Architecture of a Wireless Mesh Sensor Network for Real-time Measurements and Alarms in Low Voltage Distribution Networks

Martijn Kuipers, Tiago Pinheiro, João Pedro Taveira, Mário Nunes
INOV INESC Inovação, IST/UL

19th RTCM, July 4, 2014

inov
inesc • inovação

Outline

- Monitor BT project goals
- System architecture
- Fault detection and location
- Public Lighting faulty light bulbs detection and location
- RF Mesh Nodes
- Field Area Network (FAN) Protocol stack
- DLMS vs SEP 2.0 and Modbus
- DLMS message flows
- RF Mesh node (HW and SW)
- Conclusions

Monitor BT project (EFACEC, INOV, EDP Distribuição)

- Improve the monitoring and control of distribution LV grids:
  1. Fault detection and location in LV feeders (default current detection, voltage outage). Demonstration in the field (Batalha);
  2. Public Lighting faulty light bulbs detection and location. Demonstration in the field (Batalha);
  3. Control the power being injected by micro-generation (e.g. PV) for voltage regulation. Demonstration in lab. Environment

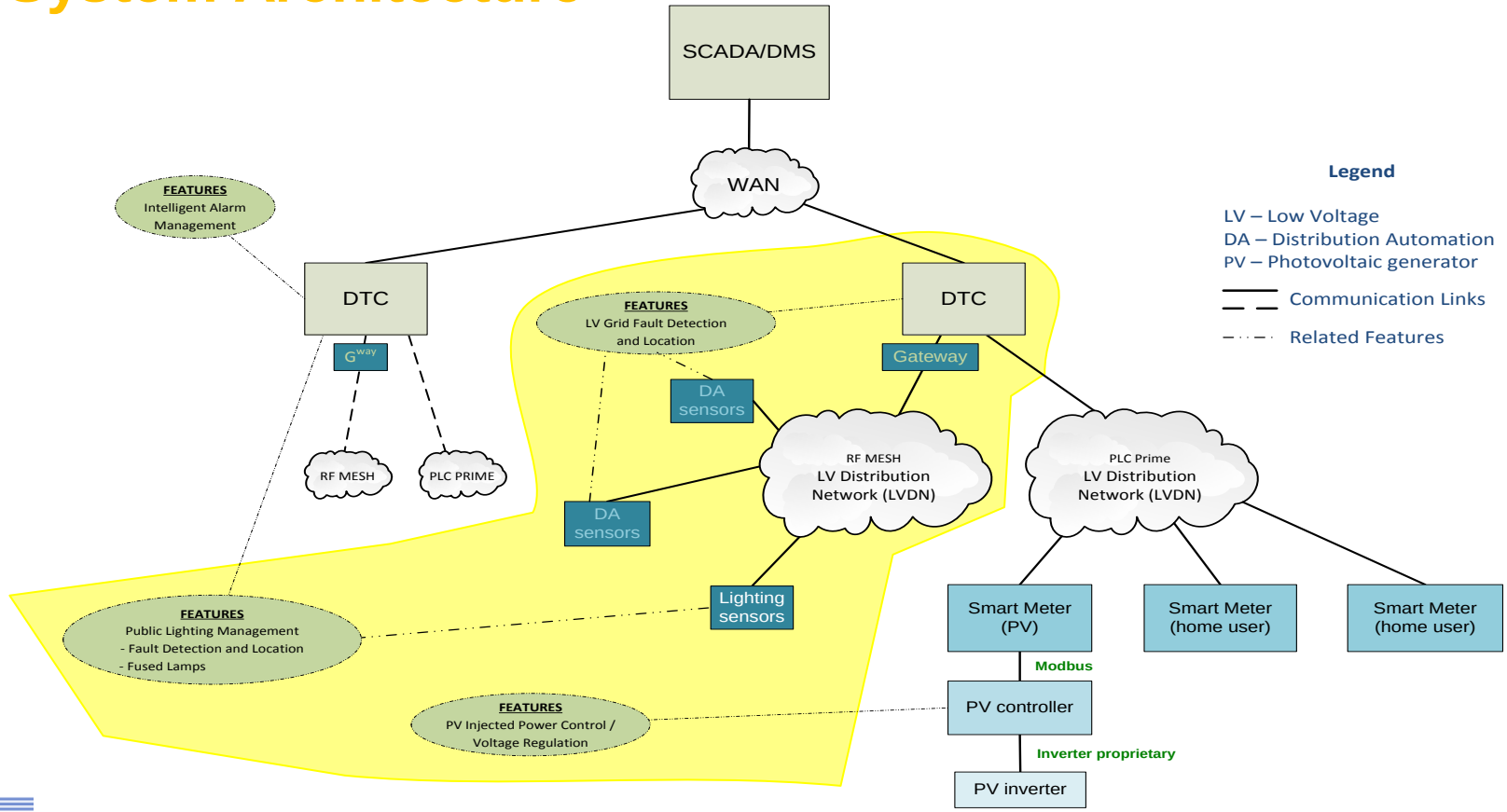
Monitor BT is financed by:

