```
/* -*- Mode:C++; c-file-style:"gnu"; indent-tabs-mode:nil; -*- */
/*
 * This program is free software; you can redistribute it and/or modify
 * it under the terms of the GNU General Public License version 2 as
 * published by the Free Software Foundation;
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA  02111-1307  USA
 */
// 运行指令，在 ns-3.29 下
// cp examples/tutorial/second.cc scratch/second.cc
// ./waf --run second
// ./waf --run second --vis #  可视化
// ./waf --run "second -- nCsma=100" --vis #  局域网中节点数 3->100

//1.头文件
#include "ns3/core-module.h"
#include "ns3/network-module.h"
#include "ns3/csma-module.h"
#include "ns3/internet-module.h"
#include "ns3/point-to-point-module.h"
#include "ns3/applications-module.h"
#include "ns3/ipv4-global-routing-helper.h"
//仿真拓扑图，有两种网络：p2p 网络和 CSMA 网络，分别有 2 个和 4 个节点。其中 n1 上安装有两种 NetDevice，i.e. p2p 和 csma。n0 通过 n1 与 n4 通信。
// Default Network Topology
//
//           10.1.1.0
// n0 -------------- n1   n2   n3   n4
//    point-to-point  |    |    |    |
//                    ================
//                      LAN 10.1.2.0

//2.命名空间
using namespace ns3;

//3.定义 LOG 模块
NS_LOG_COMPONENT_DEFINE ("SecondScriptExample");
```

```
//4.主函数
int
main (int argc, char *argv[])
{
  bool verbose = true;//定义变量，决定是否开启 2 个 applications 的 logging 组件；默认 true 开启
  uint32_t nCsma = 3;//LAN 中另外有 3 个 node

  //使用命令行声明 nCsma 变量
  CommandLine cmd;
  cmd.AddValue ("nCsma", "Number of \"extra\" CSMA nodes/devices", nCsma);//添加命令行参数。nCsma 变量表示，CSMA 节点数目
  cmd.AddValue ("verbose", "Tell echo applications to log if true", verbose);

  cmd.Parse (argc,argv);//读取命令行参数

  if (verbose)
    {
      LogComponentEnable ("UdpEchoClientApplication", LOG_LEVEL_INFO);  //打印 UdpEchoClientApplication 组件信息
      LogComponentEnable ("UdpEchoServerApplication", LOG_LEVEL_INFO); //打印 UdpEchoServerApplication 组件信息
    }

  nCsma = nCsma == 0 ? 1 : nCsma;
  //5.创建网络拓扑
  NodeContainer p2pNodes;
  p2pNodes.Create (2);//创建两个 p2p 型节点，n0----n1

  NodeContainer csmaNodes;
  csmaNodes.Add (p2pNodes.Get (1));//n1 节点既是 p2p 节点，又是 csma 节点
  csmaNodes.Create (nCsma);//创建额外的 nCsma 个 csma 节点，n2,n3,n4

  PointToPointHelper pointToPoint;//p2p 助手类，设置设备和信道属性
  pointToPoint.SetDeviceAttribute ("DataRate", StringValue ("5Mbps"));//设置设备传输速率为 5Mbps
  pointToPoint.SetChannelAttribute ("Delay", StringValue ("2ms"));//设置信道延迟为 2ms

  NetDeviceContainer p2pDevices; //创建 p2p 网络设备
  p2pDevices = pointToPoint.Install (p2pNodes); //把 p2p 网络设备分别安装在节点 n0 和 n1 上，然后共同连接至同一信道对象

  CsmaHelper csma;//csma 助手类，设置 csma 信道属性
```

```
    csma.SetChannelAttribute ("DataRate", StringValue ("100Mbps"));//设置传输速率为
100Mbps
    csma.SetChannelAttribute ("Delay", TimeValue (NanoSeconds (6560)));//设置信道延迟为
6560ns

    NetDeviceContainer csmaDevices;//创建 csma 网络设备
    csmaDevices = csma.Install (csmaNodes);//连接节点与信道

    //6.安装协议栈和分配 ip 地址

    InternetStackHelper stack;//为每个节点安装协议栈
    stack.Install (p2pNodes.Get (0));
    stack.Install (csmaNodes);

    Ipv4AddressHelper address;
    address.SetBase ("10.1.1.0", "255.255.255.0");//为 p2p 网络设备分配 ip 地址，起始地址为
10.1.1.0，终止地址为 10.1.1.254
    Ipv4InterfaceContainer p2pInterfaces;
    p2pInterfaces = address.Assign (p2pDevices);

    address.SetBase ("10.1.2.0", "255.255.255.0");//为 csma 网络设备分配 ip 地址，起始地址为
10.1.2.0，终止地址为 10.1.2.254
    Ipv4InterfaceContainer csmaInterfaces;
    csmaInterfaces = address.Assign (csmaDevices);

    //7.安装应用程序
    UdpEchoServerHelper echoServer (9);//监听 9 号端口
  //配置服务端属性
    ApplicationContainer serverApps = echoServer.Install (csmaNodes.Get (nCsma));//使用
Install 方法将 echoServer 安装在节点 n4(n3?)上，application 在第 1s 开始运行并接受 9 号端
口数据，在第 10s 结束。
    serverApps.Start (Seconds (1.0));
    serverApps.Stop (Seconds (10.0));
  //配置客户端属性
    UdpEchoClientHelper echoClient (csmaInterfaces.GetAddress (nCsma), 9);
    echoClient.SetAttribute ("MaxPackets", UintegerValue (1));//最大发送分组个数，1
    echoClient.SetAttribute ("Interval", TimeValue (Seconds (1.0)));//分组发送间隔 1s
    echoClient.SetAttribute ("PacketSize", UintegerValue (1024));//数据包大小，1024bit

    ApplicationContainer clientApps = echoClient.Install (p2pNodes.Get (0));//使用 install 方法
将 echoClient 安装在节点 n0 上。Application 在模拟，第 2s 开始运行，并接受 9 号端口的
数据，在第 10s 停止。
    clientApps.Start (Seconds (2.0));
    clientApps.Stop (Seconds (10.0));
```

```
//8.设置全局路由
  Ipv4GlobalRoutingHelper::PopulateRoutingTables ();
//9.数据追踪
//开启 P2PHelper 类对象的 pacp，second 为保存文件的前缀名。
  pointToPoint.EnablePcapAll ("second");//EnablePcapAll (“second”)函数的作用是收集这个信道上所有结点链路层分组收发记录。记录文件格式是 pcap，”second”是文件名前缀。
//开启 csmaHelper 类对象的 pcap，使用 CSMA 网段索引为 1 的设备（第二个）进行 sniff，True 开启 Promiscuous mode
  csma.EnablePcap ("second", csmaDevices.Get (1), true);//记录了一个有线节点中 CSMA 网络设备的分组收发信息
//10.启动与结束
  Simulator::Run ();
  Simulator::Destroy ();
  return 0;
}
```