ALOHA 模型构建与仿真

1. 实验原理

设计 ALOHA 网络模型将使用一个发射机节点传送数据包，并且使用一个接收器节点模型执行网络监测。模型结构如图 1 所示。

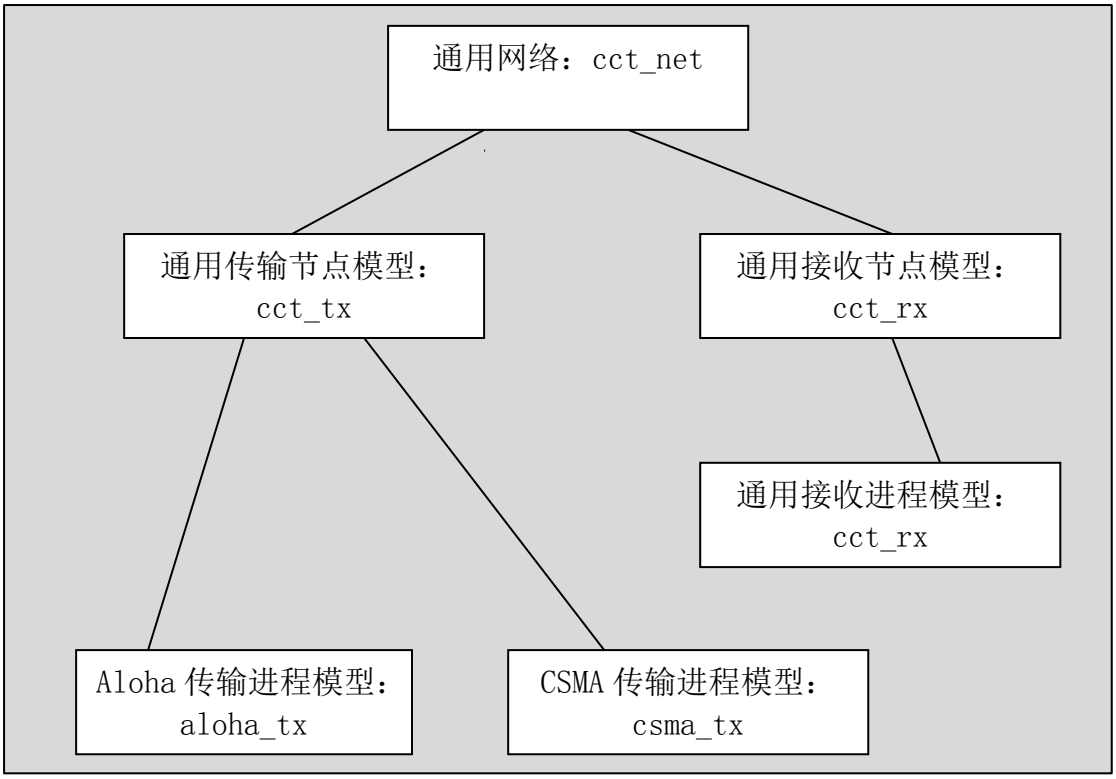

图 1

理论上，ALOHA 系统可以模拟成一个简单的资源发生器和一个总线发射器。我们在此将设计更多通用的模型，在后面设计 CSMA 时便可以在此基础上轻松实现。

2. 实验原理步骤

2.1. 创建发射节点的进程模型

1）打开 OPNET.

2）选择 File ＞ New... 然后在下拉菜单中选择 Process Model ，单击 OK。

3）使用 Create State 工具按钮，在编辑窗口放置三个状态.

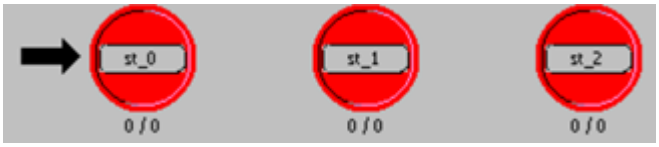

图 2-1

4）接着对这三个状态做如下设置

(a)将第一个状态的 name 属性改为 init ，将其 status 属性改为 forced.
(b）对于第二个状态，将其 name 属性改为 idle.
(c）对于第三个状态将其 name 属性改为 tx_pkt，将其 status 属性改为 forced

图 2-2

5）在状态之间添加转移线

（a）按照下图所示连接转移线

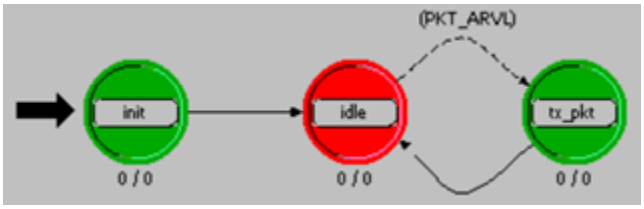

图 3-3

（b）将 idle 状态到 tx_pkt 状态转移线的 condition 属性改为 PKT_ARVL .

