Tictoc2：换图标换颜色。

Tictoc3：在C++文件中定义变量counter，初始化函数中同时初始化counter数值。

Tictoc4：在ned中定义变量和初始值，C++文件中调用该值需要用par()函数，ini文件中对值的初始化可以覆盖ned中的初始化。

Tictoc5：创建两个简单的模型类型Tic和Toc，导出的模型仅仅是修改了参数值，并添加了一些显示属性。代码上主要区别体现在搭建网络时，Tic和Toc可以直接当作模型类型使用。

Tictoc6：节点从接收到数据到发送出数据的延迟在C++文件中实现，时间到了给自己发送event，收到event后发送tictocMsg。

Tictoc7：引入随机数。将延迟从1s更改为一个随机值，该值可以从NED文件或omnetpp.ini中设置。声明模块参数语句，volatile告诉编译器该变量可能会在意料之外的时间被修改，因此编译器不应该对该变量进行优化，@unit(s) 是一个单位注解，用于说明变量的单位。模块参数能够返回随机变量;然而，为了利用这个特性，必须在每次使用handleMessage()时读取参数。在handleMessage()函数中调用的uniform()是随机数生成函数，模拟丢包情况。用par()函数调取模块参数中的随机变量，随机变量的值在ini文件中随机生成。

Tictoc8：丢包程序在C++中，与7一样。重发机制：一个计时变量用于决定多久之后重发，重发的是新创建的数据包，一个消息指针，定时给自己发送TimeoutEvent，除非自己收到的不是TimeoutEvent，此时发送下一条数据。

Tictoc9：保留原数据包，只发送副本，确认收到后删除原件。

Tictoc10：ned文件中模型需要多输入多输出，用列表、++，每个节点收到消息随机转发到某个出口，除非该节点是目标节点。

Tictoc11：定义连接类型简化代码，types。

Tictoc12：定义双向连接，< - - >，C++文件中，out变为gate$o，in变为gate$i。

Tictoc13：创建cMessage的子类，将目的地添加为数据成员。调用时include头文件，setSource()、steDestination()，在handleMessage()的参数中，以cMessage\*指针的形式获得消息。但是，如果将msg强制转换为TicTocMsg13\*，则只能访问其在TicTocMsg13中定义的字段。check\_and\_cast<TicTocMsg13 \*>转换指针。

Tictoc14：展示接收/发送包数量，在C++文件中，class里添加private变量记录，watch()查看。还可以将此信息显示在模块图标上方。t= display string标签指定文本;只需要在运行时修改显示字符串。

Ticotc15：每条message跳数统计以及延伸的简单计算，在class中添加cHistogram和cOutVector类型变量，前者计算平均值，后者将数据记录在Tictoc15-#0.vec中。当message到达目的地之后，读取message中自创变量hopCount的值，分别调用hopCountVector.record()和hopCountStats.collect()进行记录。hopCountVector.record()调用将数据写入Tictoc15-#0.vec。具有较大的仿真模型或较长的执行时间，Tictoc15-#0。Vec文件可能会变得非常大。要处理这种情况，可以在omnetpp.ini中特别禁用/启用vector，还可以指定感兴趣的模拟时间间隔(在此间隔之外记录的数据将被丢弃)。当开始一个新的模拟，现有的Tictoc15-#0。Vec /sca文件被删除。标量数据(在此模拟中由直方图对象收集)必须在finish()函数中手动记录。在模拟成功完成时调用Finish()，也就是说，当它因错误而停止时不会调用。代码中的recordScalar()调用写入Tictoc15-#0。sca文件。

Tictoc16：不需要预见到未来需要观察什么变量，可以随意地获取统计信息。C++文件class内private变量只需要一个simsignal\_t类型变量arrivalSignal，需要提前在initalize()中注册信号，arrivalSignal = **registerSignal**("arrival")。现在，当接收到数据包就可以发出信号emit()。必须在NED文件中定义发出的信号。在NED文件中声明信号允许您在一个地方拥有关于模块的所有信息。您将看到它接受的参数、它的输入和输出门，以及它提供的信号和统计信息。具体信号的内容调整在ini文件内。

Tictoc17：添加静态或动态图像、文本。在ned文件中的network中设置图形的类型、位置等参数，此时是静态图形，动态变化需要通过修改C++文件中的handleMessage()实现。获取指向画布的指针，进一步获取指向图像的指针（）根据ned文件中设定的图形名称确定指向哪个图像，最后setText()更新图像内容。