

《网络与通信》课程实验作业

实验一 无线网络中隐藏站点仿真

实验二 无线网络中暴露结点仿真

实验三 WiMax 仿真

实验四 AODV 仿真

实验五 DSR 仿真

实验一 无线网络中隐藏站点仿真

1 实验要求和目的

利用 NS2 仿真隐藏结点问题。

2 实验原理和背景知识

隐藏结点指在接收结点的覆盖范围内而在发送结点的覆盖范围外的结点。由于听不到发送结点的发送，隐藏结点可能向相同的接收结点发送分组，导致分组在接收结点处冲突。隐藏结点可以分为隐发送结点和隐接收结点。

如课件中，结点 A 和 C 同时想发送数据给结点 B，但 A 和 C 都不在对方的传送范围内。所以当 A 发送数据给 B 时，C 并未检测到 A 也在发送数据，会认为目前网络中无数据传送，会将数据发送给 B。这样，A 和 C 同时将数据发送给 B，使得数据在 B 处产生冲突，最终导致发送的数据不可用。这种因传送距离而发生误判的问题称为隐藏结点问题。

为了解决隐藏结点问题，可以通过RTS/CTS控制信息来避免冲突。当发送方发出数据前，先送出一个RTS包，告知在传送范围内的所有结点不要有任何发送操作。如果接收方目前空闲，则响应一个CTS包，告诉发送方可开始发送数据，此CTS包也会告知所有在接收方信号传输范围内的其它结点不要进行任何传输操作。

3 实验环境和模块简介

隐藏结点的仿真分析采用的实验平台为Cygwin + ns-allinone-2.34，同时需要安装mUDP模块进行实验扩展。mUDP模块是UDP的延伸，除了具有UDP的功能外，还能记录所发送的包的信息。mUdpSink可以把接收到的包的信息记录到文件中。

4 实验步骤与仿真演示

4.1 模块安装

仿真所需模块的安装过程如下：

(1) 获取 mUDP, mUdpSink 的模块文件，具体有下列几个文件；

mudp.cc mudp.h mudpsink.cc mudpsink.h

(2) 在/ns-allinone-2.34/ns-2.34/下新建 measure 文件夹，把这四个文件放入其中；

(3) 修改/ns-allinone-2.34/ns-2.34/common/下的 packet.h 文件，将如下程序添加到 struct *hdr_cmh*{}中，如图1.1 所示；

```
int frametype_; double sendtime_; unsigned int pkt_id_;  
unsigned int frame_pkt_id_;
```

```

nsaddr_t next_hop_; // next hop for this packet
int      addr_type_; // type of next_hop_addr
nsaddr_t last_hop_; // for tracing on multi-user channels

// AOMDV patch
int aomdv_salvage_count_;

int frametype_;
double sendtime_;
unsigned int pkt_id_;
unsigned int frame_pkt_id_;

// called if pkt can't obtain media or isn't ack'd. not called if
// dropped by a queue
FailureCallback xmit_failure_;
void *xmit_failure_data_;

```

图 1.1 代码添加位置

(4) 修改/ns-allinone-2.34/ns-2.34/下的 Makefile 文件，将如下程序添加到 *OBJ_CC* 中如图 1.2 所示的位置；

*measure/mudp.o measure/mudpsink.o *

```

xcp/xcpq.o xcp/xcp.o xcp/xcp-end-sys.o \
wpan/p802_15_4csmaca.o wpan/p802_15_4fail.o \
wpan/p802_15_4hlist.o wpan/p802_15_4mac.o \
wpan/p802_15_4nam.o wpan/p802_15_4phy.o \
wpan/p802_15_4sscs.o wpan/p802_15_4timer.o \
wpan/p802_15_4trace.o wpan/p802_15_4transac.o \
apps/pbc.o \
measure/mudp.o measure/mudpsink.o \
$(OBJ_STL)

