Lesson Module Status

- Slides – draft
- Properties - done
- Flashcards - na
- 1st minute quiz – done
- Web Calendar summary – done
- Web book pages – done
- Commands –
- Howtos – na
- Skills pacing - na
- X2 Lab – tested
- Depot (VMs) – na
- CD with latest rpms/ and scripts/ - not done
- Publish Lesson pdf - done
- Publish Lab X2 - done
- Publish Practice Test 2 - done

Course history and credits

Jim Griffin

- Jim created the original version of this course
- Jim's site: http://cabrillo.edu/~jgriffin/

Rick Graziani

- Thanks to Rick Graziani for the use of some of his great network slides
- Rick's site: http://cabrillo.edu/~rgraziani/

CIS 192– Lesson 8

Joe P.

Joe A.

Chuck

Kay

Joe B.

Chris H.

Edwin

Julio

Lieven

Rich

Jesus

Jack

Josh

Drew

Casady

Robert

John

Junious

Brynden

Edgar

Chris B.

Ryan

Aaron

Teach & Confer is a live interactive classroom to meet with your students.

STUDENT LOG IN

View Teach & Confer Archives

www.cccconfer.org
dial-in: 888-886-3951
passcode: 439080

VMs for tonight
(Revert, 384MB RAM and power up)
frodo sauron
elrond legolas

Email me (risimms@cabrillo.edu) a relatively current photo of your face for 3 points extra credit

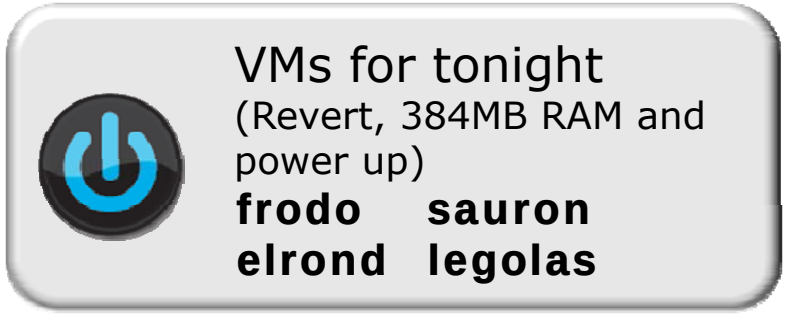
Quiz

Please take out a blank piece of paper, switch off your monitor, close your books, put away your notes and answer these questions:

- What is the Wireshark filter string to view only DHCP transactions?
- What is the DHCP service configuration file on CentOS (Red Hat) family of servers?
- When a client wishes to renew a lease does it initially send the DHCPREQUEST as a broadcast or a unicast?

PPP and WAN protocols

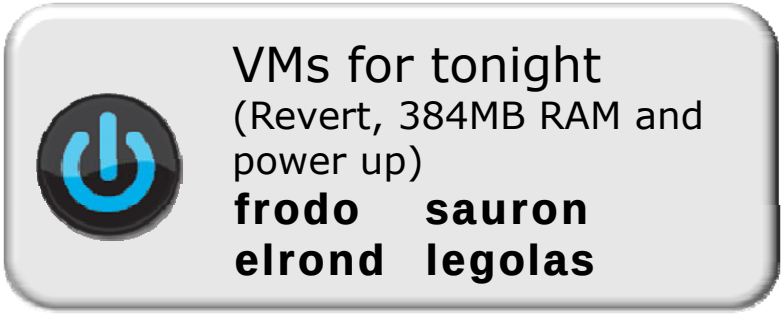
Objectives	Agenda
• Connect two computers on a serial line. • Connect two LANs together through a serial line using Point to Point protocol.	• Quiz • Questions on previous material • Housekeeping • Automation/Walkthrough classroom experiment • Review for next test on Lessons 5-8 • PPP • PPP Lab prep • Wrap



Questions on previous material

Questions?

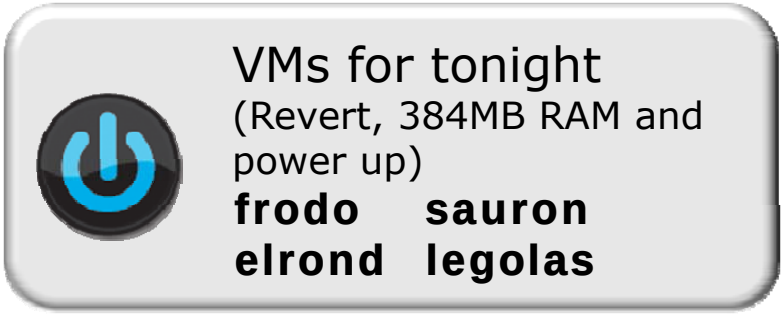
- Previous lesson material
- Lab assignment



Housekeeping

- DHCP Lab 6 due today!
- Excel and the grades page.
- Test (no quiz) next week

- 192 VM performance
  - Change RAM to 384MB
  - Minimize graphics mode
  - Use Putty for copy/paste



Automation and Peer Review Experiment

- Need a way to quickly configure multiple VMs on classroom systems
- Will try an experiment using bash scripts and peer review to "rapidly" set up lab 3 (the routing lab)

