Master
ENGINEERING (MSE)

DeSEm
Design and Specification of the DeseNET Protocol

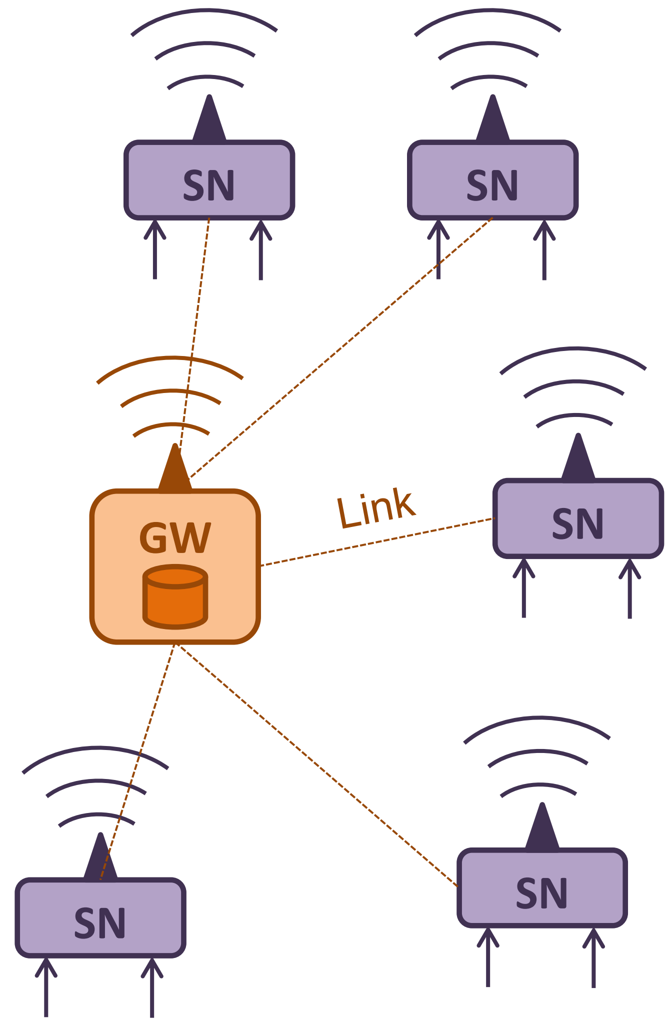
Dominique Gabioud
Michael Clausen
Thomas Sterren
Medard Rieder

HES-SO 2020/21

Table of Content

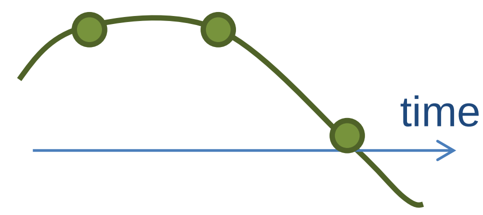
Context

- Wireless sensor network
  - GateWay (GW)
    - Role: store sensor data
      - Local storage capacity or connected to "cloud"
  - Sensor Nodes (SN)
- Star topology
  - Only GW – SN communication
- Battery powered sensor nodes
  - DeseNET protocol should enable low power implementation strategies
- Not a Plug and Play protocol
  - "Keep it simple and stupid!"
    - Manual configuration and/or implicit assumptions about data formats



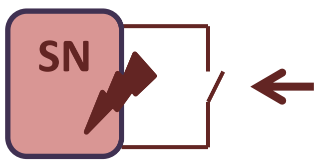
What kind of data do SNs acquire?

- Sampled Values
  - Discrete samples of a continuous signal
    - Digitised by an ADC
- Events
  - Events occur not regularly over time
    - Usually rather infrequently
  - Events can be:
    - Change of state for a binary input,
    - Continuous values above/below threshold,
    - …



