COMP9334 LAB

Session 1, 2012

1 Introduction

This lab session will highlight the use of a traffic generating application and a network emulator to perform network analysis. The traffic generating application we are using is the Distributed Internet Traffic Generator (D-ITG). We are using the NetEm network emulator, found in the Linux 2.6 kernel, to emulate the network. In addition, instead of running the experiments on *physical* network hardware (e.g. physical hosts and routers), we will run them on a number of *virtual machines* that are installed in a single host machine.

1.1 Outline

This instruction will be divided into the following sections.

Section 2 contains information on the virtual machines that we will use in this lab. Section 2.1 will give instructions on how to install and run the virtual machines on a PC or laptop using the free *VMWare Player*. Section 2.2 provides instructions on how to start up the virtual machines in the CSE Pipe or Tuba labs. Depending on whether you perform the tests on a CSE lab machine or on your own machine, you should look at the section that is relevant to you.

Section 3 provides some information regarding the tools used in this lab.

Section 4 contains the detailed instructions for this lab.

Details on how to write and submit the lab report can be found in Section 5.

2 Running the Virtual Machines

2.1 Self-Installation

In this section, we will look at how to install the applications needed to perform the lab session. If you are performing the experiments in the CSE lab (Pipe or Tuba), you can skip this section and proceed on to Section 2.2.

Note: It is advisable to ensure you have sufficient memory to run the virtual machines. I found that a typical PC/laptop with 1 GB of RAM should be enough if your host operating system is Linux and 2 GB is enough if your host operating system is running Windows. Something less may still work but there is no guarantee.

2.1.1 VMWare Player Installation

The following are instructions on how to install VMWare Player into your PC or laptop.

1. Go to the VMWare Player website –

   http://www.vmware.com/download/player/download.html

   Download the latest version of the installation file.

   (a) If the host machine is running Windows, download the file with extension `.exe`.

   (b) If the host machine is running Linux, download the file with extension `.bundle`.

2. Install VMWare Player on the host machine.

   (a) For the `.exe` file, double-click to install.

   (b) For the `.bundle` file, open a terminal console and run the command

   ```
   $ sudo sh <filename>.bundle
   ```

2.1.2 Installing and Running the Virtual Machines

1. Download the file COMP9334VM2012.tar.gz from the COMP9334 course homepage. This file contains the 3 virtual machines we need for our lab session.

2. Extract the files into a directory on your host machine. (Note: If you are running Windows, you would need the WinRAR application. If you are running Linux, do a
   $ tar xvzf COMP9334VM.tar.gz to a suitable directory.)

3. Virtual Machine 1 - **COMP9334_Router**

   (a) Run the VMWare Player application.

   (b) Click on "Open" button. Go to the directory that you have extracted the virtual machine files and open the file `COMP9334_Router.vmx`.

   (c) If you are running the virtual machine for the first time, you may be prompted with the following message: "This virtual machine may have been moved or copied...". Click the "I copied it" button.

   (d) The **COMP9334_Router** virtual machine should run.

   (e) While the Ubuntu operating system on the virtual machine is starting up, enable the interfaces by left clicking and holding on the 2 connection icons on the lower right hand corner of the window. The connection icons, starting from the left, correspond to `eth0` and `eth1` respectively. Select "Connect" and "Host-only" for both connection icons. In this way, interfaces `eth0` and `eth1` will be able to connect to the other virtual machines. You may only need to do this if you are running the virtual machine for the first time.

(f) At the login prompt, enter **comp9334user.** For the password, enter **comp9334.**

(g) To get the mouse pointer out of the virtual machine, press `<Ctrl>-<Alt>`.

4. Virtual Machine 2 - **COMP9334_Client**

(a) Run the VMWare Player application.

(b) Click on "Open" button. Go to the directory that you have extracted the virtual machine files and open the file `COMP9334_Client.vmx`.

(c) If you are running the virtual machine for the first time, you may be prompted with the following message: "This virtual machine may have been moved or copied...". Click the "I copied it" button.

