

TOPOLOGY SETUP FOR TEACUP

Router Setup:

1. Check which interfaces are managed by network-manager :

```
$ nmcli dev status
```

2. First disable all interfaces by using the command

```
$ ifdown --all
```

3. Configure interfaces :

```
$ sudo gedit /etc/network/interfaces
```

Edit:

```
auto eth0
```

```
iface eth0 inet static
```

```
address 192.168.50.2
```

```
netmask 255.255.255.0
```

```
network 192.168.50.0
```

```
broadcast 192.168.50.255
```

```
gateway 192.168.50.1
```

```
auto eth1
```

```
iface eth1 inet static
```

```
address 172.16.10.2
```

```
netmask 255.255.0.0
```

```
network 172.16.0.0
```

```
broadcast 172.16.255.255
```

```
gateway 172.16.10.1
```

4. To enable interfaces give command:

```
$ sudo ifup -all
```

5. Set iptable rules

```
$ sudo su
```

```
$ iptables -t nat -A POSTROUTING -o eth1 -j MASQUERADE
```

```
$ iptables -A FORWARD -i eth1 -o eth0 -m state --state  
RELATED,ESTABLISHED -j ACCEPT
```

```
$ iptables -A FORWARD -i eth0 -o eth1 -j ACCEPT iptables -A  
FORWARD -i eth0 -j ACCEPT
```

```
$ iptables -A FORWARD -m state --state RELATED,ESTABLISHED -j  
ACCEPT
```

```
$ iptables-save > /etc/iptables.rules
```

6. Enable packet forwarding sysctl.conf file:

```
$ sudo gedit /etc/sysctl.conf
```

```
EDIT: net.ipv4.ip-forward=1
```

7. Restart the network services by using following command

```
$ sudo /etc/init.d/networking restart
```

Host1 Setup (client):

1. Disable all interfaces

```
$ sudo ifdown --all
```

2. Configure interfaces :

```
$ sudo gedit /etc/network/interfaces
```

```
Edit:
```

```
auto eth0
```

```
iface eth0 inet static
```

```
address 192.168.50.1
```

```
netmask 255.255.255.0
```

```
network 192.168.50.0
```

```
broadcast 192.168.50.255
```

```
gateway 192.168.50.2
```

3. Enable all interfaces

```
$ sudo ifup -all
```

4. Restart network services:

```
$ sudo /etc/init.d/networking restart
```

Host2 Setup (server):

1. Disable interfaces

```
$ sudo ifdown --all
```

2. Configure interfaces as follow:

```
$sudo gedit /etc/network/interfaces
```

Edit:

```
auto eth0
```

```
iface eth0 inet static
```

```
address 172.16.10.1
```

```
netmask 255.255.0.0
```

```
network 172.16.0.0
```

```
broadcast 172.16.255.255
```

```
gateway 172.16.10.2
```

3. Enable interfaces

```
$ sudo ifup -all
```

4. Restart network services

```
$ sudo /etc/init.d/networking restart
```

5. Check connectivity by pinging from both sides.

INSTALLATION REQUIRMENTS FOR TEACUP

STEP 1: ON HOSTS (CLIENT AND SERVER) :

1. TEACUP is based on Fabric. So to install fabric commands are:

\$ sudo Install Python - 2.7

2. Pip is a package management system used to install and manage software packages written in python. So Install pip using following command:

\$ sudo apt-get install python-pip

\$ sudo apt-get install python-dev

3. Install fabric using pip command:

\$ sudo pip install fabric

4. SPP is a tool that computes RTT estimates based on tcpdump files collected at both ends of a path. First download **spp-0.3.6.tar.gz** and copy it in teacup folder.

- Extract spp-0.3.6.tar.gz using following command:

\$ sudo tar -xvzf spp-0.3.6.tar.gz

- Go to cd spp-0.3.6 folder using following command :

\$ cd spp-0.3.6

- Now run make command :

\$ sudo make

- If error comes like **Error : pcap.h: No such file or directory** then install

\$ sudo apt-get install libpcap0.8-dev

and again run make commad:

\$ sudo make installation

5. Now Install traffic generaator tools: iperf, httpperf and nttcp.

- Install iperf, httpperf and nttcp from the packages provided in the TEACUP tarball.

