iSOCS

IoT Technologies for Distributed Sensing
ISOCS Winter Short Course 2023
16-19 January 2023, Bormio, Italy

Jan Mitrovics, JLM Innovation GmbH
jan.mitrovics@jlm-innovation.de
www.jlm-innovation.de
Definition (Wikipedia):

„The Internet of Things (IoT) describes physical objects (or groups of such objects) with sensors, processing ability, software and other technologies that connect and exchange data with other devices and systems over the Internet or other communications networks.

Internet of things has been considered a misnomer because devices do not need to be connected to the public internet, they only need to be connected to a network and be individually addressable.”

Smart devices connected to a network
Basics: Communication technologies

Distributed sensing networks

• Wired Ethernet, Fieldbus, RS485, I2C, ...
• Wireless
  • Wifi / Bluetooth / GMS/UMTS/LTE/G5 / ZWave/Zigbee/Thread / 433MHz / LoRa / 6LowPAN / RFID/ IRDA / 802.15.4

Communication protocols

• TCP/IP / Zigbee / Thread / Matter
• The Things Network
<table>
<thead>
<tr>
<th>Figures of Merit for networks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Throughput / bandwidth</td>
</tr>
<tr>
<td>Range (max cable range, coverage of wireless network)</td>
</tr>
<tr>
<td>Max devices addressable</td>
</tr>
<tr>
<td>Power demand (on device)</td>
</tr>
<tr>
<td>Real time / guaranteed delivery time</td>
</tr>
<tr>
<td>Security / Secrecy</td>
</tr>
<tr>
<td>Fault tolerance / redundancy</td>
</tr>
<tr>
<td>Topology (P2P / Bus / Star / Mesh)</td>
</tr>
</tbody>
</table>
Fieldbusses for Control systems

Broad range of fieldbusses is used!

Analog interfaces: 4-20mA, 0..10V, digital, ...

Source Wikipedia
## Fieldbus vs IoT

<table>
<thead>
<tr>
<th>Fieldbus</th>
<th>IoT</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Typical Applications</strong></td>
<td></td>
</tr>
<tr>
<td>Industrial automation</td>
<td>Distributed sensing</td>
</tr>
<tr>
<td>Motor control / Automotive</td>
<td>Monitoring</td>
</tr>
<tr>
<td></td>
<td>Home automation</td>
</tr>
<tr>
<td><strong>Typical Requirements</strong></td>
<td></td>
</tr>
<tr>
<td>Reliability</td>
<td>Scalability</td>
</tr>
<tr>
<td>Deterministic (fast) timing</td>
<td>Interoperability</td>
</tr>
<tr>
<td></td>
<td>Low Cost</td>
</tr>
</tbody>
</table>
## Communication: Wireless Networks

<table>
<thead>
<tr>
<th>Type</th>
<th>Range</th>
<th>Power (on node)</th>
<th>Bandwidth</th>
<th>Topology</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>UMTS/LTE/5G</td>
<td>km / wide coverage</td>
<td>high</td>
<td>Very high</td>
<td>Star</td>
<td>high</td>
</tr>
<tr>
<td>Wifi</td>
<td>10-100m</td>
<td>high</td>
<td>Very high</td>
<td>Star (mesh)</td>
<td>low</td>
</tr>
<tr>
<td>Bluetooth LE</td>
<td>10-100m</td>
<td>Very low</td>
<td>Low</td>
<td>Star, p2p, mesh</td>
<td>low</td>
</tr>
<tr>
<td>LoraWAN</td>
<td>2km / wide coverage</td>
<td>Very low</td>
<td>Very low</td>
<td>Star</td>
<td>low</td>
</tr>
<tr>
<td>802.15.4 e.g. Zigbee</td>
<td>10-100m</td>
<td>Very low</td>
<td>Low</td>
<td>Mesh</td>
<td>low</td>
</tr>
<tr>
<td>RFID / NFC</td>
<td>1m</td>
<td>Very low / passive</td>
<td>Bytes on demand</td>
<td>p2p</td>
<td>Very low</td>
</tr>
</tbody>
</table>
Communication Bluetooth BLE

Jan Mitrovics, JLM Innovation GmbH
Communication LoraWAN
Smart Sensor Trends

- **Miniaturization**
  - Lower Size, Lower Power Consumption, Lower Cost
- **Integration**
  - Combination of different sensors in one package (e.g. BME-680, SCD-41, SEN-5x)
  - New sensor types through combination of technologies (e.g. photoacoustic sensors with MEMS microphones)
- **Software**
  - More complex algorithms
  - Product diversification via firmware / operation modes
IEEE 1451 smart transducers

