Changes made to the IEEE 802.15.4 NS-2 Implementation

Iyappan Ramachandran

February 7, 2006

In the following the changes made to the NS-2 code (release 2.28) are described along with the reasons for the modifications and list of files affected.

Changes made

- A macro called SHUTDOWN has been defined in ./wpan/p802_15_4def.h that provides the capability to shut a node down when it does not have any packet to transmit.
- Two functions Phy802_15_4::wakeupNode() and Phy802_15_4::putNodeToSleep() have been added that can be called to shutdown and wake up the radio. These functions primarily serve to decrement the correct amount of energy consumed in sleep state.
  File affected: ./wpan/p802_15_4phy.cc, ./wpan/p802_15_4phy.h
- Added a new timer called macWakeupTimer to serve as an alarm clock for the node to wake itself up. The timer on expiry calls Phy802_15_4::wakeupNode().
  Files changed: ./wpan/p802_15_4mac.cc, ./wpan/p802_15_4mac.h, ./wpan/p802_15_4timer.cc, ./wpan/p802_15_4timer.h, ./wpan/p802_15_4csmaca.h
- Added variables P_sleep_ (sleep state power consumption) and channel_sleep_time (the last time the node was put to sleep) as members of WirelessPhy.
  Files affected: mac/wireless-phy.h
- The radio if asleep should be woken up when MAC receives a packet to transmit. Similarly, a sleeping radio needs to be woken up to receive beacons whenever they are expected to arrive. If radio shutdown is activated, the radio needs to be put to sleep after transmission of a packet. Mac802_15_4::recv() does this by calling functions Phy802_15_4::wakeupNode() and Phy802_15_4::putNodeToSleep(), which decrement energy spent sleeping.

Files affected: ./wpan/p802_15_4mac.cc

- After backing off *macMaxCSMABackoffs* and being unable to transmit a packet, the MAC has to report a channel access failure. Zheng's implementation keeps attempting to transmit the packet indefinitely, instead of reporting channel access failure. This has been fixed in the Mac802_15_4::mcps_data_request() function. Also the node is put to sleep at this stage.
  Files affected: ./wpan/p802_15_4mac.cc

- After every beacon reception, the node can shut itself down if it doesn't have a packet pending to be transmitted when radio shutdown has been activated. This is done by Mac802_15_4::recvBeacon() by calling Phy802_15_4::putNodeToSleep().
  Care needs to be taken to ensure that the node doesn't go to sleep if it waiting for an ack.
  Files affected: ./wpan/p802_15_4mac.cc

- If the node is being put to sleep when not in use, the sleep-to-idle transition needs to be accounted for. This is done in CsmaCA802_15_4::start(). The number of slots to backoff at the first backoff stage is calculated as MAX(U(0,7),3), since sleep-to-idle ramp-up time is 3 backoff slots (hard coded).
  Files affected: ./wpan/p802_15_4csmaca.cc

- A new constant called aCCATime has been added, which indicates the CCA duration in symbol periods.
  Files affected: ./wpan/p802_15_4const.h

- CCA duration has been specified to be 8 symbol durations. In Zheng's implementation, CCA is done right at the end of the 8th symbol duration to determine channel idleness. As a result, if the channel is busy for the first 8 symbol durations and goes idle after that (which is likely), the implementation CCA would indicate idle while in reality it shouldn't. This has been fixed by doing the CCA at the end of the 1st symbol duration, but reporting channel idleness at the 8th. For this purpose, a new timer CSH has been added which on expiry calls CSHandler that does the CCA earlier done by CCAHandler. The CCAHandler functionality has been reduced to merely reporting the channel idleness.
  Files affected: ./wpan/p802_15_4phy.cc, ./wpan/p802_15_4phy.h

- The Phy802_15_4::PD_DATA_indication() function calls WirelessChannel::sendUp() to check if the packet has been received correctly and to decrement the energy consumed in the packet reception. The SendUp() function is already being called by recv() and calling it a second time causes energy to be decremented twice. This bug has been fixed in Phy802_15_4::PD_DATA_indication().
  Files affected: ./wpan/p802_15_4phy.cc

- Phy802_15_4::recv() function that receives packets from the channel checks if the packet has been received correctly using WirelessPhy::sendUp(), failing which the packet is freed. sendUp() returns a 0 either when the node is asleep or when the packets received power is less than the CS threshold. In the former case, the variables rxTotPower and rxTotNum need to be updated for CS purposes before dropping the packet, while in the latter case the packet simply needs to be dropped. Zheng's implementation was dropping all packets without updating the variables. This has been fixed in Phy802_15_4::recv().
Files affected: ./wpan/p802_15_4phy.cc

- The receiver has to be turned on for the carrier sensing operation and therefore receive power P_r is consumed during this period. Zheng's implementation does not decrement receive energy due to carrier sensing. This has been added in function Phy802_15_4::CarrierSenser(). Also, energy is spent during the tx-rx turnaround. This has been accounted for as well.
Files affected: ./wpan/p802_15_4phy.cc

- Information about the power consumed during the idle-receive and idle-transmit transitions are not found on data sheets. It is reasonable to assume that the power consumed during these transitions are the same those consumed during receive and transmit states respectively. Code has been changed in all energy decrements to take care of this.
Files affected: ./mac/wireless-phy.cc

- In the WirelessPhy::SendUp() function, receive energy gets decremented regardless of whether whether the radio's receiver is enabled or not. As a quick fix, a check had been added in this function to first determine if the packet is intended for this node using the function p802__4macDA(p) and then decrement either receive or idle energy. This patch renders the WirelessPhy virtually unusable with other MAC. A better way to handle this needs to be incorporated.
Files affected: ./mac/wireless-phy.cc

- A new function called DecrSleepEnergy() has been added as a member of EnergyModel class.
Files affected: ./mobile/energy-model.cc, ./mobile/energy-model.h

- We are interested in determining the performance when the packet arrivals follow a Poisson rate. We do this by attaching an application generating packets at exponential intervals. The presence of an interface queue however disturbs this Poisson arrival. This problem is circumvented by modifying the Queue::recv() code so that the queue doesn't accept any packet if it is blocked. MAC blocks the queue when a packet is dequeued and unblocks it either when the packet has been successfully transmitted or when a failure has been declared.
Files affected: ./queue/queue.cc

- In order to be able to set sleep power from the tcl interface, code has been added to create-wireless-node procedure in ns-lib.tcl and a new procedure added to ns-mobilenode.tcl
  Files affected: ./tcl/lib/ns-lib.tcl, ./tcl/lib/ns-mobilenode.tcl