PKT_ARVL 宏用来判断是否收到了流中断，在这个进程中，只会收到来自 generator，模块的流中断，所以在定义宏的时候不需要指定流中断来自于哪个包流线。

6) 在 Header Block 中添加如下代码，并保存

```
/* Input stream from generator module */
#define IN_STRM    0
/* Output stream to bus transmitter module */
#define OUT_STRM            0


/* Conditional macros */
#define PKT_ARVL (op_intrpt_type() == OPC_INTRPT_STRM)
/* Global Variable */
extern int subm_pkts;
```

7) 打开 State Variable Block 进行如图 2-4 所示设置

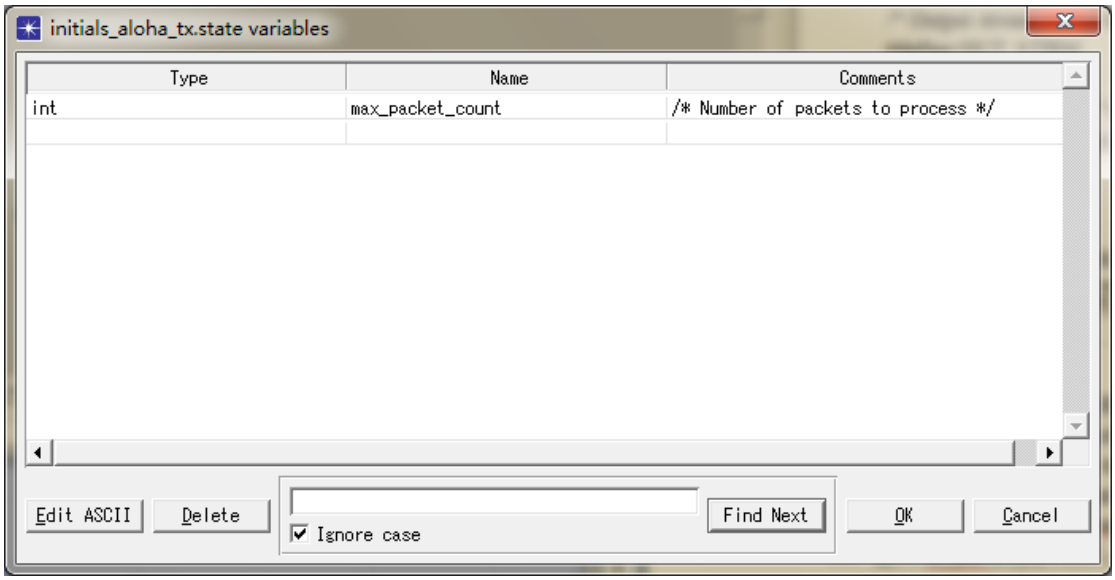

图 2-4

8) 定义 init 状态的动作，并添加如下执行代码

```
/* Get the maximum packet count, */
/* set at simulation run-time */
op_ima_sim_attr_get_int32 ("max packet count",
    &max_packet_count);
```

9) 定义 tx_pkt 状态的动作，并添加如下执行代码

```
/* Outgoing packet */
Packet *out_pkt;
/* A packet has arrived for transmission.    Acquire */
/* the packet from the input stream, send the packet */
/* and update the global submitted packet counter. */
out_pkt = op_pk_get (IN_STRM);
op_pk_send (out_pkt, OUT_STRM);
++subm_pkts;

/* Compare the total number of packets submitted with */
/* the maximum set for this simulation run.    If equal */
/* end the simulation run. */
if (subm_pkts == max_packet_count)
op_sim_end ("max packet count reached.", "", "", "");
```

保存，并关闭对话框

10）定义仿真属性

（a）选择 Interfaces > Global Attributes.
（b）输入下图所示信息

Attribute Name	Group	Type	Units	Default Value
max packet count		integer		0

图 2-5

（c）单击 OK，保存

11）设置进程模型接口

（a）选择 Interfaces > Process Interfaces。
（b）将 begsim intrpt 属性改为 enabled。
（c）将其他属性的 Status 改为 hidden。
（d）单击 OK，关闭对话框

12) 编译进程，保存名字为<initials>_aloha_tx，编译成功后，关闭进程模型编辑器。

2.2. 创建发射节点的节点模型

1）选择 File > New…，然后从下拉菜单中选择 Node Model，单击 OK。

2）编辑窗口放置两个 processor 和一个 bus transmitter。

3）按照下图所示修改每个模块的名字，并连接包流线。

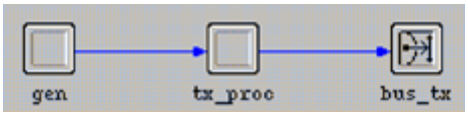

图 2-6

4）提升 gen 模块的 interarrival time 属性。

（a）右键单击 gen 模块，打开其属性对话框。
（b）将 process model 属性改为 simple_source。
（c）右键单击 Packet Interarrival Time，选择 Promote Attribute to Higher Level，则在右边 Value 会显示 promoted，如图

