FreeRTOS Monitoring on IoT devices

Daniel Guarecuco
Maja Markusson
Juan Paños

Contents

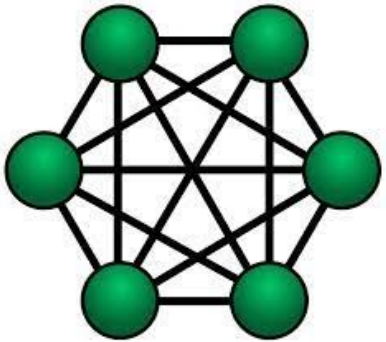
What is IoT?

“The network of devices that contain the hardware, software, firmware, and actuators which allow the devices to connect, interact, and freely exchange data and information.”

- *National Institute of Standards and Technology*

Clustering IoT devices

Why do we cluster?

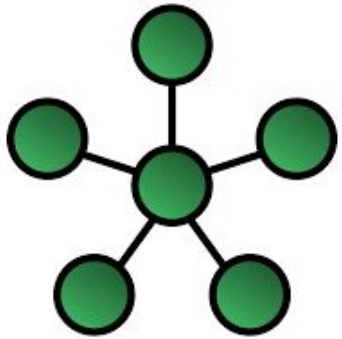


Mesh Topology

- Full or part
- All devices connected
- Inconsistent routing
- Redundancy

Star Topology

- Master node
- Independent connections
- Less fault tolerant

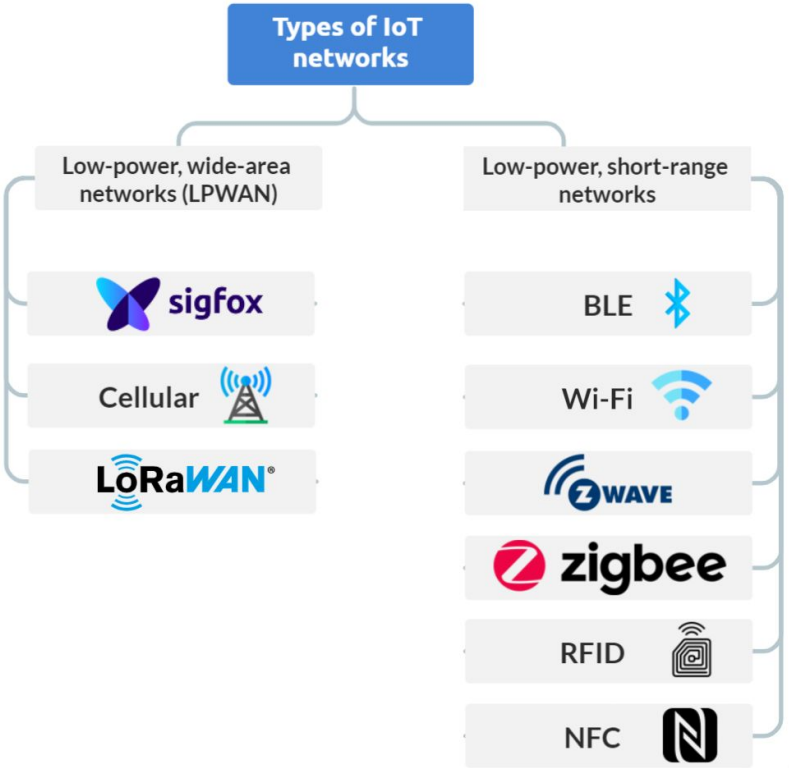


Communication protocols

Protocols vary:

- Wired vs Wireless
- Range
- Security
- Power consumption
- Topology
- Data rate

Selection depends on the use case!



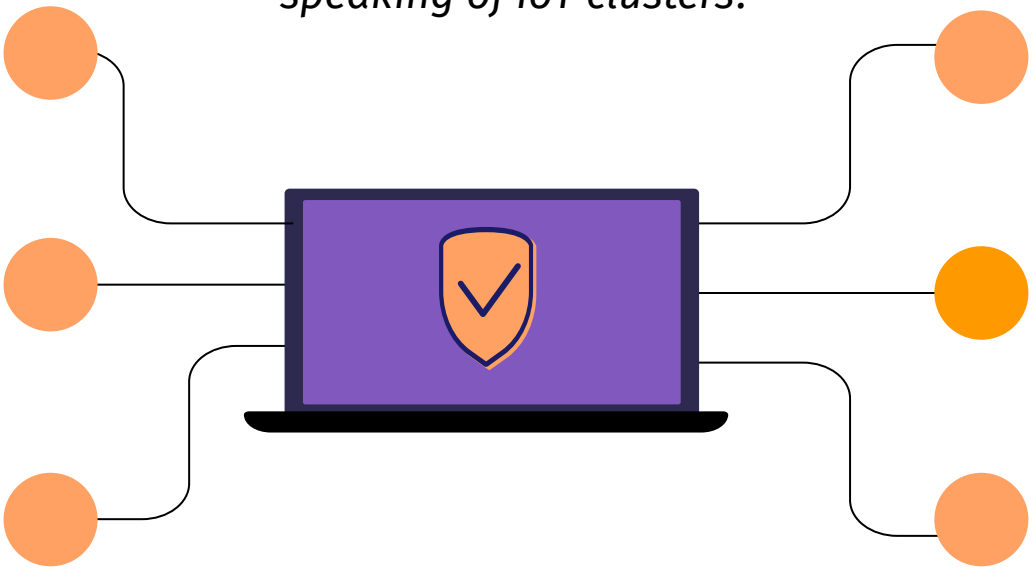
Attacks on IoT devices

Which attacks are relevant when speaking of IoT clusters?

