

## Improvements on Mac802\_11 simulation when link loss ratio or contention is considered (In NS2)

In Mac802\_11 simulations, packets are assumed to be received correctly within transmission range, hence a RTS will definitely get a CTS response. However, when you use shadowing propagation model, or consider link loss ratio in opportunistic routing, or consider using competition CTS, a sender may lose CTS due to channel contention or dynamic channel condition and a RTS will be retransmitted. In this case, the specified receiver should not refuse to respond to it; otherwise any algorithm which needs RTS/CTS negotiation may compromise its performance a lot.

Therefore, following lines need to be modified in mac/mac-802\_11.cc

```
/* [PH] following lines will be replaced =====
if(tx_state_ != MAC_IDLE) {
    discard(p, DROP_MAC_BUSY);
    return;
}
*/

/*
* If I'm responding to someone else, discard this RTS.
*/
/* if(pktCTRL_) {
    discard(p, DROP_MAC_BUSY);
    return;
}
*/
//=====replace by=====

// [PH] this node may receive a re-transmitting RTS
// from its sender who didn't receive the previous CTS
// due to collision or dynamic channel condition
// thus it should cancel the waiting timer for data
// and retransmit CTS
// However, if pktCTRL_ != CTS then I should discard the received RTS
if(tx_state_ != MAC_IDLE) {
    if(tx_state_ == MAC_CTS) {
```

```
hdr_mac802_11 *mh = HDR_MAC802_11(pktCTRL_);
if( ETHER_ADDR(mh->dh_ra) == ETHER_ADDR(rf->rf_ta) ) {
    if(mhSend_.busy()) { mhSend_.stop();}
    setTxState(MAC_IDLE);
    Packet::free(pktCTRL_);
    pktCTRL_ = 0;
} else {
    discard(p, DROP_MAC_BUSY); //received a new RTS
    return;
}
} else {
    discard(p, DROP_MAC_BUSY); //I'm not waiting for data. I'm busy in sending RTS/DATA/ACK, cannot handle another connection
    request.
    return;
}
}
```

[RETURN](#)