

**J I A N G S U U N I V E R S I T Y**

**“无线传感网与识别技术”实验报告**

|  |  |
| --- | --- |
| 学院名称： | 计算机科学与通信工程学院 |
| 专业班级： | 物联网工程18级 |
| 学生姓名： | 温凌峰  张承楷  俞旺昭 |
| 学生学号： | 3180611024  3180611023  3180611013 |
| 指导教师： | 熊书明 |

**2020年7月30日**

目录

[一、流程框图 1](#_Toc47122526)

[二、运行流程介绍 1](#_Toc47122527)

[2.1 配置初始参数 1](#_Toc47122528)

[2.2 创建节点 3](#_Toc47122529)

[2.3为节点安装网络设备 3](#_Toc47122530)

[2.4安装网络协议栈 4](#_Toc47122531)

[2.5安装应用程序 4](#_Toc47122532)

[2.6执行模拟 5](#_Toc47122533)

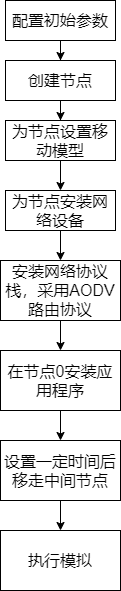
[三、学习日志： 6](#_Toc47122534)

[3.1 温凌峰 6](#_Toc47122535)

[3.2 张承楷 7](#_Toc47122536)

[3.3俞旺昭 8](#_Toc47122537)

