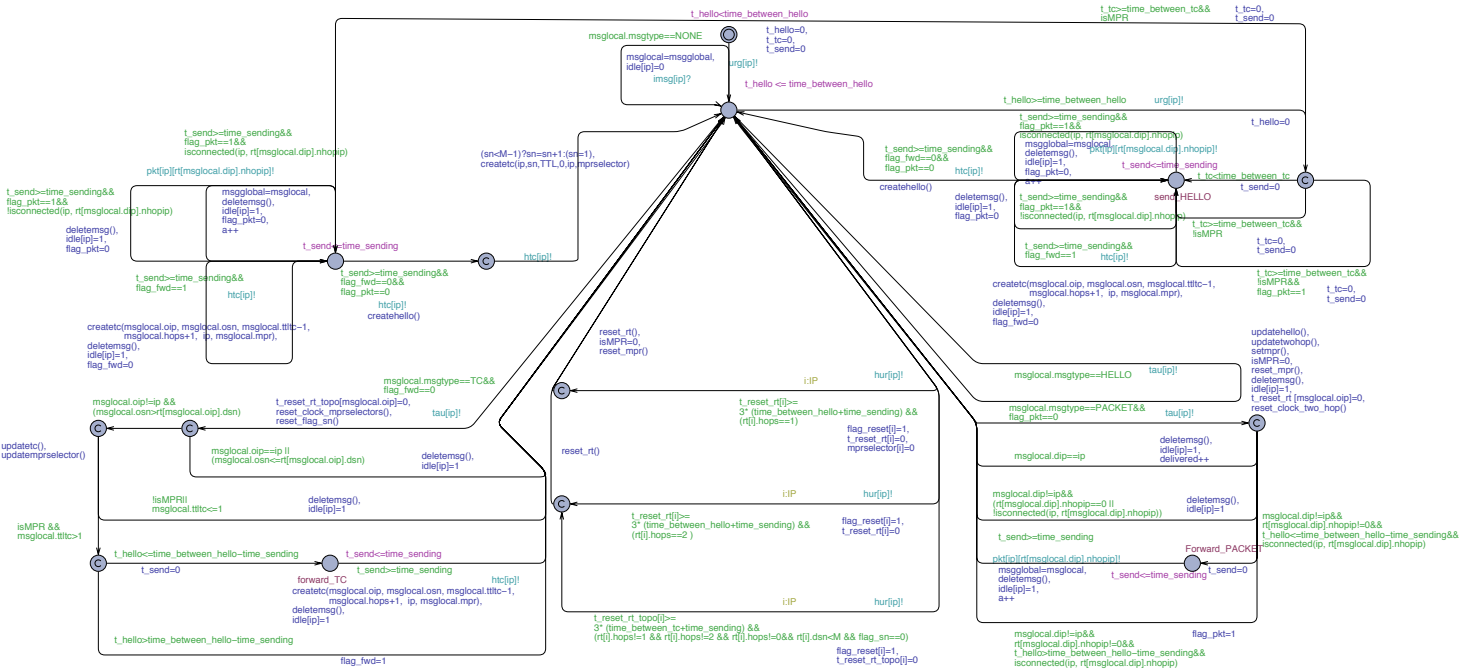
msglocal.msgtype==NONE
msglocal=msgglobal,
idle[ip]=0
imsg[ip]?
urg[ip]!
t_hello=0,
t_tc=0,
t_send=0
t_hello<time_between_hello
t_hello <= time_between_hello
t_hello>=time_between_hello
urg[ip]!
t_tc>=time_between_tc&&
isMPR
t_tc=0,
t_send=0
t_hello=0
t_send>=time_sending&&
flag_pkt==1&&
isconnected(ip, rt[msglocal.dip].nhopip)
msgglobal=msglocal,
deletemsg(),
idle[ip]=1,
flag_pkt=0,
a++
pkt[ip][rt[msglocal.dip].nhopip]!
t_send<=time_sending
t_tc<time_between_tc
t_send=0
t_send>=time_sending&&
flag_pkt==1&&
isconnected(ip, rt[msglocal.dip].nhopip)
sendHELLO
t_tc>=time_between_tc&&
isMPR
t_tc=0,
t_send=0
t_send>=time_sending&&
flag_fwd==1
htc[ip]!
