11. Mobile & Ubiquitous Computing

Sistemes Distribuïts en Xarxa (SDX)
Facultat d'Informàtica de Barcelona (FIB)
Universitat Politècnica de Catalunya (UPC)
2017/2018 Q2

Contents

- **Introduction**
- Association & interoperation
- Sensing & context-awareness
- Adaptation

Motivation

1. Miniaturization of devices

⇒ We can carry them around with us or wear them

⇒ We can embed them into many parts of the physical world

2. Prevalence of wireless connectivity

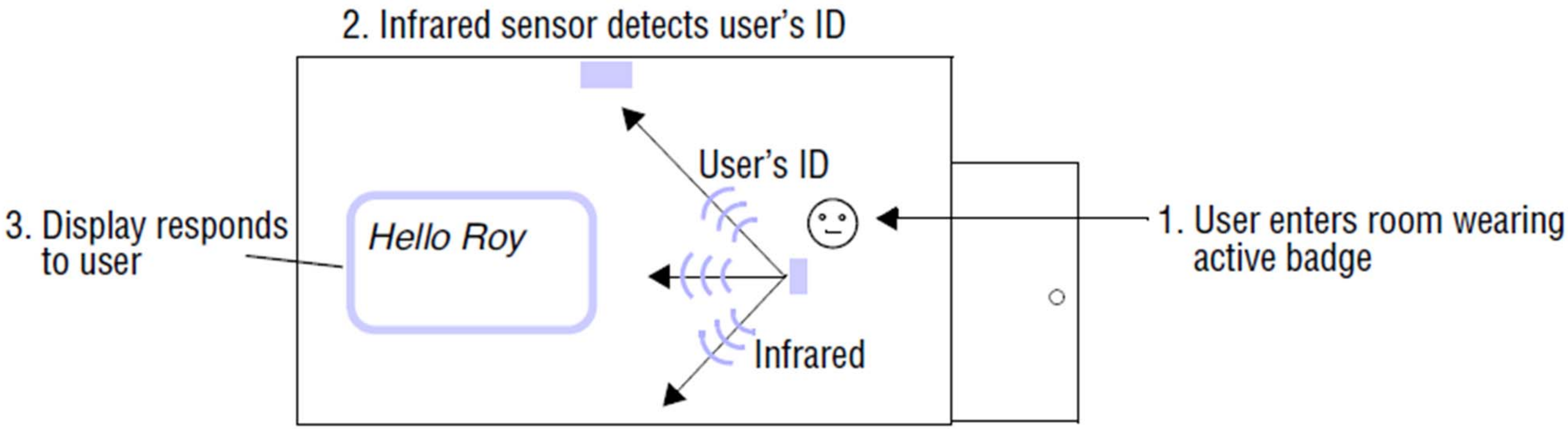
⇒ We can connect the devices to one another, and to conventional computers

Introduction

- Mobile computing
  - Users carry their computers while staying connected to other computers or the Internet
  - Aims to exploit the connectedness of portable devices (including laptops and handheld devices (e.g. smart phones, PDAs))
- Ubiquitous computing
  - Aims to exploit the increasing integration of computing devices with our everyday physical world (computing everywhere)
  - Also known as “pervasive computing”

Introduction

- Wearable computing
  - Users carry devices on their person (clothes, watches, …)
  - Typically have specialized functionality and often do not require user manipulation to operate



- Context-aware computing
  - Computer systems automatically adapt their behavior according to physical circumstances

Introduction

- Volatile systems
  - Encompass the essential distributed systems features of all of the previous systems
  - Changes are the rule instead of exception
  - Systems exhibit ALL of these forms of volatility:

1. Failures of devices and communication links
2. Changes in the characteristics of communication (e.g. bandwidth)
3. Creation and destruction of associations (i.e. logical relationships) between software components resident on the devices