Sampled value service

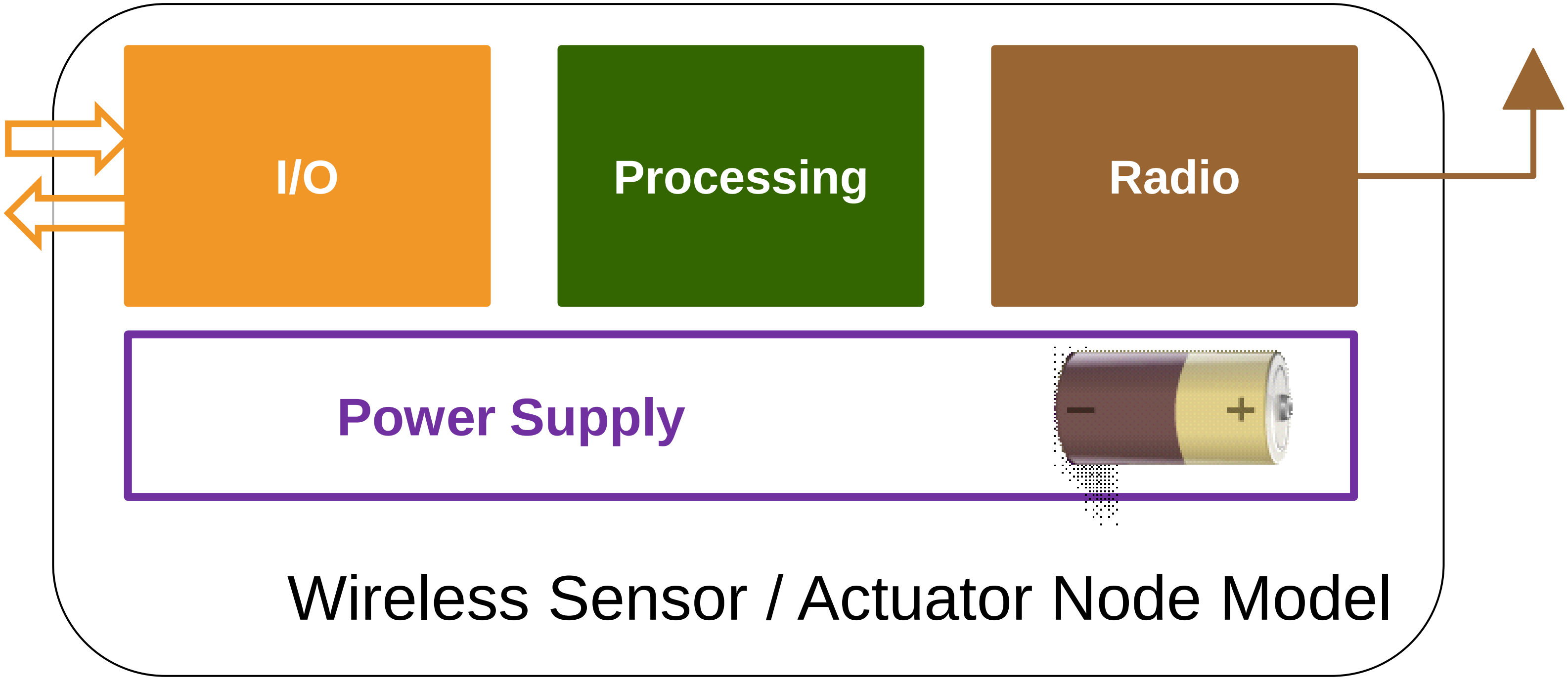
It is required that sampling is performed synchronously on all nodes, to provide a good "picture" of a process .
Frequency and phase synchronicity required!



Event service

Protocol stack should implement **a sampled value service** and an **event service**

Wireless & energy: Model

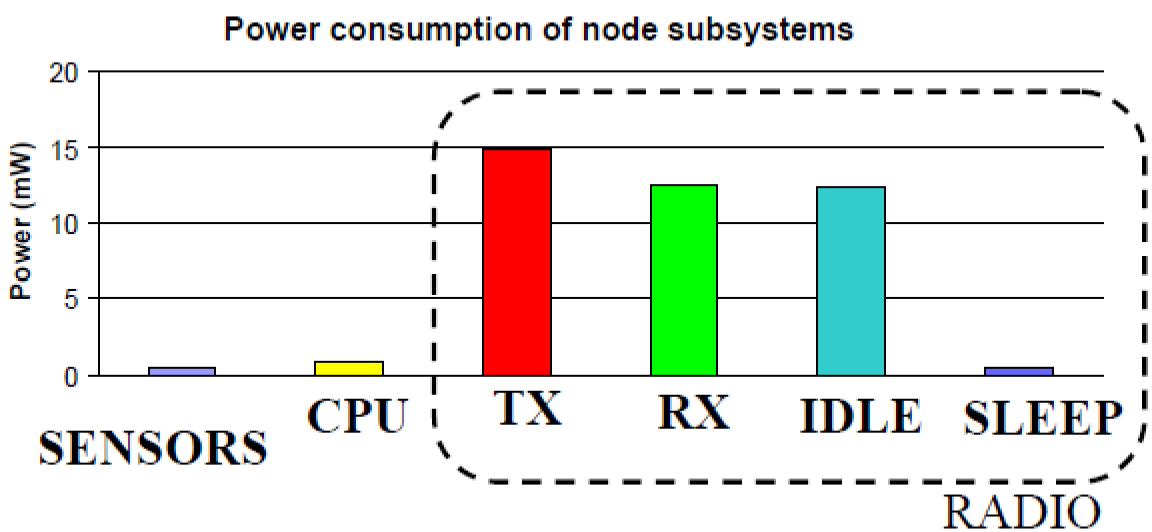


The wireless radio is often the most energy consuming unit of a node

Wireless & energy: Transceiver state

- Radio transceiver states:
  - **Transmit (TX)** : The radio is sending frames over the air to peer nodes
  - **Receive (RX)** : The radio receives frames from peer nodes
  - **Idle**: The radio receiver is turned on, but no frame is incoming
  - **Sleep**: The radio is turned off
- Radio transceiver control:
  - ON / OFF command
    - OFF: **Sleep** state
    - ON: Default state is **Idle**
      In case of incoming frame, the transceiver goes automatically in **Receive** state
      Upon frame transmission request, the transceiver goes in **Transmit** state and comes back to **Idle** state after the end of transmission