QUADRO DE REFERÊNCIA ESTRATÉGICO NACIONAL PORTUGAL 2007.2013

COMPETE PROGRAMA OPERACIONAL FACTORES DE COMPETITIVIDADE

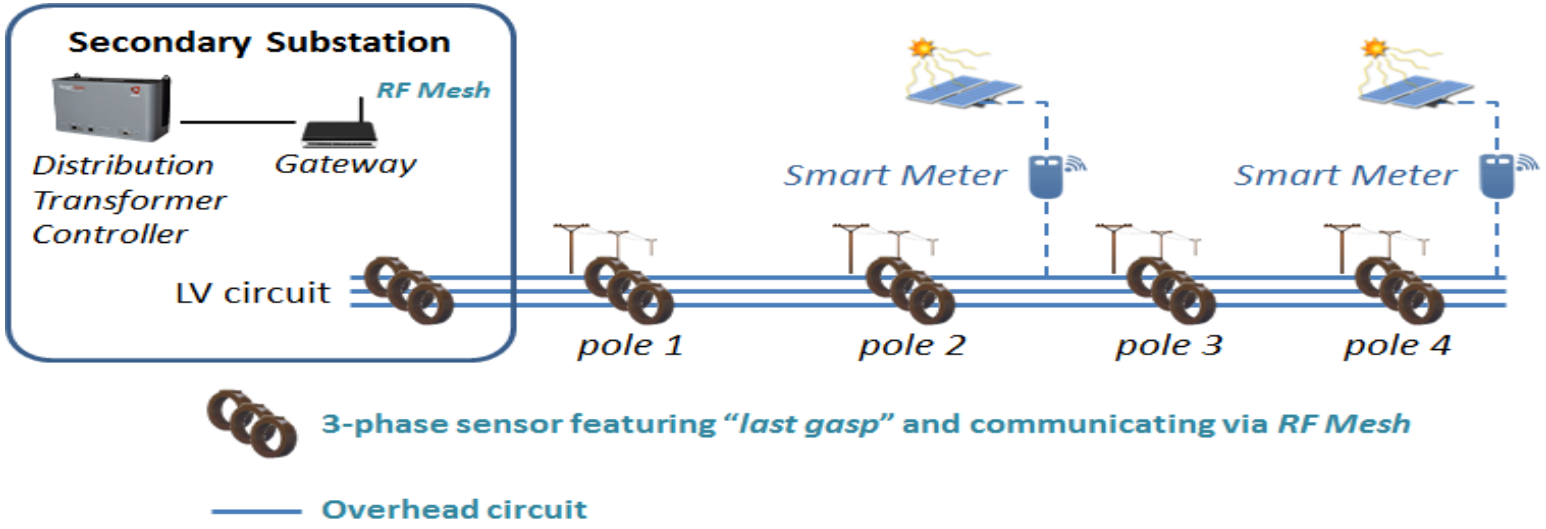
UNIÃO EUROPEIA Fundo Europeu de Desenvolvimento Regional

System Architecture



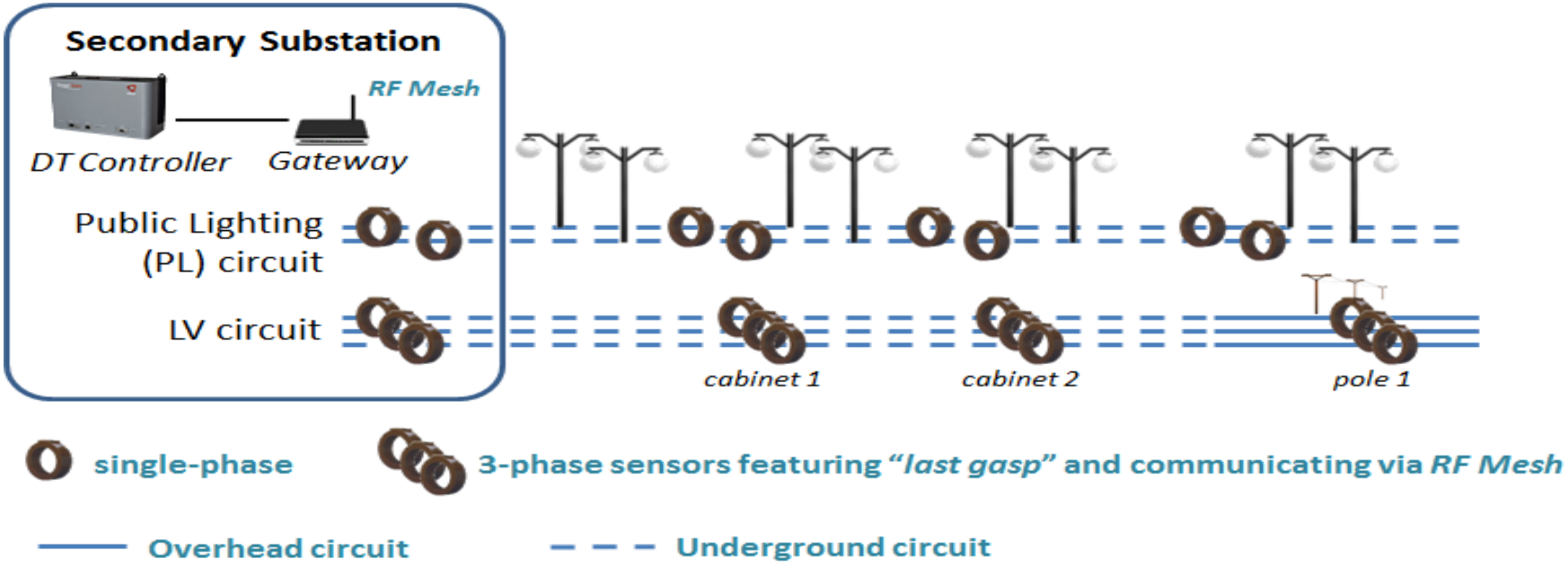
Fault Detection and Location

The project comprises sensors for monitoring the LV grid and µG. Under “Last Gasp” mode, when sensors detect outages, even without being powered, they are able to report outages upstream to the DTC, the feeder master unit.

