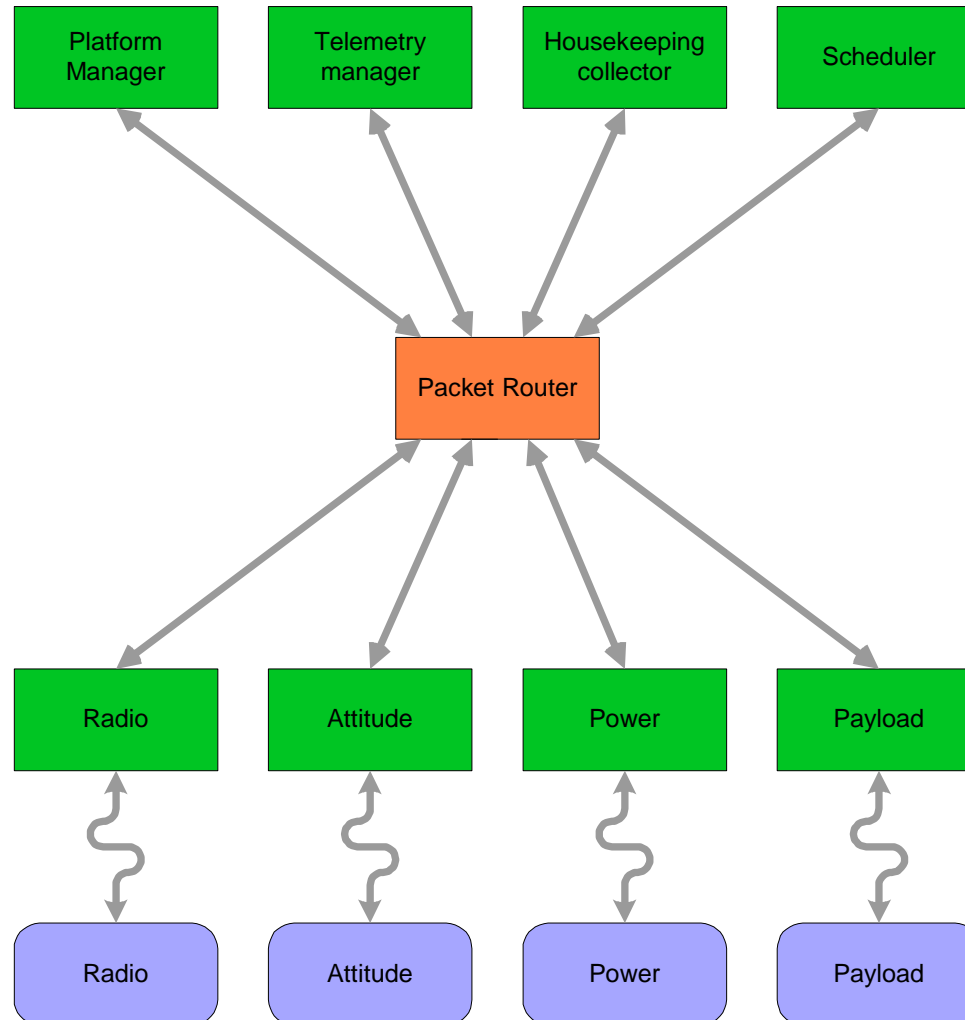
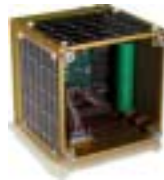


# DTUsat onboard software architecture

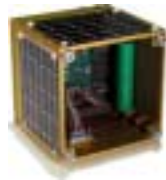




# Software elements

- One C program consisting of
  - The eCOS kernel
  - Packet router library
  - 9 software modules:
    - Platform manager
    - Scheduler
    - Housekeeping collector
    - Telemetry manager
    - Protocol (incl. Radio interface)
    - Attitude
    - Power
    - Camera
    - Tether





