Internet of Things
2017/2018

The Things

Johan Lukkien

John Carpenter, 1982
Guiding questions

- What to think about things and how are they connected?
Resource limitations

- **Memory**: available flash (‘program code’) and ram
- **Processor**: Mhz, instruction set expressive power, address width, ability to manage its power
- **Energy**: available Joules and how they are replenished
- **Communication**: required transceive power, bps, complexity of protocols

- These are connected, mainly through energy
  - Ram required power to retain state
  - Processor complexity and Mhz require energy
  - Low memory needs fewer address bits
  - Simpler network protocols and smaller bandwidths lead to lower power transceivers
RFC 7228: devices

- Three classes representing memory (hence processor) limitations

- C0: dependent on proxies for secure Internet inclusion
- C1: only low resource protocols
- C2: can run most Internet protocols
- (C9: phone, tablet, desktop)
Energy limitation and communication policies

<table>
<thead>
<tr>
<th>Name</th>
<th>Type of energy limitation</th>
<th>Example Power Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>E0</td>
<td>Event energy-limited</td>
<td>Event-based harvesting</td>
</tr>
<tr>
<td>E1</td>
<td>Period energy-limited</td>
<td>Battery that is periodically recharged or replaced</td>
</tr>
<tr>
<td>E2</td>
<td>Lifetime energy-limited</td>
<td>Non-replaceable primary battery</td>
</tr>
<tr>
<td>E9</td>
<td>No direct quantitative limitations to available energy</td>
<td>Mains-powered</td>
</tr>
</tbody>
</table>

Table 3: Classes of Energy Limitation

<table>
<thead>
<tr>
<th>Name</th>
<th>Strategy</th>
<th>Ability to communicate</th>
</tr>
</thead>
<tbody>
<tr>
<td>P0</td>
<td>Normally-off</td>
<td>Reattach when required</td>
</tr>
<tr>
<td>P1</td>
<td>Low-power</td>
<td>Appears connected, perhaps with high latency</td>
</tr>
<tr>
<td>P9</td>
<td>Always-on</td>
<td>Always connected</td>
</tr>
</tbody>
</table>

Table 4: Strategies of Using Power for Communication
### Some private taxonomy

<table>
<thead>
<tr>
<th></th>
<th>Flash</th>
<th>RAM</th>
<th>Address space</th>
<th>Processor (type)</th>
<th>OS</th>
<th>Energy</th>
<th>Operation</th>
<th>Actively reachable</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>small code memory</td>
<td>several bytes</td>
<td>&lt;= 8bits</td>
<td>~100Hz</td>
<td>no</td>
<td>External, or battery + wakeup</td>
<td>Externally activated, simple read/write</td>
<td>not designed for reachability via multi-hop</td>
<td>RFID tag, ISO 18000-6c</td>
</tr>
<tr>
<td>B</td>
<td>&lt;= 32K</td>
<td>Few hundreds</td>
<td>&lt;=16 bits</td>
<td>~1Mhz TMS430</td>
<td>no, or simple executive</td>
<td>mechanical</td>
<td>mechanically activated, just generates some data</td>
<td>no; needs proxy power switch</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>&lt;=32K</td>
<td>Few hundreds</td>
<td>&lt;=16 bits</td>
<td>~1Mhz TMS430</td>
<td>Contiki, TinyOS</td>
<td>battery</td>
<td>simple, fixed external behavior, needs proxy, simple sensing</td>
<td>duty cycled, needs proxy</td>
<td>simple sensor mote</td>
</tr>
<tr>
<td>D</td>
<td>&lt;=32K</td>
<td>~10K</td>
<td>&lt;=16 bits</td>
<td>~1Mhz TMS430</td>
<td>Contiki, TinyOS</td>
<td>battery + recharge</td>
<td>capable of managing most constrained IP protocols, sensing, actuating, processing</td>
<td>self-managed on/off behavior</td>
<td>Crossbow</td>
</tr>
<tr>
<td>E</td>
<td>&lt;=256K</td>
<td>~32K</td>
<td>&lt;=32 bits</td>
<td>~1-10Mhz ARM</td>
<td>Contiki, TinyOS</td>
<td>battery + recharge, mains</td>
<td>complete IP endpoint behavior, limited storage</td>
<td>yes</td>
<td>Jennic mote</td>
</tr>
<tr>
<td>F</td>
<td>~GB</td>
<td>~500Mb</td>
<td>32 bits</td>
<td>~Ghz ARM</td>
<td>Linux</td>
<td>battery + recharge, mains</td>
<td>full fledged embedded computer system</td>
<td>yes</td>
<td>Raspberry PI</td>
</tr>
<tr>
<td>G</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>phones, laptops, servers</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example: A battery-less light switch

- The switch is pressed.

