Enabling Multiple Controllable Radios in OMNeT++ Nodes

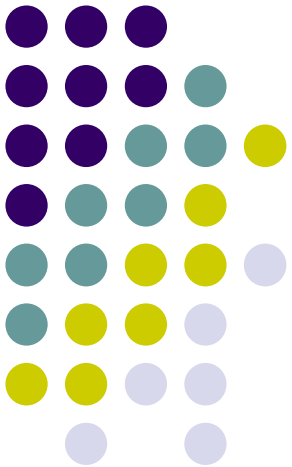
Ólafur Helgason

w. Sylvia Kouyoumdjieva and Gunnar Karlsson

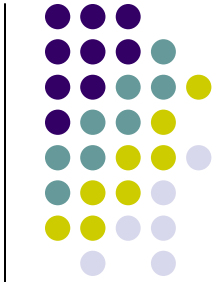
Laboratory for Communication Networks

School of Electrical Engineering

KTH - Royal Institute of Technology

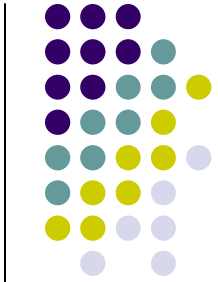


Motivation



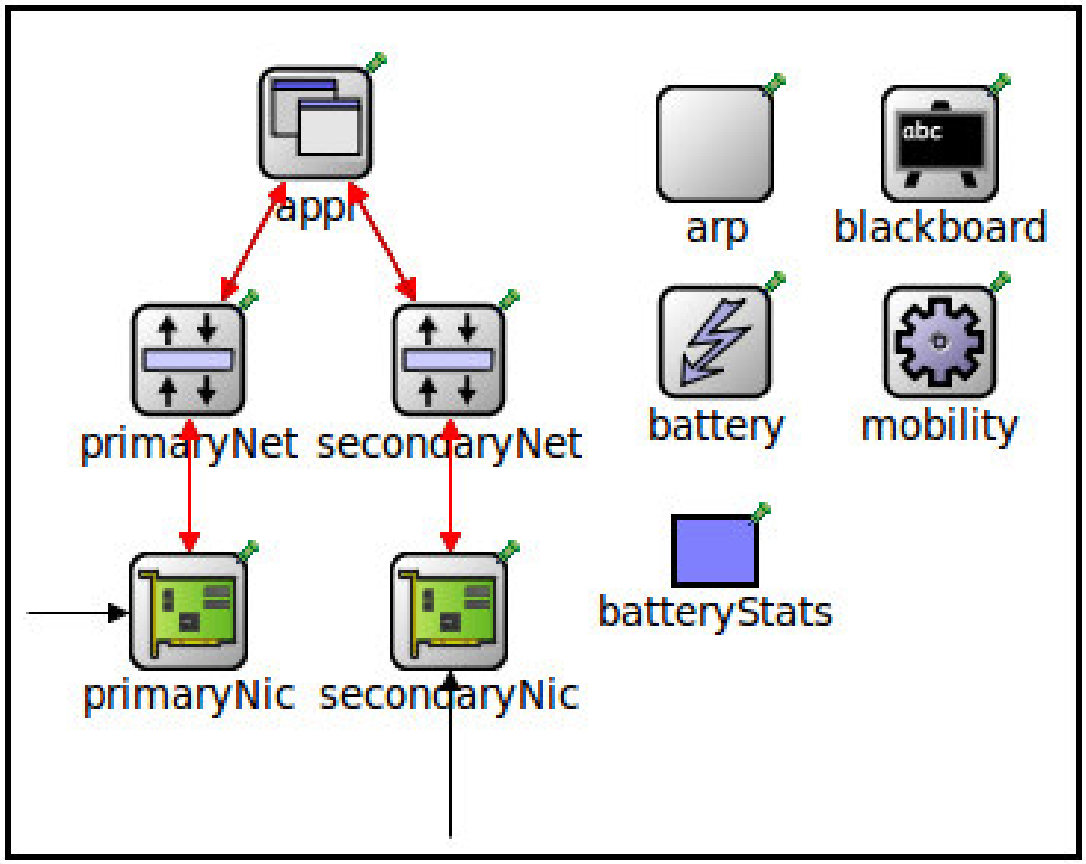
- Wireless devices commonly have multiple radios
  - Cellular, WiFi, Bluetooth, Zigbee, NFC, ...
  - Different capabilities
    - Range, rate, communication mode, discovery, energy, ...
- Dynamically exploiting radio hierarchies
  - Vertical handovers
  - Cognitive radio
  - Energy-efficiency

Energy-efficient radio subsytem



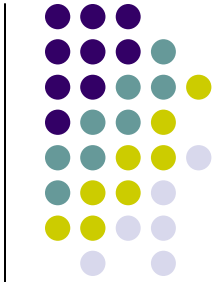
- 802.11: High energy consumption even in idle mode
- Dual controllable radios:
  - Low power, low bitrate discovery radio
  - High power, high bitrate data radio
    - HP radio suspended when idle
- Goal:
  - Enable simulation of multi-radio nodes
  - Radios should be *controllable*
  - How does it affect energy consumption
- We use *MiXiM* and the *Energy Framework*