```

图 1.2 Makefile 修改位置

(5) 修改/ns-allinone-2.34/ns-2.34/tcl/lib/下的 ns-default.tcl，将如下程序添加到文件的最后一行；

Agent/mUDP set packetSize_ 1000

(6) 在 Cygwin 中转入/ns-allinone-2.34/ns-2.34 目录下执行 *make clean*；*make* 命令，如图 1.3 所示，编译约十几分钟完成即可。

```

Administrator@CHINA-F6A383329 ~
$ cd ns-allinone-2.34

Administrator@CHINA-F6A383329 ~/ns-allinone-2.34
$ cd ns-2.34/

Administrator@CHINA-F6A383329 ~/ns-allinone-2.34/ns-2.34
$ make clean;make_

```

图 1.3 编译添加模块的切换路径

4.2 运行仿真与演示

安装完成模块后，就可以开始仿真。仿真过程如下。

(1) 在“home/<用户名>”目录下新建一个目录用于放置运行的仿真脚本，并将不同类型脚本用子目录分开，紧接着将仿真脚本 *Hidden_Terminal.tcl* 复制到该目录下(tcl 脚本的编写不再赘述，请参考相关文献)。如图 1.4 所示，这里新建了一个 *WNT* 目录，然后在目录中放置了各实验仿真代码的子目录（同学们可自行设置）。

```

Administrator@CHINA-3CDB76914 ~/WNT
$ ls
3.5.1 3.5.2 4.6.2 6.5.1 6.5.2 7.6.1 7.6.2 8.7.1 8.7.2 9.4.2 9.4.3
Administrator@CHINA-3CDB76914 ~/WNT

```

图 1.4 仿真代码位置

(2) 在 Cygwin 中利用 cd 命令切换到该新建的子目录(有关 cd 命令的使用请参考Linux 相关书籍), 如图 1.4 所示, 紧接着输入: ns Hidden_Terminal.tcl, 回车即可运行, 此时无动画显示, 如图 1.5 所示;

```
Administrator@CHINA-F6A383329 ~/WNT/3.5.2
$ ns Hidden_Terminal.tcl
num_nodes is set 3
INITIALIZE THE LIST xListHead
channel.cc:sendUp - Calc highestAntennaZ_ and distCST_
highestAntennaZ_ = 1.5, distCST_ = 200.0
SORTING LISTS ...DONE!

Administrator@CHINA-F6A383329 ~/WNT/3.5.2
$ nam: no display name and no $DISPLAY environment variable
```

图 1.5 不运行NAM方式的仿真

(3) 如果想看 nam 仿真动画, 则需要启动 XWin, 然后按(2)中的过程在 XWin 窗口中进行仿真即可, 如图 2.7 所示。XWin 的启动方式为: 在 Cygwin 中输入 startxwin.bat, 回车即可, 如图 1.6 所示;

```
Administrator@CHINA-F6A383329 ~/WNT/3.5.2
$ startxwin.bat
startxwin.bat - Starting on Wind
Administrator@CHINA-F6A383329 ~
$
```