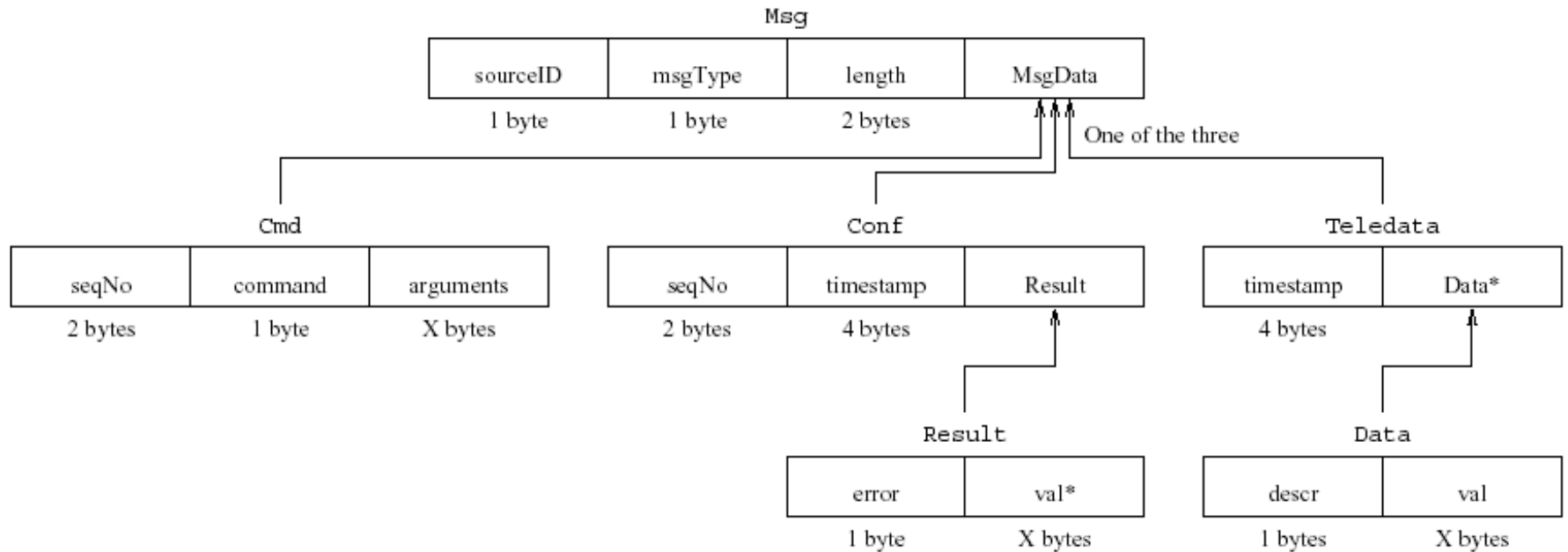
# The module concept

- What is a module?
  - The basic idea of modules is to isolate the different parts of the system in order to simplify the system development and error handling
- Properties
  - Operates autonomously
  - Communicates with the rest of the system through the Packet Router using packets and the communication facilities of the Packet Router
  - Contains several parallel threads of execution
  - All modules share the same memory space