图 2-7

（d）单击 OK，关闭属性对话框。

5)设置 tx_proc 模块的属性

（a）打开 tx_proc 的属性对话框，将 process model 属性设置为<initials>_aloha_tx。

（b）单击 OK，关闭属性对话框。

6) 对上述所创建的节点模型进行功能增强。

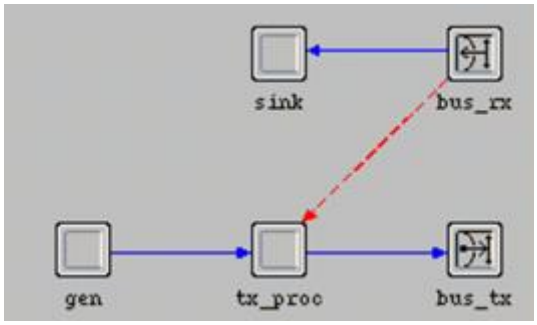

图 2-8

（a)在原有基础上增加一个 process 模块和一个 bus transmitter，分别命名为 **sink** 和 **bus_rx**，并用包流线按照下图方式连接。

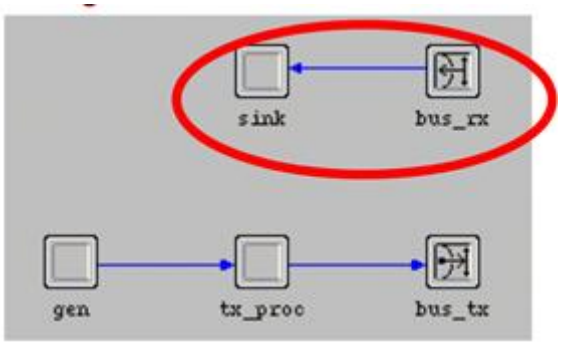

图 2-9

（b）使用 **Create Statistic Wire** 工具按钮，按照下图方式连接 **bus_rx** 和 **tx_proc**。

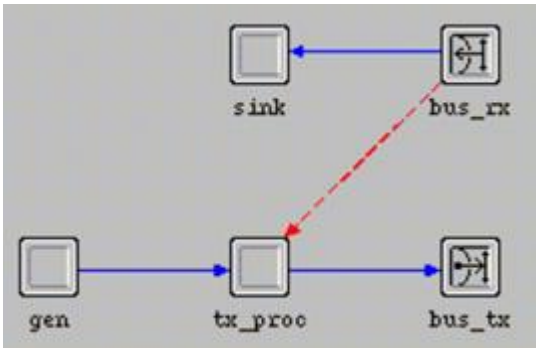

图 2-10

（c）右键单击橙色的 **statistic wire**，打开其属性对话框，将 **rising edge trigger** 和 **falling edge trigger** 属性都改为 **disabled**。单击 **OK**，关闭属性对话框。

7）检查包流线的连接是否正确

（a）右键单击 tx_proc 模块，选择 Show Connectivity，确定和下图一致。

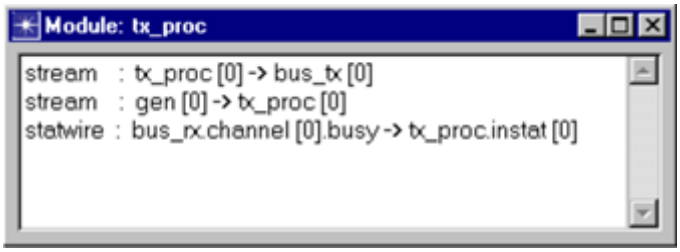

图 2-11

（b）如果不一致，则做如下修改

- 右键单击 gen 到 tx_proc 模块之间的包流线，选择 Edit Attributes，打开属性对话框。
- 将 src stream 属性的值改为 src stream [0]。
- 单击 OK，关闭属性对话框。
- 右键单击 bus_rx 到 tx_proc 之间的统计线，选择 Edit Attributes，打开属性对话框。
- 将 dest stat 属性的值修改为 instat [0]。
- 单击 OK，关闭属性对话框。

8）设置节点模型接口

（a）选择 Interfaces > Node Interfaces。
（b）在 Node types 列表中，将 mobile 和 satellite 项的 Supported 值改为 no。
（c）除了 gen.Packet Interarrival Time 属性之外，将其他属性的 Status 值都改为 hidden。
（d）单击 OK，关闭对话框。

9）将节点模型保存为<initials>_cct_tx，然后关闭节点模型编辑器。

2.3. 创建接收节点的进程模型

1) 选择 File > New…，从下拉菜单中选择 Process Model，然后单击 OK。

2) 使用 Create State 在编辑窗口中放置两个状态，分别命名为 init 和 idle，并将 init 状态设置为 forced。

3) 按照下图方式连接转移线

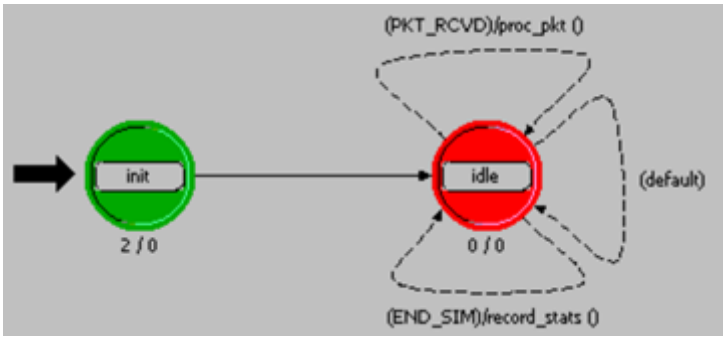

图 2-12

idle 状态到自身的转移线有三条：

（a)第一条(最上面)转移线的 condition 属性改为 PKT_RCVD，executive 属性改为 proc_pkt()。
（b）第二条（中间）转移线的 condition 属性改为 default。
（c）第三条（最下面）转移线的 condition 属性改为 END_SIM，executive 属性改为 record_stats()。