(d) The **COMP9334_Client** virtual machine should run.

(e) While the Ubuntu operating system on the virtual machine is starting up, enable the interface by left clicking and holding on the connection icon on the lower right hand corner of the window. Select "Connect" and "Host-only". In this way, interface `eth0` will be able to connect to the **COMP9334_Router** virtual machine. You may only need to do this if you are running the virtual machine for the first time.

(f) At the login prompt, enter **comp9334user.** For the password, enter **comp9334.**

5. Virtual Machine 2 - **COMP9334_Server**

(a) Run the VMWare Player application.

(b) Click on "Open" button. Go to the directory that you have extracted the virtual machine files and open the file `COMP9334_Server.vmx`.

(c) If you are running the virtual machine for the first time, you may be prompted with the following message: "This virtual machine may have been moved or copied...". Click the "I copied it" button.

(d) The **COMP9334_Server** virtual machine should run.

(e) While the Ubuntu operating system on the virtual machine is starting up, enable the interface by left clicking and holding on the connection icon on the lower right hand corner of the window. Select "Connect" and "Host-only". In this way, interface `eth0` will be able to connect to the **COMP9334_Router** virtual machine. You may only need to do this if you are running the virtual machine for the first time.

(f) At the login prompt, enter **comp9334user.** For the password, enter **comp9334.**

Proceed to Section 2.3 for details of network configuration.

2.2 Running the Virtual Machines in the CSE Lab

The following are instructions on how to start the virtual machines that have been preconfigured in the CSE lab.

1. Log onto the machine.

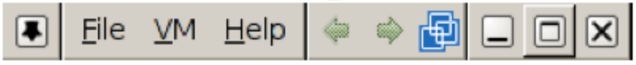
2. Open a console terminal.

3. Run `$vmware`.

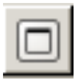
4. You would be asked to enter a number to choose which virtual machine to start. Choose the number associated with COMP9334 Router.

5. *Note:* If the virtual machine starts in full-screen mode, do the following to get it toggle it to a window mode:

   (a) Get the mouse pointer out of the virtual machine by pressing `<Ctrl>-<Alt>-<Shift>`.

   (b) Move the mouse pointer to the center top of the screen. A top menu like 

 will appear.

   (c) Click on the  button to toggle to window mode.

6. Repeat step 3 and choose the numbers associated with `COMP9334_Client` and `COMP9334_Server` each time.

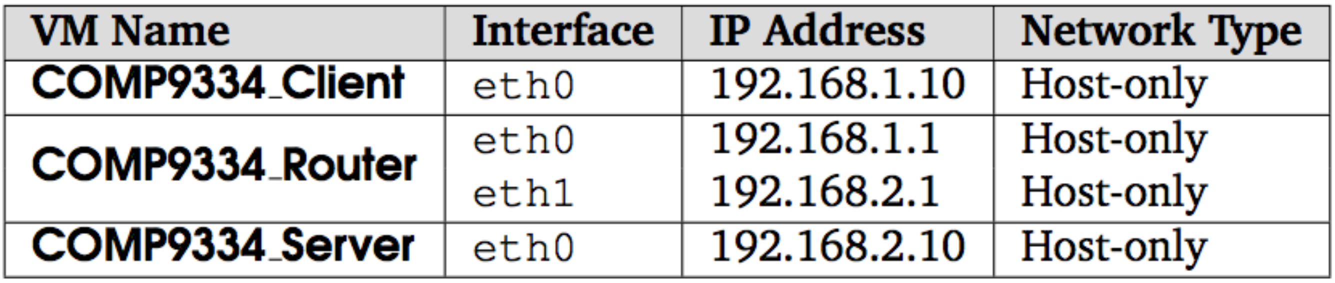
7. To get the mouse pointer out of the virtual machine, press `<Ctrl>-<Alt>-<Shift>`.