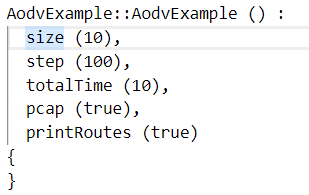
# 一、流程框图



# 二、运行流程介绍

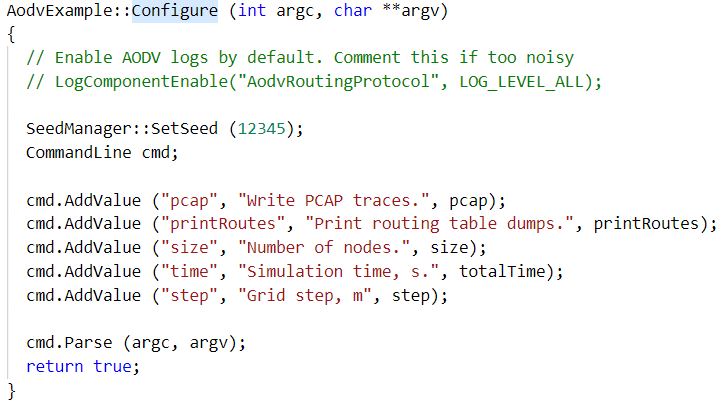
## 2.1 配置初始参数

首先在AodvExample的构造函数中设置了默认的初始参数



设置节点数为10个，节点间距离为100米，模拟时间为10秒，开启每个设备的pcap追踪，打印路由表转储

随后在Configure函数中提供通过命令行修改参数的方法

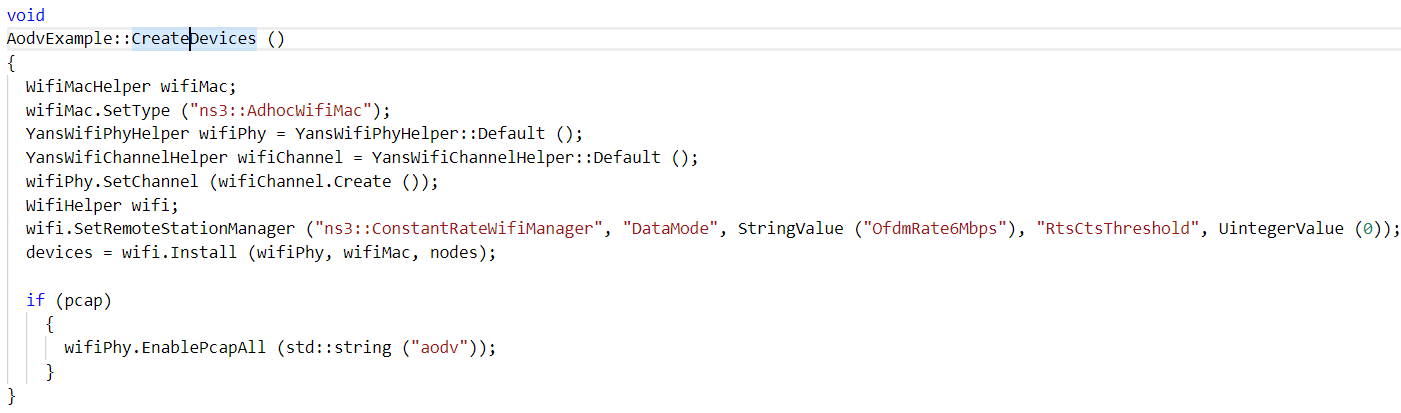


## 2.2 创建节点



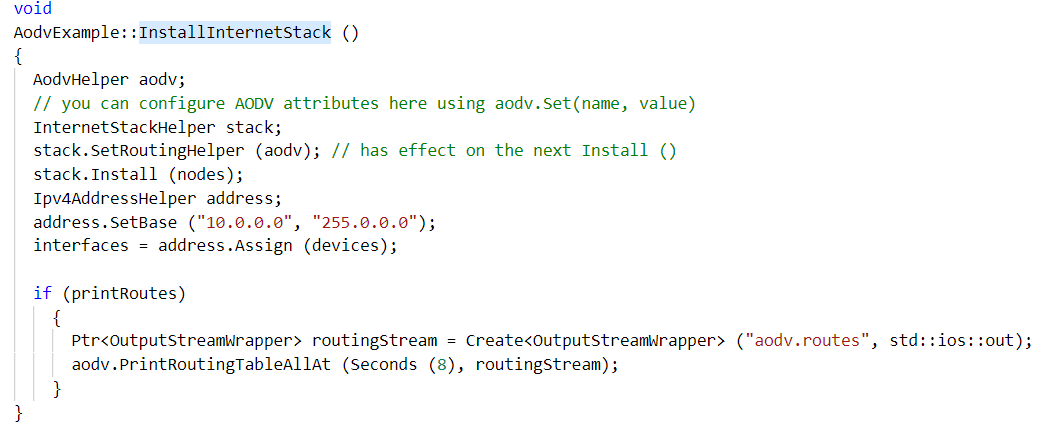
创建size个节点，并命名后为节点设置移动模型，节点分布在一个1维空间中，间隔step长度。

## 2.3为节点安装网络设备



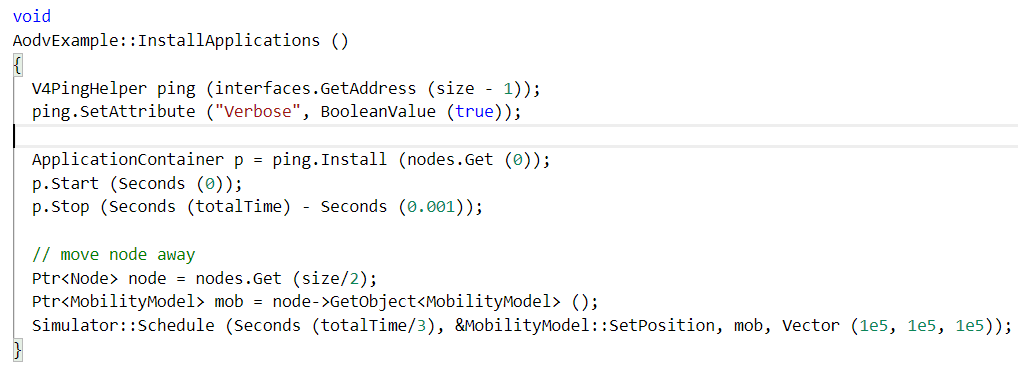
创建WiFi设备，通过物理助手与信道助手创建，安装在节点上

## 2.4安装网络协议栈



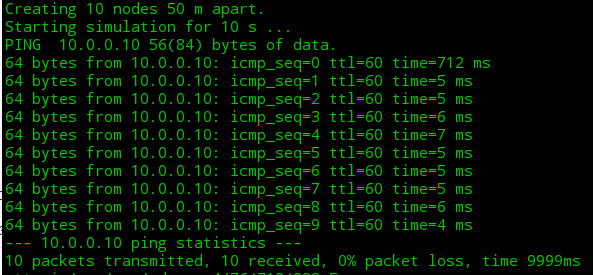
设置该网络采用AODV路由协议，以网络号为10.0.0.0为节点设置ip地址

## 2.5安装应用程序



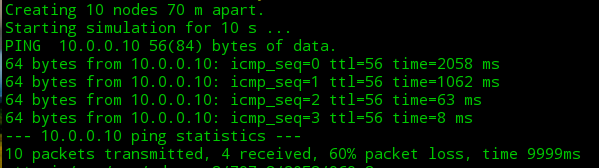
设置应用程序ping地址容器的最后一个地址，在这里是10.0.0.10，然后将该应用安装到节点0上，设置开始和结束时间。下面一段代码的作用是把中间的一个节点在模拟时间过去1/3时移到坐标（10^5, 10^5, 10^5）处，此时就超出了和其它节点的通信范围。由于这一事件的发送，根据初始参数中step设置的大小会出现3种不同的结果。