- The node *turns on* and sends a Route Request broadcast message for a known destination.
  - it boots an OS in the process!
- Using Route Reply, it finds the route to the luminaries.

- Using the discovered route, the node transmits the control signal (turn on/off) to the luminaries.

- The luminary node acknowledges the reception of the control signal.

- The switch node does multiple retries to transmit that control signal as long as the node stays on and until an ACK is received.

Running FreeRTOS and capable of transmitting compressed IP packets (6LoWPAN).

From: *6LoWPAN: IPv6 for Battery-less Building Networks*, MSc thesis of N.A. Abbasi, TU/e
Functionality of things

• ‘Things’ must be capable to perform the required sensing, actuation, computation, communication
  – functional requirements

• In addition, because they are many:
  – (secure) bootstrap, (secure) network association
    • upon (re)starting a device must load its code from a trusted source
    • it must join the correct network
  – secure communication
  – (secure) software update, over the network
    • updates are inevitable and must remain safe
  – … part of the life cycle
Concerns and management

- can it join a network?
  - secure bootstrapping
- can it be configured?
  - adapting operational parameters
    - e.g. sensing, communication frequency
- can it be updated (over the air)?
  - new firmware, new software, new application components
- can it run IP?
  - serve as IP endpoint
- can it secure itself?
  - independent node

- A,B,C: need trusted partner (proxy)
- A,B: very little; C: limited

- From D onwards
- From E onwards; D runs limited protocols
- A,B,C: need trusted partner; D: limited
## Some private taxonomy

<table>
<thead>
<tr>
<th></th>
<th>Flash</th>
<th>RAM</th>
<th>Address space</th>
<th>Processor (type)</th>
<th>OS</th>
<th>Energy</th>
<th>Operation</th>
<th>Actively reachable</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>small code memory</td>
<td>several bytes</td>
<td>&lt;= 8 bits</td>
<td>~100Hz</td>
<td>no</td>
<td>External, or battery +</td>
<td>Externally activated, simple read/write</td>
<td>not designed</td>
<td>RFID tag, ISO 18000-6c</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>wake-up</td>
<td></td>
<td>for reachability</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>&lt;= 32K</td>
<td>Few hundreds</td>
<td>&lt;=16 bits</td>
<td>~1Mhz</td>
<td>TMS430</td>
<td>no, or simple executive</td>
<td>mechanically activated, just generates some data</td>
<td>no</td>
<td>power switch</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>mechanical</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>&lt;=32K</td>
<td>Few hundreds</td>
<td>&lt;=16 bits</td>
<td>~1Mhz</td>
<td>TMS430</td>
<td>Contiki, TinyOS battery</td>
<td>simple, fixed external behavior, needs proxy, simple sensing</td>
<td>duty cycled</td>
<td>simple sensor mote</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>needs proxy</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>&lt;=32K</td>
<td>~10K</td>
<td>&lt;=16 bits</td>
<td>~1Mhz</td>
<td>TMS430</td>
<td>Contiki, TinyOS battery +</td>
<td>capable of managing most constrained IP protocols, sensing, actuating, processing</td>
<td>self-managed</td>
<td>Crossbow</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>recharge</td>
<td></td>
<td>on/off behavior</td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>&lt;=256K</td>
<td>~32K</td>
<td>&lt;=32 bits</td>
<td>~1-10Mhz</td>
<td>ARM</td>
<td>Contiki, TinyOS battery +</td>
<td>complete IP endpoint behavior, limited storage</td>
<td>yes</td>
<td>Jennic mote</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>recharge, mains</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>~GB</td>
<td>~500 Mb</td>
<td>32 bits</td>
<td>~Ghz</td>
<td>ARM</td>
<td>Linux</td>
<td>full fledged embedded computer system yes</td>
<td>yes</td>
<td>Rasberry PI</td>
</tr>
<tr>
<td>G</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>phones, laptops, servers</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Johan J. Lukkien, j.j.lukkien@tue.nl
TU/e Informatica, System Architecture and Networking
What’s new with IoT?