Activity – Remove old scripts

1. On Elrond, login as root and:
   cd bin
   rm *network

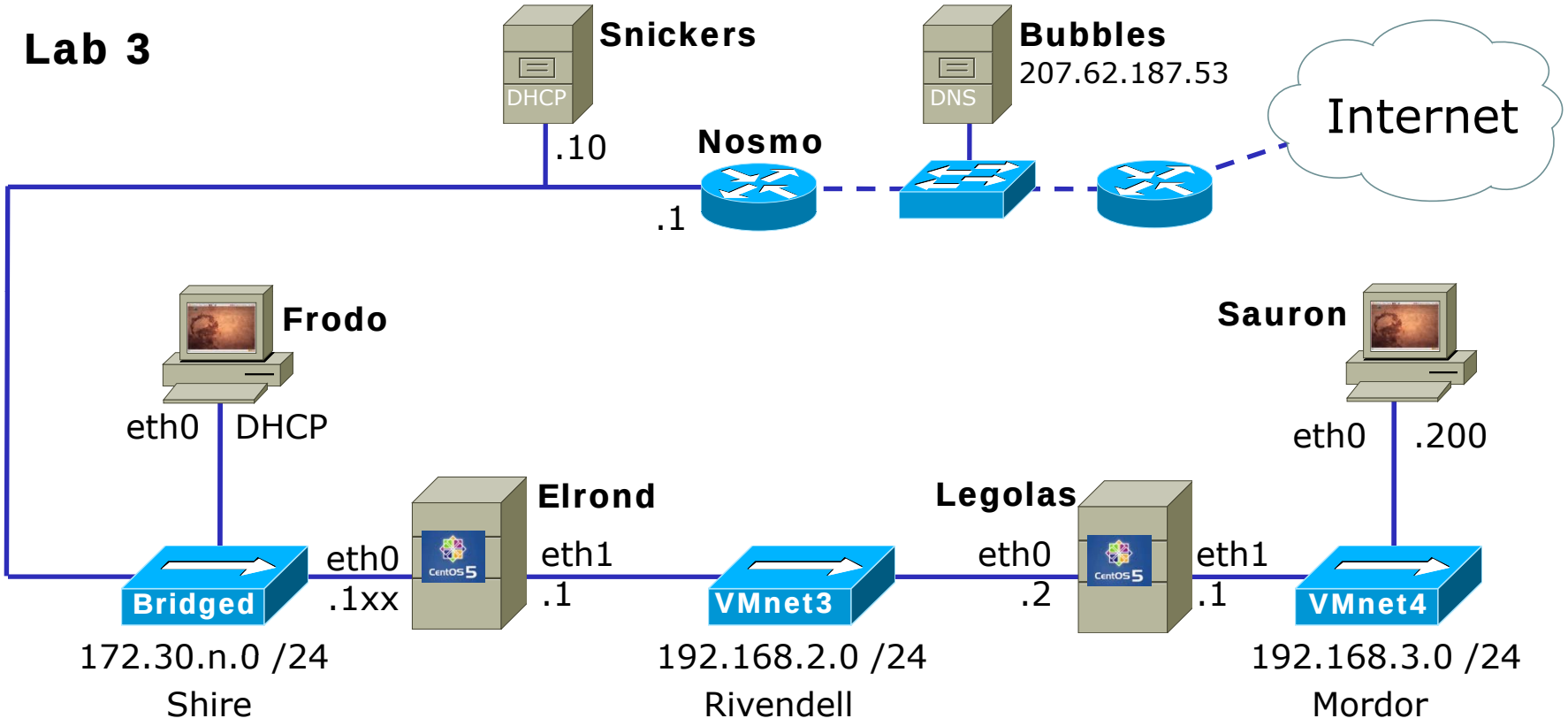
2. On Legolas, login as root and:
   cd bin
   rm *network

3. On Frodo, login as root or **su -** and:
   cd bin
   rm *network

4. On Sauron, login as root or **su -** and:
   cd bin
   rm *network

Cleaning out old scripts in the /root/bin directories

How fast can we implement this on everyone's station?



Frodo
Default GW > 172.30.n.1
192.168.2.0/24> 172.30.n.1xx
192.168.3.0/24 > 172.30.n.1xx

Elrond
Default GW > 172.30.n.1
192.168.3.0/24 > 192.168.2.2
forwarding on

Legolas
Default GW > 192.168.2.1
forwarding on

Sauron
Default GW > 192.168.3.1

Activity – Download Elrond scripts

1. Cable Elrond's eth0 to the Shire network and connect with: **dhclient eth0**
2. Change to root's bin directory if not there already with: **cd /root/bin**
3. Pull down Elrond and CentOS scripts with:

   scp *logname*@opus.cabrillo.edu:/home/cis192/scripts/*elrond .

   scp *logname*@opus.cabrillo.edu:/home/cis192/scripts/*centos .
4. Release IP address with: **dhclient –r**
5. Set execute permission with **chmod 700 ***
6. Verify files:

Make sure this matches your VM

```
[root@elrond bin]# ls -l
total 80
-rwx------ 1 root root  474 Apr  6 09:56 do-lab3-elrond
-rwx------ 1 root root 1205 Apr  6 09:57 init-network-centos
-rwx------ 1 root root  302 Apr  6 09:57 restart-network-centos
-rwx------ 1 root root  746 Apr  6 09:57 set-dns-centos
-rwx------ 1 root root  963 Apr  6 09:57 set-forwarding-centos
-rwx------ 1 root root  803 Apr  6 09:57 set-gateway-centos
-rwx------ 1 root root  927 Apr  6 09:57 set-hostname-centos
-rwx------ 1 root root 1520 Apr  6 09:57 set-interface-centos
-rwx------ 1 root root 1386 Apr  6 09:57 set-route-centos
-rwx------ 1 root root  580 Apr  6 09:57 show-network-centos
[root@elrond bin]#
```

Make sure these match your VM's distro

Activity – Download Legolas scripts

1. Cable Legolas to the Shire network and connect with: **dhclient eth0**
2. Change to root's bin directory if not there already with: **cd /root/bin**
3. Pull down Legolas and CentOS scripts with:

   scp *logname*@opus.cabrillo.edu:/home/cis192/scripts/*legolas .