图 1.6 启动 XWin

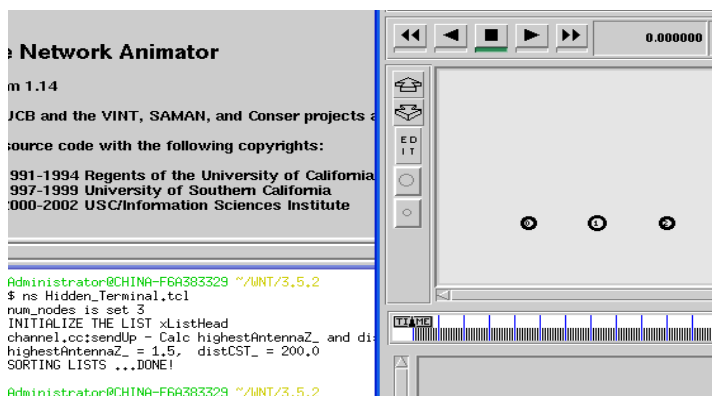


图 1.7 运行 NAM 进行仿真

5 结果分析与讨论

无线局域网隐藏结点问题对网络的传输影响较大, 极大的降低了网络的吞吐量, 增大了时延, 而 CTS/RTS 机制则较好的解决隐藏结点问题。

仿真结果可以从两个方面来分析, 即仿真动画和仿真数据。在 NS2 中演示仿真动画采用 NAM, 它将网络仿真过程中拓扑结构、数据包传输情况及队列丢包等以动画形式展现。仿真结束后, 可以通过分析产生的数据来对仿真过程中的延迟、丢包、吞吐量等进行分析。

实验二 无线网络中暴露结点仿真

1 实验要求和目的

在 NS2 中仿真实现暴露结点相关问题。

2 实验原理和背景知识

暴露结点是指在发送结点的覆盖范围内而在接收结点覆盖范围外的结点，暴露结点因监听到发送结点的发送而可能延迟发送。但是，它其实是在接收结点的通信范围之外，它的发送不会造成冲突。这就引入了不必要的延时。

暴露结点问题和隐藏结点问题不同，当一个结点要发送数据给另一结点时，因邻近结点也正在发送数据，影响了原来结点的数据传送。

为解决暴露结点问题，也采用RTS/CTS 机制。当一个结点侦听到邻近结点发出来的RTS，但却没有听到相对应 CTS 时，可判定它本身是一个暴露结点，所以允许传送数据到其它邻近结点。这个结点可以成功送出 RTS，但相应的 CTS 不一定能被成功地收到。当 S2 收到S1 发出的RTS，但未收到对应 CTS 时，S2 可推测自己是暴露结点，则可同时发送数据。但此时，虽然 S2 发出的数据不会冲突，但其它结点发给 S2 的数据(CTS/ACK)却可能与 S1 发出的数据发生冲突，影响传输系统质量。因此这种情况下，使用 RTS/CTS 并不能完全解决暴露结点问题。

3 实验环境和模块简介

暴露结点的仿真分析采用的实验平台为Cygwin + ns-allinone-2.34，同时需要安装mUDP模块进行实验扩展。mUDP 模块是 UDP 的延伸，除了具有 UDP 的功能外，还能记录所发送的包的信息。mUdpSink 可以把接收到的包的信息记录到文件中。

4 实验步骤与仿真演示

暴露结点仿真进行之前需要安装特定的模块，具体见实验一中的 1.1 节内容，在此不再重复。暴露结点的具体仿真过程如下。

(1) 在“home/<用户名>”目录下新建一个目录用于放置运行的仿真脚本，并将不同类型脚本用子目录分开，紧接着将仿真脚本 Exposed_Terminal.tcl 拷贝到该目录下。如图 2.1 所示，这里新建了一个WNT 目录，然后在目录中放置了各实验仿真代码的子目录。

```
Administrator@CHINA-3CDB76914 ~/WNT
$ ls
3.5.1  3.5.2  4.6.2  6.5.1  6.5.2  7.6.1  7.6.2  8.7.1  8.7.2  9.4.2  9.4.3
Administrator@CHINA-3CDB76914 ~/WNT
```

图 2.1 仿真代码位置

(2) 在 Cygwin 中利用 cd 命令转到该新建的目录，紧接着输入：ns Exposed_Terminal.tcl，回车即可运行，此时无动画显示，如图 2.2 所示；

```
Administrator@CHINA-F6A383329 ~/WNT/3.5.3
$ ns Exposed_Terminal.tcl
num_nodes is set 4
INITIALIZE THE LIST xListHead
channel.cc:sendUp - Calc highestAntennaZ_ and distCST_
highestAntennaZ_ = 1.5, distCST_ = 120.0
SORTING LISTS ...DONE!

Administrator@CHINA-F6A383329 ~/WNT/3.5.3
$ nam: no display name and no $DISPLAY environment variable
```