8. Once the machines have booted up, at the login prompt, enter **comp9334user.** For the password, enter **comp9334.** The usernames and passwords are the same for all the machines.

Proceed to Section 2.3 for details of network configuration.

2.3 Network Configuration of the Virtual Machines

The following table shows the network configuration of the Virtual Machines.



VM Name	Interface	IP Address	Network Type
COMP9334_Client	eth0	192.168.1.10	Host-only
COMP9334_Router	eth0 eth1	192.168.1.1 192.168.2.1	Host-only Host-only
COMP9334_Server	eth0	192.168.2.10	Host-only

Figure 1 shows how the interfaces are connected together.

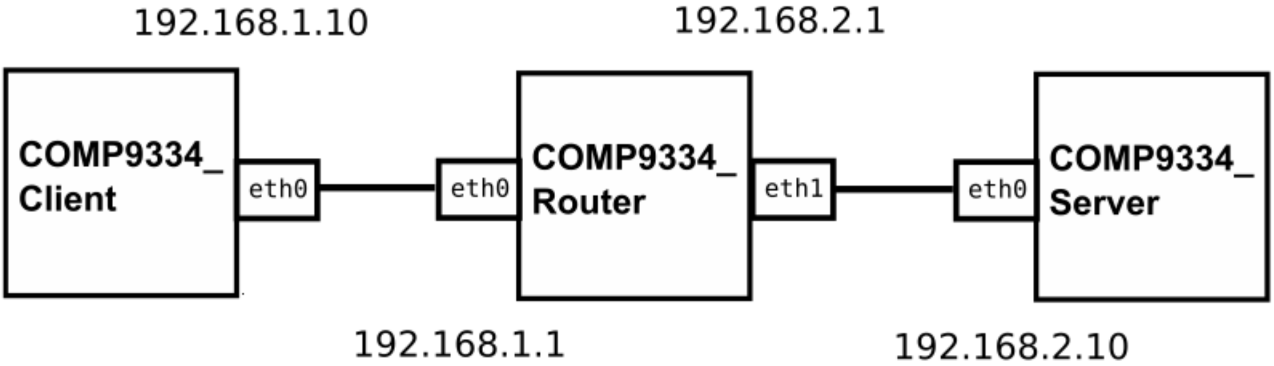


Figure 1: Connection of the virtual machines.

2.4 Testing the Virtual Machine Connections

To test the connections of the virtual machines, when all the machines are turned on, perform a ping by issuing the following command on the **COMP9334_Client** virtual machine:

```
$ ping 192.168.2.10
```

If the ping is not successful, check the connection status of the interfaces by clicking and holding the left mouse button on the corresponding connection icons.

3 Software Tools

3.1 D-ITG

The D-ITG is an application that is able to generate traffic that reproduces packet inter-arrival times and packet sizes of many common distributions, e.g. exponential, normal, pareto... It is also able to generate packets at different layers of the networking protocols, e.g. application, network and transport.

The following instructions assume that you have successfully logged into each of the virtual machines.

3.1.1 Going to the Right Directory

D-ITG has been installed on the **COMP9334_Client** and **COMP9334_Server.** To run the program, go to the directory `$HOME/Comp9334Lab/` on both virtual machines with the following command

```
$cd; cd Comp9334Lab
```

3.1.2 Synchronizing the Time

To have a relatively accurate calculation of the delay, synchronization of the source (**COMP9334_Client** virtual machine) and destination (**COMP9334_Server** virtual machine) times is important. We do this by synchronizing with the ntp server that is running on the **COMP9334_Router** virtual machine. This is done by issuing the following commands on the respective virtual machines.

For the **COMP9334_Client** virtual machine:

```
$ sudo ntpdate -b 192.168.1.1
```

For the **COMP9334_Server** virtual machine:

```
$ sudo ntpdate -b 192.168.2.1
```

If you get the message "no server suitable for synchronization found", this is most likely because the ntp server on the **COMP9334_Router** virtual machine is in the midst of synchronizing against itself (since it is not connected to the Internet). It usually takes less than a minute for this to be completed.