   scp *logname*@opus.cabrillo.edu:/home/cis192/scripts/*centos .
4. Release IP address with: **dhclient -r**
5. Set execute permission with **chmod 700 ***
6. Verify files:

Make sure this matches your VM

```
[root@legolas bin]# ls -l
total 80
-rwx------ 1 root root  475 Apr  6 10:13 do-lab3-legolas
-rwx------ 1 root root 1205 Apr  6 10:14 init-network-centos
-rwx------ 1 root root  302 Apr  6 10:14 restart-network-centos
-rwx------ 1 root root  746 Apr  6 10:14 set-dns-centos
-rwx------ 1 root root  963 Apr  6 10:14 set-forwarding-centos
-rwx------ 1 root root  803 Apr  6 10:14 set-gateway-centos
-rwx------ 1 root root  927 Apr  6 10:14 set-hostname-centos
-rwx------ 1 root root 1520 Apr  6 10:14 set-interface-centos
-rwx------ 1 root root 1386 Apr  6 10:14 set-route-centos
-rwx------ 1 root root  580 Apr  6 10:14 show-network-centos
[root@legolas bin]#
```

Make sure these match your VM's distro

Activity – Download Frodo scripts

1. Cable Frodo to the Shire network and connect with: **dhclient eth0**
2. Change to root's bin directory if not there already with: **cd /root/bin**
3. Pull down Frodo and CentOS scripts with:

   scp *logname*@opus.cabrillo.edu:/home/cis192/scripts/*frodo .

   scp *logname*@opus.cabrillo.edu:/home/cis192/scripts/*ubuntu .
4. Release IP address with: **dhclient –r**
5. Set execute permission with **chmod 700 ***
6. Verify files:

Make sure this matches your VM

```
root@frodo:~/bin# ls -l
total 40
-rwx------ 1 root root  535 2010-04-03 22:31 do-lab3-frodo
-rwx------ 1 root root  818 2010-04-03 22:31 init-network-ubuntu
-rwx------ 1 root root  323 2010-04-03 22:31 restart-network-ubuntu
-rwx------ 1 root root  739 2010-04-03 22:31 set-dns-ubuntu
-rwx------ 1 root root  906 2010-04-03 22:31 set-forwarding-ubuntu
-rwx------ 1 root root  855 2010-04-03 22:31 set-gateway-ubuntu
-rwx------ 1 root root  921 2010-04-03 22:31 set-hostname-ubuntu
-rwx------ 1 root root 1511 2010-04-03 22:31 set-interface-ubuntu
-rwx------ 1 root root 1281 2010-04-03 22:31 set-route-ubuntu
-rwx------ 1 root root  517 2010-04-03 22:31 show-network-ubuntu
root@frodo:~/bin#
```

Make sure these match your VM's distro

Activity – Download Sauron scripts

1. Cable Sauron to the Shire network and connect with: **dhclient eth0**
2. Change to root's bin directory if not there already with: **cd /root/bin**
3. Pull down Sauron and CentOS scripts with:

   scp *logname*@opus.cabrillo.edu:/home/cis192/scripts/*sauron .

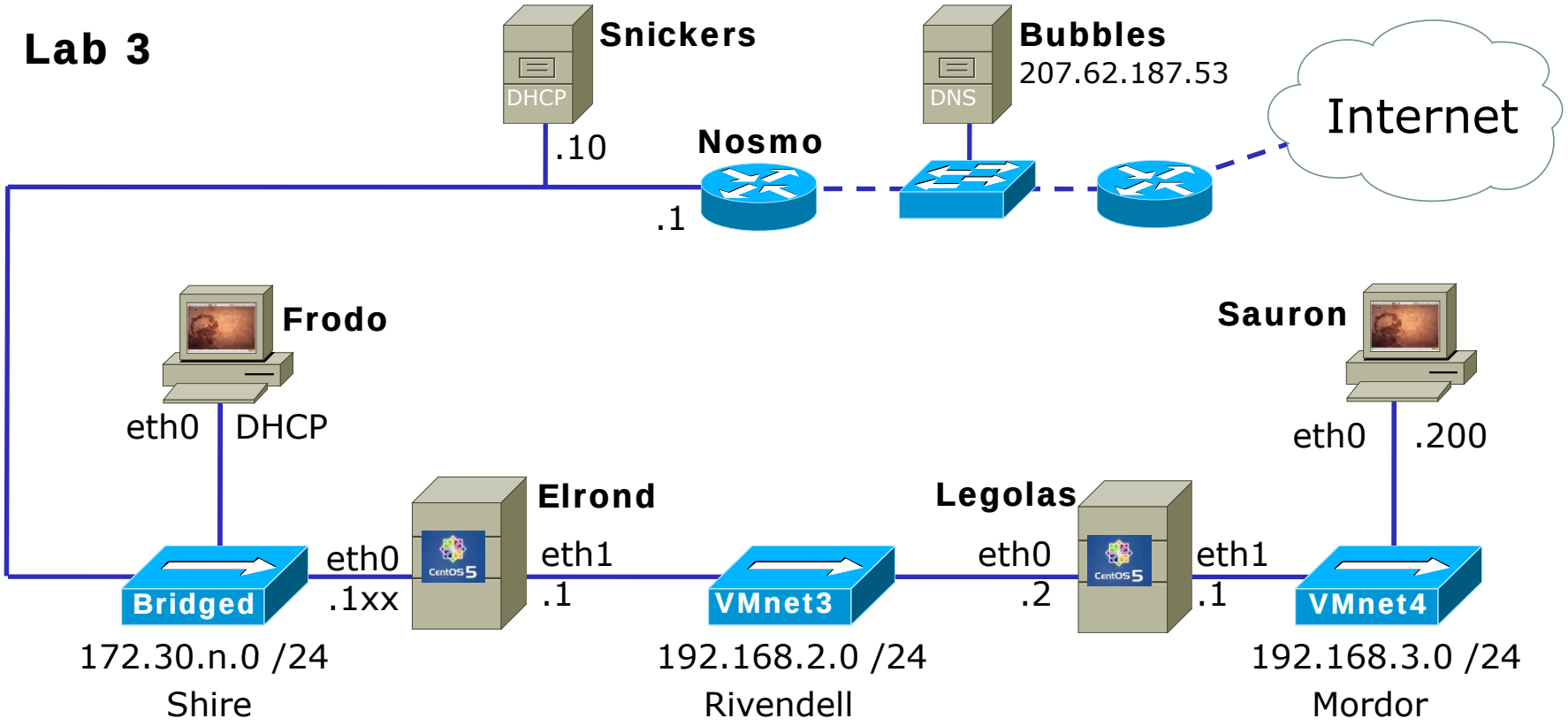
   scp *logname*@opus.cabrillo.edu:/home/cis192/scripts/*ubuntu .
4. Release IP address with: **dhclient –r**
5. Set execute permission with **chmod 700 ***
6. Verify files:

