NS2 Lab1 网络仿真环境配置与 通信场景模拟

一．实验目的

(1) 理解网络协议栈的分层机制，指明实验1中关于各层的代码定义，运行出并能解释实验结果;
(2) 掌握信道容量的计算方法。

二．实验过程

1. 安装ns2

（1）下载ns-allinone-2.35.tar.gz

（2）解压

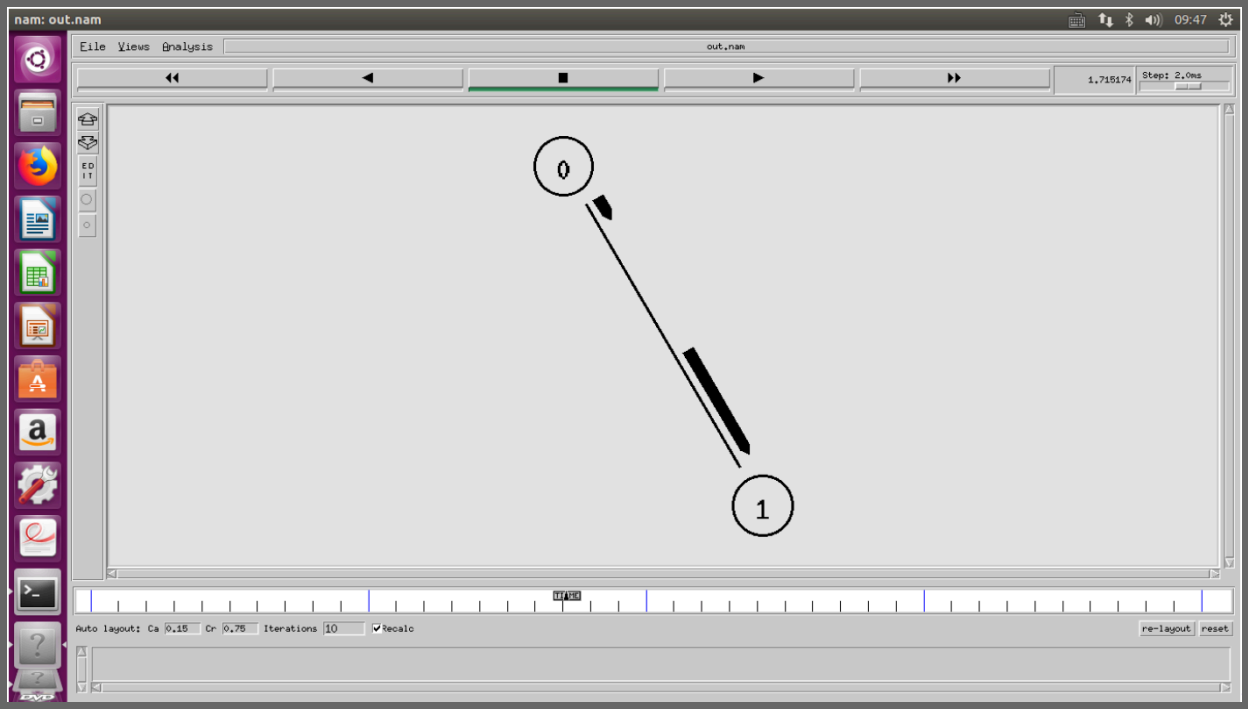
（3）修改linkstate文件夹下的ls.h文件137行的erase改为this->erase。

