IoT Framework: Principles & Practices

Fundamentals of Internet of Things

Andrea Omicini
andrea.omicini@unibo.it

Bologna Business School, Università di Bologna

Master in Digital Technology Management 2020/2021

Disclaimer

- most of the ideas here are extracted from [Chou, 2016]
- however, any mistake should be attributed to the author of these slides

Next in Line. . .

Economics

A fundamental shift in the economics of computing

- the *cost* of *computing resources* is no longer the purchase price
- the cost of computing resources comes from managing the security, availability, performance and change in the application and all of the supporting hardware and software
  - rule of thumb: managing an enterprise application costs four times the purchase price every year
- cause: *standardisation* and *automation* in enterprise application cloud services—cutting out human labor

Not Internet of People (IoP)

Things are not people

- second-generation enterprise software was focussed on *people*—individuals or groups
- to help them perform activities within the organisation
- however, things are not people
  - many more things than people
  - things can tell you more than people—as they can have as many sensors as you need them to
  - things can talk constantly—people, well, at least *most* of them can not

Next Generation Enterprise Software

Beyond transformation

- first- and second-generation enterprise software focussed on *automation* of organisation
  - with different delivery model, see SaaS (software as a service)
- most activities were automated and made more efficient and controllable
  - hardly *transformed*, however
- is third generation going to actually affect and change the nature of activities?

Internet of Things

So many different business-oriented acceptations

- we will precisely define what we mean for IoT later in the course
  - at least technically
- first, we will introduce a *conceptual framework* to provide a well-grounded overview of IoT nowadays

Next in Line. . .

Five Layers of IoT I

The IoT framework [Chou, 2016]

Five Layers of IoT II

- **Things** — the main bricks
- **Connect** — how to put things together
- **Collect** — large amounts of data
- **Learn** — infer from data
- **Do** — business outcome

Things

Things, devices, machines, . . .

- things that *compute* and *connect*
- things that *perceive* and *change*
- with increasingly faster computing and networking capabilities
- with a growing range of sensors and actuators to interact with their surrounding environment
- with software—power and vulnerability, at the same time

Connect

A diverse set of networking technologies and protocols

- depending on the the specific scenario
- which things need to *communicate* to which things
- how much data needs to be *transmitted*
- how far it needs to go
- how much power is available, how much is needed

Collect

Data from things

- a growing amount of data comes from things
- orders of magnitude larger than typical Internet applications
- that needs to be *collected* and *stored*
- using classical DB technologies
    - SQL, NoSQL databases, time series management systems, ...

Learn

Learning from data

- huge data collection can be analysed
- *inferring new knowledge*
  - via data analysis techniques, or machine learning (ML)
- either offline or online

Do

Business application

- to do something based on all data
- both acquired and inferred
- to build applications
- based on computing and connected objects

Next in Line...

Focus on. . .

Sensors

In touch with the surrounding environment

- sensors are beginning to follow Moore's Law and becoming less and less expensive each year
- sensors are more and more attached to low-cost devices
  - including small devices
- a sensor is a technical component that *directly perceives* some feature of the surrounding environment, not (yet) encoded in any way
  - so, a connection to a time server is *not* a time sensor

Sensors: Examples I

Accelerometer

- acceleration is a change in velocity
- an accelerometer is an electromechanical sensor used to detect and measure acceleration forces
- either *static* – e.g., gravity – or, *dynamic*—to sense movement or vibrations
- used to e.g., detect free fall, or, change in the device orientation along the three spatial axes

! often associated with *gyroscopes*—another device/sensor to measure/maintain orientation and angular velocity

Sensors: Examples II

Magnetometer

- a magnetometer measures the intensity and the direction of the magnetic field
  - both global (Earth's magnetic field) and local (metallic objects)
- so, it can be used both for location-based application and for metal detection

Sensors: Examples III

Proximity sensor

- proximity sensors exploit reflection
  - typically, of infrared light
- objects in the vicinity return a ray of infrared light, which is detected by the sensor
- so that a device can understand if it is near to some other physical object
  - e.g., to automatically turn off the screen of your phone when you are calling

Light sensor

- light sensors measure the brightness of the ambient light
  - e.g., to automatically adjust the brightness of a device screen and adapt to the human vision

Sensors: Examples IV

Thermometer

- a thermometer obviously measure temperature
  - either external or internal one
- e.g., can be used to detect overheating of devices

Pedometer

- a pedometer counts the number of steps taken by the user of a device
- more accurately and efficiently than using an accelerometer
- they are typically found in wearables such as tracker and smartwatches

Computer Architecture

Smart things compute

- smart objects contain some kind of central processing unit (CPU) capable of running some software
- constraints – power consumption, physical size and cost – lead to treadoffs
- first tradeoff: CPU basic architecture
- two approaches
  - at the basic level, *microcontrollers* – microprocessor with RAM, ROM, I/O ports – with simple instruction sets, limited access to memory storage and generally lower power consumption—e.g., Arduino, ATmega by Amtel
  - at the next level, processors with full instruction set yet with few transistors and low power consumption—e.g., *ARM processors* (Reduced Instruction Set Computer, RISC) such as Raspberry and Apple M1

Software I

Programming things

- *software* can add specialised behaviour to a smart thing
- different operating system or run-time environments can be used for executing software on a particular thing

Memory footprint

- *memory footprint* of an executable program is the runtime memory required by program *execution*
- it is not (mostly) the program itself
- rather, the structures introduced by the run-time environment can increase the footprint
  - e.g., in a Java program, the memory footprint is predominantly made up of the Java Virtual Machine (JVM) run-time environment (JRE)