Wireless & energy: Distribution of consumption



From Tsiatis et al. 2002

The **Transmit**, **Receive** and **Idle** states consume almost the same energy

In the **Sleep** state, the consumption is 3 orders of magnitude lower

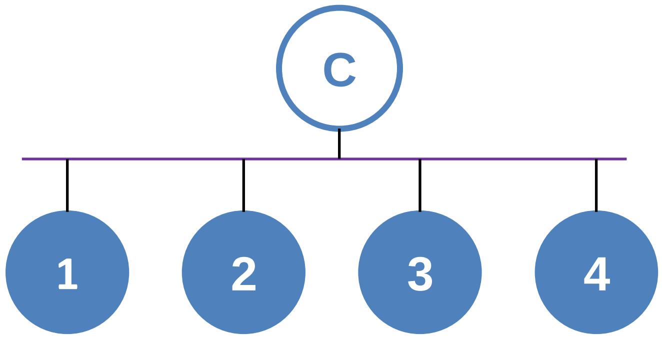
MAC & Energy

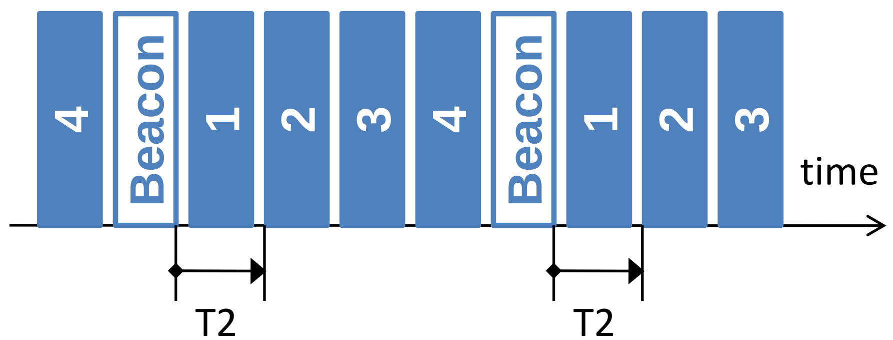
- MAC (Medium Access Control) layer role
  - Organise access to a shared communication channel by several nodes
  - Wired bus or wireless channel
- MAC & energy efficiency
  - An energy efficient MAC must control the radio transceiver states for minimum energy consumption
    - While still fulfilling the expected requirements
- Major sources of energy wastes:
  - Collision: Frames transmitted simultaneously by several nodes are corrupted.
    - Retransmission increases energy consumption
  - Overhearing: A node picks up packet destined to other nodes
  - Overhead: Sending and receiving control frames or control fields in data frames
  - Idle listening: Radio receiver turned on but no incoming frames

Principle of TDMA MAC

TDMA:
Time Division Multiple Access

- A node with coordinator role broadcasts periodic beacons
- Each node has its reserved transmission time slot
  - Fixed beacon to time slot delay
- Each node can read time slots
  - Time slot determined by beacon delay time
- Typical use:
  - Continuous bit rate “streaming applications”
  - **DECT** (Digital Enhanced Cordless Telecommunications)





TDMA MAC & energy

- Source of energy wastes
  - Collision: ++
    - No collision
  - Overhearing: ++
    - Non-existent, at least for nodes without actuator
  - Overhead: ++
    - Control frames kept to minimum
  - Idle listening: ++
    - Non-existent, at least for nodes without actuators

Performance assessment (best grade: "++")

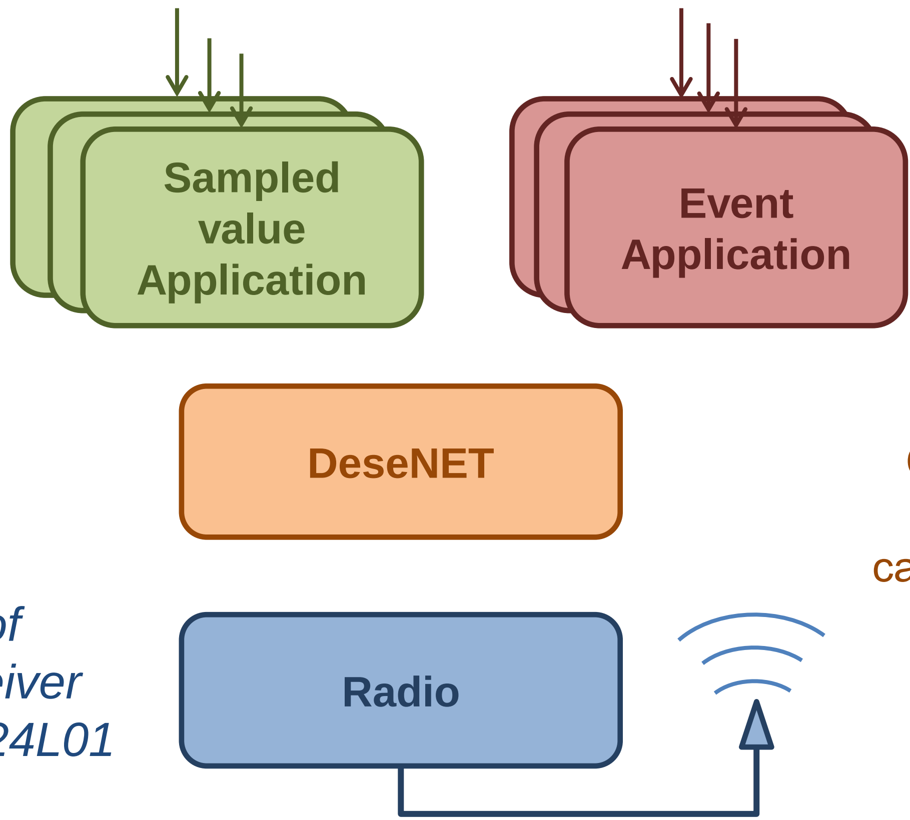
TDMA: An overall excellent behaviour regarding energy