Introduction

- Smart space
  - Computationally-enhanced physical space with services provided only or principally within it
  - Mobility in smart spaces:
    - Physical: smart spaces act as environments for devices to visit and leave them
    - Logical: a component changes some of its associations with other components
  - Component appears in smart space and becomes integrated (at least temporarily) into that space
  - Component disappears from the space, either through mobility or it is switched off or fails

Introduction

- New class of computing device due to rise of mobile and ubiquitous computing
  a) Limited energy as devices run on batteries
     - Algorithms must be sensitive to the energy they use
     - Probability of device failure is increased because of battery discharge
  b) Limited computational resources in terms of processor speed, storage capacity and network bandwidth due to energy and space limitations
     - Algorithms must finalize in reasonable time despite this
     - Augment the node capacity using resources in its environment

Introduction

c) Equipped with <u>sensors</u> to measure physical parameters and <u>actuators</u> controllable by software to affect physical world
   - Algorithms must deal with the inaccuracy typically incurred by those sensors

d) Devices have some sort of (wireless) connectivity (Bluetooth, WiFi, 4G, etc.)
   - Disconnections are far more likely since devices can exceed their operating distance from other devices or encounter radio occlusions between them
   - These factors can also lead to highly varying bandwidth and latency due to changing error rates

Introduction

e) Routinely change the set of components they communicate with, as they move or as other components appear in their environment

- *Association:* logical relationship formed when at least one of a given pair of components communicates with the other over some well-defined period of time
  - Spontaneity of associations is physically driven, as they are made and broken according to the current physical circumstances of the components, in particular, their proximity
- *Interoperation:* interactions of components during their association

Contents

- Introduction
- **Association & interoperation**
- Sensing & context-awareness
- Adaptation

Association

- A device appearing in a smart space must be able to (preferably without user intervention):
  1. Bootstrap itself onto the local network (device must acquire an address on the local network)
     a) Rely on servers accessible within the smart space
        - Device issues query to well-known broadcast address
        - DHCP server supplies IP address

     b) Serverless address assignment
        - Use zero-conf networking (e.g., Apple's Bonjour)
        - Device autoconfigures its link-local IP address after checking for conflicts by sending ARP requests
  2. Associate appropriately in the smart space

Association

- Components on the device either associate to services in the smart space, provide services to components in the smart space, or both
- Association problems
  A. Scale: how to choose what components to interoperate with?
     - There may be many devices within the smart space
  B. Scope: how to consider only components from the smart space (and all of them) rather than the ones that lie beyond?
     - Smart spaces must have 'boundaries'

Discovery services

- One solution to the association problem where services in a smart space are registered and looked up by their attributes
  - The implementation of a directory service must take account of volatile system properties:
    a) The directory data (i.e. services) are determined at runtime as a function of the client's context (the particular smart space where the queries take place)
    b) There may be no infrastructure in the smart space to host a directory server
    c) Services registered in the directory may disappear
    d) Protocols used for accessing the directory need to be sensitive to the energy and bandwidth they consume

Discovery services

- Discovery services have an interface to:
  1. Register and deregister services

  address=http://192.168.1.1/services/printer57, class=printer, type=laser, color=yes, resolution=600dpi, location=room101

  2. Look up services from those that are available
     - Ideally, <u>low-effort appropriate</u> associations
       - Without any human effort (or minimal)
       - Services returned by the query are precisely those existing in the smart space that match the query
- Usually, we have <u>network discovery services</u>
  - Bootstrap access to the local discovery service at runtime using the <u>multicast IP of the local subnet</u>

Network discovery services

A. Implemented by a directory server
   - Issue a multicast request to locate the server. It will respond with its unicast address. Then, point-to-point communication
   - Good in smart spaces providing infrastructure
   - Saves the interruption of uninvolved devices that occurs with multicast communication
   - Directory server must deal with services that disappear spontaneously
     - It maintains a service's registration only if the service periodically renews its **lease** on the entry
       - Trade-off of timeliness vs. bandwidth and energy