NODE TAMPERING
Subversion, removal or addition of nodes to the cluster

EAVESDROPPING
Monitoring information in and out of node to snap up private information

MitM
Adding an additional node that information is rerouted through but not detectable in the cluster



HARDWARE ATTACKS
Destroying or tampering with the device hardware to gain access or information

ROUTING ATTACKS
Tampering with routing algorithms to clog or deprive specific routers of network traffic

DoS
Exhausting dedicated resources to make the device neglect legitimate requests

Keylogger

Malicious software that collects system events (key pressed) to collect data (user input) which is then filtered to obtain critical information such as passwords

Can it be used as an inspiration?

Intelligent Monitoring

- **Anomaly detection:** Unexpected, anomalous data can indicate critical incidents! (Attacks, System failures...)
- **User patterns:** Crucial information can be obtained
- **System specific:** Extraction of intelligent information depends on each system and requires a thorough analysis of the collected data

FreeRTOS

- Open source real time OS
- OS components
- A very simple OS which require a lot of development to obtain the desired functionality

Monitoring the FreeRTOS

Focused on collecting OS parameters and events:

- Socket activity, Tasks running, System events
- Application independent, but to extract **intelligent** information the application needs to be considered.

Challenges with monitoring

- IoT applications are often developed in close relation to the technologies/protocols used, which makes intelligent monitoring highly application dependent
- Monitoring increases the use of resources, which often is limited in embedded applications

Oil Platforms

- Wireless Sensor Network
- Part-mesh topology
- ZigBee
- Safety aspect
- Possible attacks
- Monitoring

Security Cameras

- Fixed position sensors
- Star topology
- Wifi
- Safety aspect
- Possible attacks
- Monitoring

Health Monitors

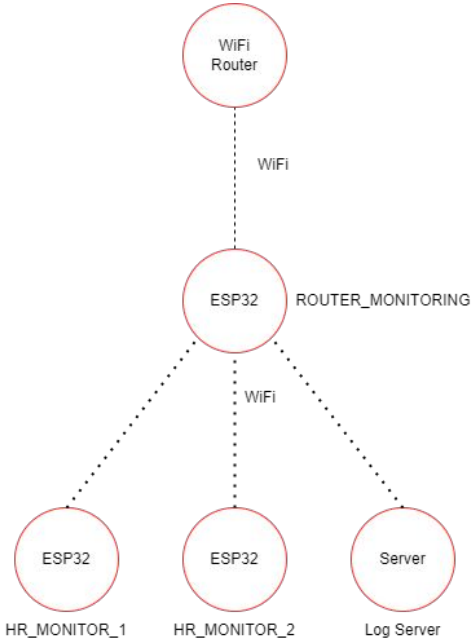
- Elderly health monitoring
- Star topology
- WiFi
- Privacy
- Possible attacks
- Monitoring

Demo

- Implementing health monitoring
- Using the ESP32 board
- Added FreeRTOS+esp-lwip support
- Monitoring sockets + Network activity + DoS attacks.

ESP32

- Dual core
- Bluetooth and WiFi support
- ESP-IDF build tool



Monitoring demo

Running tasks

- The scheduling of an embedded device should be as deterministic as possible
- An excessive amount of tasks could indicate that an DoS attack has been launched

Too many tasks == **DoS Attack!**

```
| DoS suspected, number of created tasks: 7
| Maximum number of task expected for period: 5
| Task name | Task base priority
| DoS test | 3
| DoS test | 3
| DoS test | 3
| DoS test | 3
| DoS test | 3
| DoS test | 3
| Heart Rate | 3
```

Monitoring demo

UDP Packets

- Sockets are monitored:
- If packets are being sent to unexpected ip/Have unexpected size?

Information theft! **DoS Attempt!**

```
(HR_MONITOR_1) UDP connection to Socket: 54, IP: 192.168.1.1, Port 35500, Size: 3 bytes
(HR_MONITOR_2) UDP connection to Socket: 54, IP: 192.168.1.1, Port 35500, Size: 3 bytes
(HR_MONITOR_1) UDP connection to Socket: 54, IP: 192.168.1.1, Port 35500, Size: 3 bytes
(HR_MONITOR_2) UDP connection to Socket: 54, IP: 192.168.1.1, Port 35500, Size: 2 bytes
```

Monitoring demo

Connected Nodes + MAC addresses

- Number of nodes should be known in this use case (Application level)
- Too many nodes can imply **false nodes!**

```
(ROUTER_MONITORING): WARNING: Number of connected stations above maximum threshold! Connected = 4, Maximum expected = 3
```

- Too few nodes can imply **jamming/DoS attacks on nodes...**

```
(ROUTER_MONITORING): WARNING: Number of connected stations below minimum threshold! Connected = 1, Minimum expected = 2
```

- Mac not whitelisted could mean **malicious node!**

```
(ROUTER_MONITORING): WARNING: Station MAC address not in list of expected values! (78:e3:6d:09:a0:58)
```

Conclusion

- Attacks detected by monitoring the OS is limited
- Choice of protocol and topology changes vulnerabilities and attack surface
- Intelligent monitoring is connected to knowledge of the system