Any limitation for TDMA MAC?

DeseNET layer architecture

DeseNET protocol

OSI Application Layer



TDMA MAC

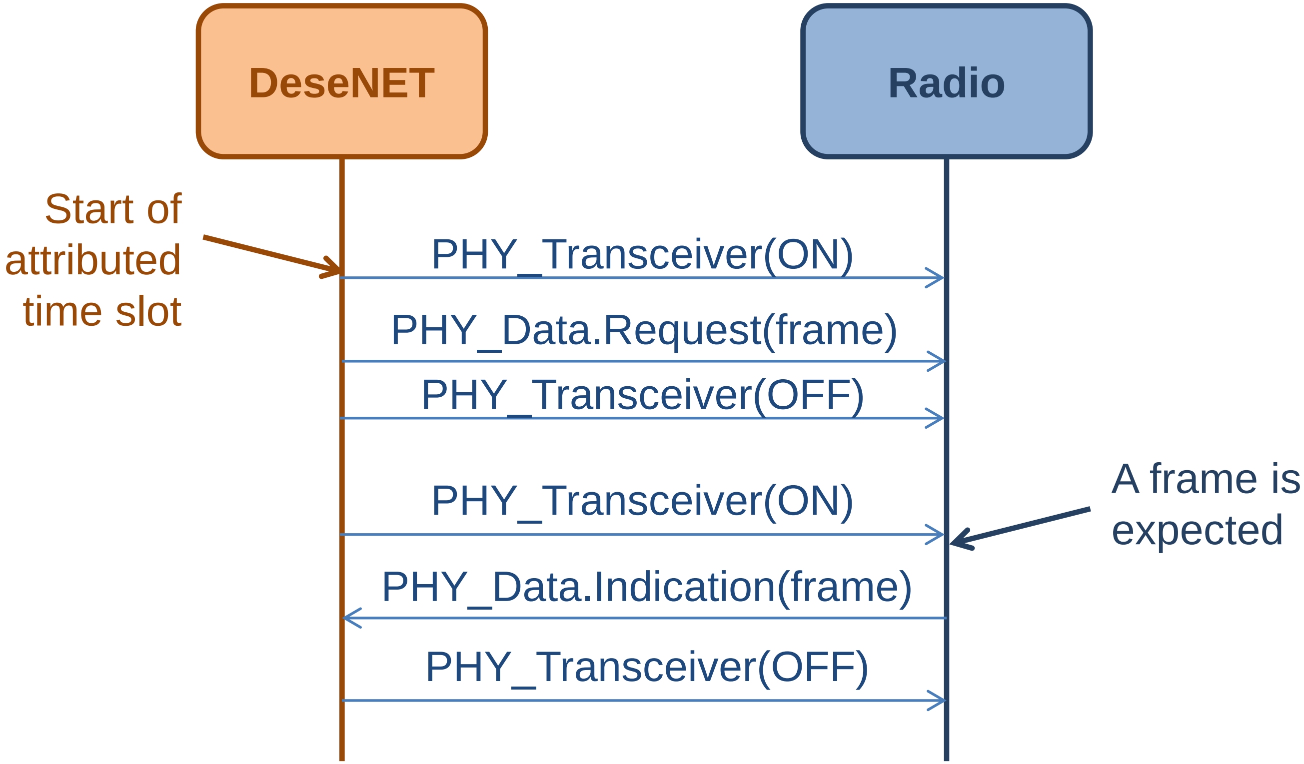
OSI Data Link Layer
(MAC sometimes called “OSI 1.5 sub-layer”)