Note: It is a good idea to do this before you perform every experiment using D-ITG as the clock in the virtual machine is known to drift over time. You may have to run the command for a couple of times to get the clock to stabilize to a reasonable accuracy.

If you are unable to synchronize with the ntp server, you would need to compute the round trip delay instead of the one-way delay, as shown in Example 3.

3.1.3 Preparing to Receive Traffic

To run D-ITG, the server must be running at the destination. In our virtual machine configuration, we will run the D-ITG server at the **COMP9334_Server** virtual machine by issuing the following command:

```
$ ./ITGRecv
```

The **COMP9334_Server** virtual machine will be ready to receive any packets from the D-ITG client application.

3.1.4 Sending Traffic

The following are some examples of basic commands to send traffic from **COMP9334_Client** (192.168.1.10) via **COMP9334_Router** (192.168.1.1 and 192.168.2.1) to **COMP9334_Server** (192.168.2.10). All the following commands are issued on **COMP9334_Client.**

Example 1

```
$ ./ITGSend -a 192.168.2.10 -sp 9400 -rp 9500 -C 100 -c 500 -t 20000 \
-x const_100_500.dat
```

This command sends traffic to IP address `192.168.2.10` on source port `9400` and destination port `9500` with a *constant* packet rate of `100` packets per second and a *constant* packet size of `500` Bytes. The traffic runs for `20000ms` (i.e. 20s) and the results are written into the log file `const_100_500`.dat at the destination.

Example 2

```
$ ./ITGSend -a 192.168.2.10 -sp 9400 -rp 9500 -E 100 -u 500 1000 -t 20000 \ -x exp 100 u5001000.dat
```

This command sends traffic to IP address 192.168.2.10 on source port 9400 and destination port 9500 with an *exponentially* distributed packet inter-departure time characterized by an average packet rate of 100 packets per second and packet sizes that are *uniformly* distributed between 500 Bytes and 1000 Bytes. The traffic runs for 20000ms (i.e. 20s) and the results are written into the log file exp 100 u5001000.dat at the destination.

Example 3

```
$ ./ITGSend -a 192.168.2.10 -sp 9400 -rp 9500 -C 100 -c 500 -t 20000 \
-m RTTM -l const_100_500.dat
```

This command is similar to Example 1 except that the round trip delay is computed and the log file is saved at the **COMP9334_Client** virtual machine (source). This is typically used if you are unable to synchronize the source and destination.

3.1.5 Interpreting the Results

To interpret the results, first close `./ITGRecv` at the destination by pressing `<Ctrl>-C`. Next, decode the log file by using

```
$ ./ITGDec <log file> -v
```

3.1.6 Additional Information

- D-ITG Website: http://www.grid.unina.it/software/ITG/
- D-ITG Manual: http://www.grid.unina.it/software/ITG/codice/D-ITG-2.7.0-Beta2-manual.pdf

3.2 NetEm Network Emulator

NetEm is a network emulator module that is enabled in the Linux 2.6 kernel. It uses the `tc` command that is part of the iprouter2 package of tools. By running the command on the network interface, it is possible to add delay, packet loss, packet duplication and rate limitations to emulate different network conditions.

The following instructions assume that you have successfully logged into the **COMP9334_Router** virtual machine, which we can use to emulate a Wide Area Network with different characteristics.

3.2.1 Adding Network Delay

To add network delay that is normally distributed with mean of `100ms` and standard deviation of `20ms`, issue the following command to **COMP9334_Router:**

```
$ sudo tc qdisc add dev eth1 root netem delay 100ms 20ms distribution normal
```

Note that this will introduce a delay to the *forward* path only. That is, traffic travelling from **COMP9334_Client** to **COMP9334_Server.** To introduce delay to the *reverse* path, the command above will be issued with `eth1` replaced by `eth0`.