Software II

Software development environment

- a *software development environment* is software that provides comprehensive facilities to computer programmers for developing (other) software
- sometimes called an integrated development environment (IDE)
- it usually includes, at least
  - a source-code editor
  - build automation tools
  - a debugger
- e.g., for programming the microcontrollers, the Arduino platform provides an IDE, which includes support for the C, C++ and Java programming languages

Software III

Operating system

- an operating system (OS) contains software that provides features and services commonly used by many different applications
  - e.g., Windows, UNIX, Linux, Mac OS X are well-known examples of general-purpose OS
- however, humans may have uses for objects that require responses in *real time*
- special-propose operating systems might be useful

Software IV

Real-time operating system

- a *real-time operating system* (RTOS) is an operating system intended to serve time-critical applications
- there, processing time requirements are measured in tenths of seconds or shorter
- application tasks are accepted and completed within a certain pre-defined amount of time
- a RTOS is valued more for how quickly or how predictably it can respond, rather than for the amount of work it can perform in a given period of
- e.g., Wind River's VxWorks

Security

Securing things

- *security* and *integrity* of software
- bad software: e.g., software viruses
- good software: updates and patches to correct software and keep it working
- critical issue: *boot*
  - the booting process stars when a machine is powered up, and loads the first software required (boot)
  - software loaded during start-up should only authorised one—e.g., digitally-signed

Packaging

Putting chips within things

- *packaging* is a major discipline within the field of electronic engineering
- it includes a wide variety of technologies
- packaging must consider protection from mechanical damage, cooling, radio frequency noise emission and electrostatic discharge
- industrial equipment made in small quantities may use standardised, commercially-available enclosures such as card cages or prefabricated boxes
- mass-market consumer devices may have highly-specialised packaging to increase consumer appeal

Packaging: An Example

System-on-a-chip (SoC)

- one of the packaging technologies used in high volume is called system-on-a-chip (SoC)
- a SoC is an integrated circuit that combines all components of a computer into a single semiconductor chip
- it may contain digital, analog, mixed-signal and often radio-frequency functions
- quite similar to microcontroller, actually—not so well-defined term

Focus on. . .

Wind Turbines

Turbines with sensors and servers

- next-generation power grids will include renewable power generation
- a large *wind turbine* is a huge machine, costing million euros, weighting several tons, generating megawatts
- no surprise they come with computer servers, have hundred sensors, and are typically interconnected with each other
- *sensors* provide information such as wind speed and direction, blade-rotation speed, power generated, component temperature, vibration, noise levels, ...
- data are sampled every few seconds, shipped into the cloud once every ten minutes over a VPN-protected network
- full-fledged OS like Windows are used, along with special software ensuring safety and security—e.g., self-diagnosis

Agricultural Machines

Farming machines

- farmers use many specialised machines
  - e.g., seed drills, fertiliser spreaders, combine harvesters
- increasingly, all this equipment is outfitted with sensors—for both machine data and agronomic data
  - e.g., oil pressure, header height, rotor speed, moisture content, nitrogen level
- more and more, agricultural machines are connected with the cloud, with specific subscription services for the farmers and the dealers
  - e.g., for getting machinery ready for the season before the season actually starts
- also, services may ensure that machines are set up to accurately record the agronomic data

Clinical Haematology Analysers I

- general scenario: Internet of Healthy Things (IoHT)
- haematology analysers are used widely in patient and research settings to count and characterise blood cells for disease detection and monitoring
- from basic
  - e.g., providing complete blood count (CBC) with a three-part differential white blood cell (WBC) count

  to advanced
  - e.g., measuring cell structure and detecting small cell populations to diagnose rare blood conditions

Clinical Haematology Analysers II

- they use physical technology
  - e.g., electrical impedance, flow cytometry, fluorescent-flow cytometry

  in combination with chemical reagents that alter blood cells to extend the measurable parameters
- they can provide large amounts of data for each patient, and store data for many patients; e.g.
  - Sysmex XE-5000 reports 31 blood parameters, 150 samples/hour
  - a small benchtop analyser can save 1,000 patient results

High Volume Mail Inserters

Stupid work that humans do not like—and, don't do well

- still "snail mail" is used a lot in some massive case—e.g., banks
- Epic Inserter (by Pitney Bowes) can stuff envelopes (22,000/hour), while allowing quick changeovers to different format types
- inserters controlled by dozens of servomotors; have hundreds of sensors to monitor the flow of mail and feedback the control system
- specifically, some sensors provide information on motor RPM, motor shaft position, velocity of mail material, position and skew of mail material, material thickness and weight, air pressure, and vacuum levels where air is used to control the machine
- sensors are all attached to local computers and storage, which are one or more industrial PCs mounted inside the machine and networked together to share data and synchronise control of the machine

Next in Line...

Networking

Connecting things is about networking

- this comes from the miniaturisation of antennas
- which can now be easily embedded into physical objects

! some fundamentals of networking is required

Main Issues

Choosing technology based on data to transmit

- size
- distance
- energy

Focus on. . .

Oxford Dictionary

The notion of network

http://en.oxforddictionaries.com/definition/network

...