Network discovery services

B. Serverless: participating devices collaborate to implement a distributed discovery service
   1. Push model: services multicast ('advertise') their descriptions regularly. Clients listen for the multicasts and run their queries against them
      - Wasted multicasts if no clients needing to discover
      - Trade-off of timeliness against bandwidth and energy
   2. Pull model: clients multicast queries. Devices respond with service descriptions that match
      - Client can discover available services as soon as it appears but it may receive several responses

   – No problem with services that disappear

Network discovery services

- Difficulties with network discovery services
  1. Using the local subnet may be a poor approximation to a smart space
  2. Association can fail due to inadequacies in the way services are described
     - Even slight variations in the service-description vocabulary could cause association to fail
       – e.g. a hotel room has a service called 'Print' whereas the guest's laptop searches for 'Printing'
     - Lost association opportunities: device cannot associate if it has no description for the service
       – e.g. 'digital picture frame' on the hotel room's wall

Discovery services

- There are options to alleviate these problems, but require more <u>human intervention</u>
  A. Human provides input to scope discovery (e.g. smart space ID such as the hotel room number)
     - Device can use the ID as an input attribute to lookups
  B. Device senses information to scope discovery
     - Smart space ID is encoded in a *glyph*, which is decoded using the camera (e.g. QR codes)
     - Smart space ID is propagated using a *physically constrained channel* (e.g. an infrared beacon)
       - Those channels are significantly attenuated by the materials at the boundaries of the smart space

Discovery services

C. Direct physical association (no discovery service): human enables the carried device to learn the network address (e.g. Bluetooth or IP address) of a ‘target’ device
   - e.g. read address from a barcode or using a short-range RFID channel (Near Field Communication)
   - e.g. target device sends its address on receiving a physical stimulus (digitally modulated laser beam)
   - e.g. two-button protocol to associate two wireless devices with each other
     – On button press, devices send their network addresses to the multicast address, and associate with any address that arrives via multicast within a small interval of the button press

Interoperation

- Once associated, devices can interoperate among them
  - In principle, any of the communication paradigms described in 'Lesson 2' could be used
  - Main difficulty that stands in the way of volatile interoperation is <u>software interface incompatibility</u>
  - Solutions:
    1. Adaptor proxies to adapt interfaces to one another by converting invocations (this approach is difficult)
    2. Use <u>standardized fixed</u> service interfaces
       - Data-oriented paradigms
       - Examples: event systems, data spaces

Interoperation

- Data-oriented models are useful for volatile computing because producers and consumers do not need to identify one another (communication is decoupled)
  - In a volatile system, keeping track of which other components are present can be difficult
- But we have traded the agreement on the set of functions in an interface against agreement on the types of data that are passed as arguments to those functions

Contents

- Introduction
- Association & interoperation
- **Sensing & context-awareness**
- Adaptation

Sensing & context-awareness

- Focus on how mobile devices are integrated with the physical world
  - About the relevance of the physical circumstances of an entity (i.e. its **context**) to system behavior
  1. Architectures for processing data collected from sensors. Typical sensed data include:
     - Location, velocity and orientation by using GPS, accelerometers and gyroscopes
     - Ambient conditions by using thermometers, microphones
     - Presence by using RFID, infrared
  2. Context-aware systems that can respond to their (sensed) physical circumstances

Sensing architectures

- Challenges designing context-aware systems
  1. Integration of idiosyncratic sensors
     - How to correctly deploy different sensors (especially those unusual) in the physical scenario
  2. Offer abstractions from raw sensor data
     - Avoid concern with the peculiarities of individual sensors
  3. Sensor outputs may need to be combined
     - Combine sensor sources to reduce errors (sensor fusion)
     - Gather several contextual attributes that an application needs to operate
  4. Context is dynamic
     - Able to respond to changes in context