Abstraction of radio transceiver Nordic nRF 24L01

OSI PHY Layer

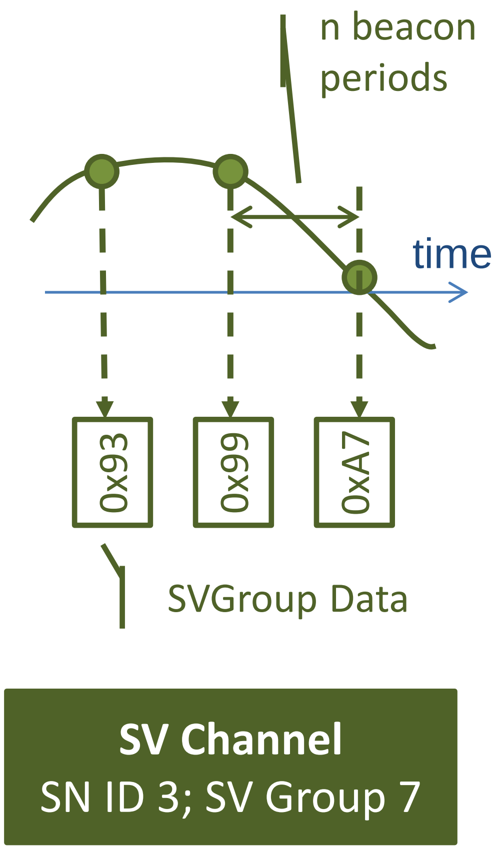
Radio & DeseNET layers

DeseNET protocol



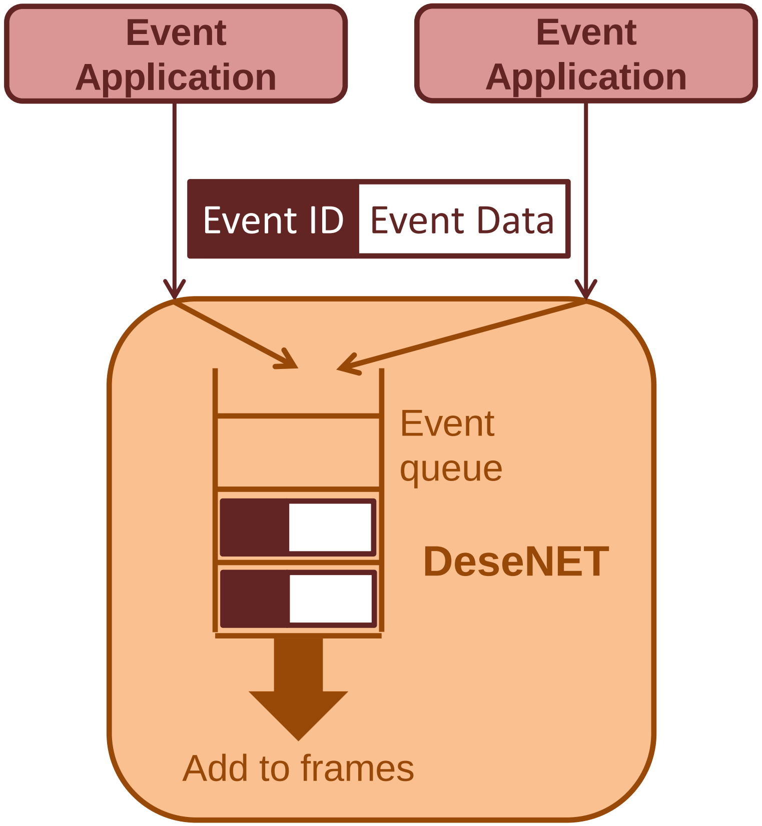
The Sampled Value service

- Chunk of data (SV Group Data) collected periodically by local sensors
  - A SV Group Data block is typically obtained through sampling (ADC) of a continuous signal
  - A SV Group Data block may contain several multiplexed sampled signals or any other periodically generated data block
    - Encoding of the SV Group Data is outside the scope of DeseNET
- Sequence of SV Group Data from the same source build a SV Channel
  - A Sampled Value Channel is identified by:
    - the SN ID of the originating SN, and
    - a so-called SV Group parameter, which identifies the local SV Group Data source
  - A SV Group identifies similar sources over all SNs
    - Example: SV Group7 -> Accelerometer sensor samples
- SV Group Data are transmitted on request of the Gateway:
  - Beacons carry Sampled Value transmission request for a given SV Group
  - A request for a given SV Group is transmitted in every n beacons (n =1, 2, 3…)
    - n may be different for each SV group