- There are **many** things
  - #things / person >> 1 (50B in 2020)
  - hence, things need to talk to *each other* or to a *database*
    - about .....?
  - **self-* properties, autonomy**
    - self management, self healing, ...
  - **scalability, at access networks**
    - many things sharing your wireless LAN
    - special infra structure outdoor
- Things have **limitations**
  - low processing power, memory, low capacity network
    - size IP packet comparable to available memory
  - sometimes battery operated
  - embedded: no UI
- Their numbers and far-reaching locations enable **entirely new applications**
  - large-scale data collection
  - data-based applications
  - manufacturers probing into the deployed systems
- Their scale and locations comes with **complex concerns**
  - device/data handling, ownership
  - security, safety, application reliability
    - at a compelling scale
  - application development, deployment, management
Guiding questions

• What to think about things and how are they connected?
How do end points communicate?
Networking approaches

- Physical neighbors:
  - Shared medium
  - Point-to-point
- In case of physical separation
  - Switching
  - Connecting networks
  - ... multiple interfaces for some nodes

- Packet oriented
- Full connectivity by
  - (intelligent) flooding
  - Routing
Flooding
<table>
<thead>
<tr>
<th></th>
<th>Wired</th>
<th>Wireless</th>
</tr>
</thead>
<tbody>
<tr>
<td>Battery</td>
<td>Energy limitations will determine uptime, communication behavior and protocols. In principle, the wire could be used for power; and at least for waking up nodes</td>
<td>Energy limitations will determine uptime, communication behavior and protocols; nodes must manage their sleeping behavior; nodes can be mobile</td>
</tr>
<tr>
<td>Mains</td>
<td>No real need for low resources except cost and energy; this class captures ‘regular’ office/home infrastructure devices</td>
<td>Wireless is there for convenience (absence of other infra - outdoors) and for connecting to mobile wireless nodes; powerful wireless protocols can be used, always on</td>
</tr>
</tbody>
</table>
The LANs

- Taken from Wikipedia (Nov. 2015)
- Important wireless:
  - the 3 group (ethernet, PoE)
  - the 11 group (‘WiFi’)
  - the 15 group (wireless PAN)
- Within IEEE 802.15
  - 1x: Bluetooth
  - 4x: PHY/MAC layer for ZigBee, 6LoWPAN, Thread, WirelessHART, MiWi, ContikiMAC, 6Tisch…
- Within the IEEE 802.11 LANs:
  - 11p: ITS
  - 11e: QoS
  - 11s: meshing
  - 11ah: low power, low interference

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>IEEE 802.1</td>
<td>Higher Layer LAN Protocols</td>
<td>active</td>
</tr>
<tr>
<td>IEEE 802.2</td>
<td>LLC</td>
<td>disbanded</td>
</tr>
<tr>
<td>IEEE 802.3</td>
<td>Ethernet</td>
<td>active</td>
</tr>
<tr>
<td>IEEE 802.4</td>
<td>Token bus</td>
<td>disbanded</td>
</tr>
<tr>
<td>IEEE 802.5</td>
<td>Defines the MAC layer for a Token Ring</td>
<td>disbanded</td>
</tr>
<tr>
<td>IEEE 802.6</td>
<td>MANs (DQDB)</td>
<td>disbanded</td>
</tr>
<tr>
<td>IEEE 802.7</td>
<td>Broadband LAN using Coaxial Cable</td>
<td>disbanded</td>
</tr>
<tr>
<td>IEEE 802.8</td>
<td>Fiber Optic TAG</td>
<td>disbanded</td>
</tr>
<tr>
<td>IEEE 802.9</td>
<td>Integrated Services LAN (iSLAN or isoEthernet)</td>
<td>disbanded</td>
</tr>
<tr>
<td>IEEE 802.10</td>
<td>Interoperable LAN Security</td>
<td>disbanded</td>
</tr>
<tr>
<td>IEEE 802.11</td>
<td>Wireless LAN (WLAN) &amp; Mesh (Wi-Fi certification)</td>
<td>active</td>
</tr>
<tr>
<td>IEEE 802.12</td>
<td>100BaseVG</td>
<td>disbanded</td>
</tr>
<tr>
<td>IEEE 802.14</td>
<td>Cable modems</td>
<td>disbanded</td>
</tr>
<tr>
<td>IEEE 802.15</td>
<td>Wireless PAN</td>
<td>active</td>
</tr>
<tr>
<td>IEEE 802.15.1</td>
<td>Bluetooth certification</td>
<td></td>
</tr>
<tr>
<td>IEEE 802.15.2</td>
<td>IEEE 802.15 and IEEE 802.11 coexistence</td>
<td></td>
</tr>
<tr>
<td>IEEE 802.15.3</td>
<td>High-Rate wireless PAN (e.g., UWB, etc.)</td>
<td></td>
</tr>
<tr>
<td>IEEE 802.15.4</td>
<td>Low-Rate wireless PAN (e.g., ZigBee, WirelessHART, MiWi, etc.)</td>
<td></td>
</tr>
<tr>
<td>IEEE 802.15.5</td>
<td>Mesh networking for WPAN</td>
<td></td>
</tr>
<tr>
<td>IEEE 802.15.6</td>
<td>Body area network</td>
<td></td>
</tr>
<tr>
<td>IEEE 802.15.7</td>
<td>Visible light communications</td>
<td></td>
</tr>
<tr>
<td>IEEE 802.16</td>
<td>Broadband Wireless Access (WiMAX certification)</td>
<td></td>
</tr>
<tr>
<td>IEEE 802.16.1</td>
<td>Local Multipoint Distribution Service</td>
<td></td>
</tr>
<tr>
<td>IEEE 802.16.2</td>
<td>Coexistence wireless access</td>
<td></td>
</tr>
<tr>
<td>IEEE 802.17</td>
<td>Resilient packet ring</td>
<td>hibernating</td>
</tr>
<tr>
<td>IEEE 802.18</td>
<td>Radio Regulatory TAG</td>
<td></td>
</tr>
<tr>
<td>IEEE 802.19</td>
<td>Coexistence TAG</td>
<td></td>
</tr>
<tr>
<td>IEEE 802.20</td>
<td>Mobile Broadband Wireless Access</td>
<td>hibernating</td>
</tr>
<tr>
<td>IEEE 802.21</td>
<td>Media Independent Handoff</td>
<td></td>
</tr>
<tr>
<td>IEEE 802.22</td>
<td>Wireless Regional Area Network</td>
<td></td>
</tr>
<tr>
<td>IEEE 802.23</td>
<td>Emergency Services Working Group</td>
<td></td>
</tr>
<tr>
<td>IEEE 802.24</td>
<td>Smart Grid TAG</td>
<td>New (November, 2012)</td>
</tr>
<tr>
<td>IEEE 802.25</td>
<td>Omni-Range Area Network</td>
<td>Not yet ratified</td>
</tr>
</tbody>
</table>
Example: which (IP) protocols occur in a lighting network?