Make sure this matches your VM

```
root@sauron:~/bin# ls -l
total 40
-rwx------ 1 root root  377 2010-04-06 11:19 do-lab3-sauron
-rwx------ 1 root root  818 2010-04-06 11:19 init-network-ubuntu
-rwx------ 1 root root  323 2010-04-06 11:19 restart-network-ubuntu
-rwx------ 1 root root  739 2010-04-06 11:19 set-dns-ubuntu
-rwx------ 1 root root  906 2010-04-06 11:19 set-forwarding-ubuntu
-rwx------ 1 root root  855 2010-04-06 11:19 set-gateway-ubuntu
-rwx------ 1 root root  921 2010-04-06 11:19 set-hostname-ubuntu
-rwx------ 1 root root 1511 2010-04-06 11:19 set-interface-ubuntu
-rwx------ 1 root root 1281 2010-04-06 11:19 set-route-ubuntu
-rwx------ 1 root root  517 2010-04-06 11:19 show-network-ubuntu
root@sauron:~/bin#
```

Make sure these match your VM's distro

How fast can we implement this on everyone's station?



Frodo
Default GW > 172.30.n.1
192.168.2.0/24> 172.30.n.1xx
192.168.3.0/24 > 172.30.n.1xx

Elrond
Default GW > 172.30.n.1
192.168.3.0/24 > 192.168.2.2
forwarding on

Legolas
Default GW > 192.168.2.1
forwarding on

Sauron
Default GW > 192.168.3.1

Activity – Recable VMs

Bridged – eth0

Frodo

Use VM Settings to recable each VM as necessary

VMnet4 – eth0

Sauron

Bridged – eth0

eth1 **- VMnet3**

Elrond

VMnet3 – eth0

eth1 **– VMnet4**

Legolas

set-dns-centos script

The shebang (#!) line references the script interpreter (e.g. bash, perl)

All other lines beginning with a # are comments

```
[root@elrond bin]# cat set-dns-centos
#!/bin/bash
#
# Rich Simms
# Spring 2010
#
# This script makes permanent DNS settings
#
# usage: set-dns [-d] server
#
```

set-dns-centos script

```
# Initialize
usage="set-dns [-d] server"

# Process debug option (-d)
while getopts ":d" opt; do
  case $opt in
     d  ) debug="y"
          echo "** Debug tracing enabled" ;;
     \? ) echo "-- Usage: " $usage >&2
          exit 1
  esac
done

# Shift out any options
if [ $OPTIND -ge 1 ]; then shift $(($OPTIND -
  1)); fi
```

getopts is predefined function for parsing options

Check for options on command line

Remove options from command line

OPTIND is an index that points to the next argument to be parsed

set-dns-centos script

```
dns=$1
```

get the next argument (follows the options)

```
if [ "$debug" = "y" ]; then
  echo "** trace: dns = " $dns
fi
```

If tracing is enabled print what the user entered

Make sure user entered something or complain

```
if [ "$dns" = "" ]; then
  echo "-- Usage: " $usage >&2
  echo "-- Missing arguments"
  exit 1
fi
```

```
> /etc/resolv.conf
echo nameserver $dns >> /etc/resolv.conf
```

The reason for deleting the entry first is to allow the script to be run repeatedly without adding redundant lines to configuration file

This could also be done in one line

```
cat /etc/resolv.conf

exit 0
[root@elrond bin]#
```

*The number following "exit" represents the exit status. 0 = success and 1 = error. Use **echo $?** to display exit status after script has been run.*

set-dns-centos script

Example: **./set-dns-centos 207.62.187.53** would correctly setup /etc/resolv.conf for that specific name server

Note: ./ is used the run a command or script that is not in the path

```
[root@elrond bin]# ./set-dns-centos 207.62.187.53
nameserver 207.62.187.53

[root@elrond bin]# ./set-dns-centos -d 207.62.187.53
** Debug tracing enabled
** trace: net =  207.62.187.53
nameserver 207.62.187.53
[root@elrond bin]#

[root@elrond bin]# echo $?
0
```

set-forwarding-centos script

The shebang (#!) line references the script interpreter (e.g. bash, perl)

