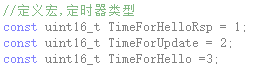
这几个常量对应定时器事件，

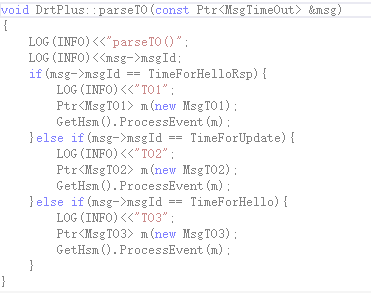


每当定时器事件触发，就发出MsgTOx事件：

MsgTO1：只有在Idle状态才会处理这个事件，需要发HelloRsp，不需要转移状态。

MsgTO2：这个事件发送说明要发更新的路由包。

MsgTO3：每当有这个事件，说明需要sendHello，并且转移到状态WaitTopo。（只有两次机会，一次是启动后十分钟，下一次是4小时后。）



路由发现大致过程：

前10 s：处于init状态。此时等待定时器事件发生（发hello包）。（定时于10s后发包）。

10 s – 42 s：处于WaitTopo状态。等待定时器事件发生（发update包）。（定时于32s后发包）此时如果收到hello包延迟处理，如果收到Update包立即更新。

42 s后：一般情况处于Idle状态。

如果在WaitTopo收到了hello包，或者当前状态收到了，则设置Rsp定时器，到时间再发Rsp。

如果收到了Update包，先更新路由包，然后设置定时器，转到WaitTopo状态。发完Update包之后回到Idle状态。

1. 事实上，处于Idle状态已经可以开始发数据包了。但是如果收到Update包，更新了路由表，则自身需要定时发送Update包，发完了才能回到Idle状态。

2. 收到hello包时会回Rsp，但不发Update包。收到Update包时回Update包。

3. WaitTopo状态，当有hello的Rsp或者Update包的时候，就更新路由表。这个状态下一直在收集路由信息，等待发送Update包。只要发完了Update包就返回Idle状态。

4. Rsp包在Idle状态发出的，不需要转移状态。Update包是在WaitTopo状态发出的，转移到Idle状态。

小结：

1. 只有该节点发出了Hello包或者受到Update包之后，才会进入WaitTopo状态。对于发了Hello包，有期望受到Rsp，从而更新路由表。

对于受到Update包，需要告诉其他节点，路由表已经更新。

2. 当在Idle状态收到了Hello包，不需要转移状态，只发送Rsp即可。可能是考虑到：

第一，对于收到Hello包的节点而言，发送Hello包的节点，发完了以后一定会发送Update包，可以等到收到该节点的Update包时，更新自己的路由包之后，顺便发送Update包。

第二，对于发送完Hello包的节点，发完Hello包后，相邻节点会发送Rsp，可以趁这个时间收集路由信息。在这个时间最好不要发数据包，这样可以更好的更新路由表。

因此，每次发完Hello包，都要进入WaitTopo状态，等到发送Update包。

3. 当Idle状态收到了Update包，需要转移到WaitTopo状态。可能是考虑到：其他节点也会收到Update包，这些节点有可能会发Update包（因为更新了路由表），可以趁它们发Update包的时候收集路由信息。

4. 总的来说，对于刚启动的节点，先发送hello包，然后进入WaitTopo状态收集路由信息（等Rsp）。等收集到一定路由后，再发送Update。对于其他节点，收到这个Update包，如果需要有更新路由，就发出去告诉其他节点。