To remove the parameters that have been added to the interface `eth1`, issue the following command:

```
$ sudo tc qdisc del dev eth1 root netem
```

You can test the above configuration using `ping` command or the D-ITG application.

3.2.2 Introducing Bandwidth Limitation

We can make use of the Hierarchy Token Bucket (HTB) to limit the bandwidth.

```
$ sudo tc qdisc add dev eth1 root handle 1: netem delay 100ms

$ sudo tc qdisc add dev eth1 parent 1:1 handle 10: htb default 1

$ sudo tc class add dev eth1 parent 10: classid 10:1 htb rate 256kbit
```

In the above example, we first introduce a delay of `100ms` to `eth1`. This is followed by limiting the rate to `256kbps`. This will emulate **COMP9334_Router** as an intermediate network will a delay of `100ms` and bandwidth of `256kbps`.

You can test the above configurations using the D-ITG application.

3.2.3 Additional Information

- Netem Website: http://www.linuxfoundation.org/en/Net:Netem
- Netem Manual: http://tcn.hypert.net/tcmanual.pdf

4 Lab Instructions

We will now perform some experiments using the set up we have described. It consists of 3 machines connected by ethernet interfaces where the intermediate router is emulating a Wide Area Network (WAN) connection using the NetEm module. In practice, this could be done using physical machines, but for the purpose of this lab, we will be using the VMWare virtual machines running inside a single host system.

4.1 Hypothetical WAN Setup

We are interested in the performance of traffic traversing between two end hosts through a WAN. A WAN typically consists of multiple intermediate networks/links with different bandwidth and delay characteristics. This is a result of the physical hardware (e.g. T-1 line, OC-1 line, router speed) as well as other traffic traversing the WAN.

4.2 Virtual Machine Emulation Setup

Often, the network characteristic of a WAN can be modeled as by the bandwidth of the bottleneck link in it, as well as the total delay needed for a packet to travel from the source host to the destination host. This is done using the NetEm network emulator module that we will run on the **COMP_Router** virtual machine. Using the emulator, we can set different bandwidth and delay parameters for the WAN.

4.3 Experiment

The following steps assume you have successfully logged into the 3 virtual machines.

1. On the **COMP9334_Router** virtual machine,

   (a) Introduce a network delay of 40ms normally distributed with standard deviation of 5ms. This emulates the total forward delay in the WAN.

   ```
   $ sudo tc qdisc add dev eth1 root handle 1: netem delay 40ms 5ms \
   distribution normal
   ```

   (b) Limit the bandwidth to 1.536 Mbps to emulate a WAN with a T1 line bottleneck link.

   ```
   $ sudo tc qdisc add dev eth1 parent 1: handle 10: htb default 1
   $ sudo tc class add dev eth1 parent 10: classid 10:1 htb rate 1536kbit
   ```

(c) Create a queue buffer of size 20kB.

```
$ sudo tc qdisc add dev eth1 parent 10:1 handle 20: bfifo limit 20kB
```

(d) The same procedures are repeated for the reverse path.

```
$ sudo tc qdisc add dev eth0 root handle 1: netem delay 40ms 5ms \
distribution normal
$ sudo tc qdisc add dev eth0 parent 1: handle 10: htb default 1
$ sudo tc class add dev eth0 parent 10: classid 10:1 htb rate 1536kbit
$ sudo tc qdisc add dev eth0 parent 10:1 handle 20: bfifo limit 20kB
```

This will emulate a WAN with a T1 line bottleneck link (approx. 1.536 Mbps) and one-way mean delay of 40ms. We apply the commands to both interfaces to get symmetric parameters for both the forward (`eth1`) and reverse (`eth0`) paths.

2. On the **COMP9334_Server** virtual machine,

(a) Synchronize to the router.

```
$ sudo ntpdate -b 192.168.2.1
```

(you should do this a few times, until the offset is small)

(b) Run ITGRecv to listen for packets from the source.

```
$ cd ~/Comp9334Lab; ./ITGRecv
```

3. On the **COMP9334_Client** virtual machine,

(a) Synchronize to the router.

```
$ sudo ntpdate -b 192.168.1.1
```

(again, do this a few times to get a small offset.)