2. A group or system of interconnected people or things.
   "the company has a network of 326 branches"
   "a trade network"

Human Networks I

The notion of network

A network consists of *two or more* entities, or objects, sharing *resources* and *information*

- *multiplicity* of entities
- that *share*—either give or get
- sharing is typically (by default) *bi-directional*

Examples

- *family* network
- community / *peer* network
- *networks of networks*: family networks within the community network

Human Networks II

Client-server networks

E.g., the restaurant network

- a customer is the *client* of the food & drink prepared by the restaurant
- the waiter is the *server* providing clients with access to resources—providing information on available resources, receiving requests, and possibly satisfying them

Contact networks

- networking is nowadays considered as the most relevant premise to career development
- accumulating and nurturing contacts is probably the most powerful tool to make the best of one's professional experience and skills

Basics Elements [Kizza, 2015]

The notion of network

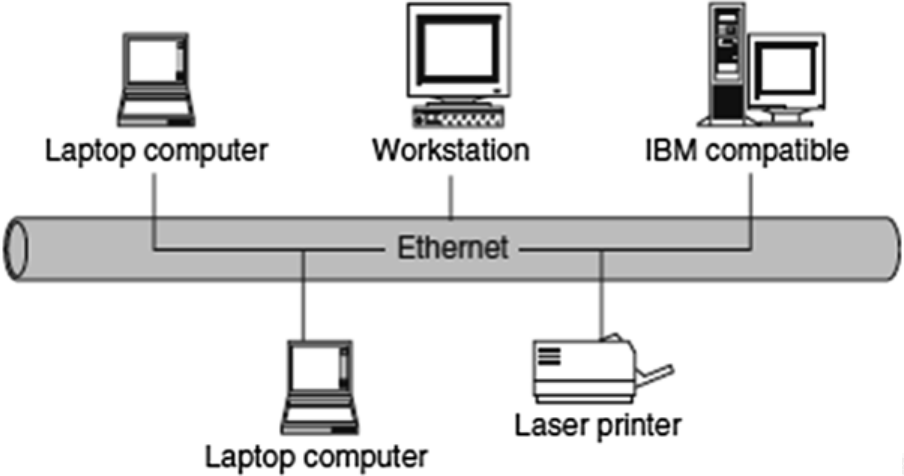
- sender & receiver
- (transmission, communication) medium, where sharable item can be channeled
- agreed-on rules of communication, or, protocols

Computer Network I

The notion of network

A *computer network* is a distributed system consisting of loosely-coupled computers and other devices.

Computer Network II

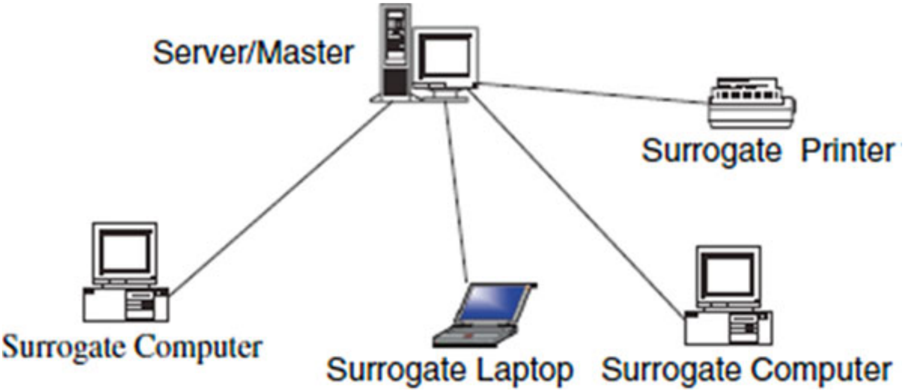


[Kizza, 2015]

Computer Network Models I

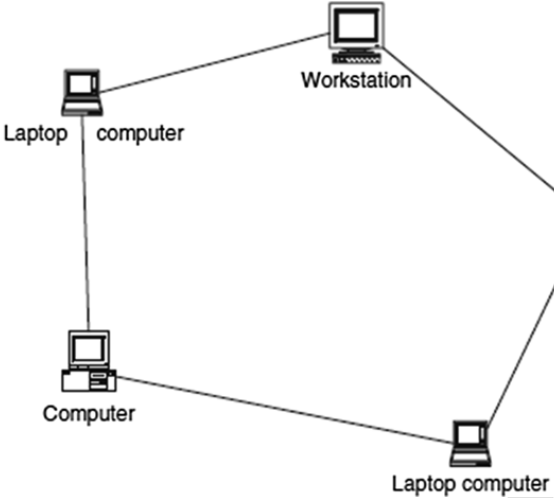
- depending on the network structure, there are several *models of networks*.
- the main ones are
  - *centralised* model
  - *distributed* model

Computer Network Models II



Centralised network [Kizza, 2015]

Computer Network Models III



Distributed network [Kizza, 2015]

The Size of a Network I

Different types on networks

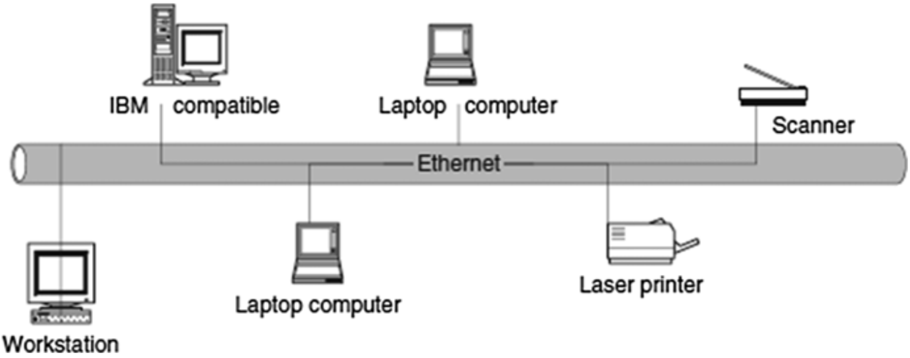
- each network is a cluster of network elements and their resources
- coming in different sizes
- determining the network type
- the main ones
  - LAN
  - WAN

Local Area Networks (LAN) I

Networks in a small geographical area

- e.g., a building floor, a building, a few adjacent buildings
- benefits
  - since all network elements are close together, data move fast through communication links
  - since communicating elements are near, high-cost and high-quality communicating elements can be used