- **1451.0–2007** Common Functions, Communication Protocols, and Transducer Electronic Data Sheet (TEDS) Formats
- **1451.1–1999** Network Capable Application Processor Information Model
- **1451.2-1997** Transducer to Microprocessor Communication Protocols & TEDS Formats
- **1451.3-2003** Digital Communication & TEDS Formats for Distributed Multidrop Systems
- **1451.4-2004** Mixed-Mode Communication Protocols & TEDS Formats
- **1451.5-2007** Wireless Communication Protocols & Transducer Electronic Data Sheet (TEDS) Formats
- **1451.7-2010** Transducers to Radio Frequency Identification (RFID) Systems Communication Protocols and Transducer Electronic Data Sheet Formats

An IoT Environment

- Distributed Nodes
- Message communication
- Application Layer / User Interface
MQTT is an **OASIS standard** messaging protocol for the Internet of Things (IoT). It is designed as an **extremely lightweight** publish/subscribe messaging transport that is ideal for connecting remote devices with a **small code footprint** and **minimal network bandwidth**. MQTT today is used in a wide variety of industries, such as automotive, manufacturing, telecommunications, oil and gas, etc.

Source: https://mqtt.org/
What is MQTT?

Main Properties:

- Lightweight and Efficient
- Scale to Millions of Things
- Support for Unreliable Networks
- Bi-directional Communications
- Reliable Message Delivery
- Security Enabled

Source: https://mqtt.org/
Topic: ws23_demo/sensor/scd41_co2/state
Topics are hierarchically ordered

MQTT has 3 defined quality of service levels:
0 - at most once,
1 - at least once,
2 - exactly once

MQTT publish messages contain a flag to retain the data

Clients can subscribe to topics using wildcards
A range of ready to use services exists to connect various sources into MQTT

- Zigbee2MQTT
- ESP32-ble2mqtt

MQTT is supported by many automation systems
  e.g. FHEM, OpenHab, Homeassistant

MQTT is easily integrated in Arduino, Python, C#, Java, ...

MQTT often directly supported by IOT devices (e.g. Tasmota, Shelly, ESPHome, ...)
IoT Tools Node Red
Node-Red Dashboard

Aquarium
- Temperature 1: 21.19
- Temperature 2: 14.56
- pH: 15.11

Envy
- Temperature: No Data
- Humidity: No Data

Shelly
- RAINBOW
- switch
- switch
- Jani LEDs On

DS18820
- Temperature
- MQ2
- No Data

Multisensor
- Temperature
- Humidity
- Pressure
- Battery: 100
IoT Tools: Time Series Databases InfluxDB2

- Influx-DB is optimized to store data (typically JSON) with TimeTags
- InfluxDB2 adds an optimized query language and integrates various services (e.g. Telegraph to automatically pull data from other sources).
- Data can be separated by Organizations and Buckets
- Buckets may contain different types of measurements
- Data can be deleted automatically after a defined period of time
- Annotations can be added to describe events at certain point of time
InfluxDB 2

Integrated Grafana to create dashboards with graphs of data from different sources.
Building a Stack for IoT using Open Source Tools

Server

- Cloud Server / Web Services: AWS / Azure / ...
- Root Server running Linux
- Single Board Computers: Raspberry Pi

Services Infrastructure

- Virtual Machines (HyperV, Proxmox, VirtualBox, ...)
- Containers (e.g. Docker)
Virtualization with Proxmox

[Image: Virtual Environment interface with various components and data displayed, including types, state, and up time of nodes and services.]
Docker Containers

Advantages over Virtual Machines:

- Lightweight (share a common kernel)
- Easy to configure complete stack (via docker-compose)
- One container per service
  - Dependencies can be met per container!
- Virtual networks connecting containers
- Configurable network bridge to the outside
- Easy to move containers (e.g. from Docker on Windows to Linux)
- Huge repository of ready to use containers [https://hub.docker.com/](https://hub.docker.com/)
- Easy to separate (and access) data via volumes
Creating a stack of services

A stack with all required services can be defined in one simple configuration file

All service can be started with one command docker-compose up –d

Containers can expose ports to the outside. Traffic between containers can be kept within the host

Images are automatically pulled from the hub.docker.com.

Data can be stored persistent in volumes that are also accessible on the host.

```yaml
services:
  node-red:
    image: nodered/node-red:latest
    restart: unless-stopped
    networks:
      - node-red-net
    ports:
      - "1880:1880"
    volumes:
      - node-red-data:/data

MQTTbroker:
  image: eclipse-mosquitto:1.6.13
  restart: unless-stopped
  networks:
    - node-red-net
  volumes:
    node-red-data:
  networks:
    node-red-net:
```
IOTstack a simple configuration tool to create a docker-compose.yml file for IOT

https://sensorsiot.github.io/IOTstack/

Optimized for Raspberry PI

Simply select different services from a menu to build a complete stack

Can run on other Linux platforms, but requires small changes to remove Raspberry Pi specific parts
Management of Docker Containers via Portainer

Jan Mitrovics, JLM Innovation GmbH