//1.头文件

```
#include "ns3/core-module.h" //all header file is at build/ns3
#include "ns3/network-module.h"
#include "ns3/internet-module.h"
#include "ns3/point-to-point-module.h"
#include "ns3/applications-module.h"
#include "ns3/netanim-module.h"

// Default Network Topology
//
//       10.1.1.0
// n0 -------------- n1
//    point-to-point
//
```

//2.命名空间

```
using namespace ns3; //if is not at namespace ns3, add xx:: , e.g. Time::NS, std::cout, std::min()
```

//3.定义 LOG 模块（允许脚本使用 log 系统中的宏定义打印辅助信息）

```
NS_LOG_COMPONENT_DEFINE ("FirstScriptExample");
```

//4.主函数

```
int
main (int argc, char *argv[])
{
  CommandLine cmd;//在命令行中可以输入参数，例如 sudo ./waf --run "hello-simulator --numbers=5"
  cmd.Parse (argc, argv);//读取命令行参数

  Time::SetResolution (Time::NS);//最小单元时间，毫秒
  LogComponentEnable ("UdpEchoClientApplication",LOG_LEVEL_INFO); //记录指定内容信息
  LogComponentEnable ("UdpEchoServerApplication", LOG_LEVEL_INFO);

  //5.创建网络拓扑
  NodeContainer nodes;
  nodes.Create (2); //创建两个网络节点，访问：nodes.Get (0)、 nodes.Get (1)

  PointToPointHelper pointToPoint; //创建 p2p 助手类
  pointToPoint.SetDeviceAttribute ("DataRate", StringValue ("5Mbps"));//设置设备属性——设备的传输速率
  pointToPoint.SetChannelAttribute ("Delay", StringValue ("2ms"));
```

```
NetDeviceContainer devices;//创建网络设备
devices = pointToPoint.Install (nodes);// pointToPoint.Install (nodes)函数创建了两个 p2p 网络设备对象也一个 p2p 信道对象，并将它们连接一块。

//6. 安装 TCP/IP 协议栈
InternetStackHelper stack;
stack.Install (nodes);//为 nodes 容器中的节点安装 TCP/IP 协议栈

Ipv4AddressHelper address;//为网络设备分配 ip 地址
address.SetBase ("10.1.1.0", "255.255.255.0");//设置起始地址，子网掩码

Ipv4InterfaceContainer interfaces = address.Assign (devices);//把生成的 ip 地址分配给每个设备

//7. 安装应用程序
UdpEchoServerHelper echoServer (9);//服务端监听端口为 9

//ApplicationContainer is to log every node's application
//this is to install application, echoServer, into the node 1 ; set star time and end time
ApplicationContainer serverApps = echoServer.Install (nodes.Get (1));//echoServer.Install()方法把 echoServer 应用安装在 1 号节点，并用 ApplicationContainer 记录
serverApps.Start (Seconds (1.0));//设置 application 在第 1s 开始运行
serverApps.Stop (Seconds (10.0));//第 10s 结束

UdpEchoClientHelper echoClient (interfaces.GetAddress (1), 9);//客户端助手类 UdpEchoClientHelper 在 0 号节点创建应用 echoClient，属性：连接到 1 号 ip 地址和端口
echoClient.SetAttribute ("MaxPackets", UintegerValue (1));//最大发送组数 1
echoClient.SetAttribute ("Interval", TimeValue (Seconds (1.0))); //分组发送间隔 1s
echoClient.SetAttribute ("PacketSize", UintegerValue (1024));//分组负载字节大小，1024bit

ApplicationContainer clientApps = echoClient.Install (nodes.Get (0));
clientApps.Start (Seconds (2.0));
clientApps.Stop (Seconds (10.0));//模拟启动后 2s 向节点 1 的 9 号端口发一个 1024bit 的 UDP 数据包，第 10s 结束

//this is set for visualization (NetAnim)
AnimationInterface anim("first.xml");
anim.SetConstantPosition(nodes.Get(0), 1.0, 2.0);
anim.SetConstantPosition(nodes.Get(1), 2.0, 3.0);

//8. 启动与结束
Simulator::Run ();
```

```
  Simulator::Destroy ();
  return 0;
}
```

```
At time 2s client sent 1024 bytes to 10.1.1.2 port 9
At time 2.00369s server received 1024 bytes from 10.1.1.1 port 49153
At time 2.00369s server sent 1024 bytes to 10.1.1.1 port 49153
At time 2.00737s client received 1024 bytes from 10.1.1.2 port 9
wyl@wyl-virtual-machine:~/tarballs/ns-allinone-3.26/ns-3.26$ sudo
```