1、step设置为50时，ping10次，所有数据包都成功接收。



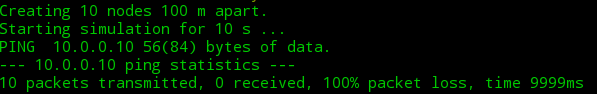
3.33s时节点5移走，节点4和节点6仍然在通信距离内，所以不会丢包

2、step设置为70时



由于3.33s时节点5移走，节点4和节点6距离140m在通信范围外，所以序号3以后的icmp数据包不能被接收。

3、step设置为100时



由于相邻节点都在通信范围外，所以丢包率是100%

## 2.6执行模拟

在调用完4个函数：

CreateNodes ();

CreateDevices ();

InstallInternetStack ();

InstallApplications ();

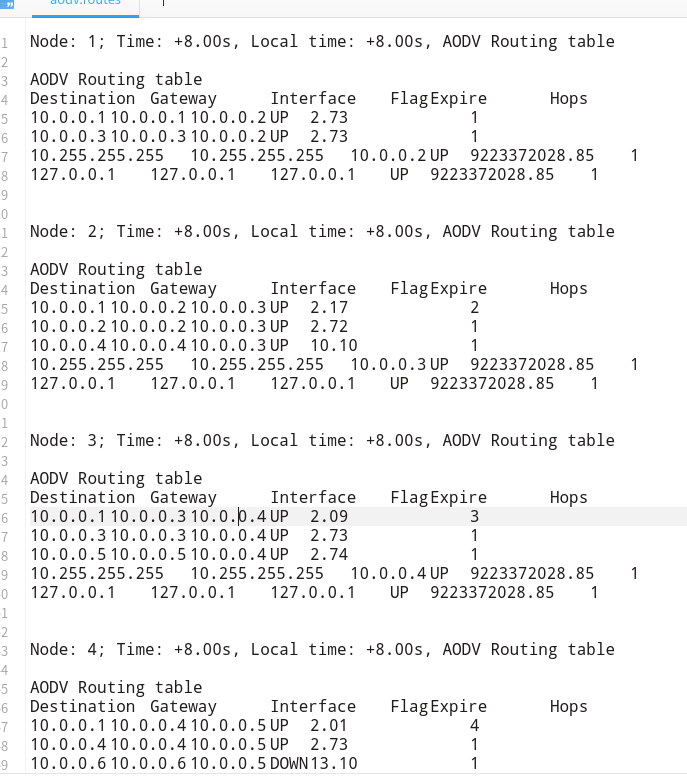
之后，程序就会调用Simulator::Run ();执行模拟

# 三、学习日志：

## 3.1 温凌峰

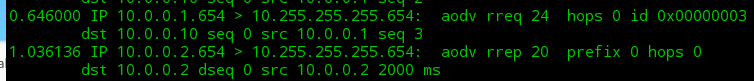
复习ppt中关于AODV路由的知识点：

AODV协议全称为无线自组网按需平面距离向量路由协议，目标是在多个移动节点间建立和维护一个动态、自启动、多跳路由的无线通信网络，其核心机制包括路由发现、路由维护、逐跳路由、目的节点序列号和路由维护阶段周期更新等。节点仅需要维护通向它信号所达范围内邻节点的路由，更远节点的路由信息则不需要维护。这一点通过分析程序导出的路由表数据文件aodv.routes可以发现



从图中可以发现各节点只维护通向它信号所达范围内邻接节点的路由，如节点3的路由表中只有网关为10.0.0.3和10.0.0.5的表项，都是节点3的邻接节点。并且也没有建立全网络的路由表，体现了AODV作为按需路由的特点。

然后是AODV路由协议的控制报文。AODV协议使用3种控制信息，分别是RouteRequest(RREQ)消息、RouteReply(RREP)消息和RouteError(RERR)消息，在IP网络中，这些消息都在UDP上使用654端口号。RREQ用于源节点发起一次在网内建立路由的请求，RREP用于对源节点发送RREQ消息的应答。打开pcap文件可以看到