4) 在 Header Block 中添加如下代码

```
/* Input stream from bus receiver */
#define IN_STRM 0


/* Conditional macros */
#define PKT_RCVD (op_intrpt_type () == OPC_INTRPT_STRM)
#define END_SIM (op_intrpt_type () == OPC_INTRPT_ENDSIM)


/* Global variable */
int subm_pkts = 0;
```

宏 IN_STRM 表示来自 bus receiver 模块的输入流；宏 PKT_RCVD 用于判断流中断是否到来；宏 END_SIM 用于判断仿真结束中断（在仿真结束时，由仿真内核自动

触发）是否到来。全局变量 subm_pkts 用于记录所有节点发送的总的数据包数。在 HB 里面定义的变量为全局变量，网络中的所有节点都可以访问。

5）定义状态变量 SV（State Variables）

打开 State Variables block，输入下图所示信息，然后保存。

Type	Name	Comments
int	rcvd_pkts	Received packet counter

图 2-13

6）打开 Function Block，添加如下代码

```
/* This function gets the received packet, destroys          */
/* it, and logs the incremented received packet total        */
static void proc_pkt (void)
        {
        Packet* in_pkt;
        FIN (proc_pkt());
        /* Get packet from bus receiver input stream */
        in_pkt = op_pk_get (IN_STRM);

        /*Destroy the received packet */
        op_pk_destroy (in_pkt);

        /* Increment the count of received packet */
        ++rcvd_pkts;
        FOUT;
        }

/* This function writes the end-of-simulation channel        */
/* traffic and channel throughput statistics to a            */
/* vector file
```

```
static void record_stats (void)
        {
        double cur_time;
        FIN (record_stats());
        cur_time = op_sim_time();
        /* Record final statistics */
        op_stat_scalar_write ("Channel Traffic G",
                (double) subm_pkts / cur_time);
        op_stat_scalar_write ("Channel Throughput S",
                (double) rcvd_pkts / cur_time);
        FOUT;
    }
```

7）双击 init 状态打开 Enter Executives block，添加如下代码

```
/* Initialize accumulator */
rcvd_pkts = 0;
```

8）设置进程接口

（a）选择 Interfaces > Process Interfaces。
（b）将 begsim intrpt 和 endsim intrpt 属性的 initial value 值都改为 enabled。
（c）将所有属性的 Status 值都改为 hidden。
（d）单击 OK，关闭对话框。

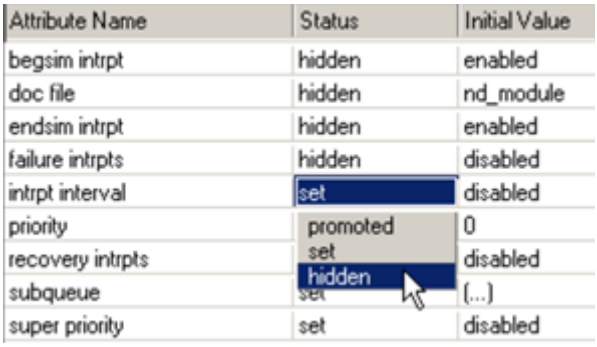

图 2-14

9）编译进程模型

（a）单击 Compile Process Model 工具按钮。
（b）将进程模型保存为<initials>_cct_rx，然后单击 Save 按钮。
（c）关闭编译对话框和进程模型编辑器。

2.4. 创建接收节点的节点模型

1) 选择 File > New… 然后从下拉菜单中选择 Node Model 然后单击 OK。

2) 在编辑窗口放置一个 processor 模块和一个 bus receiver 模块，并分别改名为 rx_proc 和 bus_rx。

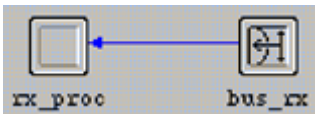

图 2-15

3) 按照上图方式使用包流线连接这两个模块。其中输入包流线的索引号默认为 0，这和 cct_rx 进程的 HB 中定义的一致。

4) 右键单击 rx_proc 模块，打开其属性对话框，将 process model 属性设置为 <initials>_cct_rx。

5) 设置节点模型接口

（a）选择 Interfaces > Node Interfaces。
（b）在 Node types 列表中，将 mobile 和 satellite 类型的 Supported 值改为 no。
（c）在 Attributes 列表中，将所有属性的 Status 值改为 hidden。