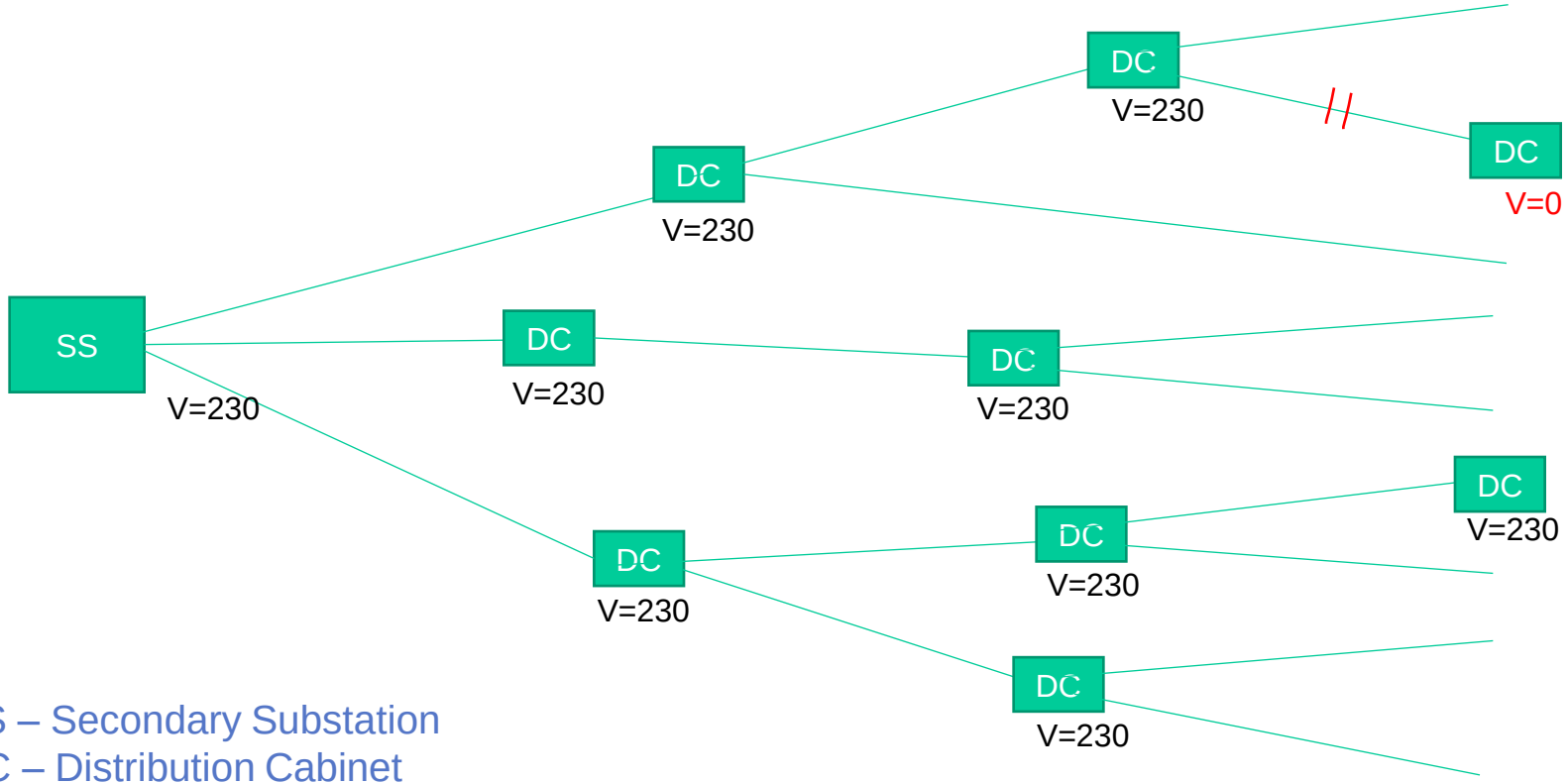


Public Lighting faulty light bulbs detection and location

The project also includes sensors for monitoring the LV grid, as well as PL feeders.



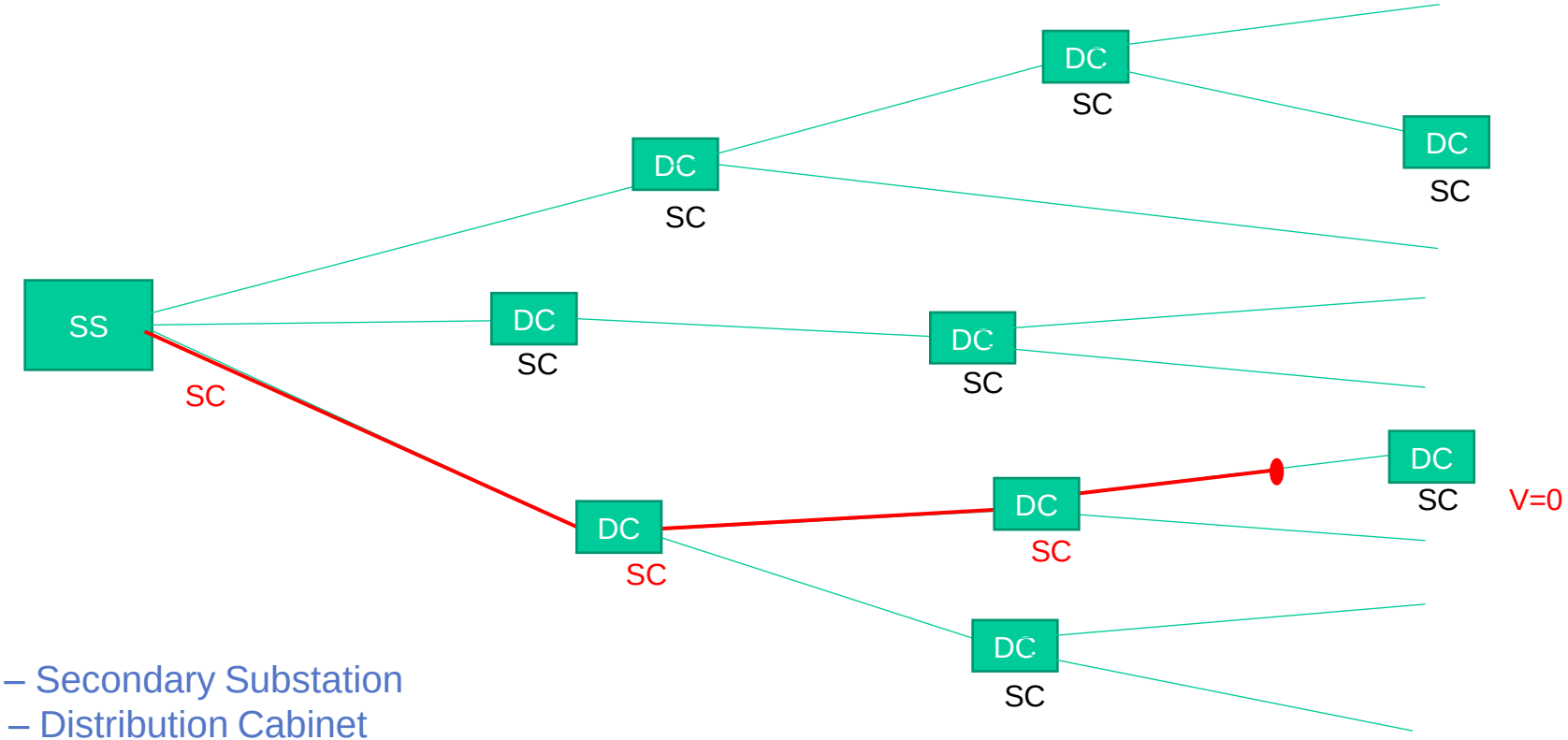
Fault detection and location: line interruption