Example: wireless sensor networks

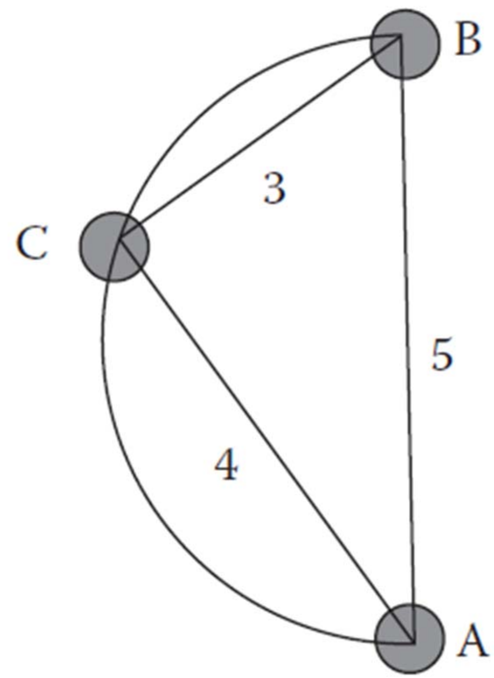
- Consist of a (large) number of small, low-cost devices (i.e. nodes) with facilities for sensing, computing, and wireless communication
- They function without any global control
  - Each node bootstraps itself by discovering its wireless neighbors and communicating via them
- In general, they are dedicated to detecting certain conditions of interest
  - Include at least one more powerful device (root node) for longer-range communication with a conventional system that reacts to the alarms

Example: wireless sensor networks

- Multihop communication between nodes
  - They communicate over multiple wireless hops
- They are added to an existing environment and function independently of it
  - Physically arranged ‘randomly’ but at a sufficient density to enable communication and allow sensing significant phenomena
- Direct communication restricted to neighbors
  - Reduce network contention
  - WiFi is costly in power consumption
    - Increases exponentially with distance

Example: wireless sensor networks

- Minimum energy needed to communicate with a node at a distance d is $E_d = K \cdot d^n$
  - $2 \leq n \leq 4$ depending on environmental parameters
  - K depends on the characteristics of the transmitter
  - The shortest Euclidean path between a pair of nodes is not necessarily the minimum energy path
    - If $n > 2$, the path ACB between A and B will consume lesser energy than the direct path AB



Example: wireless sensor networks

- Energy conservation & continuous operation despite volatility have driven architectural features for sensor networks
  - a) Traditional methods for data collection such as flooding or gossiping are not attractive from the energy efficiency point of view
  - ⇒ **Directed diffusion:** data-centric model based on publish/subscribe and reinforcement learning to adapt routing to changing network conditions
    - e.g. data-centrism provides space decoupling, which allows to deal with node volatility

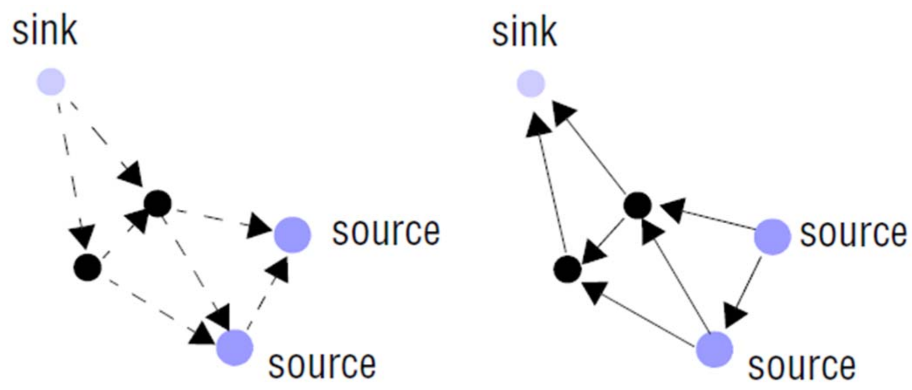
Example: wireless sensor networks

- **Directed diffusion**

1. Sink node expresses an interest (description of a sensing task) using attribute-value pairs

type = wheeled vehicle	// detect vehicle location
interval = 20 ms	// send events every 20 ms
timestamp = 01:20:40	// for the next 10 minutes
expiresAt = 01:30:40	
rectangle = [-100,100,200,400]	// from sensors within rectangle