图 2.2 未运行 NAM 进行仿真

(3) 如果想看 nam 仿真动画，则需要启动 XWin，然后按(2)中的过程进行仿真即可，如图 2.4 所示。XWin 的启动方式为：在Cygwin 中输入 startxwin.bat，回车即可，如图 2.3 所示；

```
Administrator@CHINA-F6A383329 ~/WNT/3.5.3
$ startxwin.bat
startxwin.bat - Starting on Wind
Administrator@CHINA-F6A383329 ~$
```

图 2.3 启动 XWin

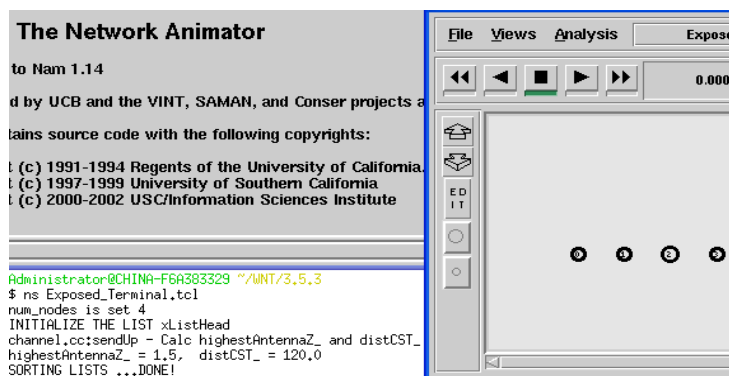


图 2.4 运行 NAM 进行仿真

5 结果分析与讨论

无线局域网暴露结点问题对网络的传输影响较大，极大的降低了网络的吞吐量，增大了时延，而CTS/RTS 机制则较好的解决隐藏结点问题。

仿真结果可以从两个方面来分析，即仿真动画和仿真数据。在 NS2 中演示仿真动画采用 NAM，它将网络仿真过程中拓扑结构、数据包传输情况及队列丢包等以动画形式展现。仿真结束后，可以通过分析产生的数据来对仿真过程中的延迟、丢包、吞吐量等进行分析。

6 仿真扩展与分析

在本实验的基础上，还可自己动手设计不同网络场景下的仿真，分析网络的性能，分析解读实验相关的程序代码。

7 注意事项与资源

为保证仿真代码的正常运行，仿真之前一定要正确安装本手册提供的模块。仿真主要涉及

mUDP 模块和相关的仿真脚本。

实验三 WiMax 仿真

1 实验要求和目的

- 掌握 WiMax 的工作原理
- 利用 NS2 仿真 WiMax 的数据传输和通信过程

2 实验原理和背景知识

802.16 也称 WiMax(World wide Interoperability for Microwave Access, 全球微波接入互操作性), 但 WiMax 是一个商业化的名称, 其产业联盟称为 WiMax 论坛。WiMax 能达到最高70Mbps 的速率, 比 WLAN 要高得多。

WMAN 能有效解决有线方式无法覆盖地区的宽带接入问题, 有较完备的 QoS 机制, 可根据业务需要提供实时、非实时不同速率要求的数据传输服务, 为居民和各类企业的宽带接入业务提供新的方案。本实验仿真拓扑包含了 11 个结点, 其中, 中心结点 0 与其周围的 10 个结点分别进行了通信, 周围结点之间不直接进行通信, 详细拓扑结构如图3.1。