Design overview: Host



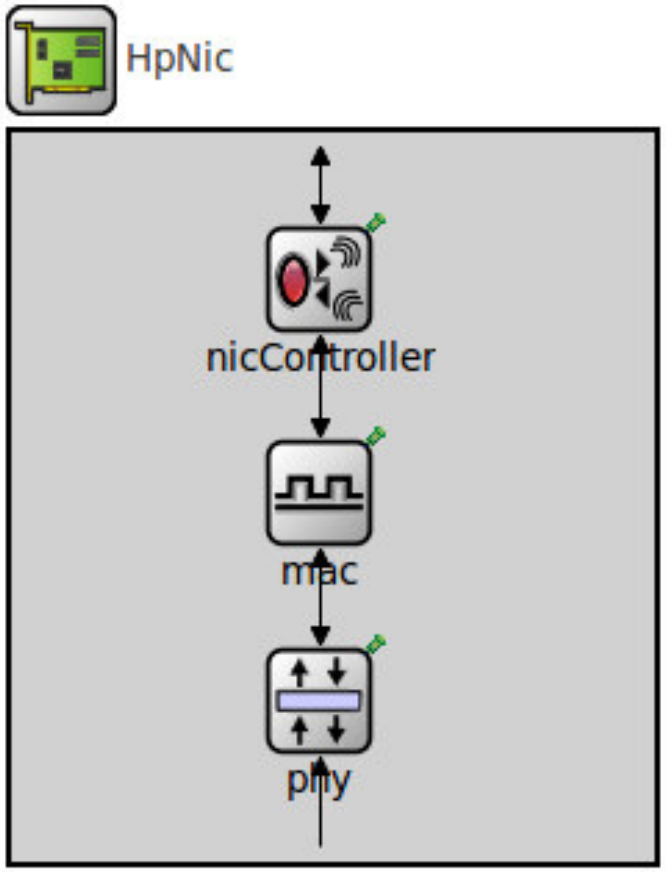
NICs draw energy from Battery

Controllable radios



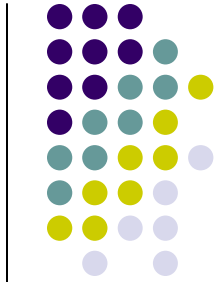
- Three states per radio
  - *ON: Full energy consumption*
  - *SLEEP: Low energy consumption, short wakeup*
  - *OFF: No energy consumption, long wakeup*
- NIC is controlled via Blackboard
- Facilitates flexibility in control
  - Application
  - Session layer

Implementation



- NicController
  - Receive ctrl commands from BB
  - Simulate wakup delay
  - Turn on/off mac & phy
  - Publish state changes on BB

IControllable interface



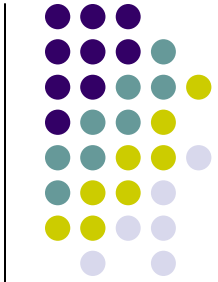
```
class IControllable {
public:
  enum Controls {TURN_ON, SLEEP, WAKE_UP, TURN_OFF};
  enum Status {TURNED_ON, SLEEPING, TURNED_OFF};

  virtual bool isOn();
  virtual bool isSleeping();
  virtual bool isOff();

protected:
  virtual bool turnOn() = 0;
  virtual bool sleep() = 0;
  virtual bool wakeUp() = 0;
  virtual bool turnOff() = 0;
};
```

- Interface implemented by NIC modules
  - Extend existing MiXiM mac & phy classes
  - Does not break any existing code

Extending PHY & MAC

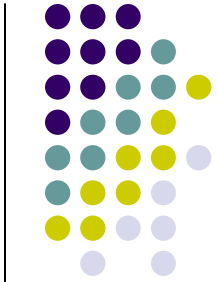


```
class PhyLayerControl
    : public PhyLayerBattery, public IControllable
{
public:
  virtual void initialize(int stage);
  virtual void finish();
  virtual void receiveBBItem(int category, const
       BBItem *details, int scopeModuleId);

protected:
  virtual void handleUpperCtrlMessage(cMessage* msg);

  virtual bool turnOn();
  virtual bool turnOff();
  virtual bool sleep();
  virtual bool wakeUp();
};
```

Conclusion



- Dual radio for opportunistic networking
  - MiXiM 802.15.4 for control radio
  - MiXiM 802.11 for data radio
- Evaluate content distribution performance
  - Energy savings vs performance decrease
  - Effect of range disparency in control & data radios
  - Effect of neighbor discovery delay
  - See paper for prel. results on a simplified system
- Generic framework applicable to different mobile wireless services/applications
- Our MiXiM fork available at

  `https://github.com/olafur/mixim`