2. Sink node forwards interest to neighboring nodes, which update an interest-cache and forward interest
   - This dissemination sets up gradients for each interest (denoting preferred paths)



(a) Interest propagation (b) Gradients set up

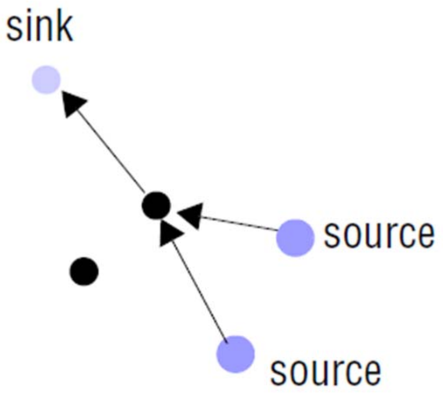
Example: wireless sensor networks

- **Directed diffusion**

3. Matching source nodes turn on their sensors as required and generate the data asked by the sink node

type = wheeled vehicle	// type of vehicle seen
instance = truck	// instance of this type
location = [125, 220]	// node location
intensity = 0.6	// signal amplitude measure
confidence = 0.85	// confidence in the match
timestamp = 01:20:40	// event generation time

4. Data is sent back to the sink along the reverse path set up with the gradients
5. Nodes receiving data reinforce some of the gradients depending on the responsiveness of these paths (reinforced gradients will be the preferred links for drawing down data)



(c) Data delivery

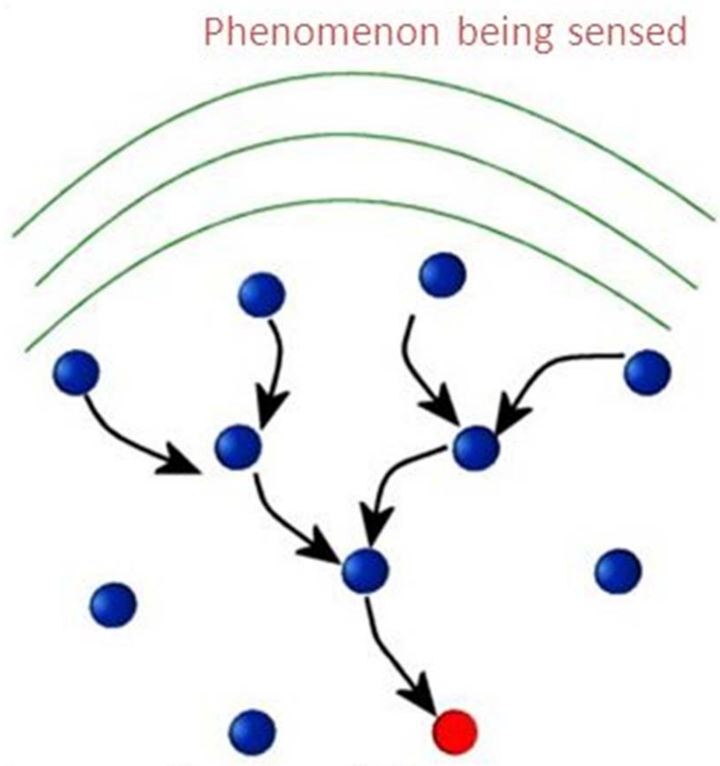
Example: wireless sensor networks

b) Wireless communication is expensive compared to processing in terms of energy consumption

   – A processor could execute 3 million instructions for the same energy used to transmit 1 kbit of data 100 m by radio

⇒ **In-network processing:** sensors have a processing capability and there is some processing within the wireless network