SS – Secondary Substation
DC – Distribution Cabinet

V– Voltage Detector

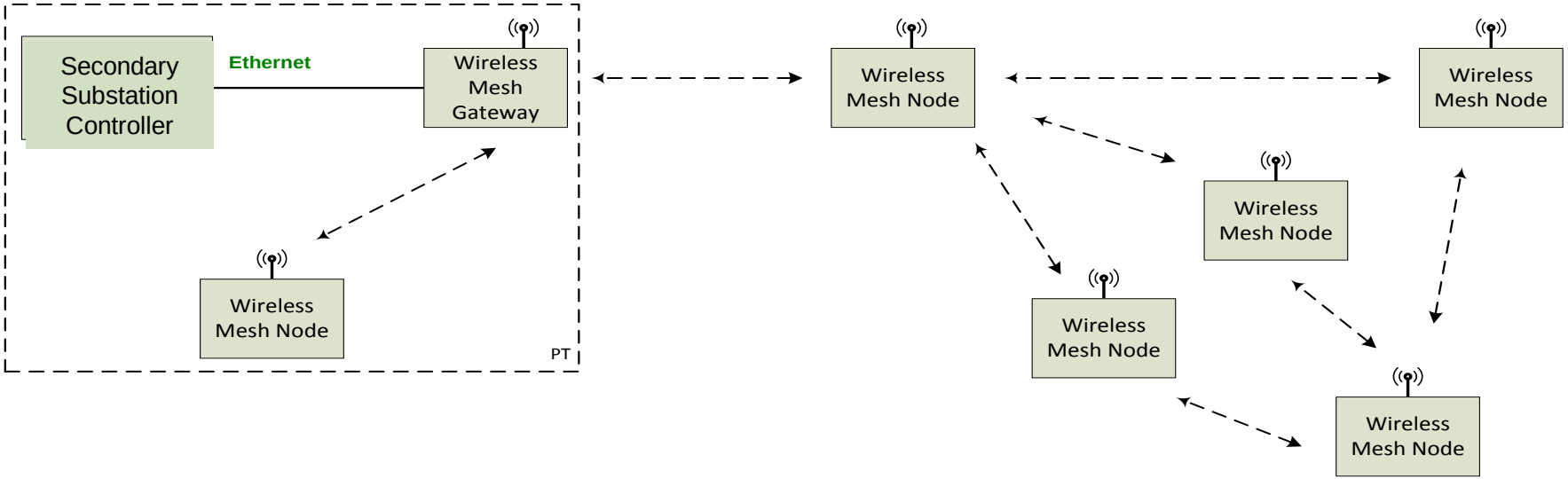
Fault detection and location: short-circuit in the line