Local Area Networks (LAN) II



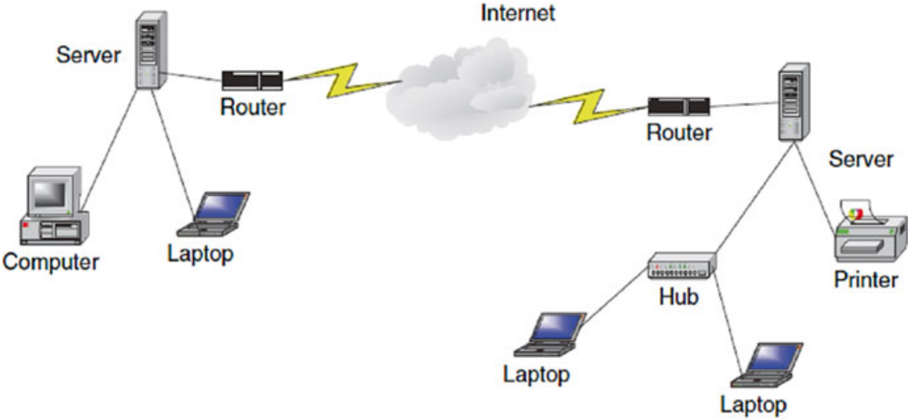
[Kizza, 2015]

Wide Area Networks (WAN) I

Devices or clusters scattered in a wide geographical area

- e.g., a region of a country, across the whole country, several countries, the entire globe like the Internet
- benefits
    - distributing services to a wider community
    - availability of a wide array of both hardware and software resources that may not be available in a LAN
- drawbacks
    - communication media may be slow and often unreliable

Wide Area Networks (WAN) II



[Kizza, 2015]

Metropolitan Area Networks (MAN) I

In the middle between LANs and WANs

- middle network: wider than LANs, smaller than WANs
  - e.g., civic networks
- e.g., civic networks

Basic Elements of Transmission

The performance of a network type depends on

- transmission *technology*
- transmission *medium*

Transmission Technology

- *transmission* is the propagation and processing of data signals between network elements
- *representation* of data for transmission is the *encoding scheme*
- two encoding schemes: analog and digital

Transmission Medium I

- characteristic quality, dependability, and overall performance of a network depend on its *transmission medium*
- transmission medium determines expected network traffic, reliability, size, transmission rate
- transmission media: wired and wireless

Transmission Medium II

Wired

- copper wires
- twisted pair
- coaxial cable
- optical fiber

Wireless

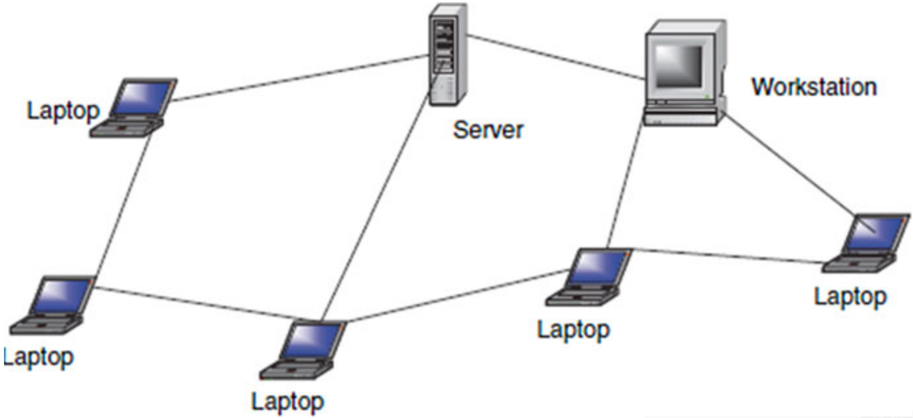
- infrared
- high-frequency radio
- microwaves
- laser

Network Topology

Computer networks, whether LANs, MANs, or WANs, are constructed based on a topology. There are several topologies including the following popular ones