6) 保存节点模型为<initials>_cct_rx，然后关闭节点模型编辑器。

2.5. 创建链路模型

1） 选择 File > New…，然后从下拉菜单中选择 Link Model，然后单击 OK。

2） 在 Supported link types 列表中，将 ptsimp 和 ptdup 类型的 Supported 值改为 no。

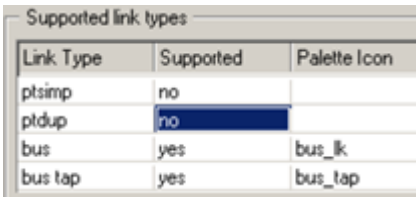

图 2-16

3） 将该链路模型保存为<initials>_cct_link，然后关闭链路模型编辑器。

2.6. 创建网络模型

1) 选择 File > New..，然后从下拉菜单中选择 Project，然后单击 OK。

2) 在 Startup Wizard 向导中，按照下表所示设置。

Dialog Box Name	Value
Initial Topology	Default value: **Create empty scenario**
Choose Network Scale	**Office** (“Use metric units” selected)
Specify Size	**700 x 700 Meters**
Select Technologies	None
Review	Check values, then click Finish

3) 创建对象功能栏（object palette）

（a）单击左上角的 **Open Object Palette** 工具按钮  。

（b）单击左上角的 **Open Palette in Icon View** 按钮  ，切换到图标视图。

（c）单击 **Configure Palette…**按钮。

（d）在 **Configure Palette** 对话框中，单击 **Clear** 按钮，然后单击 **Node Models** 按钮。

（e）包含**<initials>_cct_rx** 和**<initials>_cct_tx**，然后单击 **OK**。

（f）在 **Configure Palette** 对话框中，单击 **Link Models** 按钮。

（g）包含**<initials>_cct_link**，然后单击 **OK**。

（h）选择 **Save As**，将其命名为**<initials>_cct**，然后单击 **Save**。

4) 创建网络拓扑

（a）选择 Topology > Rapid Configuration…。

（b）从 Configurations 的下拉菜单中选择 Bus，然后单击 Next…。

（c）按照下图所示来设置 Rapid Configuration: Bus 对话框。

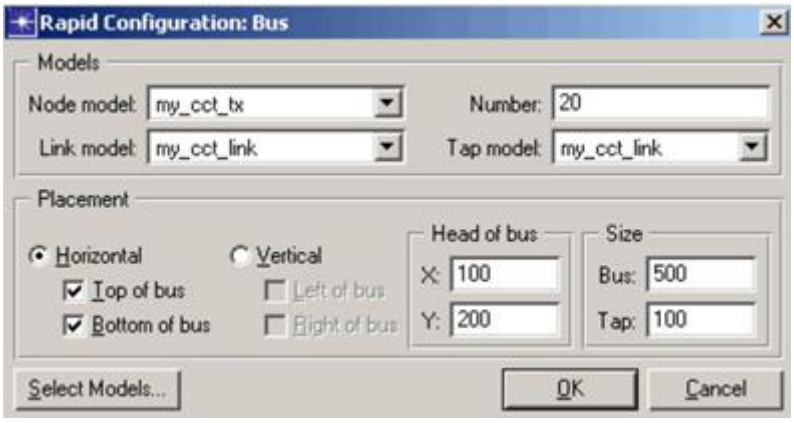

图 2-17

（d）设置完成后，单击 OK。然后下图所示的总线网络会被创建。

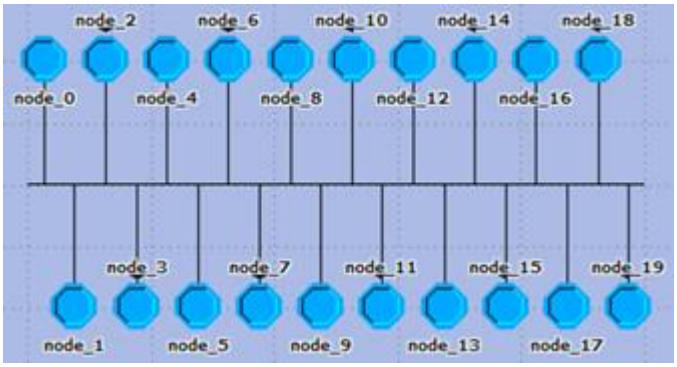

图 2-18

（e）从刚创建的对象功能栏中拖一个<initials>_cct_rx 节点到网络中。

（f）在对象功能栏中单击<initials>_cct_link tap 链路，确保使用的是 tap link。

图 2-19

