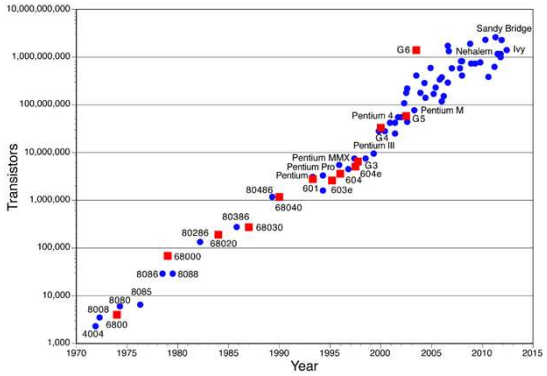
The Moore's law

"The number of transistors that can be (inexpensively) embedded in a chip grows exponentially"

(it doubles every two years)

6





Moore's Law

7

Moore's law growth

INTEL 4004 (1971)	INTEL CORE I9-7980XE (Q3'17)
2.3 K transistors	1.8 G transistors
740 KHz	4.4 GHz
	165 W power consumption
4KB program memory	128 GB

4

8

IoT * context of CS

The Moore's law and IoT

The Moore's law offers three different interpretations:

1. The performance doubles every two years at the same cost
   - Up to now this is true for processors of servers/desktops
2. The chip's size halves every two years at the same cost
   - Consequently also the energy consumption is reduced
3. The size and the processing power remain the same but the cost halves every two years

same rule ... power || space

9

The Moore's law and IoT

In IoT all the three interpretations are true…

There are applications that:

- Require small-sized sensors and/or that have low power consumption
- Require higher processing capabilities to the single sensor
- The cost is important in (almost) all applications
- considering (size > power || cost)

(in server performance is important, not critical in IoT but huge # of devices, sensing 60% of time => need to reduce costs

(performance) processing images is crucial for cameras

5

10

The Moore's law and IoT

Nowadays there exist several IoT HW platforms with different capabilities in terms of processing and energy consumption

Differently than server/desktop applications the IoT devices use low-power, cheap processors

- Even old or refreshed processors that are still on the market

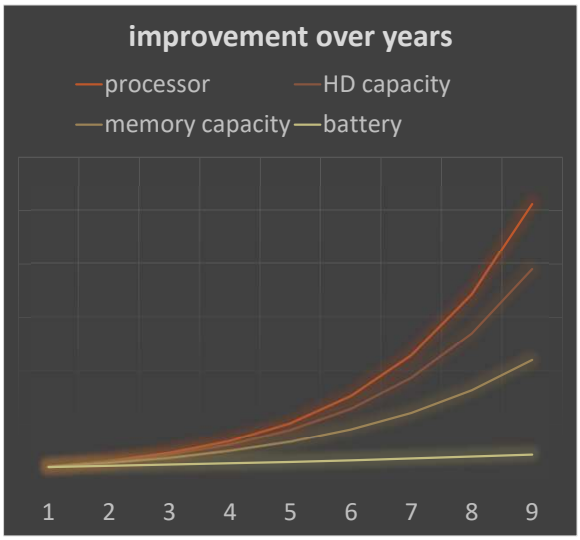
Normally used the cheapest HW that meets the application requirements

- considering the scale factor due to the large number of IoT devices, that has considerable effects on the final costs

11

Energy efficiency

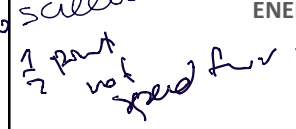
12



Energy efficiency

INTEL VS DURACELL

13

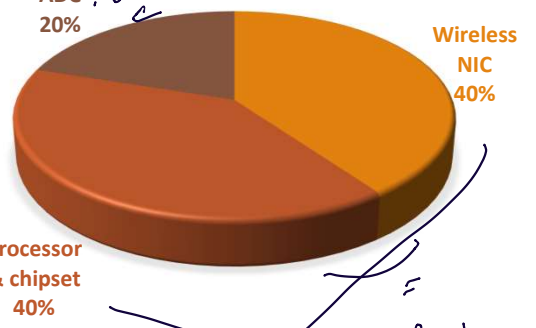


Energy efficiency

14

Energy efficiency

ENERGY USAGE OF A WIRELESS SENSOR



15

Energy efficiency - radio

Example of energy consumption of a WiFi Network Interface	
Sleep mode:	10mA
Listen mode:	180mA
Receive mode:	200 mA
Transmit mode:	280 mA