- For installing iperf follow the steps as given below :

```
cd teacup-1.0/tools  
$ sudo tar -xvzf iperf-2.0.5-mod.tar.gz  
cd iperf-2.0.5-mod/  
$ sudo ./configure  
$ sudo make  
$ sudo make install
```

- For installing httpperf follow the steps as given below :

```
$ cd teacup-1.0/tools  
$ sudo tar -xvzf httpperf-0.8-mod.tar.gz  
$ cd httpperf-0.8-mod/  
$ sudo ./configure
```

If error comes after running make then modify the make file as given below:

```
INCLUDES = -I$(top_srcdir)/include -I$(top_builir) -I$ (top_srcdir) -I$  
(top_srcdir)/lib -I/usr/kerberos/include
```

```
to INCLUDES =-I$(top_srcdir)
```

then again apply make :

```
$ sudo make  
$ sudo make install
```

- For installing nttcp follow the steps as given below :

```
$ cd teacup-1.0/tools  
$ sudo tar -xvzf nttcp-1.47-mod.tar.gz  
$ cd nttcp-1.47-mod/  
$ sudo make
```

6. INSTALL R AND PDFJAM

These are needed for plotting graphs.

```
$ sudo apt-get install r-base
```

```
$ sudo apt-get install pdffjam
```

7. INSTALL OTHER REQUIRED PYTHON MODULES

```
$ sudo pip install pexpect 3-1
```

8. INSTALL TCP LOGGERS

These are needed on the experiment hosts (client and server).

Download kernel source (linux-3.17) from Internet linux-3.17-4.tar.xz.

(Kernel version for all the PCs should be same and updated. If it is not linux -3.17 version then apply following commands to upgrade to 3.17 . If kernel is 3.17 then no need to do follow below procedure)

- For 32 bit architecture system:

```
$ sudo wget http://kernel.ubuntu.com/~kernel-
```

```
ppa/mainline/v3.17-utopic/linux-headers-3.17.0-031700-
```

```
generic_3.17.0-031700.201410060605_i386.deb
```

```
$ sudo wget http://kernel.ubuntu.com/~kernel-
```

```
ppa/mainline/v3.17-utopic/linux-headers-3.17.0-031700_3.17.0-  
031700.201410060605_all.deb
```

```
$ sudo wget http://kernel.ubuntu.com/~kernel-
```

```
ppa/mainline/v3.17-utopic/linux-image-3.17.0-031700-  
generic_3.17.0-031700.201410060605_i386.deb
```

```
$ sudo dpkg -i linux-headers-3.17.0-*.deb linux-image-  
3.17.0-*.deb
```

- For 64-bit system, run:

```
$ sudo wget http://kernel.ubuntu.com/~kernel-
```

```
ppa/mainline/v3.17-utopic/linux-headers-3.17.0-031700-  
generic_3.17.0-031700.201410060605_amd64.deb
```

```
$ sudo wget http://kernel.ubuntu.com/~kernel-ppa/mainline/v3.17-utopic/linux-headers-3.17.0-031700\_3.17.0-031700.201410060605\_all.deb
```

```
$ sudo wget http://kernel.ubuntu.com/~kernel-ppa/mainline/v3.17-utopic/linux-image-3.17.0-031700-generic\_3.17.0-031700.201410060605\_amd64.deb
```

```
$ sudo dpkg -i linux-headers-3.17.0-*.deb linux-image-3.17.0-*.deb
```

Now restart system using following command:

```
$sudo reboot
```

Copy linux-3.17-4.tar.xz kernel code in /usr/src and untar it using following command :

```
$ sudo cp /home/aishwarya/Downloads/linux-3.17-4.tar.xz /usr/src
```

```
$ cd /usr/src
```

```
$ sudo tar -xvf linux-3.17.4.tar.xz
```

```
$ cd linux-3.17.4
```