（g）从总线开始，连接刚放入网络中的<initials>_cct_rx 节点，确保是从总线开始的。

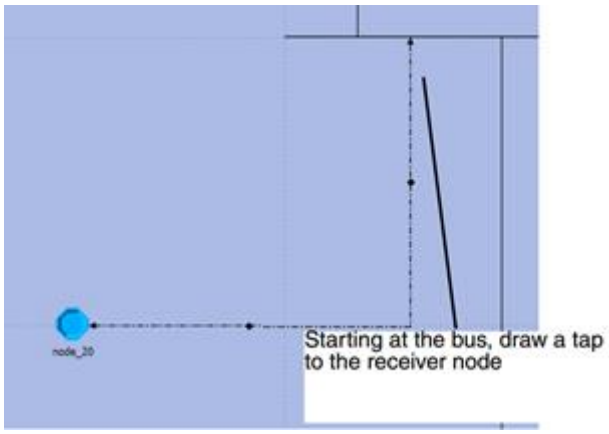

图 3-20

（h）确认最后的网络模型如下图所示。

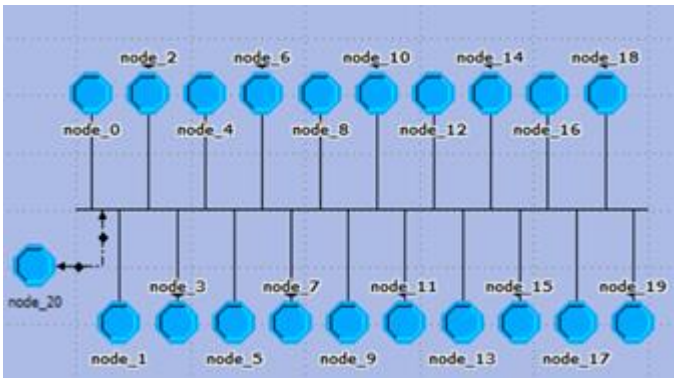

图 2-21

（i）将网络模型保存为<initials>_cct_network，并关闭对象功能栏，不要关闭网络模型编辑器。

2.7. 运行 ALOHA 仿真

1）配置仿真序列

（a）选择 Scenarios > Scenario Components > Import…。
（b）从下拉菜单中选择 Simulation Sequence，然后选择 cct_network-CSMA，单击 OK。
（c)选择DES > Configure/Run Discrete Event Simulation (Advanced)，则Simulation Sequence对话框会被打开，其中包含 12 次仿真运行，每次仿真运行使用不同的参数。

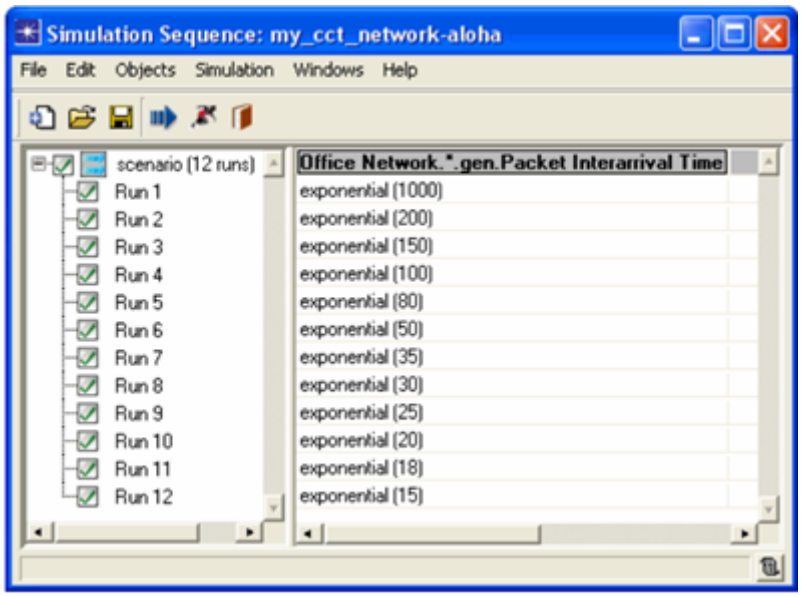

图 2-22

（d）右键单击 scenario (12 runs)节点，选择 Edit Attributes，展开 Execution 节点，然后展开 Advanced 节点，选择 Application。单击 Application 节点，则应用面板会被打开。

（e）确认 Network model 设置为<initials>_cct_network-aloha。
（f）单击 Outputs 节点，然后单击 Statistics Collection 节点。
（g）将 Probe file 设置为<NONE>。
（h）确认 Vector file 设置为<initials>_cct_network-aloha。这个文件收集 op_stat_scalar_write()写入的结果。每次仿真运行都会产生一个以-DES-<run#>为后缀的矢量文件。在这个教程里面，该文件只包含了标量数据。
（i）单击 Inputs 节点，然后单击 Global Attributes 节点，确认 max packet count 设置为 1000。
（j）单击 Object Attributes 节点，在属性 Office Network.*.gen.Packet Interarrival Time 的 Value 一栏，有 12 个值。
（k）单击 OK，保存，并关闭 Simulation Sequence 对话框。
（l）选择 File > Save。

2) 运行仿真

（a）确认上图中的 Run1-Run12 全部被选上。
（b）单击 Running Man 工具按钮。

图 2-24

（c）随后弹出 Confirm Execution 对话框，单击 Yes。

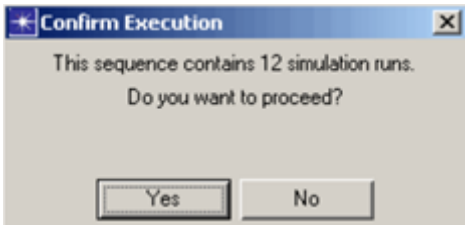

图 2-25

（d）随后会弹出 DES Execution Manager 对话框，显示仿真运行的进度。每次仿真运行都产生 1000 个数据包，当产生的数据包达到 1000 个，则仿真会被终止（单击 View Details 可以看到相应的提示信息）。

