Lab 3 Ad Hoc Routing

March 31, 2011

Contents

1 Description of Ad Hoc routing Testbed

In this experiment you will use 3 laptops to configured to communicate in ad hoc mode. The laptops directly communicate with each other in peer-to-peer mode, without going through 802.11g access points. The topology of the ad hoc routing testbed is shown as figure 1.

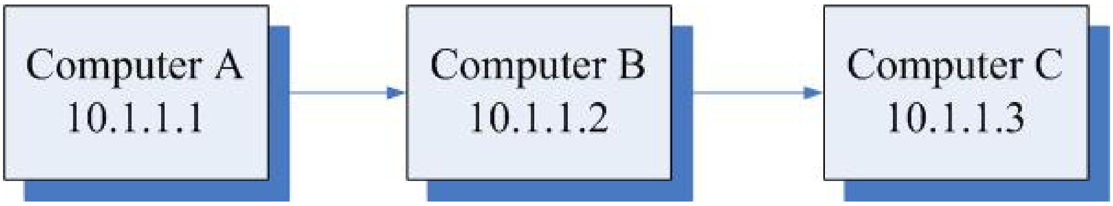

Figure 1: Ad Hoc Routing Setup. Since there are several groups of testbeds, please use 10.1.<group ID>.x as your IP address.

In the above setup, you will use MAC address blocking feature of the *iptables* utility such that computer A cannot communicate directly with computer C and computer C cannot communicate directly with computer A. This is done to simulate a real-world ad hoc network scenario where computer A is out of the transmission range of computer C, but computer B can reach both computer A and computer C (thus it can act as intermediate hop to forward packets from computer A to computer C) within the confines of the lab.

Though we can decrease the transmission power of WLAN card to really simulate a multi-hop scenario, we will leave it to further research.

2 Exercises

Identify the devices you will be using in the exercises (Laptop hardware and software). Login into the laptop. The account is ”sjtu”, and password is ”sjtu” too.

2.1 Initial Experiment Setup

1. Set up all laptops in ad hoc mode as shown in figure 1 as follows.

   For each of the laptops, open a terminal and type the following command to become root(super administrator).

   ```
   sudo su
   ```

   the password is: sjtu

   Type the following command:(**Note**: the essid should be different for different testbed groups. If your testbed group ID is X, please use ”test_groupX”. We take test_group1 as an example in the following code.)

   ```
   ifconfig wlan0 down
   iwconfig wlan0 essid test_group1 mode ad−hoc
   ifconfig wlan0 up
   ```

   Note that the above command should be typed twice. Type the following command for all laptops. Assign IP address in any order.

```
ifconfig wlan0 10.1.<group ID>.x netmask 255.255.255.0
iwconfig wlan0 essid test_group1 mode ad−hoc
```

Now type the following in all laptops to see output.

```
iwconfig wlan0
```

Sample Output:

```
wlan0     IEEE 802.11bg  ESSID:"test_group1"
          Mode:Ad−Hoc  Frequency:2.412 GHz  Cell: C6:89:FE:B3:3C:80
          Tx−Power=20 dBm
          Retry  long limit:7    RTS thr:off    Fragment thr:off
          Encryption key:off
          Power Management:off
```

Please note that the 'Cell' value above must be the same on all laptops to make sure that all of them are part of the same ad-hoc network.

Now type the following in all laptops to see output.

```
ifconfig wlan0
```

Sample Output:

```
wlan0     Link encap:Ethernet  HWaddr c8:3a:35:c3:45:95
          inet addr:10.1.1.1  Bcast:10.1.1.255  Mask:255.255.255.0
          inet6 addr: fe80::ca3a:35ff:fec3:4595/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:39 errors:0 dropped:0 overruns:0 frame:0
          TX packets:20 errors:0 dropped:0 overruns:0 carrier:0
           collisions:0 txqueuelen:1000
          RX bytes:7399 (7.3 KB)  TX bytes:3827 (3.8 KB)
```

Note down the MAC address(Hwaddr field) for each of the laptops.

2. Make sure all laptops are reachable from each other using ping.

   Next use iptables to block computer C from computer A and computer A from computer C as follows:

   On computer A type the following:

```
iptables −A INPUT −m mac −−mac−source <MAC address of computer C>
          −j DROP
```

   On computer C type the following:

```
iptables −A INPUT −m mac −−mac−source <MAC address of computer A>
          −j DROP
```

   Now ping computer C from computer A. You should not be able to send any packets. Next ping computer B from computer A. Then, ping computer B from computer C. In both cases, the ping should be successful.

The ad-hoc routing setup is now ready.

2.2 Evaluation of Route Discovery

Purpose: Preliminary investigation regarding the details of discovering a route to a nonadjacent node.

Instructions: You are advised to answer questions as and when they appear. You may find it harder or answer them at a later stage.

Steps:

1. On computer A, type in

   ```
   route −n
   ```

   Save the results as olsrresult1.1. (Note: you can redirect output to save files ”route -n > olsrresult1.1”).

2. Start the OLSR daemon on all computer by typing in the following on each of them and keep it running (do not press ”ctrl+C” or close the window, if you need to use the terminal, open a new one):

   ```
   olsrd −i wlan0
   ```

   Redirect output to temp.txt on Computer A. Stop after 10 seconds. On computer A, observe the debug messages printed. If you want to make the olsr run in the background, just type: olsrd -i wlan0 -d 0

3. On computer A, type in again

   ```
   route −n
   ```

   Save results as olsrresult1.2.