16

8

Energy efficiency - radio

Energy consumption of a sensor (Mote-clone)	
Sleep mode:	0.016 mW
Listen mode:	12.36 mW
Receive mode:	12.50 mW
Transmit mode	0.1 power level, 19.2kbps: 12.36 mW
	0.4 power level, 19.2kbps: 15.54 mW
	0.7 power level, 19.2kbps: 17.76 mW

17

Energy efficiency - radio

Energy consumption of a sensor (Mote-clone):

- In some cases transmit power < receive power!
- listen power ≈ receive power
- Radio should be turned off as much as possible

18

- Turn ON/OFF a component requires ENERGY and TIME
- IoT ≠ from LAPTOP: are not general purpose & real time interactions with humans
  => BUSINESS LOGIC (SW) for each specific application

Energy efficiency - processor

Processor power around 30%-50% of total power
- Processor as well should be turned off!

Turning on and off processor and radio consumes power as well…

19

- Duty cycle: 1) Read data (from sensor, "sense"): sampling physical data: efficient if periodically, at constant time.
2) Analyze: process & store
3) Transmit/receive
...
4) Wait for the next sample
...

Duty cycle

$$\text{Duty cycle} = \frac{\text{Active time}}{\text{Period of activity}}$$

Saving energy by reducing the period of activity of a sensor:
- The activity of a sensor is (mostly) repetitive:
  - Sense
  - Process & store
  - Transmit/receive
- A sensor alternates a period of activity and a period of inactivity (defines a duty cycle)
- During inactivity the energy consumption is very low
  - But processor, radio and I/O need to be freezed!

- activity of dev. is periodical
- period define duty cycle

10

20

can exploit this fact: put component in sleep mode (sensors, storage, radio, processor)

2 prob.
1) How save energy ⇒ during inactivity period (during duty cycle and waiting in next period)
2) Exception when you want NETWORK ACTIVE (always)

Example of duty cycle: code

Lasts 4 milliseconds → processor and sensor active

Lasts 1 milliseconds → only processor active

Lasts 15 milliseconds → processor and radio active (tx)

Lasts 380 milliseconds → All components idle

Total: 400 milliseconds

```
...
void loop() {
 // reads the input from analog pin 0:
 int sensorValue = analogRead(A0);

 // converts value into a voltage (0-5V):
 float voltage = sensorValue * (5.0/1023.0);

 // transmits voltage over the radio
 Serial.println(voltage);

 // waits for next loop
 delay(380);
}
```

**NOTE: the milliseconds are not the real ones… they are just to illustrate the concept

Example of duty cycle: code

However, this code does not turn off any component when not in use…

Let's say you have calls like:

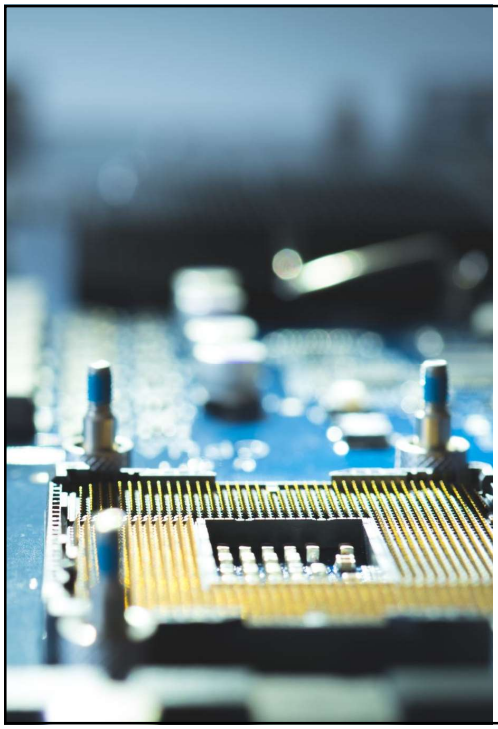
- `turnOn(x)` / `turnOff(x)` that turn on/off the component `x`
- `Idle(y)` that puts the processor in idle state (with low power consumption) for `y` milliseconds

```
void loop() {
 // reads the input from analog pin 0:
 turnOn(analogSensor);
 int sensorValue = analogRead(A0);
 turnOff(analogSensor);

 // converts value into a voltage (0-5V):
 float voltage = sensorValue * (5.0 / 1023.0);

 // transmits voltage over the radio
 turnOn(radioInterface);
 Serial.println(voltage);
 turnOff(radioInterface);

 // waits for next loop
 idle(380);
}
```



Question

What is now the duty cycle of the processor, of the radio and of the sensor?