All other lines beginning with a # are comments

```
[root@elrond bin]# cat set-forwarding-centos
#!/bin/bash
#
# Rich Simms
# Spring 2010
#
# This script makes permanent forwarding settings on CentOS
#
# usage: set-forwarding [-d] value
#
```

set-forwarding-centos script

```
# Initialize
usage="set-forwarding [-d]  0 | 1"

# Process debug option (-d)
while getopts ":d" opt; do
  case $opt in
     d  ) debug="y"
          echo "** Debug tracing enabled" ;;
     \? ) echo "-- Usage: " $usage >&2
          exit 1
  esac
done

# Shift out any options
if [ $OPTIND -ge 1 ]; then shift $(($OPTIND - 1)); fi
```

getopts is predefined function for parsing options

Check for options on command line

Remove options from command line

OPTIND is an index that points to the next argument to be parsed

set-forwarding-centos script

```
value=$1
```

get the next argument (follows the options)

```
if [ "$debug" = "y" ]; then
  echo "** trace: value = " $value
fi
```

If tracing is enabled print what the user entered

Make sure user entered something or complain

```
if [ "$value" = "" ]; then
  echo "-- Usage: " $usage >&2
  echo "-- Missing argument"
  exit 1
fi
```

Set up forwarding (both permanently and temporarily.

```
if [ "$value" = "0" ]; then
  sed -i 's/forward = 1/forward = 0/' /etc/sysctl.conf
  echo 0 > /proc/sys/net/ipv4/ip_forward
else
  sed -i 's/forward = 0/forward = 1/' /etc/sysctl.conf
  echo 1 > /proc/sys/net/ipv4/ip_forward
fi

cat /etc/sysctl.conf | grep forward
cat /proc/sys/net/ipv4/ip_forward

exit 0
[root@elrond bin]#
```

set-forwarding-centos script

Example: **./set-forwarding-centos 1** would enable forwarding on the server

```
[root@elrond bin]# ./set-forwarding-centos 1
# Controls IP packet forwarding
net.ipv4.ip_forward = 1
1
[root@elrond bin]# ./set-forwarding-centos -d 1
** Debug tracing enabled
** trace: value =  1
# Controls IP packet forwarding
net.ipv4.ip_forward = 1
1
[root@elrond bin]#
```

set-gateway-centos script

The shebang (#!) line references the script interpreter (e.g. bash, perl)

All other lines beginning with a # are comments

```
[root@elrond bin]# cat set-gateway-centos
#!/bin/bash
#
# Rich Simms
# Spring 2010
#
# This script makes permanent gateway settings on Centos
#
# usage: set-gateway [-d] gw
#
```

set-gateway-centos script

```
# Initialize
usage="set-gateway [-d] gw"

# Process debug option (-d)
while getopts ":d" opt; do
  case $opt in
     d  ) debug="y"
          echo "** Debug tracing enabled" ;;
     \? ) echo "-- Usage: " $usage >&2
          exit 1
  esac
done

# Shift out any options
if [ $OPTIND -ge 1 ]; then shift $(($OPTIND - 1)); fi
```

getopts is predefined function for parsing options

Check for options on command line

Remove options from command line

OPTIND is an index that points to the next argument to be parsed

set-gateway-centos script

```
gw=$1
```

get the next argument (follows the options)

```
if [ "$debug" = "y" ]; then
  echo "** trace: gw = " $gw
fi
```

If tracing is enabled print what the user entered

Make sure user entered something or complain

```
if [ "$gw" = "" ]; then
  echo "-- Usage: " $usage >&2
  echo "-- Missing argument"
  exit 1
fi
```

The reason for deleting the entry first is to allow the script to be run repeatedly without adding redundant lines to configuration file

```
sed -i '/GATEWAY/ d' /etc/sysconfig/network
sed -i '$aGATEWAY='$gw'' /etc/sysconfig/network
cat /etc/sysconfig/network
```

Set up default gateway (both permanently and temporarily.

```
route del default gw $gw
route add default gw $gw
```

```
exit 0
[root@elrond bin]#
```

set-gateway-centos script

Example: **./set-gateway-centos 172.30.1.1** would make that the default gateway

```
[root@elrond bin]# ./set-gateway-centos 172.30.1.1
NETWORKING=yes
NETWORKING_IPV6=no
HOSTNAME=elrond.localdomain
GATEWAY=172.30.1.1
[root@elrond bin]#

[root@elrond bin]# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.3.0     192.168.2.2     255.255.255.0   UG    0      0        0 eth1
172.30.1.0      0.0.0.0         255.255.255.0   U     0      0        0 eth0
192.168.2.0     0.0.0.0         255.255.255.0   U     0      0        0 eth1
169.254.0.0     0.0.0.0         255.255.0.0     U     0      0        0 eth1
0.0.0.0         172.30.1.1      0.0.0.0         UG    0      0        0 eth0
[root@elrond bin]#
```

set-hostname-centos script

The shebang (#!) line references the script interpreter (e.g. bash, perl)

All other lines beginning with a # are comments

```
[root@elrond bin]# cat set-hostname-centos
#!/bin/bash
#
# Rich Simms
# Spring 2010
#
# This script makes permanent hostname settings on CentOS
#
# usage: set-hostname [-d] hostname
#
```

set-hostname-centos script

```
# Initialize
usage="set-hostname [-d] hostname"

# Process debug option (-d)
while getopts ":d" opt; do
  case $opt in
     d  ) debug="y"
          echo "** Debug tracing enabled" ;;
     \? ) echo "-- Usage: " $usage >&2
          exit 1
  esac
done

# Shift out any options
if [ $OPTIND -ge 1 ]; then shift $(($OPTIND - 1)); fi
```

getopts is predefined function for parsing options

Check for options on command line

Remove options from command line

OPTIND is an index that points to the next argument to be parsed

set-hostname-centos script

```
host=$1
```

get the next argument (follows the options)

```
if [ "$debug" = "y" ]; then
  echo "** trace: host = " $host
  echo "** trace: gw = " $gw
fi
```