SS – Secondary Substation
DC – Distribution Cabinet

SC – Short Circuit sensor

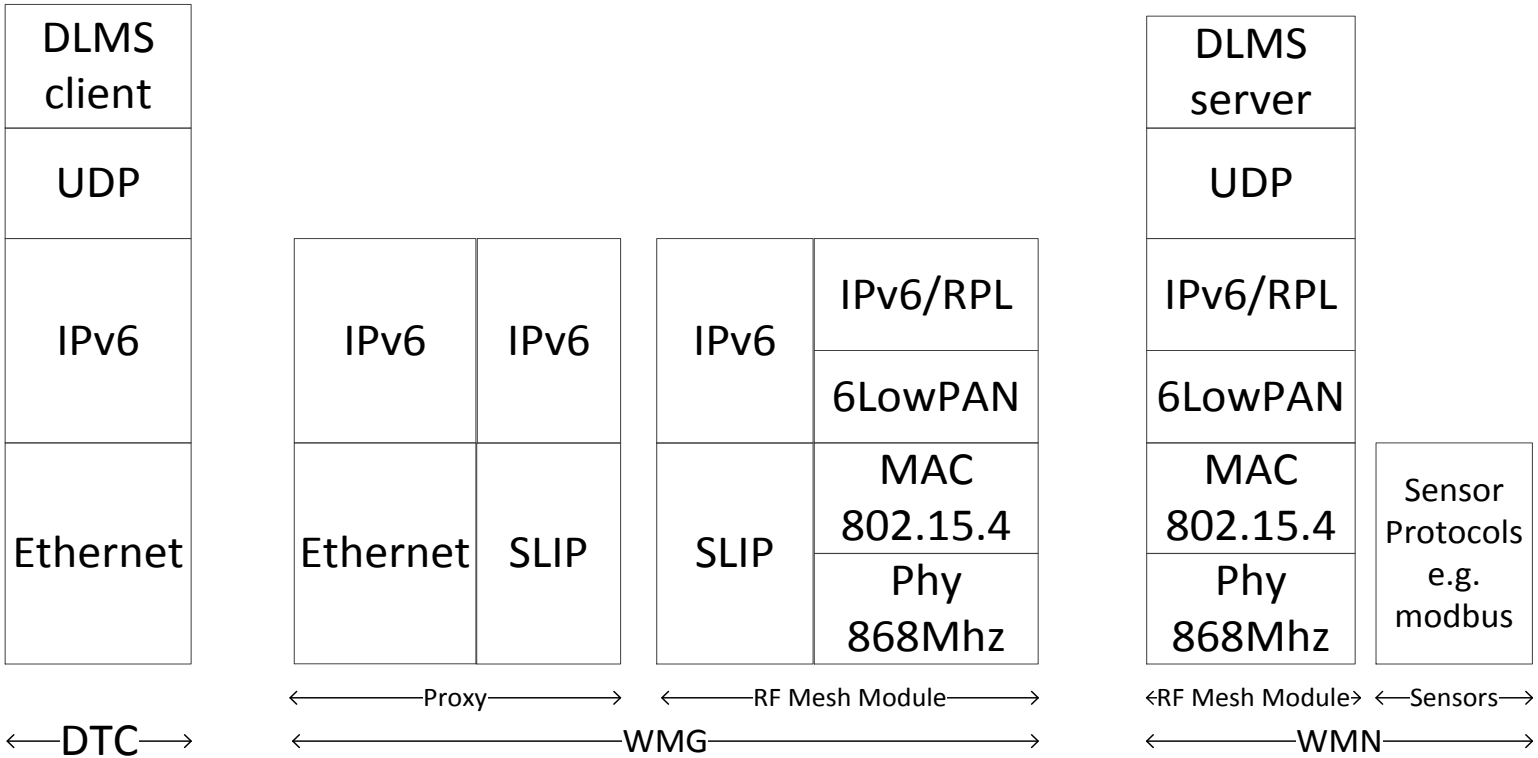
RF Mesh Architecture



RF Mesh Nodes

	High performance node	Low performance node
OS	Linux	Contiki
CPU	Cortex A8 @ 1GHz RAM 512MB, 2GB Flash	ATMega1284p @ 16MHz, 16kB RAM, 128KB flash
Power consumption:		
Idle	~ 2.5W (500mA @ 5V)	~0.1W (35mA @ 3.3V)
Receiving	~2.9W (580mA @ 5V)	~0,33W (100mA @ 3.3V)
Transmitting	~4.5W (900mA @ 5V)	~1.7W (500mA @ 3.3V)
Flexibility	Supports many interfaces, via SPI, UART and USB host	Supports many platforms and radio modules

Field Area Network (FAN) Protocol Stack



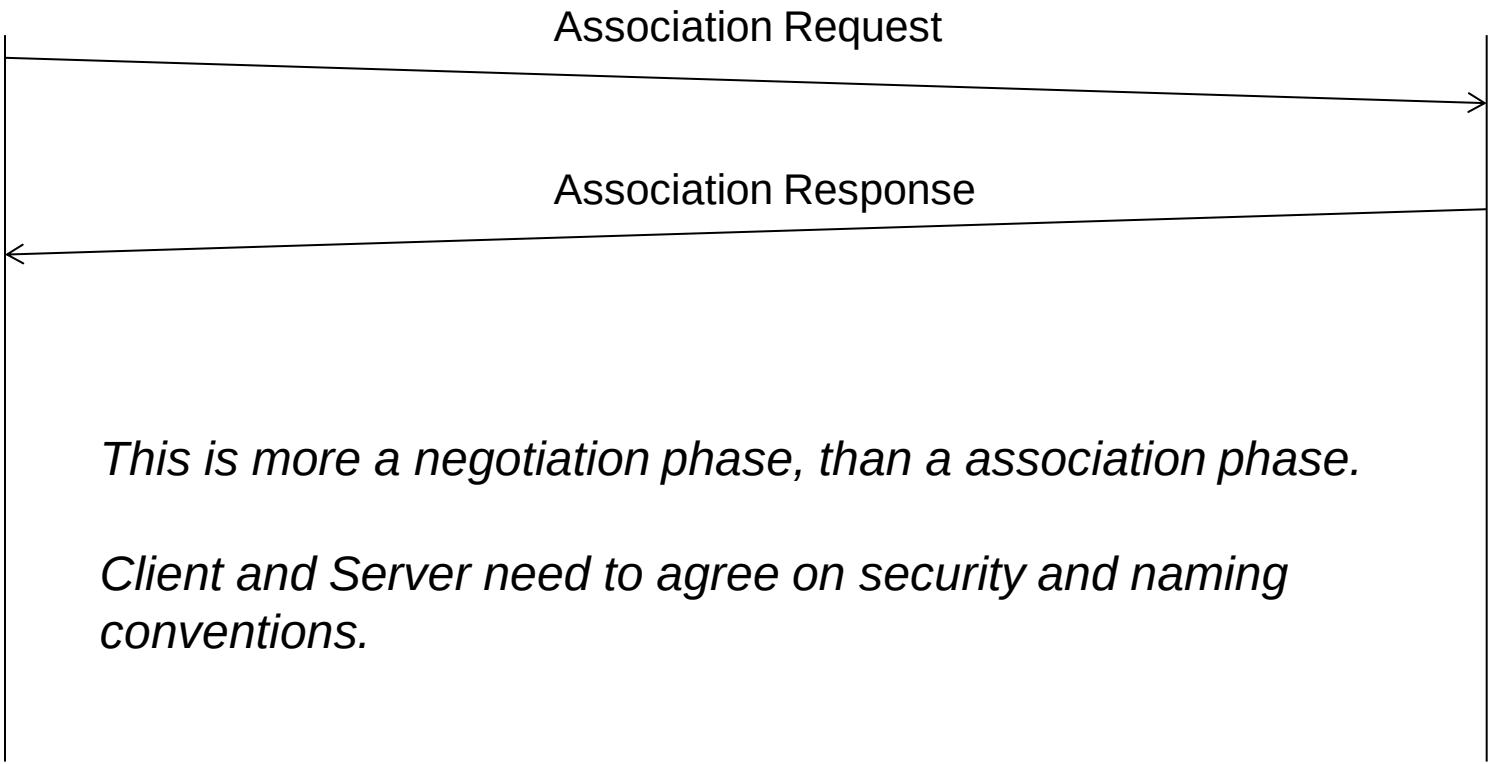
DLMS – Device Language Message Specification