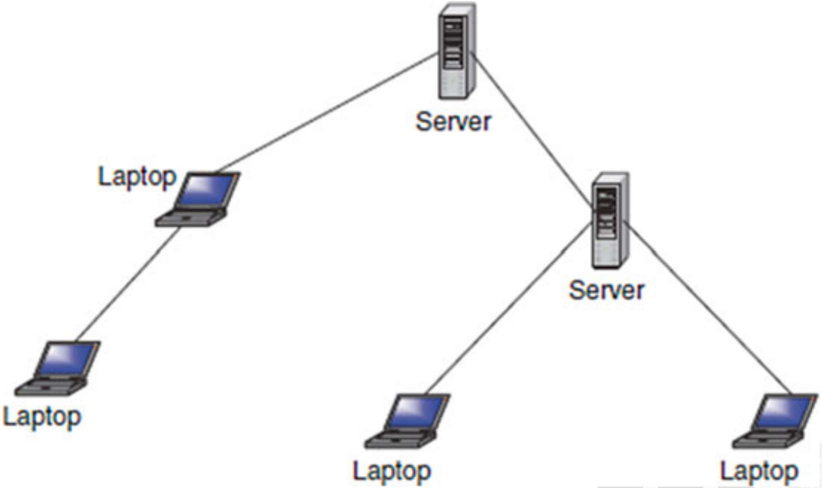
- mesh
- tree
- bus
- star
- ring

Mesh



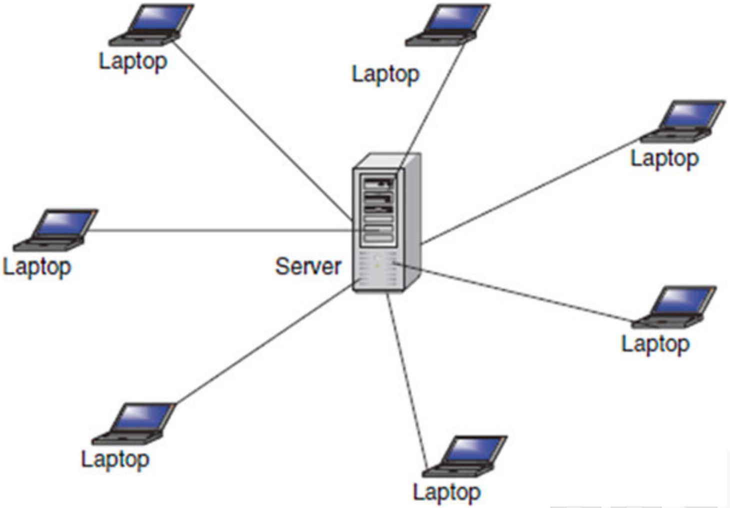
[Kizza, 2015]

Tree



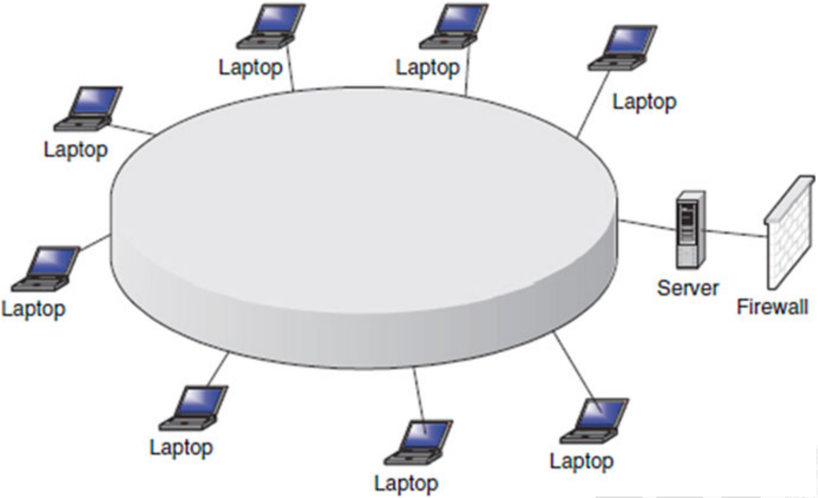
[Kizza, 2015]

Star



[Kizza, 2015]

Ring



[Kizza, 2015]

Standards

- historically, networks grew along with the need to connect different sorts of computers
- independently of diversity
- *standards* were required to make diverse computers interconnect

OSI I

OSI & ISO

- to help computers communicate, the *International Organization for Standardization* (ISO, http://www.iso.org) developed the *Open System Interconnection* (OSI) model
- OSI is an *open architecture model*—ISO/IEC 7498, four parts

ISO/IEC 7498-1 basic model
https://www.iso.org/standard/20269.html

ISO 7498-2 security architecture
https://www.iso.org/standard/14256.html

ISO/IEC 7498-3 naming and addressing
https://www.iso.org/standard/25022.html

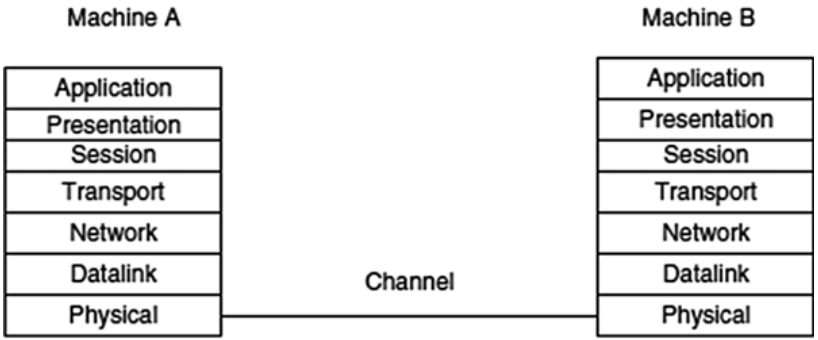
ISO/IEC 7498-4 management framework
https://www.iso.org/standard/14258.html

OSI II

Layers

- OSI model based on the idea that communication over a network can be broken into *seven layers*
- each layer represents a different portion of the communication task
- different layers of the protocol provide different services
- each layer can communicate only with its own neighbouring layers
  - e.g., the protocols in each layer are based on the protocols of the previous layers

OSI III



ISO logical peer communication model [Kizza, 2015]

TCP/IP I

Transport Control Protocol/Internet Protocol (TCP/IP) Model

- the OSI model works as the network communication protocol standard
- however, it is not the most widely used one
- the Transport Control Protocol/Internet Protocol (TCP/IP) model is the *de facto* standard
- developed for the US Department of Defense Advanced Research Projects Agency (DARPA)

TCP/IP II

Layers

- application
- transport
- network
- data link
- physical

TCP/IP III

Layer	Delivery unit	Protocols
Application	Message	Handles all higher-level protocols including File Transfer Protocol (FTP), Name Server Protocol (NSP), Simple Mail Transfer Protocol (SMTP), Simple Network Management Protocol (SNMP), HTTP, Remote file access (telnet), Remote file server (NFS), Name Resolution (DNS), HTTP, TFTP, SNMP, DHCP, DNS, BOOTP
		Combines application, session, and presentation layers of the OSI model
		Handles all high-level protocols
Transport	Segment	Handles transport protocols including Transmission Control Protocol (TCP), User Datagram Protocol (UDP)
Network	Datagram	Contains the following protocols: Internet Protocol (IP), Internet Control Message Protocol (ICMP), Internet Group Management Protocol (IGMP)
		Supports transmitting source packets from any network on the internetwork and makes sure they arrive at the destination independent of the path and networks they took to reach there
		Best path determination and packet switching occur at this layer
Data link	Frame	Contains protocols that require IP packet to cross a physical link from one device to another directly connected device
		It included the following networks
		WAN – wide area network
		LAN – local area network
Physical	Bit stream	All network card drivers