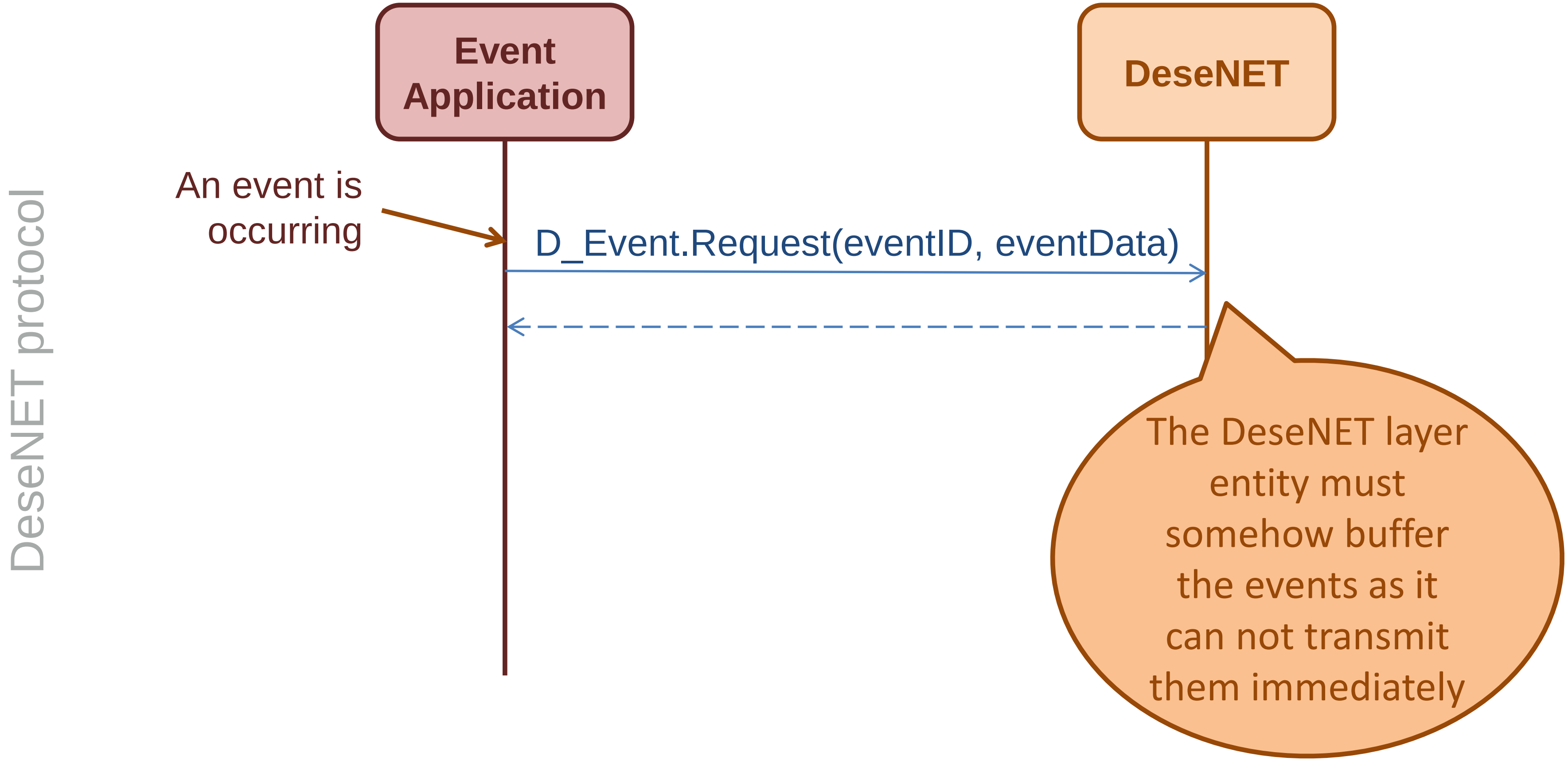


The Event service

- An event is characterised by:
  - an Event ID, and
  - associated Event Data
- There is no restriction on the pace of occurrence of event. Hence, DeseNET:
  - queues events generated by local applications
  - empties the event queue as fast as possible. Extracted events are sent using the DeseNET protocol