t_tc>=time_between_tc&&
isMPR&&
flag_pkt==1
t_tc=0,
t_send=0
t_send>=time_sending&&
flag_fwd==0&&
flag_pkt==0
htc[ip]!
createhello()
deletemsg(),
idle[ip]=1,
flag_pkt=0
t_send>=time_sending&&
flag_pkt==1&&
isconnected(ip, rt[msglocal.dip].nhopip)
msgglobal=msglocal,
deletemsg(),
idle[ip]=1,
flag_pkt=0,
a++
pkt[ip][rt[msglocal.dip].nhopip]!
t_send>=time_sending&&
flag_pkt==1&&
isconnected(ip, rt[msglocal.dip].nhopip)
deletemsg(),
idle[ip]=1,
flag_pkt=0
t_send>=time_sending
htc[ip]!
(sn<M−1)?sn=sn+1:(sn=1),
createtc(ip,sn,TTL,0,ip,mprselector)
t_send>=time_sending&&
flag_fwd==1
htc[ip]!
t_send>=time_sending&&
flag_fwd==0&&
flag_pkt==0
htc[ip]!
createhello()
deletemsg(),
idle[ip]=1,
flag_fwd=0
reset_rt(),
isMPR=0,
reset_mpr()
msglocal.msgtype==HELLO
tau[ip]!
updatehello(),
updatetwohop(),
setmpr(),
isMPR=0,
reset_mpr(),
deletemsg(),
idle[ip]=1,
t_reset_rt [msglocal.oip]=0,
reset_clock_two_hop()
msglocal.msgtype==PACKET&&
flag_pkt==0
tau[ip]!
deletemsg(),
idle[ip]=1,
delivered++
msglocal.dip==ip
msglocal.dip!=ip&&
(rt[msglocal.dip].nhopip==0 ||
isconnected(ip, rt[msglocal.dip].nhopip))
deletemsg(),
idle[ip]=1
msglocal.dip!=ip&&
rt[msglocal.dip].nhopip!=0&&
t_hello<=time_between_hello−time_sending&&
isconnected(ip, rt[msglocal.dip].nhopip)
t_send>=time_sending
pkt[ip][rt[msglocal.dip].nhopip]!
Forward_PACKET
msgglobal=msglocal,
deletemsg(),
idle[ip]=1,
a++
t_send<=time_sending
t_send=0
flag_pkt=1
msglocal.dip!=ip&&
rt[msglocal.dip].nhopip!=0&&
t_hello<time_between_hello−time_sending&&
isconnected(ip, rt[msglocal.dip].nhopip)
msglocal.msgtype==TC&&
flag_fwd==0
t_reset_rt_topo[msglocal.oip]=0,
reset_clock_mprselectors(),
reset_flag_sn()
tau[ip]!
msglocal.oip!=ip &&
(msglocal.osn>rt[msglocal.oip].dsn)
msglocal.oip==ip ||
(msglocal.osn<=rt[msglocal.oip].dsn)
deletemsg(),
idle[ip]=1
updatetc(),
updatemprselector()
isMPR &&
msglocal.ttltc>1
lisMPRII
msglocal.ttltc<=1
deletemsg(),
idle[ip]=1
t_hello<=time_between_hello−time_sending
t_send=0
t_send<=time_sending
t_send>=time_sending
forward_TC
htc[ip]!
createtc(msglocal.oip, msglocal.osn, msglocal.ttltc−1,
msglocal.hops+1, ip, msglocal.mpr),
deletemsg(),
idle[ip]=1
t_hello>time_between_hello−time_sending
flag_fwd=1
i:IP
hur[ip]!
t_reset_rt[i]>=
3* (time_between_hello+time_sending) &&
(rt[i].hops==1)
flag_reset[i]=1,
t_reset_rt[i]=0,
mprselector[i]=0
reset_rt()
i:IP
hur[ip]!
t_reset_rt[i]>=
3* (time_between_hello+time_sending) &&
(rt[i].hops==2)
flag_reset[i]=1,
t_reset_rt[i]=0
i:IP
hur[ip]!
t_reset_rt_topo[i]>=
3* (time_between_tc+time_sending) &&
(rt[i].hops!=1 && rt[i].hops!=2 && rt[i].hops!=0&& rt[i].dsn<M && flag_sn==0)
flag_reset[i]=1,
t_reset_rt_topo[i]=0