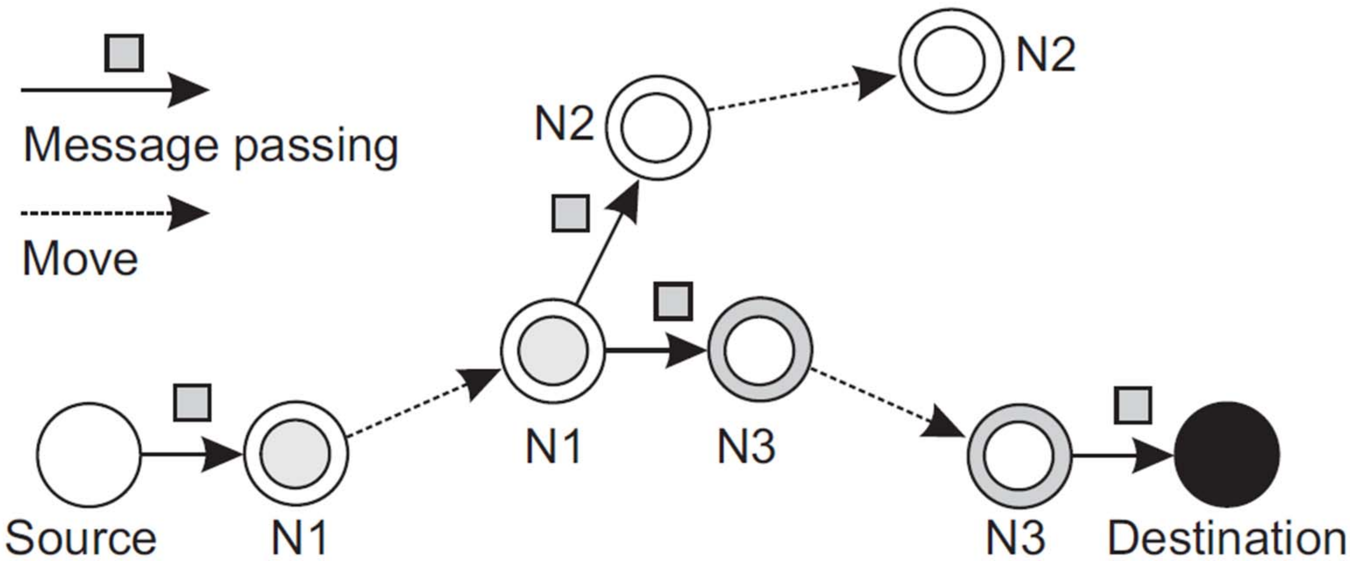
   – e.g. aggregation of data from different sensors to eliminate redundant transmissions to the base station



Example: wireless sensor networks

c) Lack of continuous connectivity precludes stable end-to-end paths $\rightarrow$ traditional routing could fail

$\Rightarrow$ **Disruption-tolerant networking:** nodes take on successive responsibilities to move data in a store-and-forward fashion (store message until finding another node to pass it on)



Location sensing

- Location is an obvious parameter for mobile computing, so it has received high attention
  - e.g. devices behave depending on where the user is ; devices assist users in navigation
- Goal: obtain data about the position of entities (also orientation and velocity)
- An entity can determine its own location, or someone else can do it (tracking)
- Let's introduce some location technologies

Location sensing

- **Global Positioning System (GPS)**
  - Device with a receiver calculates its absolute geographic coordinates from satellite radio signals using a trigonometric operation (multilateration)
    - Derives its distance from several satellites using the difference between the time of arrival of the signal and the time it was broadcasted (which is encoded within)
  - Three satellites must be visible to obtain latitude and longitude. More satellites allow to calculate also altitude
  - Works only outdoors because of signal attenuation inside buildings