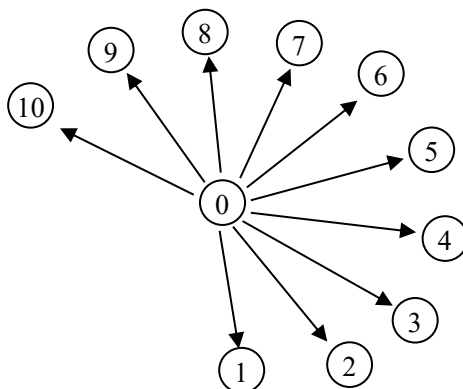


图3.1 WiMax仿真实验网络拓扑图

3 实验环境和模块简介

WiMax 仿真采用的实验平台为 Cygwin + ns-allinone-2.34, 在标准的 ns2 中尚未集成WiMax 模块, 因此在实验前需要进行安装, 详细步骤见 4.1 节。

4 实验步骤与仿真演示

4.1 模块安装

具体安装过程如下:

- (1) 获取 WiMax 的补丁文件(见<http://www.mamicode.com/info-detail-1340005.html>), 文件命名为“WiMax.diff”;

(2) 将获得的补丁文件拷贝到 ns-allinone-2.34 文件夹中，并在 Cygwin 窗口中用“cd”命令进入到 ns-allinone-2.34 目录下，执行打补丁命令：patch -p0 < WiMax.diff。注意：ns-allinone-2.34 必须为未更改过的官方(标准)版本，否则 patch 不一定成功；

(3) 在Cygwin 窗口中用“cd”命令进入到ns-allinone-2.34/ns-2.34 目录下，执行命令：./install 命令，大约编译十几分钟 (具体时间依据机器性能) 即可完成。

4.2 仿真运行

WiMax 仿真的具体过程如下：

(1) 在“home/<用户名>”目录下新建一个目录用于放置运行的仿真脚本，并将不同类型脚本用子目录分开，紧接着将仿真脚本 WiMax.tcl 拷到该目录下。如图 3.2 所示。这里新建一个 WNT 目录，然后在该目录中放置了各实验仿真代码的子目录。

```
Administrator@CHINA-3CDB76914 ~/WNT
$ ls
3.5.1 3.5.2 4.6.2 6.5.1 6.5.2 7.6.1 7.6.2 8.7.1 8.7.2 9.4.2 9.4.3
Administrator@CHINA-3CDB76914 ~/WNT
```

图 3.2 仿真代码位置

(2) 在 Cygwin 中利用 cd 命令转到该新建的目录，紧接着输入：ns WiMax.tcl，回车即可运行，此时无动画显示，如图 3.3 所示；

```
ns send UL-MAP
ns send DL-MAP
ns send UL-MAP
ns send DL-MAP
ns send UL-MAP
ns send DL-MAP
ns send UL-MAP
ns send DL-MAP
ns send UL-MAP
ns send DL-MAP
ns send UL-MAP
Administrator@CHINA-F6A383329 ~/WNT/4.6.2
$ nam: no display name and no $DISPLAY environment variable
```

图 3.3 非NAM 方式运行仿真

(3) 如果想要显示 NAM 仿真动画，则需要启动 XWin，然后按(2)中所述的过程进行仿真即可，如图 3.5 所示。XWin 的启动方式为：在Cygwin 中输入 startxwin.bat，回车即可，如图 3.4 所示；

```
Administrator@CHINA-F6A383329 ~/WNT/4.6.2
$ startxwin.bat
startxwin.bat - Starting on Wind
Administrator@CHINA-F6A383329 ~/WNT/4.6.2
$
```