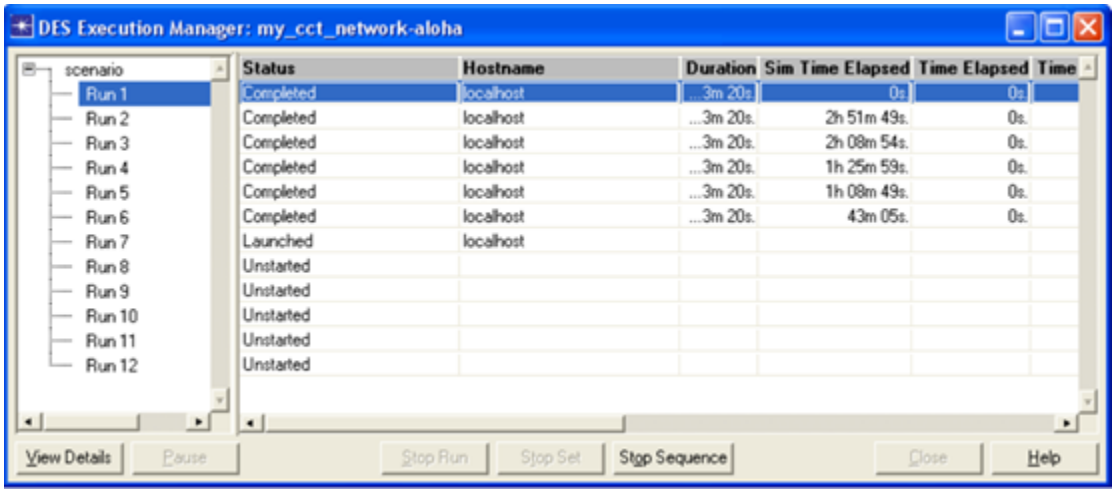

图 2-26

```
Beginning simulation of my_cct_network-aloha at 18:10:30 Fri Apr 13 2007
----
Kernel: development (not optimized), sequential, 32-bit address space
----
Simulation terminated by process (ss_aloha_tx) at module (top.Office
Network.node_13.tx_proc), T (10308.6), EV (100887)
max packet count reached.



----
Simulation Completed - Collating Results.
Events: Total (100,889); Average Speed (925,593 events/sec.)
Time  : Elapsed (0.11 sec.); Simulated (2 hr. 51 min. 48 sec.)
DES Log: 2 entries
```