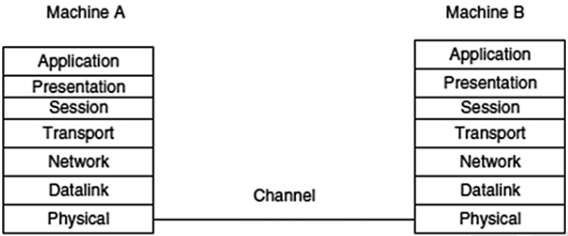
[Kizza, 2015]

Focus on. . .

Layers for the IoT I

Layered architecture

- each layer serves the layer above
- and is served by the layer below
- each layer provides and ensures some specific features



[Kizza, 2015]

Layers for the IoT II

ISO/OSI for the IoT

- a useful framework to understand the diverse levels of connectivity
- front the physical to the application one

Data-link Layer

The data-link layer. . .

- provides node-to-node *data transfer*, which is a link between two directly connected nodes
- detects and possibly *corrects errors* that may occur in the physical layer
- defines the *protocol* to establish and terminate a connection between two physically connected devices and defines the protocol for flow control between them
- is responsible for controlling how devices in a network gain access to data and permission to transmit it
  - e.g., Ethernet, WiFi, ZigBee all encompass operation at the data-link layer

Range vs. Power I

Wireless as the most general option

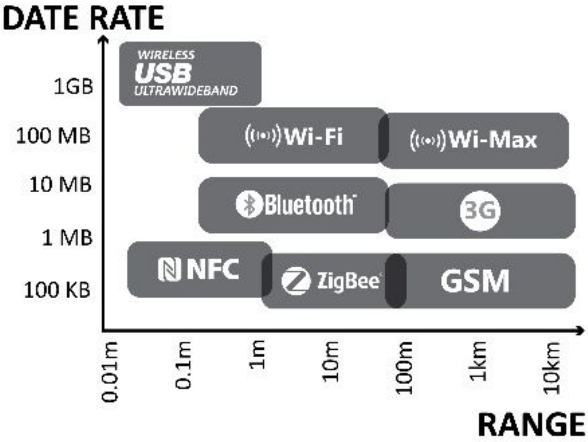
- tradeoff between communication range and power consumption
- basically because of the dependancy between distance and energy at the physical level
  - energy required for transmission is proportional to the *square* of the distance to reach
- many alternatives: WiFi, 3G, 4G, ZigBee, Bluetooth, NFC, ...
- choice: based on power consumption, range requirements, data rates, costs, size of antenna and the environment

Range vs. Power II

	WIFI	ZigBee	Bluetooth	NFC
POWER	High	Low	Classic: Mid LE/Smart: Low	Tag: Zero Reader: Very Low
RANGE	30-100 m	10-20 m	10 m	<0.1 m

[Chou, 2016]

Range vs. Data Rate I



[Chou, 2016]

Range vs. Data Rate II

Range vs. speed tradeoff

- as frequency rises
  - available bandwidth typically increases
  - distance and ability to overcome obstacles decrease
- higher frequencies have higher bandwidth capability, but require more power to achieve the same range
- lower frequencies have lower bandwidth, but can achieve longer distance
- lower frequencies require larger antennas to achieve the same gain

Range vs. Data Rate III

Different size of antennas means different chances for embedding

[Chou, 2016]

Environment

The role of environment

- humidity obstacles transmissions
  - e.g., network performance is typically bad when it rains a lot
- other physical obstacles hinder transmission
- range figures typically mention *line of sight* between antennas
- more generally, the performance of any wireless connection depends on its *situation*

Focus on. . .

IEEE 802.15.4 Suite / ZigBee I

IEEE 802.15.4

- a technical standard for low-rate wireless personal area networks (LR-WPANs)
- it defines the physical layer and MAC (media access control, data layer)
- by providing proof of origin that is difficult to deny
- it is the basis for the Zigbee, ISA100.11a, WirelessHART, MiWi, 6LoWPAN, Thread, SNAP specifications
- which complete the definition of the upper layers

IEEE 802.15.4 Suite / ZigBee II

ZigBee

- IEEE 802.15.4-based specification for a suite of high-level communication protocols
- used to create personal area networks with small, low-power digital radios
- meant to be simpler and less expensive than Bluetooth or WiFi
- low-power consumption limiting transmission distances to 10-100 meters (line of sight)
- can transmit data over long distances via a mesh network of intermediate devices
- typically used in low-data-rate applications that require long battery life and secure networking
    - e.g., disposable health sensors

Wi-Fi

Based on IEEE 802.11 standard

- Wi-Fi is a family of wireless networking technologies
- based on the IEEE 802.11 family of standards
- commonly used for local area networking of devices and Internet access
- w.r.t. ZigBee and Bluetooth, Wi-Fi consumes a lot of power
- yet, long range, hundreds of MB/s
- used extensively in both office and domestic settings
- becoming ubiquitous nowadays
    - e.g., domestic appliances

LPWAN

Low Power Wide Area Networks (LPWAN)

- wireless telecommunication wide area network designed for long-range communications at a low bit rate among things—e.g., sensors operated on a battery
- designed for wireless, battery-powered things
  - e.g., LoRaWAN
  - data rates range from 0.3kbps to 50kbps.
  - lower frequency (sub-GHz bands) to penetrate the core of large structures and subsurface deployments within a range of 2 Km

Other Technologies

Satellite

- to operate outside standard coverage
- like, in rural areas
  - e.g., locomotives

Cellular network

- a permanent and pervasive network
- worldwide
- standard—4G, 5G
- providing a secure and reliable mobile connection between things and devices

Next in Line. . .