4 milliseconds
1 milliseconds
15 milliseconds
380 milliseconds

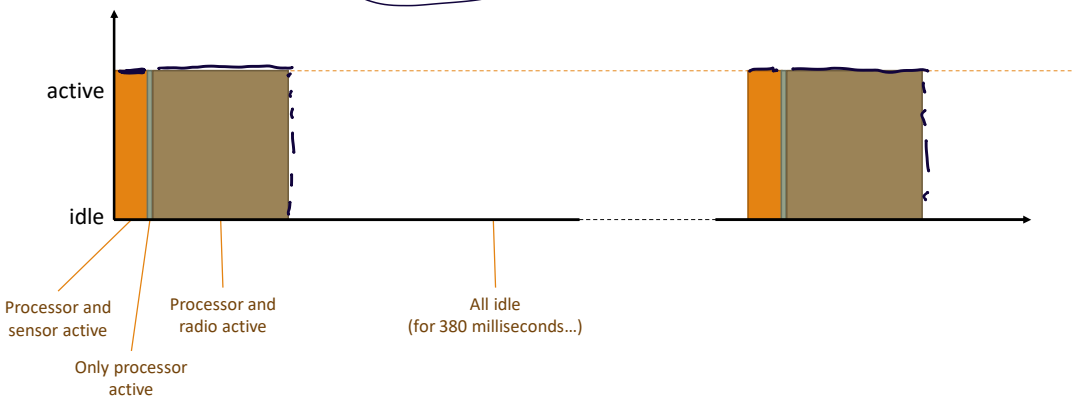
```
...
void loop() {
  turnOn(analogSensor);
  int sensorValue = analogRead(A0);
  turnOff(analogSensor);

  float voltage = sensorValue * (5.0 / 1023.0);

  turnOn(radioInterface);
  Serial.println(voltage);
  turnOff(radioInterface);

  idle(380);
}
```


Example of duty cycle: diagram



current awarded by each component in the main SENSEY:

Specs. of a mote class-sensor

	Value	units
Micro Processor (Atmega128L)		
current (full operation)	8	mA
current sleep	15	µA
Radio		
current in receive	19,7	mA
current xmit	17,4	mA
current sleep	20	µA
Logger (storage in the flash memory)		
write	15	mA
read	4	mA
sleep	2	µA
Sensor Board		
current (full operation)	5	mA
current sleep	5	µA
Battery Specifcations		
Capacity Loss/Yr	3	%

- battery: keeps charge
  part of energy is lost cuz components exist

25

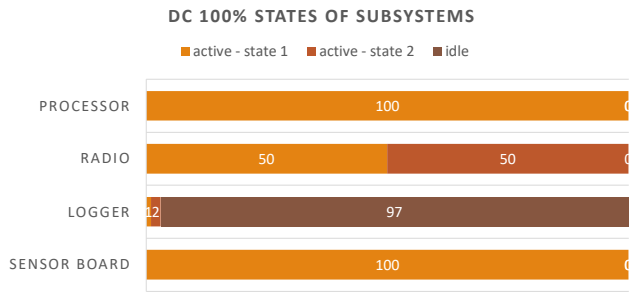
Example: two DC models

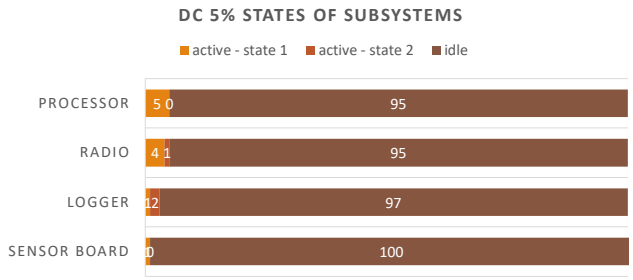
	model 1: 100%DC	model 2: 5%DC	units
Micro Processor			
current (full operation)	100	5	%
current sleep	0	95	%
Radio			
current in receive	50	4	%
current xmit	50	1	%
current sleep	0	95	%
Logger			
write	1	1	%
read	2	2	%
sleep	97	97	%
Sensor Board			
current (full operation)	100	1	%
current sleep	0	99	%

Some components work in parallel

Logger duty cycle of 3%

26





Example: two DC models

27

Measuring energy – a note

Energy and power are measured in Joule (J) and Watt (W), respectively:

$$1J = 1W \cdot sec$$

In electromagnetism, $1W$ is the work performed when a current of $1Ampere$ $(1A)$ flows through an electrical potential difference of $1Volt$ $(1V)$:

$$1W = 1V \cdot 1A$$

Since we use direct current and the electrical potential difference is (almost) constant, the power and the energy only depend on the current (Ampere).