If tracing is enabled print what the user entered

Make sure user entered something or complain

```
if [ "$host" = "" ]; then
  echo "-- Usage: " $usage >&2
  exit 1
fi
```

The reason for deleting the entry first is to allow the script to be run repeatedly without adding redundant lines to configuration file

Set up new hostname (both temporarily and permanently

```
hostname $host
sed -i '/HOSTNAME/ d' /etc/sysconfig/network
sed -i '$aHOSTNAME='$host.localdomain'' /etc/sysconfig/network
cat /etc/sysconfig/network | grep $host
```

Don't forget to update /etc/hosts

```
sed -i '/127.0.0.1/ d' /etc/hosts
sed -i '$a127.0.0.1 '$host'.localdomain '$host'
  localhost.localdomain localhost' /etc/hosts
cat /etc/hosts | grep $host

exit 0
[root@elrond bin]#
```

set-hostname-centos script

Example: **./set-hostname-centos elrond** would make that the new hostname

```
[root@elrond bin]# ./set-hostname-centos elrond
HOSTNAME=elrond.localdomain
127.0.0.1 elrond.localdomain elrond localhost.localdomain localhost
[root@elrond bin]#

[root@elrond bin]# hostname
elrond.localdomain
[root@elrond bin]#
```

set-interface-centos script

The shebang (#!) line references the script interpreter (e.g. bash, perl)

All other lines beginning with a # are comments

```
[root@elrond bin]# cat set-interface-centos
#!/bin/bash
#
# Rich Simms
# Spring 2010
#
# This script makes permanent interface settings
#
# usage: set-interface [-d] ethx ip mask
#
```

set-interface-centos script

```
# Initialize
usage="set-interface [-d] ethx ip mask"

# Process debug option (-d)
while getopts ":d" opt; do
  case $opt in
     d  ) debug="y"
          echo "** Debug tracing enabled" ;;
     \? ) echo "-- Usage: " $usage >&2
          exit 1
  esac
done

# Shift out any options
if [ $OPTIND -ge 1 ]; then shift $(($OPTIND - 1)); fi
```

getopts is predefined function for parsing options

Check for options on command line

Remove options from command line

OPTIND is an index that points to the next argument to be parsed

set-interface-centos script

```
eth=$1
ip=$2
mask=$3
```

Get the next arguments off the command line

```
if [ "$debug" = "y" ]; then
  echo "** trace: eth = " $eth
  echo "** trace: ip = " $ip
  echo "** trace: mask = " $mask
fi
```

If tracing is enabled print what the user entered

Make sure user entered something or complain

```
if [ "$eth" = "" ] || [ "$ip" = "" ] || [ "$mask" = "" ]; then
  echo "-- Usage: " $usage >&2
  exit 1
fi
```

Check for correct interfaces

```
if [ "$eth" = "eth0" ] || [ "$eth" = "eth1" ] || [ "$eth" =
  "eth3" ]; then
  echo "Interface OK"
else
  echo "-- Usage: " $usage >&2
  echo "-- Interface $eth is bogus"
  exit 1
fi
```

set-interface-centos script

Configure the interface temporarily

```
ifconfig $eth $ip netmask $mask
```

Then update the appropriate ifcfg file

```
sed -i '/ONBOOT/ d' /etc/sysconfig/network-scripts/ifcfg-$eth
sed -i '$aONBOOT=yes' /etc/sysconfig/network-scripts/ifcfg-$eth
sed -i '/BOOTPROTO/ d' /etc/sysconfig/network-scripts/ifcfg-$eth
sed -i '$aBOOTPROTO=static' /etc/sysconfig/network-scripts/ifcfg-$eth
sed -i '/IPADDR/ d' /etc/sysconfig/network-scripts/ifcfg-$eth
sed -i '$aIPADDR='$ip'' /etc/sysconfig/network-scripts/ifcfg-$eth
sed -i '/NETMASK/ d' /etc/sysconfig/network-scripts/ifcfg-$eth
sed -i '$aNETMASK='$mask'' /etc/sysconfig/network-scripts/ifcfg-$eth
```

```
cat /etc/sysconfig/network-scripts/ifcfg-$eth

exit 0
[root@elrond bin]#
```

The reason for deleting the entry first is to allow the script to be run repeatedly without adding redundant lines to configuration file

set-interface-centos script

Example: **./set-interface-centos eth0 172.30.1.125 255.255.255.0**
would setup eth0 as 172.30.1.125/24

```
[root@elrond bin]# ./set-interface-centos eth0 172.30.1.125 255.255.255.0
Interface OK
# Intel Corporation 82543GC Gigabit Ethernet Controller (Copper)
DEVICE=eth0
ONBOOT=yes
BOOTPROTO=static
IPADDR=172.30.1.125
NETMASK=255.255.255.0
[root@elrond bin]#

[root@elrond bin]# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 08:00:27:7A:14:7F
          inet addr:172.30.1.125  Bcast:172.30.1.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe7a:147f/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1361 errors:0 dropped:0 overruns:0 frame:0
          TX packets:910 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:206448 (201.6 KiB)  TX bytes:77582 (75.7 KiB)
          Memory:f0000000-f0020000

[root@elrond bin]#
```

set-route-centos script

The shebang (#!) line references the script interpreter (e.g. bash, perl)