- **Connectivity:**
  - 6LoWPAN (= adaptation for IP/802.15.4), UDP, TCP [sometimes]
  - RPL, RIP, MPL: routing, multicasting
  - DTLS: packet based security

- **Application**
  - Trickle: application protocol for dissemination to all devices in a network
  - RESTful style
    - REST plus HTTP methods
    - CoAP – constrained application protocol
  - DNS-SD using mDNS, or CoAP directory: for service discovery
  - M2M protocols, e.g. MQTT/TCP

![Diagram of Lighting/Building Control Application](image-url)
Characteristics of wireless communication

• A wireless network is like a shared channel to put data on
  – standards must address this sharing, avoiding destructive interference

• Wireless communication is intrinsically subject to errors
  – received signal is the energy collected over time
  – can always be disturbed by an external party (a different protocol family in same frequency), or by another neighbor than the sender
  – quality depends on environment properties (e.g. reflections)

• Wireless communication is energy-hungry, particularly when compared to the energy used by embedded processors

• Standards must address the unreliability, sharing and energy use
What techniques are applied?

- **FDMA: multiple channels** (frequency division multiple access)
  - channel hopping with synchronization
    - e.g. 802.15.1 (Bluetooth), 6tisch,
  - cell formation / independent domains
    - e.g. 3/4G, IEEE 802.11 (WiFi)

- **TDMA: division in timeslots** (time division multiple access)
  - timeslots with schedules, time synchronization
    - e.g. WirelessHART, DECT, IEEE 802.15.4 in superframe mode
  - in combination with cell formation

- **Master-Slave**
  - e.g. an access point as master
  - beacon-based signaling
    - e.g. IEEE 802.11 (WiFi), Bluetooth, IEEE 802.15.4

- **CSMA: ’try before you cry’** (carrier sense multiple access)
  - Clear Channel Assessment: Aloha
    - with collision avoidance, detection or resolution
      - e.g. IEEE 802.15.4 (ZigBee): random distribution of retries
    - with priority
      - e.g. IEEE 802.11e (WiFi + QoS)

- **CDMA** (code division multiple access, as in 4G)
  - concurrent access of the medium with coded packets
  - admits decoding even with interference