Now download web10g-userland-2.0.9 from web10g.com website. And extract web10g-userland-2.0.9.tar.xz by following command:

```
$ sudo tar -xvf web10g-userland-2.0.9.tar.xz
```

Now apply web10g-0.11-3.17.diff patch to linux-3.17.4 kernel using following command :

```
$ sudo patch -p1 < /home/aishwarya/Downloads/web10g-userland.0.9/web10g-0.11-3.17.diff
```

9. STEPS TO UPDATE UBUNTU KERNEL

```
$ sudo add-apt-repository ppa:kernel-ppa/ppa
```

```
$ sudo apt-get update
```

-Check whether you added kernel properly or not using following command :

\$ sudo apt-cache showpkg linux-headers

10. STEP TO ENABLE TCP_ESTAT

Copy existing .config to new kernel directory and run make oldconfig.

So existing .config file is in /boot folder

**\$ sudo cp /boot/config-3.17.4-28-generic
/usr/src/linux-3.17.4**

\$ sudo make oldconfig

\$ sudo make menuconfig

If error comes: ERROR: Unable to find the ncurses libraries or the required header files, then Install ncurses libraries by command:

\$ sudo apt-get install ncurses-dev

Then run the command :

\$ sudo make menuconfig

Further, it will gives options.

Now Go to :

1. **"Networking support"->"Networking options"**
2. Enable **"TCP: Extended TCP statistics(TCP ESTATS) MIB"**.

This will automatically enable **"TCP: ESTATS netlink module"**

Go to :

1. **"General setup"->"Local version"**
2. change the string to **'-vanilla-web10g'**

11. Build the kernels and modules

(Note: DON'T DO make mrproper. It cleans all the changes made.)

Steps to build module are as follows :

\$ sudo make

If *error* comes while doing make do the following thing :

The module built will not work, because some symbols are not exported.

Modify **tcp_input.c** to **export sysctl_tcp_timestamps and sysctl_tcp_sack:**

```
int sysctl_tcp_timestamps __read_mostly = 1;  
EXPORT_SYMBOL(sysctl_tcp_timestamps);  
  
int sysctl_tcp_window_scaling __read_mostly = 1; int  
sysctl_tcp_sack__read_mostly = 1; EXPORT_SYMBOL(sysctl_tcp_sack);
```

Do the following steps :

```
$ sudo make
```

```
$ sudo make INSTALL_MOD_STRIP=1 modules_install
```

```
$ sudo make install
```

added the new kernel to the boot manager:

Check in /boot if it is grub2 or grub : then run the following command:

```
$ sudo grub-mkconfig -o /boot/grub/grub.cfg
```

12. Install Web10g:

```
$ cd /home/student/web10g (specify the path of web10g)
```

```
$ sudo apt-get install libmnl-dev
```

```
$ sudo ./configure
```

```
$sudo make
```

```
$ sudo make install
```

*the commands to check if **web10g** is installed or not*

```
$ modprobe tcp_estats_nl
```

```
$ web10g-listconns #list connections
```

```
$ web10g-readvars 1
```

```
$ web10g-logger -i 500
```

lighttpd is open source web server install it by using command :

\$ sudo apt-get install lighttpd

13. check tcpdump, tc, ethtool are all installed on all the PCs

ON BOTH HOSTS AND ROUTER :

Turn on ssh daemon :

\$ sudo apt-get install openssh-server

take backup of sshd_config file:

\$ sudo cp /etc/ssh/sshd_config

\$ /etc/ssh/sshd_config.factory-defaults

\$ sudo chmod a-w /etc/ssh/sshd_config.factory-defaults

check ssh daemon is running or not :

ps -A | grep sshd

If no line appears it means it's not running check whether ssh is listening for incoming connections :

\$ sudo ss -lnp | grep sshd

try logging in from your own computer :

\$ ssh -v localhost

\$ exit

Disable boot splash:

\$ sudo gedit /etc/default/grub

change: **GRUB_CMDLINE_LINUX_DEFAULT="quiet splash"** to **GRUB_CMDLINE_LINUX_DEFAULT="quiet"**

\$ sudo update-grub

Turn off automatic updates:

\$ sudo gedit /etc/apt/apt.conf.d/10periodic

change:

From **APT::Periodic::Update-Package-Lists "1";**

to **APT::Periodic::Update-Package-Lists "0";**

NTP(Network Time Protocol) server setting.