图 3.4 启动 XWin

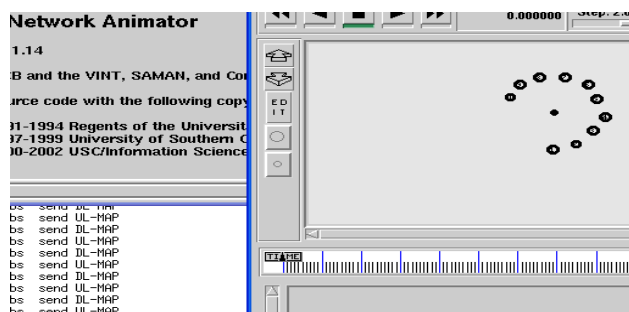


图 3.5 运行 NAM 仿真

5 结果分析与讨论

无线城域网的应用广泛，提供了相对速率更高、范围更广的无线通信能力，本文主要对无线城域网

中的WiMax 模块进行场景仿真。

无线城域网(WiMax)的仿真分析可从两个方面来分析实现结果，即仿真动画和仿真数据。在 NS2 中演示仿真动画的工具采用了 NAM，它将网络仿真过程中拓扑结构、数据包传输情况及队列丢包等以动画的形式展现。仿真结束后，可以通过数据来对仿真过程中的延迟、丢包、吞吐量等问题进行分析。

本实验的结果分析和代码说明详见教材 4.6.2 节，相关电子资源可到本书出版社网站下载。在仿真分析的过程中，需要注意以下两个问题：

- (1) WiMax 在传输过程中存在哪些特点和不足？
- (2) 无线城域网技术和其它同类技术相比有何不同？

6 仿真扩展与分析

在本实验的基础上，可自己动手设计不同网络场景下的仿真，分析网络的性能，分析解读实验相关的程序代码。

7 注意事项与资源

仿真之前必须安装 WiMax 的模块，已有的ns-allinone-2.34 中不包含该模块。仿真主要涉及 WiMax 模块和相关的仿真脚本。

实验四 AODV 仿真

1 实验要求和目的

- 掌握无线自组织网络的组网方式
- 掌握 AODV 路由协议的工作过程
- 利用 NS2 仿真实现 AODV 路由协议

2 实验原理和背景知识

针对 AODV 路由协议仿真，NS2 中已有丰富的模块，能够满足基本的仿真需求，本实验的仿真拓扑结构如图4.1所示，共需用到 13 个移动结点，分别是结点 0~12。其中，结点 8 是源结点，结点 2 是目的结点。

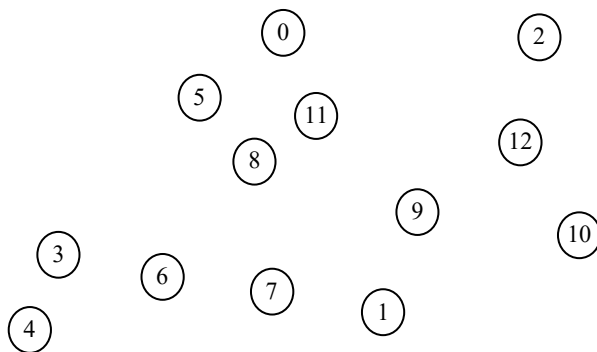


图4.1 AODV/DSR仿真实验拓扑图

3 实验环境和模块简介

AODV 仿真采用的实验平台为Cygwin + ns-allinone-2.34，在标准的ns2 中已集成了相应的模块。

4 实验步骤与仿真演示

具体仿真过程如下：

(1)在“home/<用户名>”目录下新建一个目录用于放置运行的仿真脚本，并将不同类型的脚本用子目录分开，紧接着将仿真脚本 AODV.tcl 和 AODV_topo.scn 拷到该目录下，如图 4.2 所示。这里仿真新建了一个 WNT 目录，然后在目录中放置了各实验仿真代码的子目录。

```

Administrator@CHINA-3CDB76914 ~/WNT
$ ls
3.5.1  3.5.2  4.6.2  6.5.1  6.5.2  7.6.1  7.6.2  8.7.1  8.7.2  9.4.2  9.4.3
Administrator@CHINA-3CDB76914 ~/WNT