Hence we can «express» both the energy stored in a battery (battery charge) and the energy consumed in mAh

28

Example: computing consumption per duty cycle

- Energy cost microprocessor E_μ (per cycle):

$$E_\mu = C_\mu^{full} \cdot dc_\mu + C_\mu^{idle} \cdot (1 - dc_\mu)$$

  - C_μ^{full} full energy cost microprocessor
  - C_μ^{idle} idle energy cost microprocessor
  - dc_μ % duty cycle microprocessor

- Energy cost radio E_ρ (per cycle):

$$E_\rho = C_\rho^T \cdot dc_\rho^T + C_\rho^R \cdot dc_\rho^R + C_\rho^{idle} \cdot \left(1 - dc_\rho^T - dc_\rho^R\right)$$

  - C_ρ^T radio transmission energy cost
  - C_ρ^R radio receival energy cost
  - C_ρ^{idle} idle energy cost
  - dc_ρ^T % transmit duty cycle radio
  - dc_ρ^R % receive duty cycle radio

29

Example: computing lifetime

- Energy cost logger E_λ and sensor board E_σ (same as before)
- Total energy cost (per duty cycle):

$$E = E_\mu + E_\rho + E_\lambda + E_\sigma$$

- Lifetime (in number of duty cycles):

$$Lifetime = \frac{B_0 - L}{E}$$

  - Where B_0 is the initial battery charge and L is the battery charge lost during the lifetime due to **battery leaks**
  - Note that L depends on lifetime!
  - Based on the charge loss/cycle ε

30

Example: computing lifetime

- We thus have a recurrence equation:

$$B_n = B_{n-1} \cdot (1 - \varepsilon) - E$$

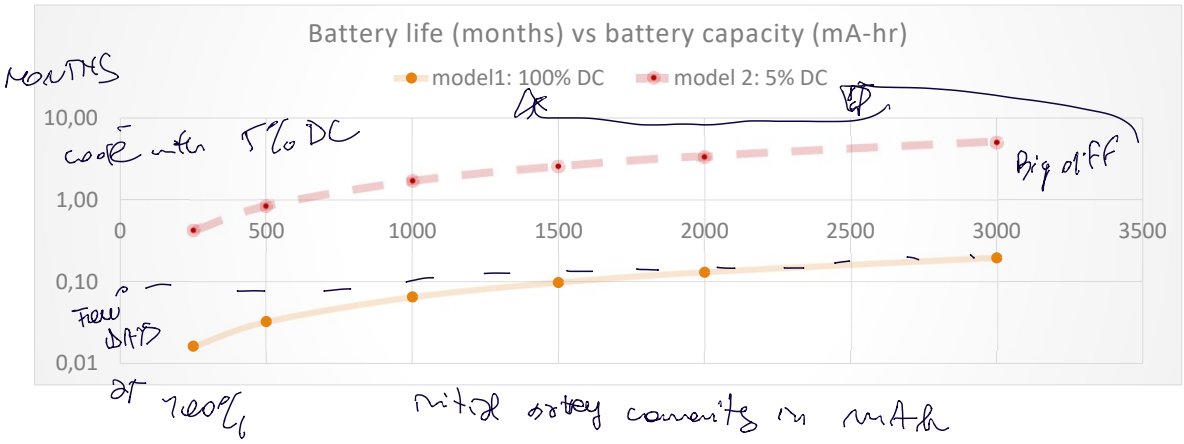
  - Where B_n is the battery charge at cycle n

- Solving the recurrence equation:

$$B_n = B_0 \cdot (1 - \varepsilon)^{n-1} + \frac{E\left((1 - \varepsilon)^n - 1\right)}{\varepsilon}$$