Server side:

Disable ntpdate service:

\$ sudo update-rc.d -f ntpdate remove sudo apt-get install ntp

Edit ntp.conf to setup standalone NTP server :

\$ sudo gedit /etc/ntp.conf

comment all pool servers and edit:

server 192.168.50.1 this should be the ip address of your-server: save and exit

\$ sudo service ntp restart

Client side:

\$ sudo apt-get install ntp

\$ sudo gedit /etc/ntp.conf :

edit: server 192.168.50.1:

\$ sudo service ntp restart

Turn off firewall :

\$ sudo ufw disable

TCP segmentation offloading is disabled (tso off) automatically when we run experiment using TEACUP.

Also no need to enable/disable ECN, adding delay using NetEm, specify buffer size all the things will be done by TEACUP.

HOW TO START WITH TEACUP-1.0

1. Download tar file of teacup-public-1.0.tar.gz
2. mkdir Documents/TEACUP
3. cp
/home/hostname/Downloads/teacup1.0.tar.gz/home/hostname/Documents/TEACUP/
4. cd /home/hostname/Documents/TEACUP/
5. tar -xvzf teacup-public-1.0.tar.gz
6. mkdir -p experiment
7. cp teacup-1.0/example_configs/config-scenario1.py /experiment/config.py
8. cp teacup-1.0/example_configs/run.sh /experiment/
9. cp teacup-1.0/fabfile.py /experiment/
10. cd experiment/
11. Now make changes in config file as per your set up of testbed.
12. env.user='your-username'
13. env.password='your-password'
14. put the path of your teacup directory
TPCONF_script_path='/home/hostname/Documents/TEACUP/teacup-1.0'
15. SAVE and EXIT
16. To check config file correctness
17. fab check_config
18. ./run.sh

Following error may come while running TEACUP then solutions are as below :

1. Fatal error: Name lookup failed for router

Solution : make changes in config.py as below :

```
TPCONF_router = ['192.168.50.2', ]
```

```
TPCONF_hosts = [ '192.168.50.1', '172.16.10.1', ]
```

```
TPCONF_host_internal_ip = {  
'192.168.50.2': ['172.16.10.2','192.168.50.2'],  
'192.168.50.1': ['192.168.50.1'],  
'172.16.10.1': ['172.16.10.1'],  
}
```

2. If it is not able to run sudo commands

Solution :

ssh-keygen (on router and other host)

ssh-copy-id -i [root@ip_address](#)

ssh [root@ip_address](#)

Now you should be able to login without any password.

In fabfile.py, add following line :

```
env.hosts = [ '192.168.50.1','172.16.10.1' , '192.168.50.2']
```

On router and host2 run command

```
$ sudo su
```

```
$ visudo
```

```
$ hostname ALL =(ALL) NOPASSWD: ALL
```

3. In sanitycheck.py TEACUP uses put() command to copy file. But it is not able to run on host. It gives error Fatal error: put() encountered an exception while uploading

'/home/pglab/Documents/TEACUP/teacup-1.0/runbg_wrapper.sh'.

Hence copy manually the runbg_wrapper.sh file and comment that command on both hosts and enable the executable permissions.

4. If there is difference between clock timing of all three hosts it will give error. Fatal error: Host 192.168.50.2 time synchronization error (difference > 1 seconds. Then synchronize the timings of all the hosts and router.

5. If TEACUP does not able to create pseudo interfaces on router then create it by using following command

```
$ modprobe ifb
```

```
$ set link ip add dev ifb0
```

```
$ set link ip add dev ifb1
```

The two pseudo interfaces are needed, one for apply delay using NetEm and other for AQMs.

6. If duplicate process id are created then **kill -0 pid** will give error.

So first kill that process using **killall** command and then again run the experiment.

7. It will be better if you keep same password for all the hosts.