All other lines beginning with a # are comments

```
[root@elrond bin]# cat set-route-centos
#!/bin/bash
#
# Rich Simms
# Spring 2010
#
# This script makes permanent route settings
#
# usage: set-route [-d] network prefix ethx gw
#
```

set-route-centos script

```
# Initialize
usage="set-route [-d] network prefix ethx gw"

# Process debug option (-d)
while getopts ":d" opt; do
  case $opt in
     d  ) debug="y"
          echo "** Debug tracing enabled" ;;
     \? ) echo "-- Usage: " $usage >&2
          exit 1
  esac
done

# Shift out any options
if [ $OPTIND -ge 1 ]; then shift $(($OPTIND - 1)); fi
```

getopts is predefined function for parsing options

Check for options on command line

Remove options from command line

OPTIND is an index that points to the next argument to be parsed

set-route-centos script

```
net=$1
prefix=$2
eth=$3
gw=$4
```

Get the next arguments off the command line

```
if [ "$debug" = "y" ]; then
  echo "** trace: net = " $net
  echo "** trace: prefix = " $prefix
  echo "** trace: eth = " $eth
  echo "** trace: gw = " $gw
fi
```

If tracing is enabled print what the user entered

Make sure user entered something or complain

```
if [ "$net" = "" ] || [ "$prefix" = "" ] || [ "$eth" = "" ] ||
  [ "$gw" = "" ]; then
  echo "-- Usage: " $usage >&2
  echo "-- Missing arguments"
  exit 1
fi
```

Check for correct interfaces

```
if [ "$eth" = "eth0" ] || [ "$eth" = "eth1" ] || [ "$eth" =
  "eth3" ]; then
  echo "Interface OK"
else
  echo "-- Usage: " $usage >&2
  echo "-- Interface $eth is bogus"
  exit 1
fi
```

set-route-centos script

Configure the route temporarily

```
route del -net $net/$prefix gw $gw
route add -net $net/$prefix gw $gw
```

Then create or modify the appropriate route file

```
touch /etc/sysconfig/network-scripts/route-$eth
sed -i '/'$net'/ d' /etc/sysconfig/network-scripts/route-$eth
echo $net/$prefix via $gw >> /etc/sysconfig/network-scripts/route-$eth

echo "file /etc/sysconfig/network-scripts/route-$eth" contains:
cat /etc/sysconfig/network-scripts/route-$eth

exit 0

[root@elrond bin]#
```

The reason for deleting the entry first is to allow the script to be run repeatedly without adding redundant lines to configuration file

set-route-centos script

Example: **./set-route-centos 192.168.3.0 24 eth1 192.168.2.2**
would setup a static route for 192.168.3.0/24 via 192.168.2.2

```
[root@elrond bin]# ./set-route-centos 192.168.3.0 24 eth1 192.168.2.2
Interface OK
file /etc/sysconfig/network-scripts/route-eth1 contains:
192.168.3.0/24 via 192.168.2.2
[root@elrond bin]#

[root@elrond bin]# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.3.0     192.168.2.2     255.255.255.0   UG    0      0        0 eth1
172.30.1.0      0.0.0.0         255.255.255.0   U     0      0        0 eth0
192.168.2.0     0.0.0.0         255.255.255.0   U     0      0        0 eth1
169.254.0.0     0.0.0.0         255.255.0.0     U     0      0        0 eth1
0.0.0.0         172.30.1.1      0.0.0.0         UG    0      0        0 eth0
[root@elrond bin]#
```

init-network-centos script

```
[root@elrond bin]# cat init-network-centos
#!/bin/bash

# Remove network settings on interfaces
sed -i '/ONBOOT/ d' /etc/sysconfig/network-scripts/ifcfg-eth0
sed -i '/BOOTPROTO/ d' /etc/sysconfig/network-scripts/ifcfg-eth0
sed -i '/IPADDR/ d' /etc/sysconfig/network-scripts/ifcfg-eth0
sed -i '/NETMASK/ d' /etc/sysconfig/network-scripts/ifcfg-eth0
sed -i '/DHCP_HOSTNAME/ d' /etc/sysconfig/network-scripts/ifcfg-eth0
sed -i '/HWADDR/ d' /etc/sysconfig/network-scripts/ifcfg-eth0
sed -i '$aONBOOT=no' /etc/sysconfig/network-scripts/ifcfg-eth0
sed -i '$aBOOTPROTO=none' /etc/sysconfig/network-scripts/ifcfg-eth0

#
sed -i '/ONBOOT/ d' /etc/sysconfig/network-scripts/ifcfg-eth1
sed -i '/BOOTPROTO/ d' /etc/sysconfig/network-scripts/ifcfg-eth1
sed -i '/IPADDR/ d' /etc/sysconfig/network-scripts/ifcfg-eth1
sed -i '/NETMASK/ d' /etc/sysconfig/network-scripts/ifcfg-eth1
sed -i '/DHCP_HOSTNAME/ d' /etc/sysconfig/network-scripts/ifcfg-eth1
sed -i '/HWADDR/ d' /etc/sysconfig/network-scripts/ifcfg-eth1
sed -i '$aONBOOT=no' /etc/sysconfig/network-scripts/ifcfg-eth1
sed -i '$aBOOTPROTO=none' /etc/sysconfig/network-scripts/ifcfg-eth1
```