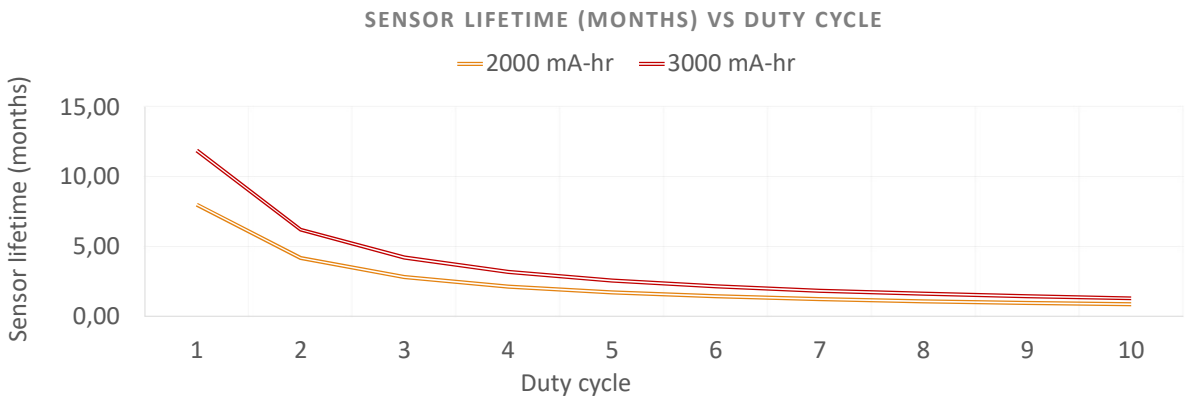
- Device lifetime is given by $n : B_n = 0$
  - … in practice the device stops working before than that…

31



Battery life vs DC (I)

32



Battery life vs DC (II)

33

Energy efficiency

- Hence the solution is to reduce the DC

However:

Turning off the processor is a local decision
- the node scheduler knows what are the activities and when the processor should run

Turning off the radio is a global decision:
- A device with its radio off does not communicate
- Cannot receive incoming messages/commands
- Cannot act as router in multihop network…
  … cannot receive requests, commands, updates…

17

34

MAC Protocols

Low-level communication protocols
- send/receive packets to/from in-range sensors

In conventional networks, MAC protocols arbitrate the access to the shared communication channel

In IoT they also implement strategies for energy efficiency
- synchronize the devices
- turn off the radio when it is not needed
  - turning off the radio means excluding a device from the network

35

Exercise

Consider this program and the table of energy consumption in the different states. Compute:

- the energy consumption of the device per single hour
- the expected lifetime of the device

```
…
void loop() {
 turnOn(analogSensor);
 int sensorValue = analogRead(A0);
 turnOff(analogSensor);

 float voltage = sensorValue*
         (5.0 / 1023.0);

 turnOn(radioInterface);
 Serial.println(voltage);
 turnOff(radioInterface);

 idle(380);
}
```

4 milliseconds

1 milliseconds

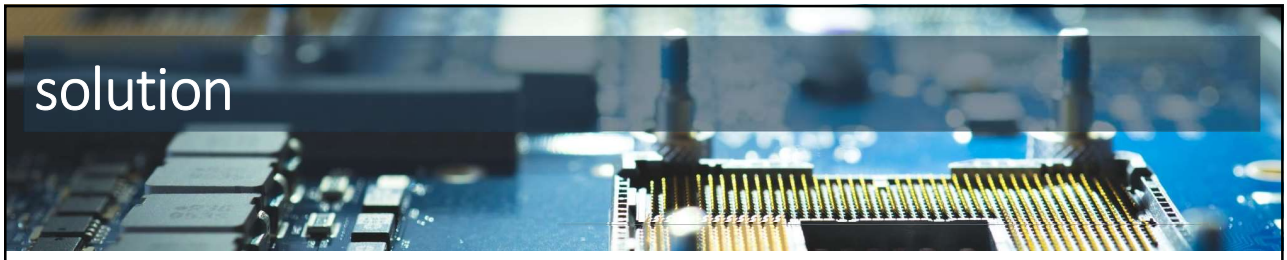
15 milliseconds

380 milliseconds

	value	units
Micro Processor (Atmega128L)		
current (full operation)	8	mA
current sleep	15	µA
Radio		
current xmit	1	mA
current sleep	20	µA
Sensor Board		
current (full operation)	5	mA
current sleep	5	µA
Battery Specifcations		
Capacity	2000	mAh

36

solution



energy consumption per hour:

- The processor has a duty cycle of _____
- The radio has a duty cycle of _____
- The sensor has a duty cycle of _____

Energy consumption in one hour: _____

Lifetime of the device: _____

	idle	Active
Processor		
Radio		
Sensor		
Total:		

	value	units
Micro Processor (Atmega128L)		
current (full operation)	8	mA
current sleep	15	µA
Radio		
current xmit	1	mA
current sleep	20	µA
Sensor Board		
current (full operation)	5	mA
current sleep	5	µA
Battery Specifcations		
Capacity	2000	mAh

37

Summary

- Issues in IoT design
- Energy efficiency
- Duty Cycle
- Energy consumption of an IoT device
- Lifetime of an IoT device

40

19