- a generalized concept for abstract modeling of communication entities
- Defines universal definitions
- Mostly a request/response architecture (exception for alarm events).
- Sensors are seen as “objects”, identified by a 6 byte identifier, e.g., 1.0.32.7.0.255 (Instantaneous Voltage value Phase 1).
- The different attributes of the object have a integer index for the get and set functions.
  - 1. Logical Name (just the 6 byte object id of the object)
  - 2. Value (all fields carry a field-type identifier, long unsigned, etc.)
  - 3. Scaler/unit (scaler is the exponent to the base of 10 of the multiplication factor / the unit can be an enumerator or a string “V”)

DLMS and the competition

- SEP2 (Bluetooth Smart Energy Profile 2):
  - XML message format with exi as compressed binary format.
    This is not easy to implement in 128KB flash, which also has to do other things.
  - Intelligence is in the format
- Modbus
  - There are no alarm events. The client needs to have deep knowledge of the exact sensor (manufacturer, type, etc) in order to be able to make sense of the data
- DLMS
  - Supports events (recent additions)
  - Intelligence is in the standard (probably why it is so long and in several “books”)
  - Our project partners have their control software around DLMS

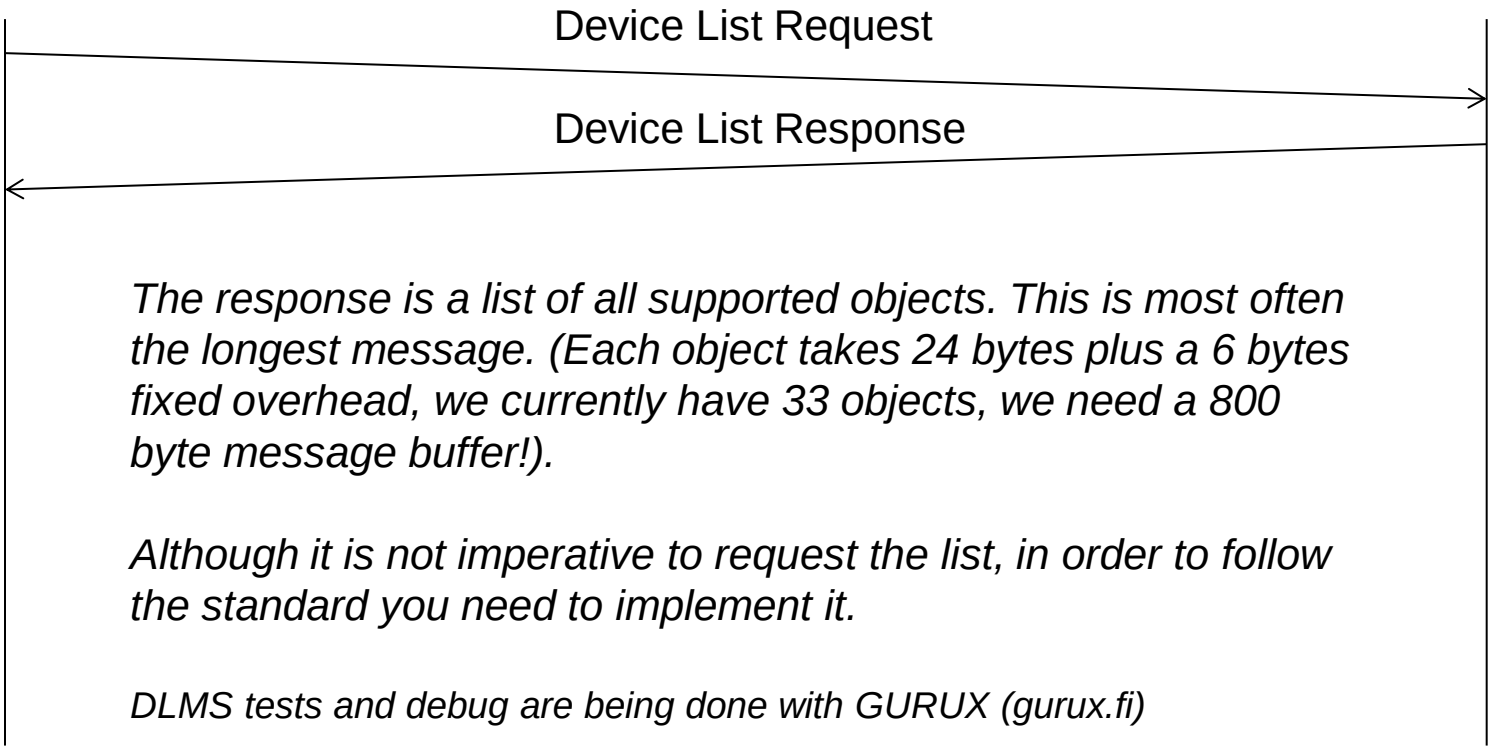
DLMS Message Flow



This is more a negotiation phase, than a association phase.

Client and Server need to agree on security and naming conventions.