当修改节点间距为70时，会有RERR消息：路由应答消息RERR用于链路断开或节点不可达时的错误消息通知



在程序中，有关AODV路由协议的地方有以下部分：

AodvHelper aodv;

// you can configure AODV attributes here using aodv.Set(name, value)

InternetStackHelper stack;

stack.SetRoutingHelper (aodv); // has effect on the next Install ()

stack.Install (nodes);

通过查阅官方文档，AodvHelper是一个负责添加AODV路由到节点的助手类，在以上程序中主要起作用的是Create()函数，在语句stack.Install (nodes);中被调用。该函数的参数是将在其上运行路由协议的节点node，负责在所有可用IP接口的子集上安装AODV协议。

## 3.2 张承楷

学习内容：

ns3中的AODV协议：

在ns3中，通过AODVHelper建立AODV路由，AODVHelper继承了IPV4RoutingHelper，其在ns3中的拓扑图如下所示：



可以看到AODVHelper继承了对象工厂，并通过各个类设置了不同的属性。

AODVHelper主要分为以下函数：

1.copy()

返回AODVHelper的复制指针,拓扑图如下：



可以看到，copy函数依赖AODVHelper，但实现复制主要通过其余的helper来实现。

2.create()

在实例中，通过下面的语句调用函数

 AodvHelper aodv;  
 // you can configure AODV attributes here using aodv.Set(name, value)  
 InternetStackHelper stack;  
 stack.SetRoutingHelper (aodv); // has effect on the next Install ()  
 stack.Install (nodes);

在stack.SetRoutingHelper (aodv)这句语句中，调用了create()函数，返回一个路由协议。

综上所述，而在AODV例中，AODVHelper起了重要的作用，负责将AODV的网络路由表建立等功能，而在使用中，要注意的是各个节点间的距离不能过长，否则无法通信。

心得：

通过这次实验课，我更加了解了ns3这个网络仿真应用的使用，对各种网络协议也有了更加深刻的理解。

## 3.3俞旺昭

主函数代码如下所示：

int main (int argc, char \*\*argv)

{

AodvExample test;

if (!test.Configure (argc, argv))

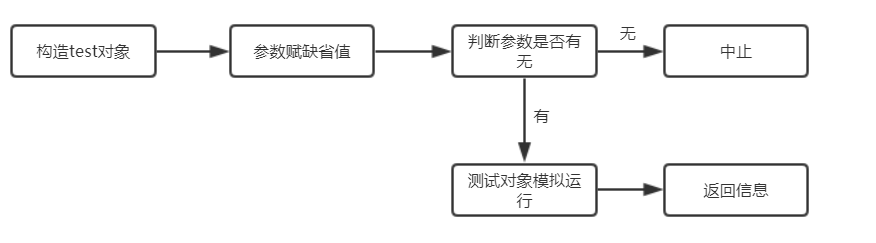
NS\_FATAL\_ERROR ("Configuration failed. Aborted.");

test.Run ();

test.Report (std::cout);

return 0;

}



则由主函数代码可得出主函数流程图大致如下所示：

1.构造对象test

AodvExample test;

在该例程中有AodvExample类，主函数中用此类创建了test的对象。

2.参数赋缺省值

在创建对象时，会有构造函数且由缺省值赋值给参数。缺省值初始如下所示（在第一次进入aodv.cc中时）：

AodvExample::AodvExample () :

size (10),

step (100),

totalTime (10),

pcap (true),

printRoutes (true)

{}

3.判断参数有无

if (!test.Configure (argc, argv))

NS\_FATAL\_ERROR ("Configuration failed. Aborted.");

在该语句中，test.Configure (argc, argv)是作为bool类型判断的，即如果括号里的参数有值，则Configure为ture则可继续运行，反之则为false则终止例程运行。

4.测试对象模拟运行

void Run();

Run()函数如下定义所示：

AodvExample::Run ()

{

CreateNodes ();

CreateDevices ();

InstallInternetStack ();

InstallApplications ();

std::cout << "Starting simulation for " << totalTime << " s ...\n";

Simulator::Stop (Seconds (totalTime));

Simulator::Run ();

Simulator::Destroy ();

}

该语句则为运行仿真的函数。而该函数定义中，首先进行了节点、服务、因特栈和应用的创建。然后输入模拟将进行多少秒（该数值由之前的totalTime

缺省值决定）之后开始模拟的仿真运行。

1. 返回信息

返回的信息用输出流输出即可。