Location sensing

- **Radio Beaconing**
  - Device with a receiver calculates its proximity to a fixed wireless base station (cellular, Bluetooth, WiFi) with a limited transmission range depending of the strength of the received signal
- **Active Bat**
  - Base station calculates the relative coordinates in a room of a device with a 'bat' transmitter using multilateration
    - Derives the bat distance from several ceiling-mounted ultrasound receivers using the time elapsed between the reception a ultrasound pulse and its emission by the bat

Location sensing

- **Ultra Wide Band (UWB)**
  - Base station calculates the relative coordinates in a room of a device with a UWB transmitter using multilateration
    - Derives the device distance from several receivers of short radio pulses sent at low power over a wide frequency spectrum
  - Signals propagate at high bit rates over short ranges (up to 10 m) and can go through walls
- **EasyLiving**
  - Cameras calculate the relative coordinates in a room of a device by using vision algorithms

Location sensing

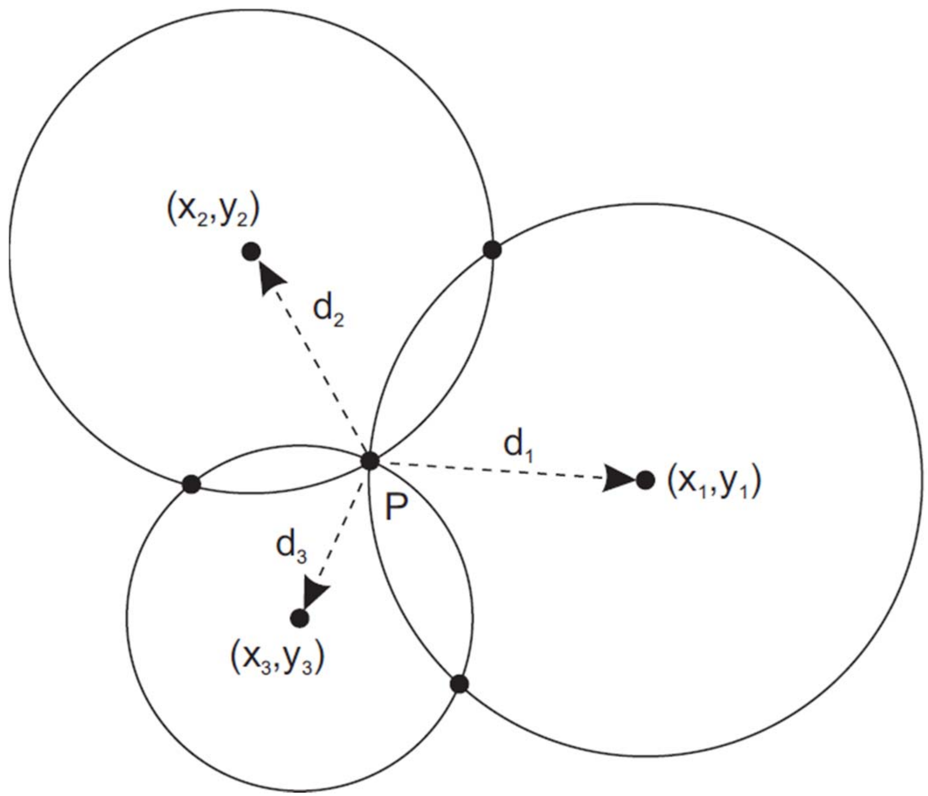
- **Active Badge**
  - Base station polls infrared sensors to calculate the semantic location of a device with a badge that regularly broadcasts its identity through infrared
  - Infrared signals are strongly attenuated by building materials
- **Automatic Identification Tag**
  - Reader calculates semantic location of a device with an attached tag (electronically readable ID)
  - Tags include RFID, Near Field Communication (NFC) and visual symbols (glyphs, barcodes, …)

Multilateration

- A node needs d+1 landmarks to compute its own position in a d-dimensional space