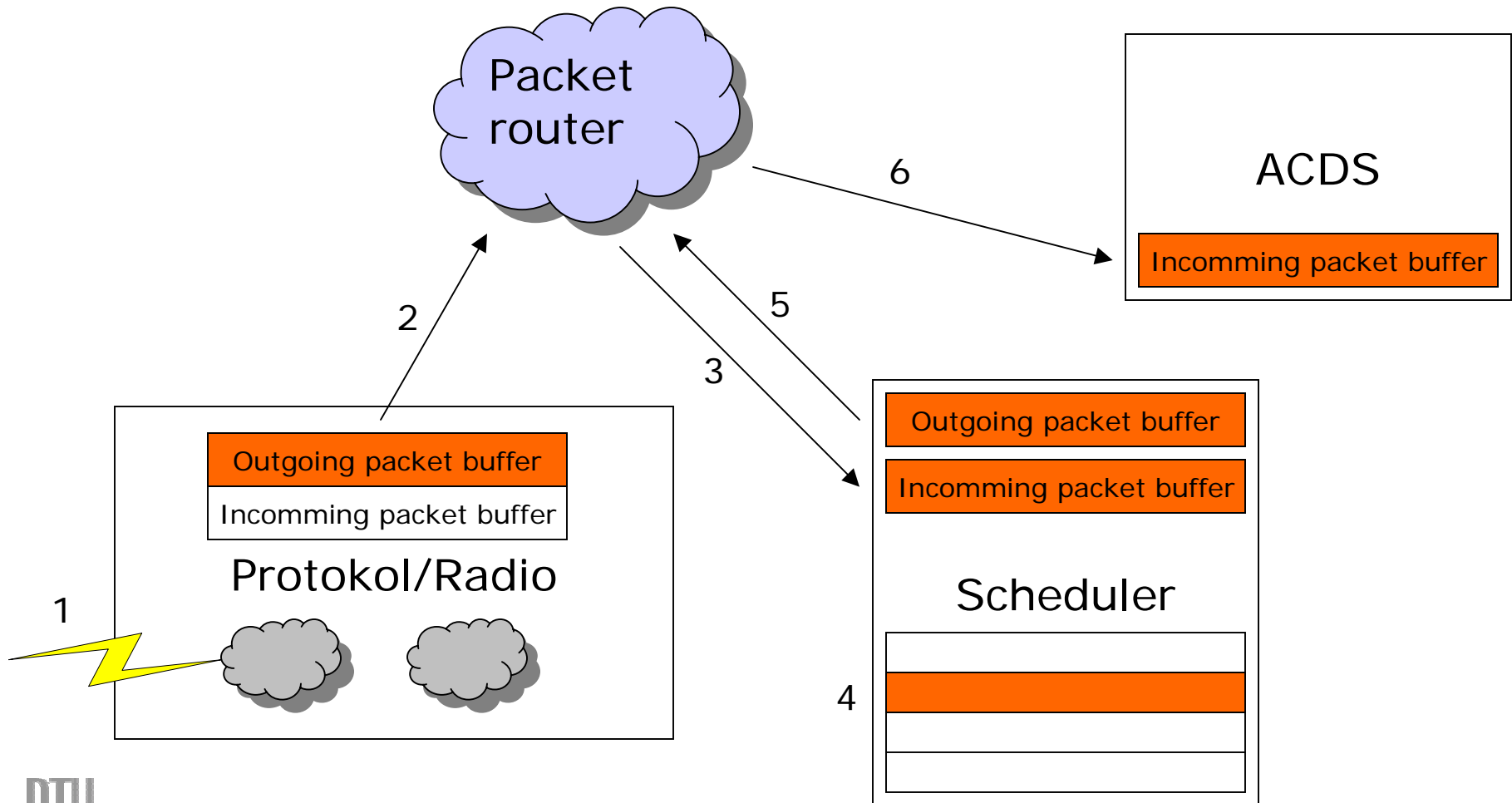
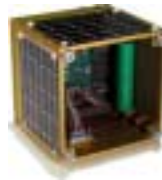