图 2-27

（e）仿真完成后，关闭 DES Execution Manager 对话框和 Simulation Sequence 编辑器。

2.8. 分析 ALOHA 仿真结果

1) 在项目编辑器的工具栏中单击 View Results 工具按钮，则 Results Browser 会被打开。

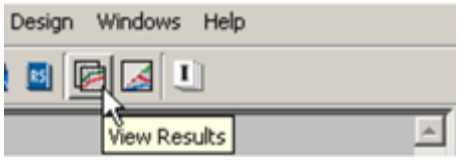

图 2-28

2) 单击 DES Parametric Studies 选项卡。

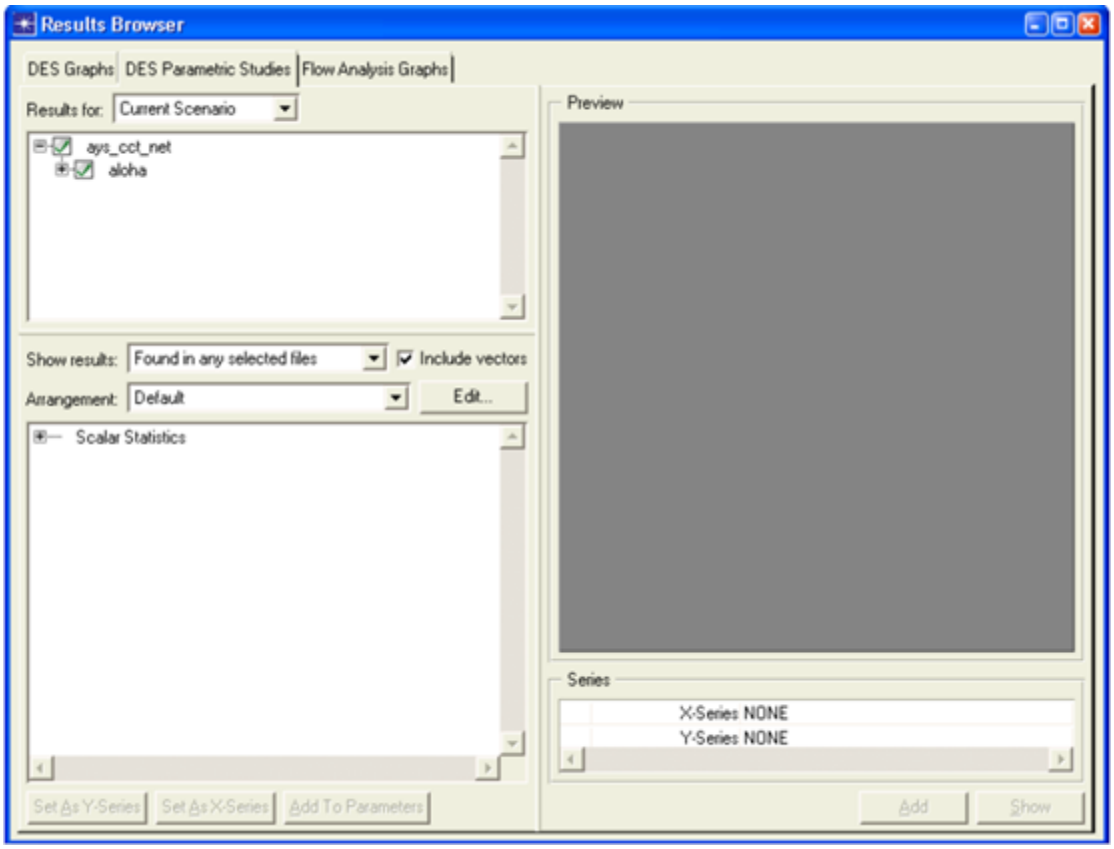

图 2-29

3) 展开 Scalar Statistics 节点，右键单击 Channel Throughput S，然后选择 Set as Y-Series。右边的 Preview 框会显示下图所示的曲线。

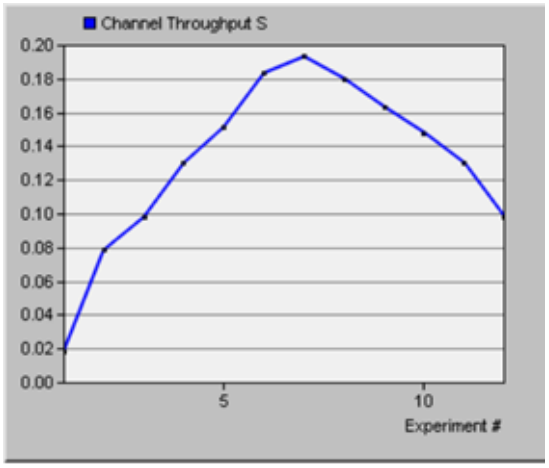

图 3-30

4）右键单击 Channel Traffic G，然后选择 Set as X-Series。

5）单击 Show，则曲线如下图所示。

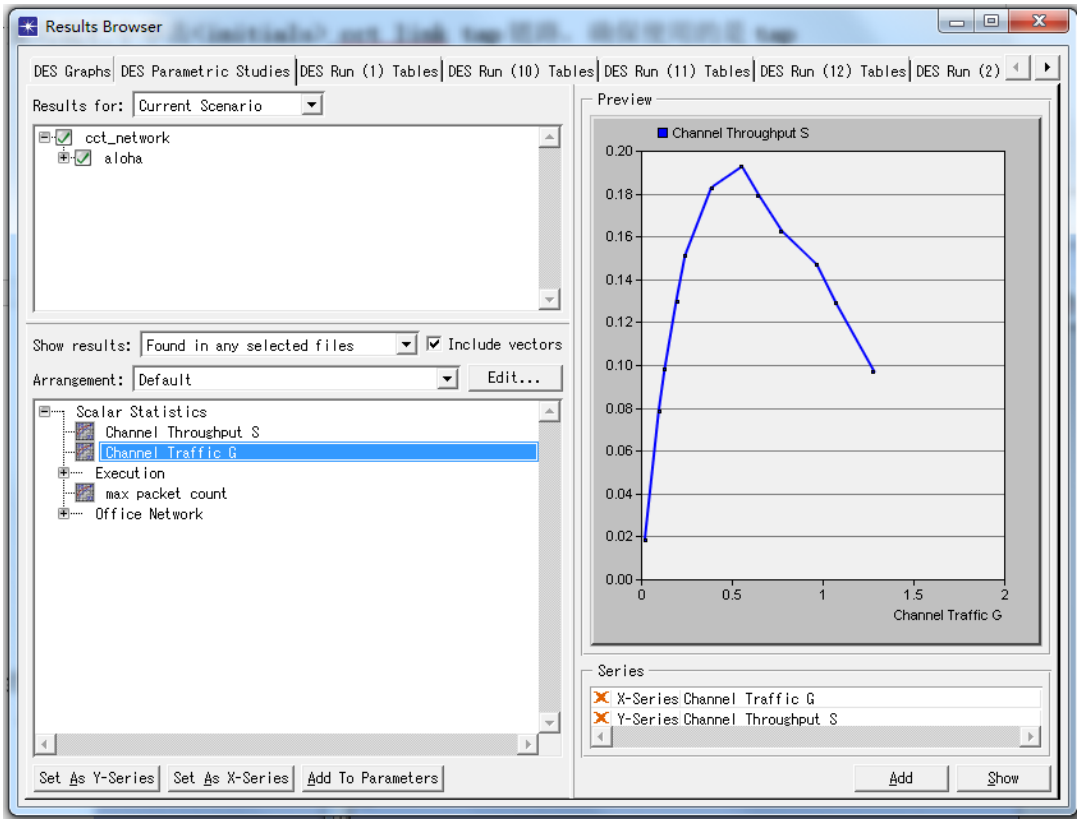

图 3-31

在理论上，ALOHA 协议的吞吐量 S 和负载 G 的关系为 $S=Ge^{-2G}$，则最大的吞吐量为 $S_{max} = 1/（2e）≈0.18$。

6）关闭曲线图（点击 delete）和 Results Browser 窗口。