(b) Run ITGSend to send packets to the destination host.

```
$ cd ~/Comp9334Lab
$ ./ITGSend -a 192.168.2.10 -sp 9400 -rp 9500 -C 100 -c 1500 \
-t 50000 -x const-10kB-run1.dat
```

The **COMP9334_Client** virtual machine will send packets of size 1460 bytes at a *constant* rate of 100 packets per second, running for 60s. The log file, `const-10kB-run1.dat` will be saved at the **COMP9334_Server** virtual machine.

4. On the COMP9334 Server virtual machine,

(a) Extract the results from the test.
Press `<Ctrl>-C` to end `ITGRecv`.

```
$ ./ITGDec const-20kB-run1.dat -v
```

Note: As the virtual machines are in *Non-persistent* mode, i.e. all changes are not saved when they are powered off, you might like to redirect this output to a text file by running e.g. `$ ./ITGDec const-20kB-run1.dat -v > const-20kB-run1.txt`. You can transfer the file to an external usb thumb drive later.

(b) Take particular note of the `Average delay`, `Average bitrate` and `Packets dropped` results from the output. You will need them to present and discuss your results.

5. **Repeating the experiment**
As you have learned in your lecture, statistical methods are often used to verify the consistency and confidence of the experimental results. This is usually done by repeating an experiment for a number of independent runs. You should now repeat steps 2–4 **six** times. If you wish to save the log files for all the experiments, make sure that you give a different file name for each run of the experiment, e.g. `const-10kB-run2.dat`.

6. **Stochastic Traffic Characteristics**
Real world traffic is never deterministic; i.e. they do not have constant packet rate and packet size. We will now introduce some randomness into the traffic characteristic by using two types of probability distributions to model the packet inter-arrival time. We will keep the mean packet departure rate constant at 100 packets per second so as to make meaningful comparisons with the deterministic traffic.

Exponential Inter-Arrival Time

(a) Restart `ITGRecv` on **COMP9334_Server** as in step 2.

(b) On **COMP9334_Client,** send packets with exponential inter-departure time using

```
$ ./ITGSend -a 192.168.2.10 -sp 9400 -rp 9500 -E 100 -c 1500 \
-t 50000 -x exp-20kB-run1.dat
```

The `-E` option specifies to the traffic generator to send packets at exponential inter-departure time, with a mean packet rate of 100 packets per second.

(c) Extract the results on **COMP9334_Server** as in step 4.

(d) Again, repeat the experiment six times to get six independent runs.

Pareto Inter-Arrival Time

As you have learned in your lecture, the Pareto probability distribution is characterized by two parameters: shape γ and scale δ. The mean of the distribution is defined by

$$\frac{\gamma\delta}{\gamma - 1}$$

In the D-ITG traffic generator, traffic with Pareto inter-arrival time (expressed in milliseconds) can be generated by specifying the appropriate γ and δ values.

(a) Restart `ITGRecv` on **COMP9334_Server** as in step 2.

(b) On **COMP9334 Client,** send packets with Pareto inter-departure time using

```
$ ./ITGSend -a 192.168.2.10 -sp 9400 -rp 9500
-V <shape> <scale> -c 1500 -t 50000 -x par1-20kB-run1.dat
```

Send traffic with a shape (i.e. γ) value of **2.** (**Hint**: To get the appropriate scale (δ) value, note that the shape and scale need to define a Pareto distribution that has a mean inter-departure time of 1000 × 1/100 = 10ms.)

(c) Extract the results on **COMP9334_Server** as in step 4.

(d) Again, repeat the experiment six times.

(e) Next, send the traffic with shape value of 1.2. Run the same experiment six times and extract the results as before.