```

图 4.2 仿真代码位置

(2)在 Cygwin 中利用 cd 命令转到该目录，紧接着输入：ns AODV.tcl，回车即可运行，此时无动画显示，如图 4.3所示；

```

Administrator@CHINA-F6A383329 ~/WNT/6.5.1
$ ns AODV.tcl
num_nodes is set 13
INITIALIZE THE LIST xListHead
channel.cc:sendUp - Calc highestAntennaZ_ and distCST_
highestAntennaZ_ = 1.5, distCST_ = 35.9
SORTING LISTS ...DONE!

Administrator@CHINA-F6A383329 ~/WNT/6.5.1
$ nam: no display name and no $DISPLAY environment variable

```

图 4.3 非NAM 方式仿真

(3)如果想以 NAM 方式运行仿真动画，则需要启动 XWin，然后按(2)中叙述的方式行仿真即可，如图 4.5 所示。XWin 的启动方式为：在Cygwin 中输入 startxwin.bat，回车即可，如图 4.4 所示；

```

Administrator@CHINA-F6A383329 ~/WNT/6.5.1
$ startxwin.bat
startxwin.bat - Starting on Wind
Administrator@CHINA-F6A383329 ~
$

```

图 4.4 启动 XWin

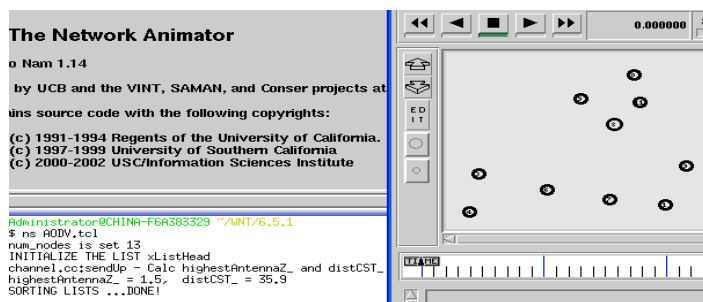


图 4.5 运行 NAM 仿真

5 结果分析与讨论

AODV 路由协议是主动按需路由的代表，在无线自组织网络中应用广泛，受到广泛的研究。

本实验的AODV 仿真可以从两个方面来分析结果，即仿真动画和仿真数据。在NS2 中演示仿真动画的工具采用了 NAM，它将网络仿真过程中拓扑结构、数据包传输情况及队列丢包等以动画的形式展现，相关的仿真动画详见电子资源。仿真结束后，可以通过产生的数据来对仿真过程中的延迟、丢包、吞吐量等问题进行分析。

仿真分析的过程中还需要注意以下两个问题：

- (1) AODV 协议如何进行路由发现和路由建立？
- (2) AODV 与其它按需路由协议相比有何不同？试仿真分析之。

实验五 DSR 仿真

1 实验要求和目的

- 掌握无线自组织网络的组网方式和网络结构
- 掌握 DSR 路由协议的实现过程和原理
- 利用 NS2 平台仿真实现 DSR 协议

2 实验原理和背景知识

为便于仿真实现，本实验采用的仿真拓扑与 AODV 协议仿真拓扑类似。在 NS2 中，已有丰富的模块仿真实现 DSR 路由协议，能够满足常见的仿真需求，本实验采用与图 4.1 相同的仿真拓扑，共需用到 13 个移动结点，分别是结点 0~12。其中，结点 8 是源结点，结点 2 是目标结点。

3 实验环境和模块简介

DSR 仿真采用的实验平台为 Cygwin + ns-allinone-2.34，在标准的 ns2 中已集成了相应的模块。

4 实验步骤与仿真演示

本实验不需要添加模块，在 ns-allinone-2.34 版本的 NS2 中已经集成了所需仿真模块，仅仅编写仿真脚本运行即可。具体仿真过程如下：

(1) 在“home/<用户名>”目录下新建一个目录用于放置运行的仿真脚本，并将不同类型脚本用子目录分开，紧接着将仿真脚本 DSR.tcl 和 DSR_topo.scn 拷到相应的子目录下，如图 5.1 所示。这里新建了一个 WNT 目录，然后在目录中放置了各实验仿真代码的子目录。

```
Administrator@CHINA-3CDB76914 ~/WNT
$ ls
3.5.1 3.5.2 4.6.2 6.5.1 6.5.2 7.6.1 7.6.2 8.7.1 8.7.2 9.4.2 9.4.3
Administrator@CHINA-3CDB76914 ~/WNT
```

