ECS: An enhanced carrier sensing mechanism (In NS2)

By simply allowing wireless nodes to transmit at different power levels under the DCF of IEEE 802.11, the number of hidden terminals is likely to increase, which, in turn, results in more collisions and retransmissions due to the very nature of DCF's contention-based access mechanism, and hence, more energy is consumed eventually.

The main problem is that **the hidden terminals detect a transmission but cannot decode it, they set their NAVs for the EIFS duration and then may not sense any signal during the DATA transmission** for the data packet is not transmitted using the maximum transmission power level.

Jung and Vaidya proposed a **Power Control MAC** which ***periodically transmits with the maximum transmission power during the transmission of DATA frame*** so that the throughput would not be degraded. A better solution is to ***exploit transmission power control for spatial reuse of wireless channel***. Developing efficient medium access protocols that optimize spectral reuse, increase channel efficiency and hence maximize aggregate channel utilization and achievable throughput is another important research issue in wireless networks.

Here we incorporate an "ECS: An enhanced carrier sensing mechanism for wireless ad hoc networks, Zhifei Lia, Sukumar Nandib, Anil K. Gupta". It was proposed to address the problem that whenever a node detects an erroneous frame due to the wireless transmission errors or the fact that carrier sensing range is normally greater than the transmission range, how long the node should defer for.

ECS improves the fairness as well as the throughput substantially ***by distinguishing the type of erroneous frames based on their lengths*** and ***deferring nodes' transmissions accordingly***. We found that this scheme can be considerably helpful for alleviating the severe hidden/exposed terminal problem which results from using transmission power control to achieve the goal of energy saving. The future work will include utilizing transmission power control to optimize channel reuse and completely solve the hidden/exposed terminal problem.

1. First, we need to define the **Max_Data Length** so that we can calculate how long to defer for.

In mac-802_11.cc

```
MAC_MIB::MAC_MIB(Mac802_11 *parent)
{
   ....
   ....
   //add below lines
   #ifdef ECS
         MaxDataLen = 512;     //[HP] if forget to assign the value, default will be used (512 Byte)
         parent->bind("MaxDataLen_", &MaxDataLen);    //[HP]
   #endif
}
```

2. Secondly, when you receive an error packet, you should decide which type of the packets (orts, ccts, data, ack) you are receiving.

```
void Mac802_11::recv(Packet *p, Handler *h)
{
   ....
```

```
   ....
   #ifdef ECS
      //[HP]implement Enhanced Carrier Sensing
      //using the time length to distinguish different types of SR(sensing range) frames
      //and adopt different values of EIFS accordingly
      if( hdr->error() && (rx_state_ == MAC_IDLE) ) {
          sr_frame_ = SR_UNKNOWN;
          if( txtime(p) == txtime(phymib_.getORTSlen(),basicRate_) ) {sr_frame_ = SR_ORTS;}
          if( txtime(p) == txtime(phymib_.getCCTSlen(),basicRate_) ) {sr_frame_ = SR_CCTS;}
          if( txtime(p) == txtime(phymib_.getACKlen(),basicRate_) ) {sr_frame_ = SR_ACK;}
      }
      if ( hdr->error() && (rx_state_ == MAC_RECV) && (pktRx_->txinfo_.RxPr / p->txinfo_.RxPr <
p->txinfo_.CPThresh) ){
          sr_frame_ = SR_UNKNOWN;     //reception encounters collisions and thus the time length will be
extended. We cannot figure out which type of SR it is.
      }
   #endif
   ....
   ....
}

3. Then you can defer each node's transmission accordingly.
void Mac802_11::recv_timer()
{
   ....
   ....
   if( ch->error() ) {
     Packet::free(pktRx_);
     #ifdef ECS
        switch(sr_frame_) {
            case SR_ORTS:
                set_nav(usec(phymib_.getDIFS() + txtime(phymib_.getCCTSlen(), basicRate_) + 2 *
phymib_.getSIFS() +
                txtime((macmib_.getMaxDataLen() + phymib_.getHdrLen11()),dataRate_) +
txtime(phymib_.getACKlen(), basicRate_)));
                break;
            case SR_CCTS:
                set_nav(usec(2 * phymib_.getSIFS() + txtime((macmib_.getMaxDataLen() +
phymib_.getHdrLen11()),dataRate_) +
                txtime(phymib_.getACKlen(), basicRate_)));
                break;
            case SR_UNKNOWN:
                set_nav(usec(phymib_.getEIFS()));
                break;
        }
        goto done;
    #else
        set_nav(usec(phymib_.getEIFS()));
        goto done;
    #endif
```

```
  }
  ....
  ....
}
```

Note: Using the method above cannot guarantee that all erroneous control packets could be distinguished because the time length will be extended in collisions. The paper does not provide solutions, but it indeed can alleviate the hidden terminal problem to some extent.

RETURN