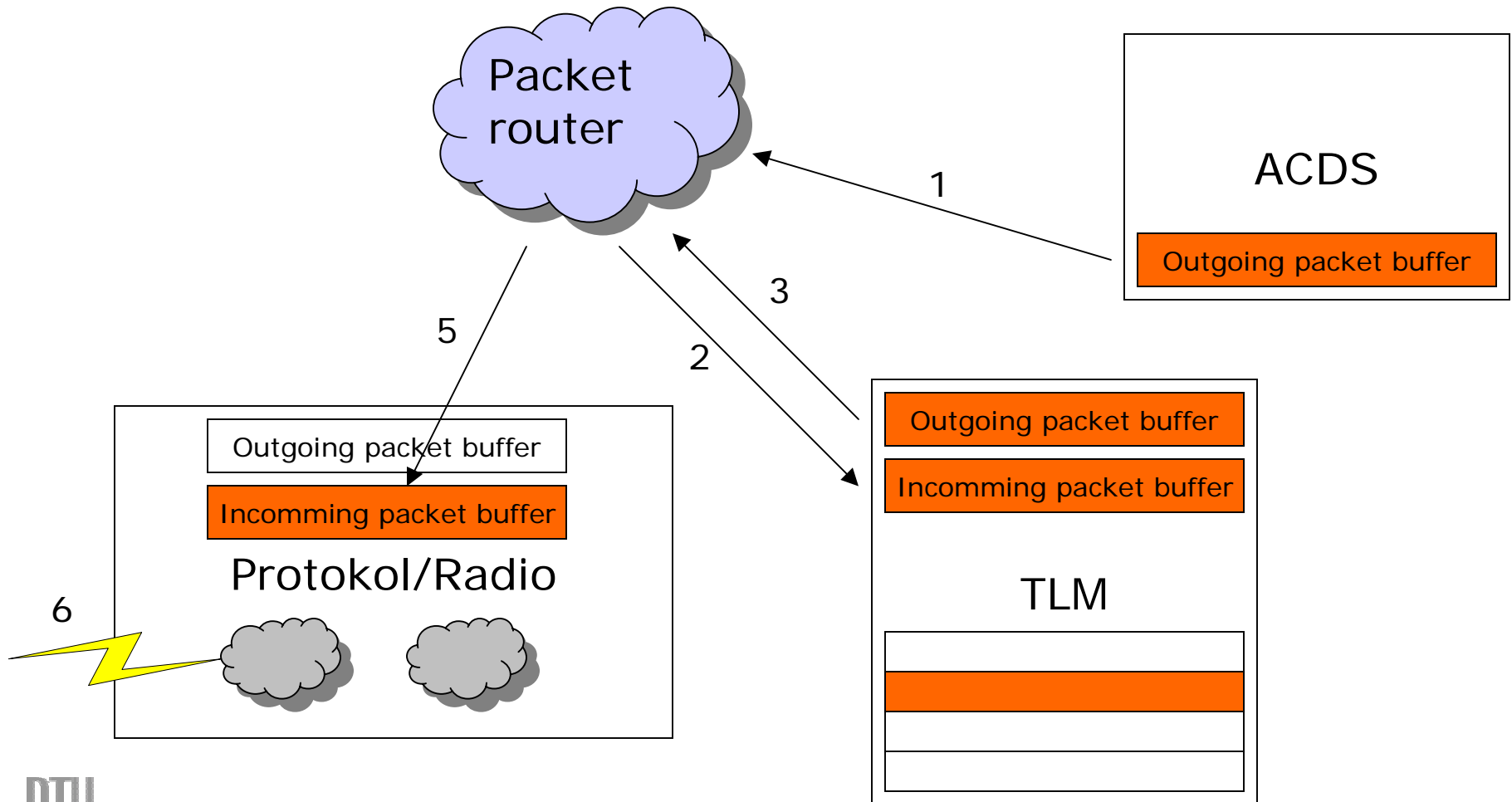
# Packetformat



# Telecommand packet flow



# Telemetry packet flow





# About Redhat eCOS

- Small memory footprint (10-100 kb)
- Open source, royalty-free, license-free
- Highly customizable
- Real-time kernel including:
  - Interrupt handling
  - Exception handling
  - Thread support
  - Rich set of synchronization primitives
  - Timers, counters, and alarms





# Developing modules

- In order to co-exist with other modules a module must comply to the rules regarding:
  - Naming conventions
  - Memory usage & allocation
  - Software watchdog
  - Receiving/sending packets
    - Telecommands
    - Telemetry
    - Inter-module communication





# Developing modules - Tools



- Module development and testing using the Linux synthetic target
- Common CVS (Concurrent Versions System) server
  - [cvs.dtusat.dtu.dk](http://cvs.dtusat.dtu.dk)
- System documentation (SDK)
  - [www.dtusat.dtu.dk](http://www.dtusat.dtu.dk) > software > architecture





# Software projects

- Camera module
  - JPEG compression of images
  - Telecommands
- Attitude module
  - Implementing Matlab model in C
  - Telecommands
- Testing the onboard software
  - Scenarios
  - Developing methods for automated testing
- . . .



# Q & A