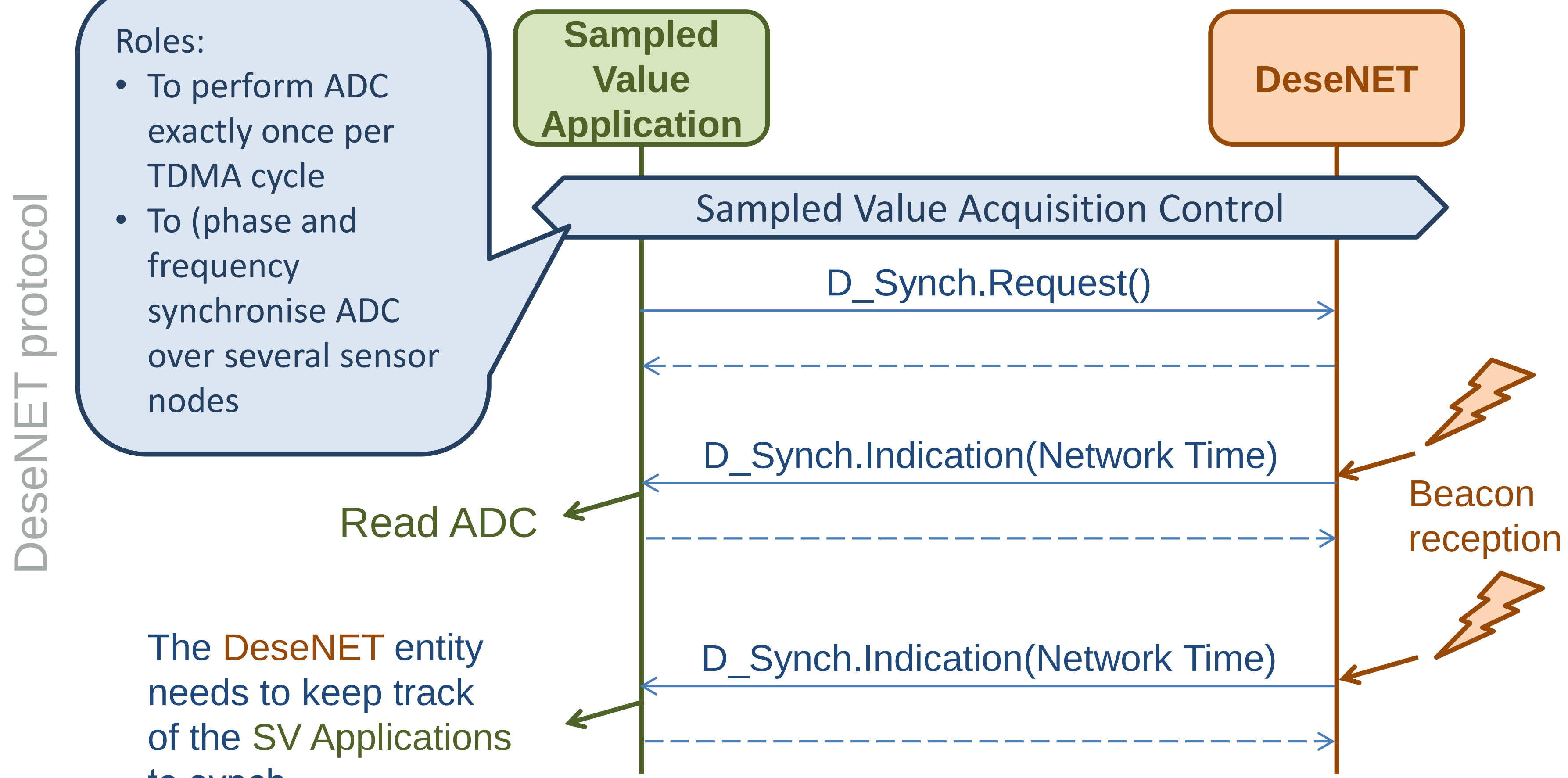
DeseNET & Event Application layers



DeseNET & SV Application layers: Synchronising ADC

Roles:

- To perform ADC exactly once per TDMA cycle
- To (phase and frequency synchronise ADC over several sensor nodes



The DeseNET entity needs to keep track of the SV Applications to synch…

DeseNET & SV Application layers: Synchronising ADC

- DeseNET enables a Gateway to trigger ADC reading on a per cycle basis
  - The Gateway may ask to read & transmit a selected subset of the SV Groups at each cycle

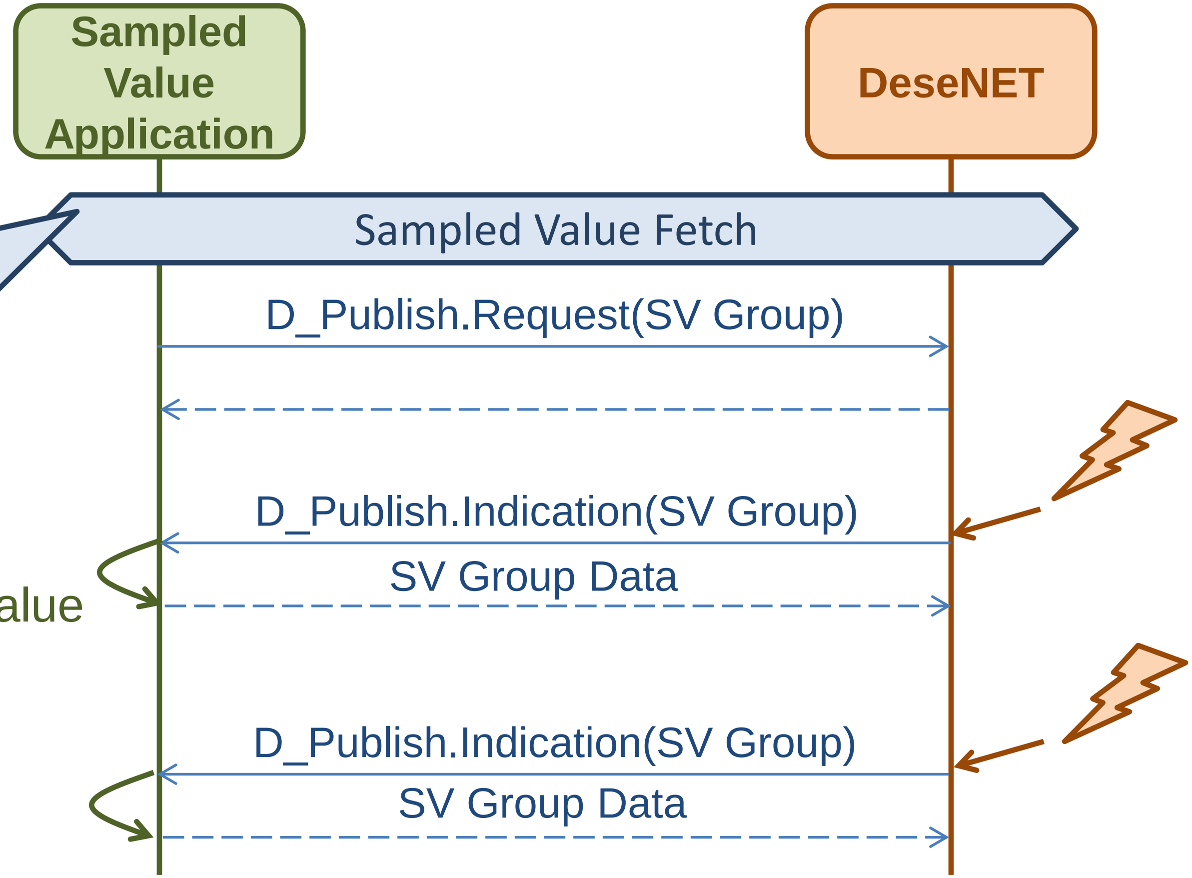
- Synchronisation service is optional
    - ADC read could be performed cyclically for example