- **Regulation**
  - duty cycle restrictions (e.g. 1% per station)
  - frequency assignment
<table>
<thead>
<tr>
<th></th>
<th>Wired</th>
<th>Wireless</th>
</tr>
</thead>
</table>
| **Shared medium** | CSMA/CD: coax ethernet  
               CSMA/CR: CAN  
               TDMA  
               token passing: profibus  
               Master/Slave: ethercat (, profibus) | MA/CA: (pure) Aloha  
               CSMA/CA: Wifi, Zigbee  
               TDMA: Wireless Hart  
               Master/Beaconing: Wifi, Zigbee  
               FDMA + channel hopping: bluetooth (low energy), 6tisch  
               CDMA: virtual point to point (4G), Chirp Spread Spectrum UWB |
| **Point to point** | Switched ethernet: PoE  
                Common way to set up larger networks using switches and routers | FDMA  
                Dedicated frequencies for long-haul transport: satellite            |
<table>
<thead>
<tr>
<th>Wired</th>
<th>Wireless</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Outdoor</strong></td>
<td><strong>City WiFi</strong></td>
</tr>
<tr>
<td>No common infra structure available; some initiatives via light poles or other relatively dense infra</td>
<td>4G/5G: LTE adaptation, LongRange low power WAN technologies</td>
</tr>
<tr>
<td><strong>Indoor</strong></td>
<td><strong>WiFi</strong></td>
</tr>
<tr>
<td>Standard wired infra structure using UTP, fiber, switches, bridges and routers</td>
<td>Bluetooth IEEE 802.15.4 based technologies</td>
</tr>
</tbody>
</table>
Energy, in low capacity environment

- Radio communication is a major contributor to energy consumption
  - don’t use it 😊

- Applied techniques (combinations, need support at multiple layers)
  - asymmetry
    - low-power nodes are 1 hop away from well-powered infra
  - control the transmit / receive power
    - trade connectivity for energy
  - duty cycle
    - switch node or radio on and off periodically
  - demand driven
    - event driven control of radio (wakeup radio, or push mode) with asymmetry
  - push / pull strategies
    - let the low power partner always be the first
  - trade power for range and throughput
    - special network technologies, special physical layers
Metrics (which is how to judge all this)

- **throughput**
  - number of bytes per time unit
  - for a station: bytes per interval

- **latency or delay**
  - time difference in initiation of a transmission and the start of the receipt
  - consist of processing delay, transmission delay and queueing delay

- **jitter**
  - variations in timing, e.g. delay jitter, throughput jitter

- **fairness, ability to prioritize**
  - fairness: bound on delay in access to a transmission channel
  - or fair share in competition

- **overhead**

- **scalability**
  - as utilization
    - increasing the amount of communication
    - increasing # stations
    - e.g. CSMA/CA scales badly when increasing # stations while TDMA scales well
  - as dimensioning
    - LoRaWan can scale the number of gateways with the number of low-power nodes

- **predictability**

- **reliability**
  - e.g. resilience against interference

- **power**

- **range**
# Qualitative metrics

<table>
<thead>
<tr>
<th></th>
<th>Range</th>
<th>Power</th>
<th>Resilience against interference</th>
<th>Scalability (utilization)</th>
<th>Overhead</th>
<th>Prioritization</th>
<th>Fairness</th>
<th>Jitter</th>
<th>Latency</th>
<th>Throughput</th>
</tr>
</thead>
<tbody>
<tr>
<td>FDMA</td>
<td>o</td>
<td>o</td>
<td>+</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
</tr>
<tr>
<td>TDMA</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>+</td>
<td>-</td>
<td>o/+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>o</td>
</tr>
<tr>
<td>CSMA/CA (+QoS)</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>-</td>
<td>-</td>
<td>o(+))</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>o</td>
</tr>
<tr>
<td>CDMA, spread spectrum</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
</tr>
<tr>
<td>Master / Slave</td>
<td>o</td>
<td>o</td>
<td>o (+)</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>o</td>
<td>o</td>
<td>o</td>
</tr>
<tr>
<td>Cell formation / domains</td>
<td>o</td>
<td>o</td>
<td>+</td>
<td>o</td>
<td>-</td>
<td>o</td>
<td>o</td>
<td>-</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>Asymmetry</td>
<td>o</td>
<td>+</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>+ o</td>
</tr>
<tr>
<td>Duty Cycling</td>
<td>o</td>
<td>+</td>
<td>o</td>
<td>o</td>
<td>+</td>
<td>o</td>
<td>o</td>
<td>+</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

- **+: improves** (e.g. ‘Latency +’ means improves (reduces) Latency)
- **-**: makes worse
- **o**: no effect/not applicable
Guiding questions

• What to think about things and how are they connected?