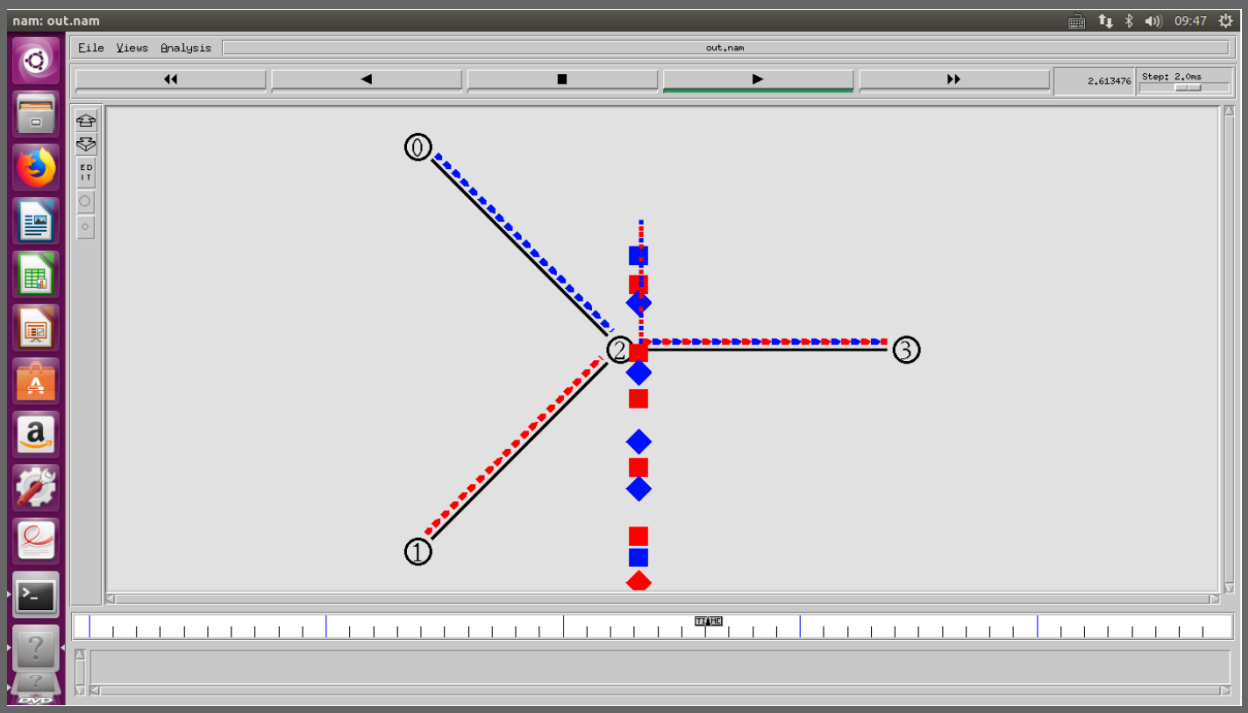
（4）运行./install进行安装

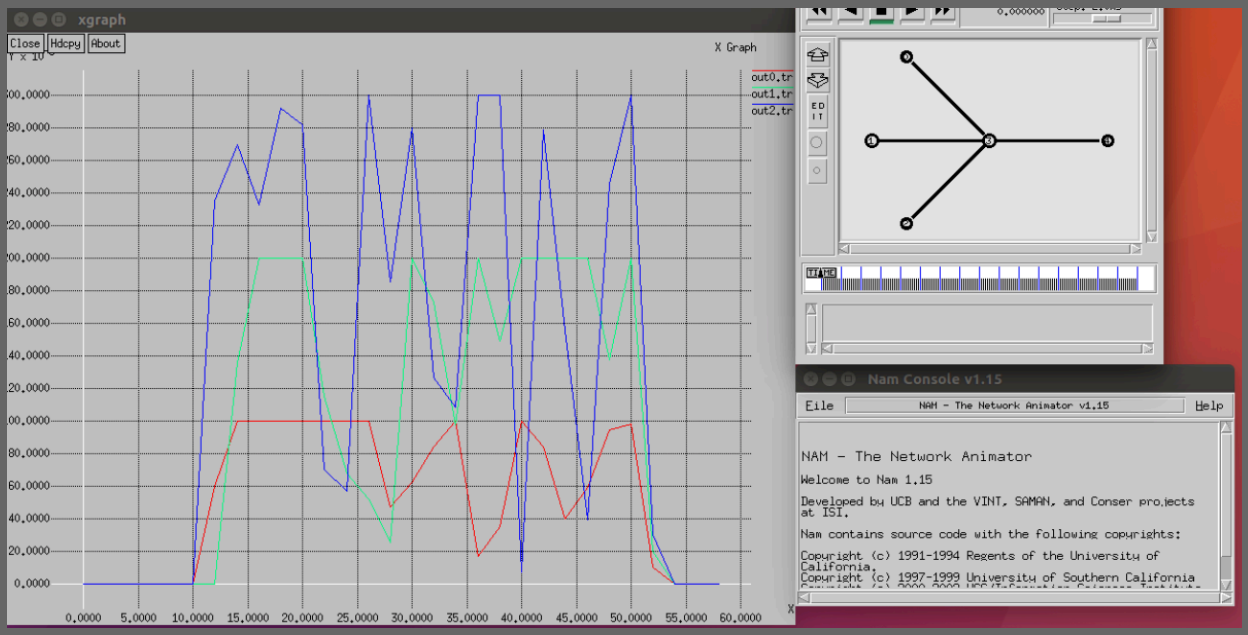
（5）将bin加入环境变量PATH中

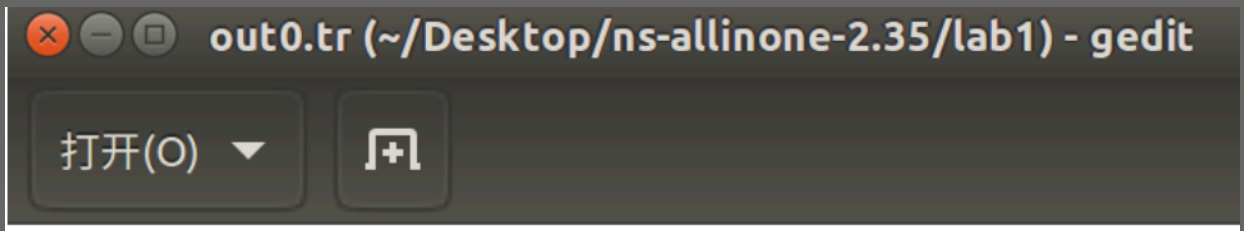
（6）Ns可以使用

2. 运行示例代码example1.tcl，example2.tcl，example3.tcl，lab1.tcl，效果分别如下图




nam: out.nam
09:47
File Views Analysis
out.nam
2.613476
Step: 2.0ms
0
1
2
3


xgraph
Close Hdcpy About
X Graph
out0.tr
out1.tr
out2.tr
0.0000 5.0000 10.0000 15.0000 20.0000 25.0000 30.0000 35.0000 40.0000 45.0000 50.0000 55.0000 60.0000
Nam Console v1.15
File
NAM - The Network Animator v1.15
Help
NAM - The Network Animator
Welcome to Nam 1.15
Developed by UCB and the VINT, SAMAN, and Conser projects at ISI.
Nam contains source code with the following copyrights:
Copyright (c) 1991-1994 Regents of the University of California.
Copyright (c) 1997-1999 University of Southern California




```
0 0.0
2 0.0
4 0.0
6 0.0
8 0.0
10 0.0
12 0.05999999999999998
14 0.10000000000000001
16 0.10000000000000001
18 0.10000000000000001
20 0.10000000000000001
22 0.10000000000000001
24 0.10000000000000001
26 0.10000000000000001
28 0.04719999999999999
30 0.06239999999999997
32 0.08400000000000005
34 0.10000000000000001
36 0.01679999999999999
38 0.03520000000000002
40 0.10000000000000001
42 0.08400000000000005
44 0.04000000000000001
46 0.05920000000000003
48 0.09439999999999998
50 0.09840000000000001
52 0.0104
54 0.0
56 0.0
58 0.0
```