7. **Changing the queue buffer size**
On the **COMP9334_Router** virtual machine, run

```
$ sudo tc qdisc change dev eth1 parent 10:1 handle 20: bfifo limit 25kB
$ sudo tc qdisc change dev eth0 parent 10:1 handle 20: bfifo limit 25kB
```

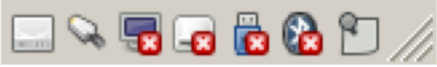
This will cause the queue buffer size to increase to 25kB.

8. Repeat all the experiments (constant, exponential, Pareto with shape=2, Pareto with shape=1.2) for queue buffer sizes of 20kB, 25kB and 30kB.

9. **How do I get the data out of the virtual machine?**
You will need a usb thumb drive.

(a) Insert the thumb drive into the host machine. (On the CSE machines, the usb ports can be found behind the Mac monitor.)

i. *Own personal PC/laptop:* On the **COMP9334_Server** window, enable the thumb drive detection by clicking on the relevant option of the "Devices" dropdown menu at the top of the window.

ii. *CSE lab machines:* At the bottom the **COMP9334_Server** window, you will be able to see a row like this: . Click on the icon and choose "Connect" to enable the inserted usb drive.

(b) Once drive is detected, run
`$ sudo mount /dev/sdb1 /media/usb/` on the virtual machine. The drive would be mounted on the `/media/usb` directory.

(c) Transfer the files to the drive by running

```
$ sudo cp
  ~/Comp9334Lab/*.txt /media/usb/
```

(We assume here that you have directed the results to a number of `.txt` files).

(d) Unmount the usb drive by running

```
$ sudo umount /media/usb
```

(e) You may now unplug the usb drive from the host machine.
(**Note**: The original log files are binary files. They require `ITGDec` decode to view. There is

no necessity to save them. It is better to redirect the output to a text file and save that instead.)

10. Plot the graphs of the **average delay, effective bandwidth, packet dropped rate** and **utilization** for the different queue buffer sizes. You should use what you have learned in the lectures to include not just the mean of the six independent experimental runs, but also the 95% confidence intervals. If your graph plotting application supports errorbars, the confidence intervals can be included directly on the graphs. Otherwise, you should include the results in a table.

11. Discuss the significance of the results you have gotten in this lab.

5 Lab Report

For lab assessment, you are required to submit a report. The total marks for this report is 25 marks. This component is an individual work. You are expected to complete the report on your own.

5.1 Report Format

This section provides a basic format for the lab report that you are required to submit after you have performed these lab exercises. Note that this format only serves as a guide and you are free to submit your report in any appropriate format so long as you detail what you have done, include the results and discuss them.

1. **Title Page**
   This should include the course name, course code, your name and student id.

2. **Introduction**
   Provide a brief introduction of the experiment you are performing. It need not be longer than a paragraph.

3. **Results**
   Along with the Discussion section, this is the main core and most important part of the report. You include the results – in tables, graphs or plain description to show the results you have acquired.

4. **Discussion**
   The other main part of the report. You will need to discuss the results that you have gotten, to show that you are able to relate them to what you have learned in the lectures.

5. **Conclusion**
   A short section to conclude your report.

5.2 Submission

Submit your report as a file called "**lab.pdf**". The system will only accept this file name. Make sure you are in the same directory as your work, then do the following:

- When you are ready to submit, at the bash prompt type ***9334***;
- Next, type: ***give cs9334 lab lab.pdf*** (You should receive a message stating the result of your submission).

  Your report must be submitted by **23:59:59 Friday 13 April 2012**. Two marks per day will be subtracted for late submission. Submissions which are overdue for more than five days will receive ZERO mark.

5.3 Lab Related Queries

Please consult Lab-coordinator Lukas regarding lab related queries (Monday 11.00 - 12.00, K17-501(17). You can also send your query via class email account cs9334@cse.unsw.edu.au or request for appointment to kail@cse.unsw.edu.au (TEL: +61 2 9385 7225) if you can come to consultation.