DLMS Device List Request

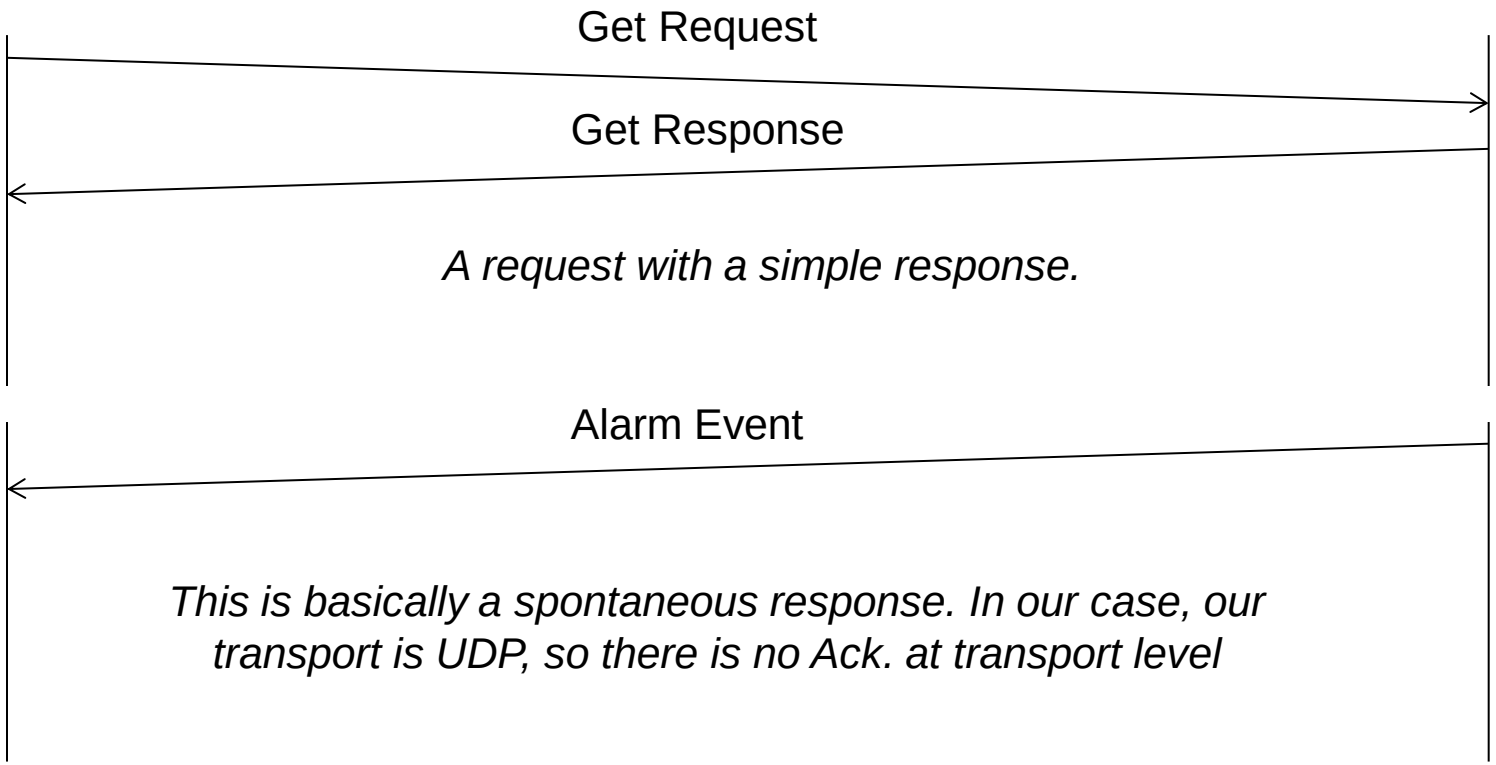


The response is a list of all supported objects. This is most often the longest message. (Each object takes 24 bytes plus a 6 bytes fixed overhead, we currently have 33 objects, we need a 800 byte message buffer!).

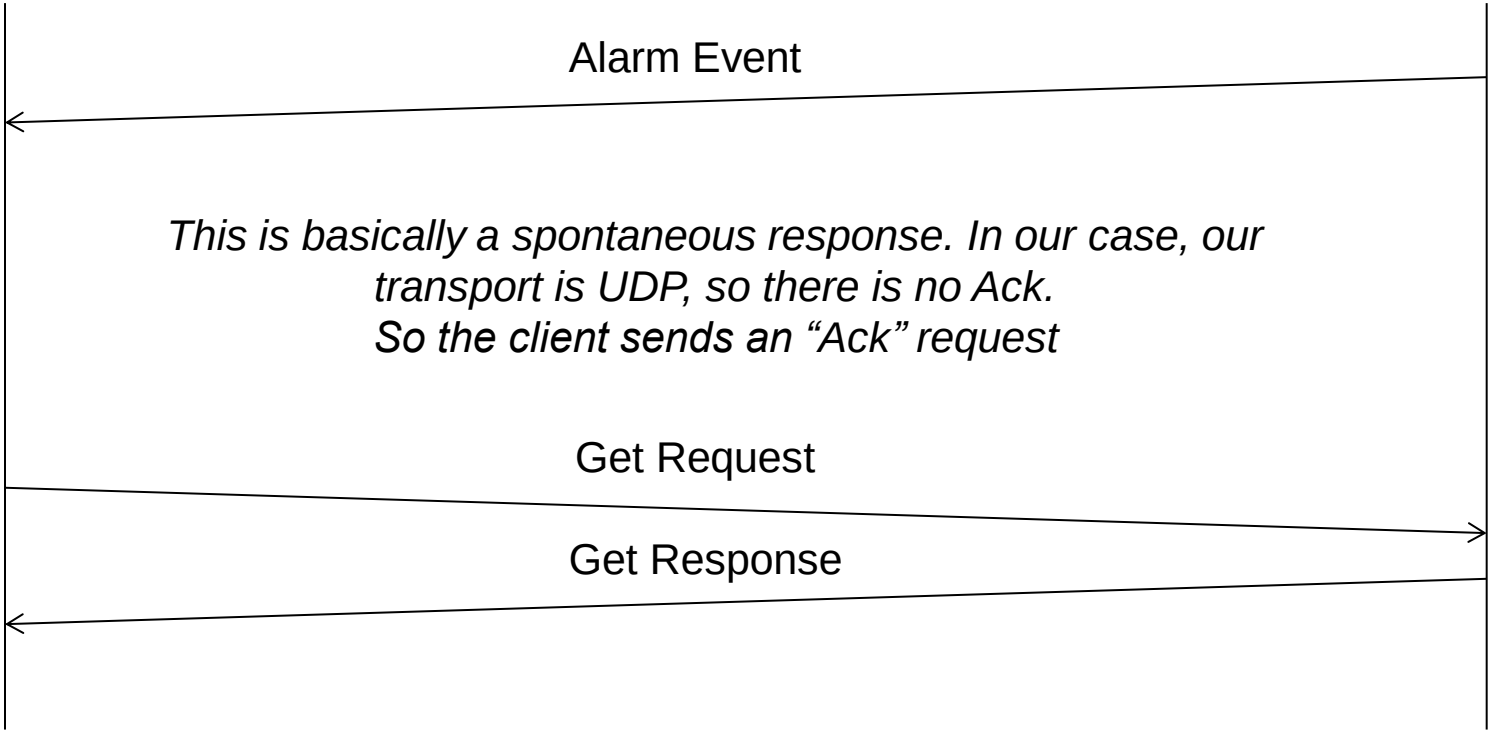
Although it is not imperative to request the list, in order to follow the standard you need to implement it.