图 5.1 仿真代码位置

(2) 在 Cygwin 中利用 cd 命令转到该新建的目录，紧接着输入：ns DSR.tcl，回车即可运行，此时无动画显示，如图 5.2 所示：

```
Administrator@CHINA-F6A383329 ~/WNT/6.5.2
$ ns DSR.tcl
num_nodes is set 13
INITIALIZE THE LIST xListHead
channel.cc:sendUp - Calc highestAntennaZ_ and distCST_
highestAntennaZ_ = 1.5, distCST_ = 35.9
SORTING LISTS ...DONE!
tupan/p802_15_4phy.cc::PD_DATA_request110.2037031(node 8) Invalid PSDU/MPDU le
ch: type = tcp, src = 8, dst = 9, uid = 8, mac_uid = 9, size = 155
tupan/p802_15_4phy.cc::PD_DATA_request110.3498881(node 12) Invalid PSDU/MPDU l
gth: type = DSR, src = 12, dst = -1, uid = 13, mac_uid = 11, size = 159
tupan/p802_15_4phy.cc::PD_DATA_request110.4384311(node 4) Invalid PSDU/MPDU le
ch: type = tcp, src = 4, dst = 6, uid = 19, mac_uid = 20, size = 151
Administrator@CHINA-F6A383329 ~/WNT/6.5.2
$ nam: no display name and no $DISPLAY environment variable
```

图 5.2 非NAM 方式仿真

(3)如果想看 NAM 仿真动画,则需要启动 XWin,然后按(2)中叙述的过程进行仿真即可,如图 5.4 所示。XWin 的启动方式为:在 Cygwin 中输入 startxwin.bat,回车即可,如图 5.3 所示。

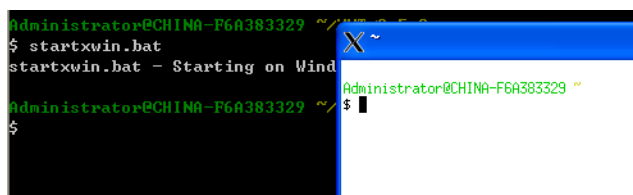


图 5.3 启动 XWin

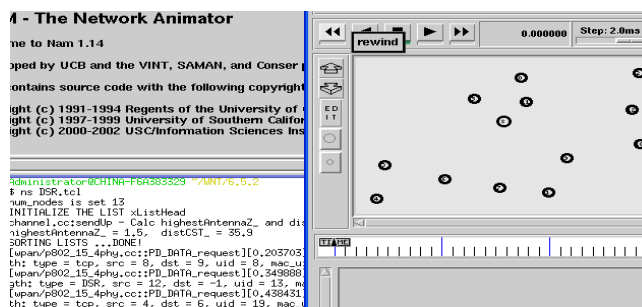


图 5.4 运行 NAM 进行仿真

5 结果分析与讨论

DSR 的仿真可从两个方面来分析实现结果,即仿真动画和仿真数据。在 NS2 中演示仿真动画的工具采用了 NAM,它将网络仿真过程中拓扑结构、数据包传输情况及队列丢包等以动画的形式展现。相关的仿真动画详见电子资源。仿真结束后,可以通过产生的数据来对仿真过程中的延迟、丢包、吞吐量等问题进行分析。

仿真分析的过程中还需要注意以下两个问题:

- (1) DSR 协议如何进行路由发现和路由建立?
- (2) DSR 与其它自组织网络路由协议相比有何不同?试仿真分析之。