Things are not People

Data from things

- huge amounts data comes from things
- much more than IoP
- to be collected and stored
- with different models and technologies

Focus on...

RDBMS

Relational databases with SQL

- databases with rows (one for each record) and columns (one for each attribute)
- *schema* as the DB structure
- SQL is the language to *query* a relational DB
  - e.g., to get all the records matching a criteria
- cost in space used, in search, in changes
- ACID properties (Atomicity, Consistency, Isolation, Durability) for transactions
- DBMS–DB management systems, different technologies available
- whose performances are also determined by the physical support

NoSQL

No schema

- no tabular relations
- other data structures (e.g., key-value, graph, document, tuples)
- typically, no ACID
- eventual consistency
- simple design and simpler horizontal scaling to clusters of machines
  - e.g., MongoDB

SQL vs. NoSQL

SQL better when...

- logical, related or discrete data requirements can be identified up-front
- data integrity is essential
- standards-based proven technology with wide developer experience is needed

NoSQL better when...

- data requirements are unrelated, indeterminate or evolving
- simpler or looser project objectives enable coding efforts to start immediately
- speed and scalability are imperative

Time Series

Collecting data from things

- data is produced with a pace (hours, minutes, seconds, ...)
  - e.g., level of CO_2 by a gas sensor
- problems with RDBMS: 1000 machines with 10 seniors sampling 20 times per second generate 17 billions rows per day—if each sample is a row
- *historians* (time-series DB) record data as flat files—with name, value, quality and timestamp for each data point
- e.g., Apache Cassandra, OpenTSDB

Heterogeneous Data

Hadoop

- software technology designed for storing and processing large volumes of data distributed across a cluster of commodity servers and commodity storage
- from Google published design principles to handling large volumes of data—Hadoop implements them
  - faults are the norm, fault tolerance and automatic recovery should be embedded
  - files are huge by default, standard DB assumptions do not hold
  - default for update is append
- MapReduce (for parallel data processing and execution) and other tools

Cloud Computing

Reducing costs, increasing QoS

- most of the cost is managing security, availability, performance, change of storage environment
- cloud represents an architectural solution
- cloud service providers can dedicate engineering resources to automate the key processes to manage the security, availability and performance of the compute and storage environments
- through automation, they can both dramatically reduce the cost and increase the overall quality of service

Focus on. . .

Health Care

Heterogeneous data in hospitals

- hospital data typically deal with huge amounts of unstructured and structured data
  - e.g., to continuously monitor a patient's vital signs—heart rate, respiration rate and temperature
- Hadoop is known to be used—and, properly

Power

Real-time analysis

- power industry requires huge amounts of data
- from production, from consumers, from weather
- turbine sensors generate data
- weather affect photovoltaic solar panels
- heterogeneous data, again—e.g., Splunk

Car Insurance

Data from cars

- driving data essential for today insurance
  - e.g., run-time data such as engine start, barometric pressure, accelerator pedal position, hybrid battery pack remaining life, engine-oil temperature, . . .
- a provider (Progressive) adopted Hadoop to record and process all data from its drivers

Next in Line...

Learn from Data

Once data are made available...

- we can use them
- to infer new knowledge

Machine learning (ML)

- unsupervised learning—finding regularities in data to devise out *clusters* in a set of input data
- supervised learning—learn a predictive nodel mapping input patterns into output values, based on a set of *training data*

Focus on. . .

Database Query

When we know what to look for

- a query defines a set of properties of data that some record in a DB may satisfy
- to select those records—e.g., via SQL, or, visual tools
- useful when we know which part of the population to investigate

On-line Analytical Processing (OLAP)

- provides a GUI to query large data collections and visualise the results to facilitate data exploration
- the dimensions of analysis for OLAP must be pre-programmed—unlike data mining tools

Prediction I

Predictive models

- a predictive model provides a mapping from the input-data vector to the desired output value(s)
- mapping in the form of a mathematical function, governed by a set of parameters (weights)
- the learning algorithm generates the optimal set of parameters generating accurate predictions of the output values for each input vector

Prediction II

Continuous vs. discrete output values

- if the outputs are continuous values then the prediction problem is known as *regression*—the predictive model is known as a regression model
- if the outputs are discrete values, then the prediction problem is known as *classification* (the predictive model is known as a classifier)
  - e.g., predicting the heart rate of the user — a continuous value — would require a regression model; predicting whether the user's heart-rate patterns were normal (class 0) or abnormal (class 1) would require a classifier

Prediction III

Labelled dataset

- both regression and classification require a *labelled* dataset
- there, each example is an input vector coupled with the desired output value
- in supervised learning, the dataset is partitioned into *training set* and *test set*—the first used to set the model parameters, the second to assess the model performance

Novelty Detection I

Predicting anomalies

- sometimes, we need to detect if everything is working normally
- and, we have not enough data about anomalous behaviour
- and, we would like to detect also behaviours never before occurred

Novelty / anomaly detection

- data on normal behaviour
- used to build a model for *normal* behaviour
- whenever data is labelled with low probability by the model, a potential anomaly is detected

Clustering I

Discovering patterns

- *clustering* is a set of techniques for discovering patterns or *clusters* within input data
- a cluster is defined as a collection of input data vectors that are similar according to some metric
- cluster basically partitions the input data space in distinct classes or groups
- which are not easily detected by a human

Clustering II

A form of unsupervised learning

- it only uses the input-data vectors
- does not make use of any labels or output values that we may have
- distinct from novelty detection in that the aim of clustering is to discover groupings within the data where each grouping represents input patterns with high similarity—whereas the aim of novelty detection is to build a faithful statistical model of the full range of normal input data

Focus on...