DLMS tests and debug are being done with GURUX (gurux.fi)

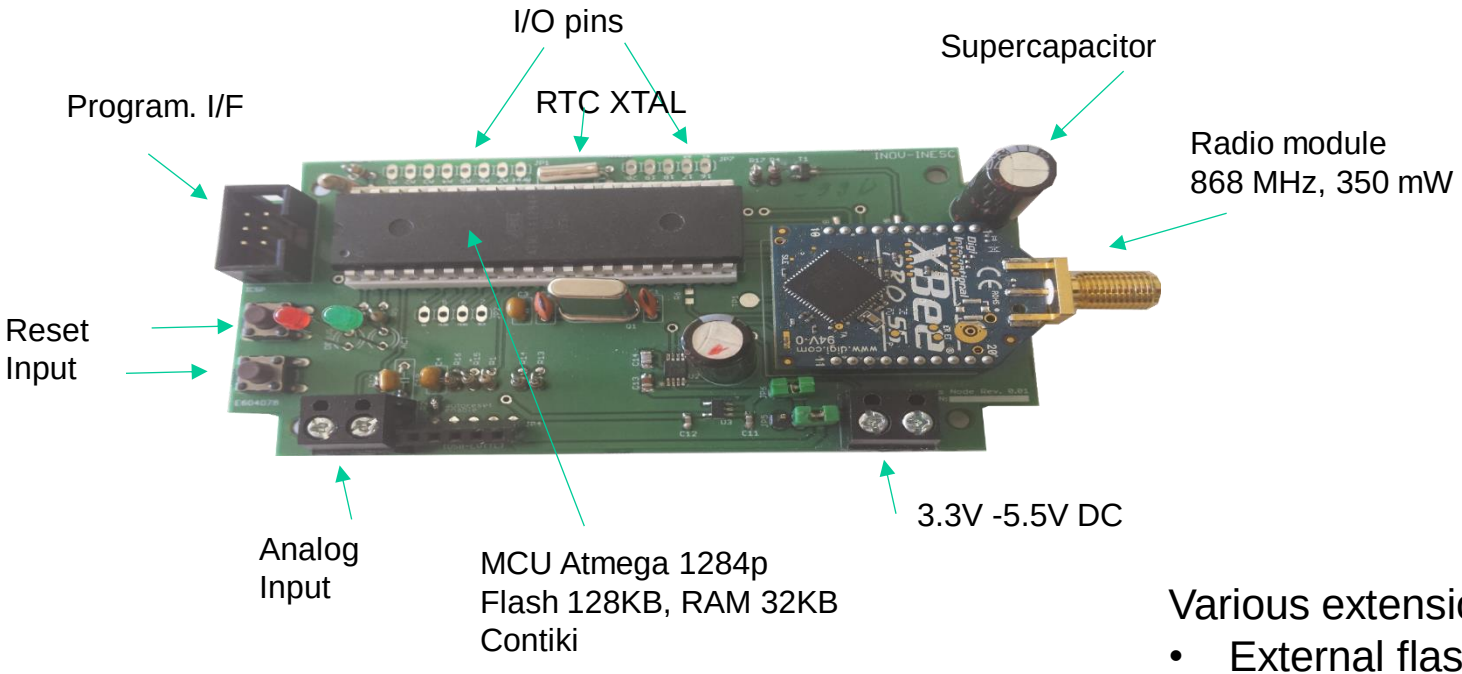
DLMS Request/Response



DLMS Alarm Events



“iBee” HW details



Various extensions:

- External flash (for OTA)
- Temperature sensor via SPI
- Etc.

iBee Software developments

- Contiki changes/additions
  - Add Atmega1284p support
  - Add xBeePro 868 driver
  - Update atmega1284p bootloader to support external flash
  - Add contiki RTC sync to xBeePro driver
- Protocol developments
  - Time sync protocol (NTP like)
  - CoAP
  - Observe (alarm configuration/subscription)
  - DLMS Server

Status and Conclusions

- The project will demonstrate the benefits of extending wireless sensors for active fault management and voltage control on LV grids, also able to integrate SM via RF Mesh or PLC.
- One gateway and one node already tested and deployed in Batalha
- First results allowed to detect removal of 1lamp in a segment of 20
- Communication between gateway and nodes achieved 400m