   Before proceeding, try answering the following questions:

   Question.1 Compare olsrresult1.1 and olsrresult1.2. Do you have routes to both computer B and computer C? If not, why? Is it because OLSR does not function as intended?

   Question.2 From your observations of step-2, enumerate the OLSR control messages transmitted/received by computer A and their respective sources. What is the purpose of each kind of message?(You can answer this part later if you are not sure).

4. On computer A, repeat step 2 but do not stop after 10 seconds. Then open xterm terminal on computer A and try to ping computer C by entering the following command:

   ```
   ping 10.1.<group ID>.3
   ```

5. As soon as you get a ping response, open a third xterm terminal and print

   ```
   route −n
   ```

   Save the result as olsrresult1.3.

   Stop immediately both olsr command (hyper terminal) and ping command (xterm terminal) on computer A.

Question.3 From your observation of step-3, enumerate the OLSR control messages transmitted/received by computer A in temp.txt and their respective sources. What is the purpose of each kind of new messages?

Question.4 What difference do you observe between the routing table entries of olsrresult1.3 and olsrresult1.2? What is the result for this difference?

Question.5 What difference do you observe between the routing table entries of olsrresult1.3 and olsrresult1.2? What is the reason for this difference?

6. Repeat step4. As soon as you get response, stop pinging.

7. Wait for around 1 minute. Then again type on computer A:

   ```
   route −n
   ```

   Save results as olsrresult1.4

   Question.6 What difference do you observe between olsrresult1.4 and olsrresult1.3? What is the reason for this difference, and what is the mechanism that OLSR uses to accomplish it?

8. Stop all programs running on the laptops.

9. Restart the OLSR daemon on computer B, computer A and computer C (in the same order) by typing in the following on each of them:

   ```
   olsr −i wlan0
   ```

   On computer C, observe the debug messages printed.

10. Open xterm on computer A and try to ping computer C.

    ```
    ping 10.1.<group ID>.3
    ```

    Stop olsr on computer C after getting ping response. Observe the debug messages printed by the OLSR daemon on the terminal.

    Question.7 From your observations of step 10, enumerate the OLSR control messages transmitted/received by computer C and their respective sources. Try to explain the meaning of the message.

2.3 Open Question

It is not mandatory, you will get extra points for this part.

1. Is it feasible to simulate a 3 hops chain(or star topology) using 4 laptops in a lab confine? Give your experiment plan about olsr of 3 hops, and test your idea.

2. Is there a feasible way to quickly estimate the throughput or latency of multi-hops transmission.(You can use any software tools under linux) Give you estimation value and your measure method.

A Some basics of Linux

1. If you don't familiar with any command, you could use man command to help you. For example, you don't understand how to use ifconfig or iwconfig, you could simply type:

   ```
   $ man ifconfig
   $ man iwconfig
   ```

2. If you encounter errors like Operation not permitted, you may use sudo before your command.

   ```
   $ sudo ifconfig wlan0 up
   ```

   or you can also into the su mode:

   ```
   $ sudo su
   ```

3. If anything happends unanticipated, you could use Ctrl+C to turn off the current command:

   ```
   wl@wl−desktop:~$ ping 10.1.2.3
   PING 10.1.2.3 (10.1.2.3) 56(84) bytes of data.
   ^C
   −−− 10.1.2.3 ping statistics −−−
   4 packets transmitted, 0 received , 100% packet loss, time 3024ms

   wl@wl−desktop:~$
   ```

B The installation of olsr

If you want to complete lab3 in your own laptops, you need to install OLSR(an implementation of the Link State Routing Protocol) and ipta-bles(a firewall rule specifies criteria for a packet and a target).

1. Download the olsr package from http://www.olsr.org/mediawiki/index.php/Releases.

2. Unpack the package olsrd-0.9.0.3.tar.bz2 downloaded.

3. Get to the path.

   ```
   $ cd Downloads/olsrd-0.9.0.3/
   ```

4. Watch the readme file.

   ```
   $ ls
   $ gedit README
   ```

5. The install

   ```
   $ make
   $ make install
   ```

If the ubuntu lack some pacakage, use "sudo apt-get install " to install those missing pacakeg. After that you can install the olsr.

C About the name of the wireless card

For different distributions of Linux OS or different desktops, you may notice that the name of the wireless card could be different.

You may change the name to your expectation by editing the file /etc/udev/rules.d/70-persistent-net.rules manually:

```
$ sudo gedit /etc/udev/rules.d/70-persistent-net.rules
```

You may open the file similar as follows:

```
# This file maintains persistent names for network interfaces.
# See udev(7) for syntax.
#
# Entries are automatically added by the 75-persistent-net-generator.rules
# file; however you are also free to add your own entries.


# PCI device 0x14e4:0x169b (tg3)
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*", ATTR{address}=="
        00:1d:09:14:73:8a", ATTR{dev_id}=="0x0", ATTR{type}=="1", KERNEL
        =="eth*", NAME="eth0"

# PCI device 0x1814:0x0301 (rt61pci)
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*", ATTR{address}=="
        c8:3a:35:c3:45:95", ATTR{dev_id}=="0x0", ATTR{type}=="1", KERNEL
        =="wlan*", NAME="wlan0"

# PCI device 0x10ec:0x8139 (8139too)
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*", ATTR{address}=="
        50:78:4c:54:15:66", ATTR{dev_id}=="0x0", ATTR{type}=="1", KERNEL
        =="eth*", NAME="eth1"
```

The device name could be modified manualy. Note that different device has different Mac address.