- e.g. 2D
  - Solve 3 equations in 2 unknowns: (x,y)
    - (x_i, y_i): coordinates of landmark i
    - d_i: distance to landmark i

$$d_i = \sqrt{(x_i - x)^2 + (y_i - y)^2}$$



Multilateration

- e.g. 3D (GPS)
  - Solve 4 equations in 4 unknowns: (x,y,z) and Δ_r
    - Each satellite $\mathbf{i}$ continually broadcasts a signal including:
      - (x_i, y_i, z_i): coordinates of satellite $\mathbf{i}$
      - ToT_i: time of transmission of the signal from satellite $\mathbf{i}$
    - Receiver measures the time of arrival of the signal (ToA_i) and calculates the time of flight: $\boldsymbol{ToF_i = ToA_i - ToT_i}$
    - Given the speed of light $\boldsymbol{c}$, the measured distance to satellite $\mathbf{i}$ (a.k.a. pseudorange) is $\boldsymbol{c * ToF_i}$
    - As the receiver's clock has an offset Δ_r w.r.t. the satellite's clock, the real distance to satellite $\mathbf{i}$ is $\boldsymbol{c * (ToF_i - \Delta_r)}$

$$c * (ToF_i - \Delta_r) = \sqrt{(x_i - x)^2 + (y_i - y)^2\ (z_i - z)^2}$$

Contents

- Introduction
- Association & interoperation
- Sensing & context-awareness
- **Adaptation**

Adaptation

- Devices in the volatile systems are highly heterogeneous in terms of processing power, input/output capabilities, network bandwidth, memory, and energy capacity
- **Adaptive** systems can adapt their runtime behavior to device heterogeneity and to the current resource availability
  - Ideally, without sacrificing crucial application properties

Context-aware adaptation of content

- The content that a service needs to deliver to a given device is a function of the context
- Dynamic adaptation of the original content programmatically into a suitable form
  - e.g. transcoding of multimedia data
    - Reduce image resolution
    - Convert text to speech or vice versa
- A lot of attention for client-server systems on the Internet
  - Adaptation to take place in the resource-rich server/proxy, not in the resource-poor client

Context-aware adaptation of content

- Adaptation more demanding in smart spaces
  - Requires adaptation between any pair of dynamically associated devices
  - Content providers may be too resource-poor to perform some adaptations themselves
- Provide **proxies** in the smart space to adapt content between the volatile components
- Device has to send its data to the proxy
  - It may be most energy-efficient to compress data prior to transmission due to the energy tradeoff between communication and processing

Adapting to changing resources

- Hardware resources such as screen size are heterogeneous across devices, but at least they are stable and well known
- Other resources are subject to change at runtime and may be hard to predict
  - e.g. available energy and network bandwidth
- Techniques for dealing with those changes to resource levels at runtime:
  1. Middleware support
  2. Cyber foraging

Adapting to changing resources

1. Middleware support
   a) Notify the user of reduced resource availability so can adapt to use less of that resource
      - e.g. on low bandwidth, the user of a video player (or the player itself) could switch the frame rate or resolution
   b) Allow reservations that guarantee a certain level of a resource
      - Guarantees are difficult to achieve in volatile systems (impossible in cases such as energy depletion)
   c) Suggest a corrective action to the user to get an adequate resource supply
      - e.g. change location to get better wireless coverage

Adapting to changing resources

2. Cyber foraging: A processing-limited device discovers a compute server in a smart space and offloads some of its processing load to it
   - Device should still function correctly (albeit more slowly or with reduced fidelity) if no server is available
   - Offloading should incur low communication between the server and the portable device
     - Time taken by communication over a low-bandwidth connection could outweigh the processing time gains
     - Energy costs of communication with the server could outweigh the energy savings from offloading

Summary

- Volatility is the root of the main challenges with mobile and ubiquitous systems
  - Components in a given smart space are subject to unpredictable change
- Integration of devices with our physical world involves sensing and context awareness
- Physical integration also means new degrees of resource constraints which force devices to adapt to them
- Further details:
  - [Tanenbaum]: chapter 1.3.3
  - [Coulouris]: chapter 19