But it's a good practice to read samples from ADC at the TDMA transmission rate

DeseNET & SV Application layers: Getting the sampled values

DeseNET protocol

Roles:
- To fetch the most recent Sampled Value buffer for wireless transfer

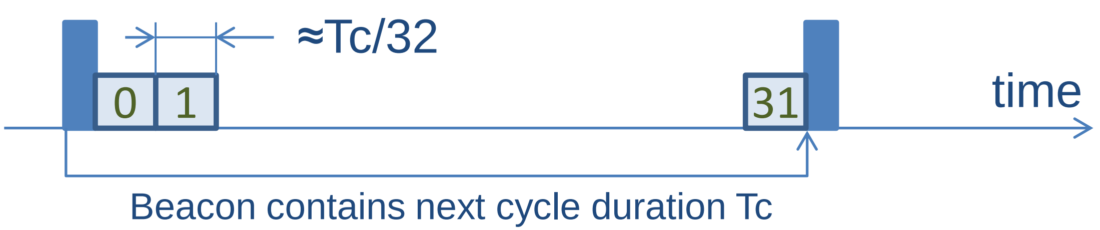


The DeseNET entity needs to keep track of the SV Applications to poll…

D_Publish.Indication() follows D_Synch:Indication()

TDMA implementation

- DeseNET features:
  - 32 time slots numbered 0… 31
  - A variable period Tc, indicated by the Gateway in each Beacon
    - Hence the time position in cycle and the slot duration can be calculated
- The slot number (“the address”) is statically configured in each sensor node

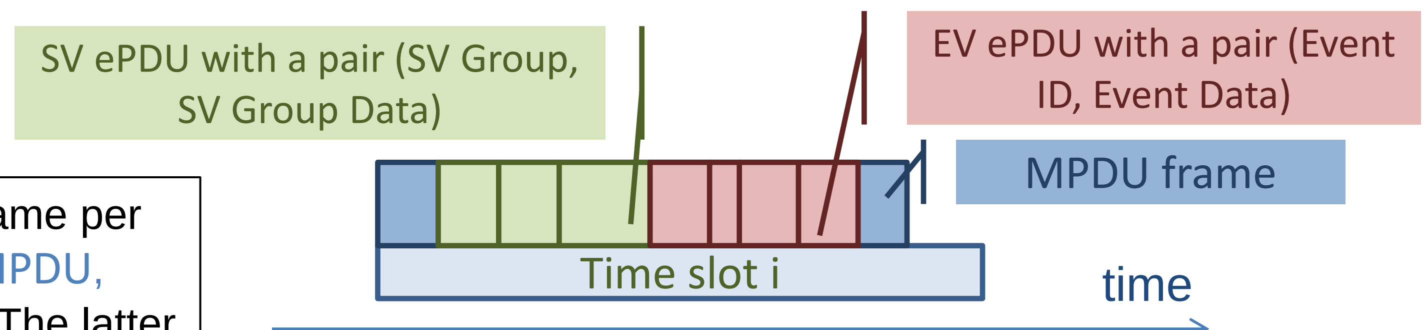


Overview of DeseNET frame format

DeseNET frame format

- Within a cycle period Tc, a sensor node can have gathered
  - Sampled values: A constant number of pairs (SV Group, SV Group Data)
    - Unless new subscriptions occurred
  - Events: A variable number of pairs (Event ID, Event Data)
- All these data have to be send in the attributed time slot
  - DeseNET chose the option to send a unique frame per time slot
    - There is one and only one receiver (the Gateway)
    - Less overhead with one frame

DeseNET frame is called MPDU (Multiple PDU)



There is one frame per time slot (the MPDU, Multiple PDU). The latter contains several ePDUs (embedded PDUs)

A simple, conservative, suboptimal algorithm is to be implemented to keep the MPDU frame duration shorter than the time slot

ePDU format

- An SV ePDU contains mainly:
  - The SV Group and the SV Group Data
- An EV ePDU contains mainly
  - The Event ID and the Event Data
- SV Groups and Event IDs must be managed at DeseNET level
  - But their management is outside the scope of the DeseNET protocol
  - For the DeseNET protocol:
    - SV Groups and Event IDs are just numbers
    - SV Group Data and Event Data are just byte arrays