三. 代码理解

```
1  #The source file for NS2 Warm Up
2  ##################################################################
   ###
3  # Create a simulator object
```

```
4  set ns [new Simulator]
5
6  # Define colors
7  $ns color 1 Blue
8  $ns color 2 Red
9  $ns color 3 Green
10
11 # Open the output files for recording
12 # 定义文件并打开文件以写入数据
13 set f0 [open out0.tr w]
14 set f1 [open out1.tr w]
15 set f2 [open out2.tr w]
16
17 # Open a file for the nam trace data
18 # 打开out.nam文件
19 set nf [open out.nam w]
20 $ns namtrace-all $nf /
21
22 # Define the 'finish' procedure
23 # 定义结束程序，包括关闭输出文件，运行nam，确定xgraph窗口大小及要显示的数据集
   实现数据可视化
24 proc finish {} {
25     # 全局变量声明
26     global f0 f1 f2 ns nf
27     close $f0
28     close $f1
29     close $f2
30     close $nf
31     # 效果等价于：将exec后面的命令输入到命令行执行
32     exec nam out.nam &
33     #Call xgraph to display the results
34     exec xgraph out0.tr out1.tr out2.tr -geometry 800x600 &
35     exit 0
36     }
37
38 # Define the 'record' procedure
39 # 定义“记录”程序
40 proc record {} {
41     global sink0 sink1 sink2 f0 f1 f2
42     #Get an instance of the simulator
43     set ns [Simulator instance]
44     #Set the time after which the procedure should be called again
45     # 定义再次发送的间隔时间
46     set time 2.0
47     #How many bytes have been received by the traffic sinks?
48     # 记录传输的字节数
49     set bw0 [$sink0 set bytes_]
50     set bw1 [$sink1 set bytes_]
51     set bw2 [$sink2 set bytes_]
```

```
52      #Get the current time
53      #获取当前时间
54      set now [$ns now]
55      # Calculate the bandwidth (in MBit/s) and write it to the
    files
56      # 计算带宽换算成MBit/s单位并写入输出文件
57      puts $f0 "$now [expr $bw0/$time*8/1000000]"
58      puts $f1 "$now [expr $bw1/$time*8/1000000]"
59      puts $f2 "$now [expr $bw2/$time*8/1000000]"
60      # Reset the bytes_ values on the traffic sinks
61      # 重置接收数据
62      $sink0 set bytes_ 0
63      $sink1 set bytes_ 0
64      $sink2 set bytes_ 0
65      #Re-schedule the procedure
66      $ns at [expr $now+$time] "record"
67  }
68
69  # Define the attach-expoo-traffic procedure
70  # 定义相关的传送程序
71  proc attach-expoo-traffic { node sink size burst idle rate color }
    {
72      #Get an instance of the simulator
73      set ns [Simulator instance]
74
75      # Create a UDP agent and attach it to the node
76      # 创建UDP（传输层）
77      set source [new Agent/UDP]
78
79      # 在node节点（物理层、数据链路层）上搭建source（传输层）
80      $ns attach-agent $node $source
81      $source set fid_ $color
82
83      # Create an Expoo traffic agent and set its configuration
    parameters
84      # 创建信源（应用层）并设定相关参数
85      set traffic [new Application/Traffic/Exponential]
86      $traffic set packetSize_ $size
87      $traffic set burst_time_ $burst
88      $traffic set idle_time_ $idle
89      $traffic set rate_ $rate
90
91      # Attach traffic source to the traffic generator
92      # 在传输层上搭建信源
93      $traffic attach-agent $source
94
95      #Connect the source and the sink
96      $ns connect $source $sink
97
```

```
98      return $traffic
99  }
100
101 # Nodes definition
102 # 定义节点（物理层）
103 set n0 [$ns node]
104 set n1 [$ns node]
105 set n2 [$ns node]
106 set n3 [$ns node]
107 set n4 [$ns node]
108
109 # Nodes connection
110 # 将节点之间连接起来，全双工，带宽1Mb，延时100ms（物理层）
111 $ns duplex-link $n0 $n3 1Mb 100ms DropTail
112 $ns duplex-link $n1 $n3 1Mb 100ms DropTail
113 $ns duplex-link $n2 $n3 1Mb 100ms DropTail
114 $ns duplex-link $n3 $n4 1Mb 100ms DropTail
115
116 # Nodes position for nam
117 # 定义网络的拓扑结构，即各个节点的排列位置和逻辑连接
118 $ns duplex-link-op $n0 $n3 orient right-down
119 $ns duplex-link-op $n2 $n3 orient right-up
120 $ns duplex-link-op $n1 $n3 orient right
121 $ns duplex-link-op $n3 $n4 orient right
122
123 # 这里没有3，因为3节点是个中转节点，本身没有发出数据
124 set sink0 [new Agent/LossMonitor]
125 set sink1 [new Agent/LossMonitor]
126 set sink2 [new Agent/LossMonitor]
127
128 # 将这些发送数据点 与 节点4 联系起来
129 $ns attach-agent $n4 $sink0
130 $ns attach-agent $n4 $sink1
131 $ns attach-agent $n4 $sink2
132
133 # 调用attach-expoo-traffic过程
134 # 输入参数（$n0 $sink0 200 2s 1s 100k 1）->（节点0，发送数据量，一个包的
    大小，忙碌时间，空闲时间，传输速率，颜色）
135 # 忙碌时间：允许使用最大速率传输的时间
136 # 空闲时间：允许不发包的最长时间
137 set source0 [attach-expoo-traffic $n0 $sink0 200 1s 30s 100k 1]
138 set source1 [attach-expoo-traffic $n1 $sink1 200 1s 1s 200k 2]
139 set source2 [attach-expoo-traffic $n2 $sink2 200 200s 1s 300k 3]
140
141 # 定义各节点开始/停止传输数据的时间
142 $ns at 0.0 "record"
143 $ns at 1.0 "$source0 start"
144 $ns at 2.0 "$source1 start"
145 $ns at 3.0 "$source2 start"
```

```
146  $ns at 40.0 "$source0 stop"
147  $ns at 45.0 "$source1 stop"
148  $ns at 40.0 "$source2 stop"
149  $ns at 120.0 "finish"
150
151  # 开始运行
152  $ns run
153
154
```