Oil and Gas

The case of SmartSignal

- goal: predicting pump failures at nuclear facilities
- each thing represented with its own unique historical data
- models for normal behaviour is established
- case: predictive monitoring
  - diesel generator monitored with anomaly detection
  - e.g., worked, allowing identification of a defective oil pump before actual failure

Transportation

The case of NYAB

- New York Air Brake (NYAB) provides train control systems to railroads
- with the ability to instrument trains, connect, and collect the data, they exploited both supervised and unsupervised machine-learning algorithms
- supervised machine learning for fuel-efficiency analysis
- unsupervised machine learning to implement train-handling risk mitigation

Power Grids

Way too much data

- huge blackouts in the US have made it clear that intensive monitoring is required
- PMU (phasor measurement units) measure power-frequency voltage, current and phasor angle—60 times per second, with hundred sensors in the grid
- way too much data for a human operator
- time series, with similarity analysis and clustering have been used to detect equipment failure

Health Care

Monitoring patients

- a single patient can generate many different streams of data
- sources unrelated, data related—possibly, to some health condition
- anomaly detection to keep critical conditions under control

Next in Line...

Outcomes

Which outcomes?

- higher degrees of services for machines
- to both the producers and the consumers of things
- reshaping the business models for producers

Focus on. . .

Enterprise Applications

SAP & friends

- most organisational functions (finance, accounting, sales, ...) "bundled" in *enterprise applications*
- on client/server architectures
- in the nineties, Oracle, SAP, PeopleSoft, ...
- after a decade, on cloud-based architectures
- Taleo, Salesforce.com, Concur, NetSuite, ...
- caveat: they actually cover a small percentage of the total application footprint
- as a result, most enterprise applications are custom

Middleware

Custom enterprise application

- middleware—between machines and applications
- providing services or software required as common functionalities by different applications
  - e.g., naming service, authentication, workflow management, coordination, ...

Precision Machines: Improving QoS

Improving quality of service

- maintenance service traditional in computing industry—e.g., hardware "calling home" in case of problems
- once a thing is connected, it can do the same—for better availability, security, performance and change management
- thus improving QoS for each thing, and for the overall IoT system

Precision Machines: Reducing Cost of Service

Reducing cost of service

- sensors, learning, and communication make it possible to diagnose problems remotely
- and, to know exactly which parts are needed when repair is required
- visits and time by human technician are optimised, thus reducing costs

Precision Machines: New Business Models

Three new machine business models—from software companies

- *service and support contracts* — machine sales bundled with warranty or service contracts: service contracts allow the manufacturer to keep service in-house and capture more of the value from service efficiencies
- *assisted services* — connect to the machine and offer advice to improve the availability, security or performance of the machine
- *machine-as-a-service* — the machine manufacturer retains ownership and takes full responsibility for the security, availability, performance and change of the product—in return for a recurring charge

Machine vs. Nomic Data

Who owns the data?

- do data belong to the maker or to the user?
- two sorts of data: *machine* data and *nomic* data
- machine data refer to the machine itself
- nomic data are produced by the machine operating its function
- said that who owns what?
- is there a universal answer, or, does it change from case to case?

Precision Service: Higher Quality Product or Service

The case: mining machines

- vibration sensors have been added to the wall, for predicting roof collapses
- service can proceed with no interruption for the mining system
- with no extra costs

Precision Service: Improved Health

The case: precision farming

- lowering the cost of consumable – fuel, fertiliser, pesticides – by exploiting data and learning from them
- not just cost: less pesticides mean healthier environment for a farmer

Precision Service: Improved Safety

The case: automatic trains

- constant control over the status of an automatic train can prevent most of the issues—including human errors
- not just cost, but safer railroads, too

Summary

Transforming business

- tech is cool, yet its goal is to *transform business*
- IoT has the potential to do so for both producers and consumers
- in a number of application areas—such as agriculture, healthcare, power, etc.

Focus on. . .

Early Stage

Too early to see it all

- the potential of IoT has a long way to go
- yet some examples can be mentioned, already

Precision Machines

- improved quality of service — wind turbine business
- reduced cost of service — high-speed printers and inserters
- new business model — General Electric moving to service contracts with machines sold

Precision Service Industries

- lower consumable cost — e.g. fuel savings, energy savings
- higher quality product — e.g., more environment control in agriculture
- improved health & safety — e.g., data from cars to insurance companies improve driving behaviour

Next in Line. . .

The Point

IoT changes the processes

- by providing more data and more knowledge on processes
- that can be improved under many aspects

IoT allows for precision

- much more data and knowledge on process – in particular, on line – reduce waste and improve precision—of machines and services

IoT Framework: Principles & Practices

Fundamentals of Internet of Things

Andrea Omicini
andrea.omicini@unibo.it

Bologna Business School, Università di Bologna

Master in Digital Technology Management 2020/2021

References

[Chou, 2016] Chou, T. (2016).
Precision: Principles, Practices and Solutions for the Internet of Things.
CrowdStory Publishing.

[Kizza, 2015] Kizza, J. M. (2015).
Guide to Computer Network Security.
Computer Communications and Networks. Springer, 3rd edition.