Disable eth0 interface

Now disable eth1 interface

This script will disable networking after restarting the network service

init-network-centos script

```
# remove default and static routes
sed -i '/GATEWAY/ d' /etc/sysconfig/network
rm /etc/sysconfig/network-scripts/route-eth[01]

# disable forwarding
sed -i 's/forward = 1/forward = 0/' /etc/sysctl.conf
cat /etc/sysctl.conf | grep forward
echo 0 > /proc/sys/net/ipv4/ip_forward

# Remove DNS settings
> /etc/resolv.conf

echo To use new settings: service network restart
[root@elrond bin]#
```

Remove default and static routes

Disable packet forwarding

Remove name servers

Remind user to restart network service

init-network-centos script

```
[root@elrond bin]# ./init-network-centos
# Controls IP packet forwarding
net.ipv4.ip_forward = 0
To use new settings: service network restart
[root@elrond bin]#

[root@elrond bin]# cat /etc/sysconfig/network-scripts/ifcfg-eth0
# Intel Corporation 82543GC Gigabit Ethernet Controller (Copper)
DEVICE=eth0
ONBOOT=no
BOOTPROTO=none
[root@elrond bin]#

[root@elrond bin]# cat /etc/sysconfig/network
NETWORKING=yes
NETWORKING_IPV6=no
HOSTNAME=elrond.localdomain
[root@elrond bin]#
```

Interface eth0 is disabled

Default gateway removed

This script will disable networking after restarting the network service

restart-network-centos script

```
[root@elrond bin]# cat restart-network-centos
#!/bin/bash
#
# Ubuntu VM network restart
#
echo -n "Restart network service? (y to confirm): "
read answer
if [ "$answer" = "y" ]; then
  echo Using: service network restart
  service network restart
else
  echo Network service has not been restarted
  echo To restart use: service network restart
fi
[root@elrond bin]#
```

Interactive script to ask you whether network service should be restarted

restart-network-centos script

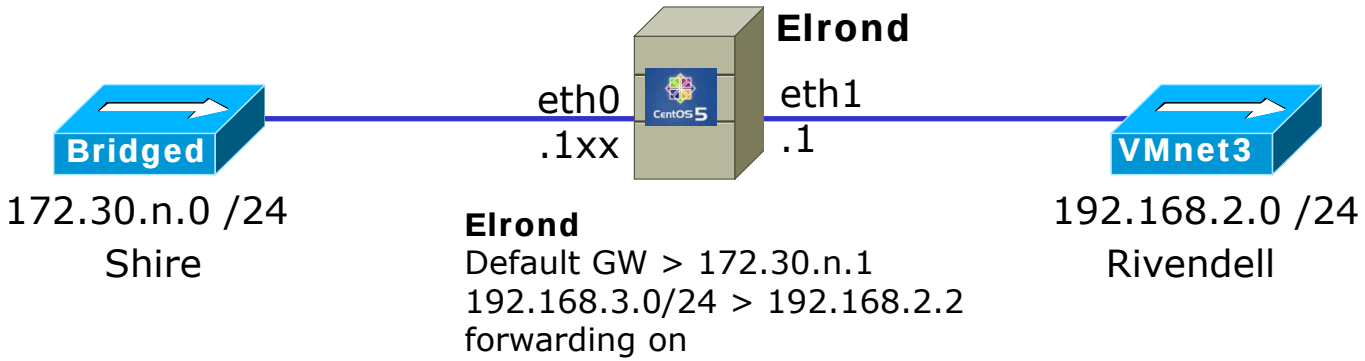
```
[root@elrond bin]# ./restart-network-centos
Restart network service? (y to confirm): y
Using: service network restart
Shutting down interface eth0:                              [  OK  ]
Shutting down interface eth1:                              [  OK  ]
Shutting down loopback interface:                          [  OK  ]
Bringing up loopback interface:                            [  OK  ]
[root@elrond bin]#
```

Interactive script to ask you whether network service should be restarted

do-lab3-elrond script

```
[root@elrond bin]# cat do-lab3-elrond
echo "Setting up Lab 3 on Elrond"

/root/bin/init-network-centos
/root/bin/set-interface-centos eth0 172.30.1.125 255.255.255.0
/root/bin/set-interface-centos eth1 192.168.2.1 255.255.255.0
/root/bin/set-gateway-centos 172.30.1.1
/root/bin/set-route-centos 192.168.3.0 24 eth1 192.168.2.2
/root/bin/set-dns-centos 207.62.187.53
/root/bin/set-hostname-centos elrond
/root/bin/set-forwarding-centos 1
/root/bin/restart-network-centos

echo Use init 6 if hostname was changed

[root@elrond bin]#
```

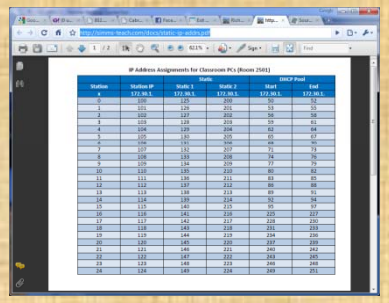
This script calls the other scripts to configure Elrond for Lab 3!



do-lab3-elrond script

```
[root@elrond bin]# cat do-lab3-elrond
echo "Setting up Lab 3 on Elrond"

/root/bin/init-network-centos
/root/bin/set-interface-centos eth0 172.30.n.1xx 255.255.255.0
/root/bin/set-interface-centos eth1 192.168.2.1 255.255.255.0
/root/bin/set-gateway-centos 172.30.1.1
/root/bin/set-route-centos 192.168.3.0 24 eth1 192.168.2.2
/root/bin/set-dns-centos 207.62.187.53
/root/bin/set-hostname-centos elrond
/root/bin/set-forwarding-centos 1
/root/bin/restart-network-centos

echo Use init 6 if hostname was changed

[root@elrond bin]#
```

Modify to your unique static IP address from

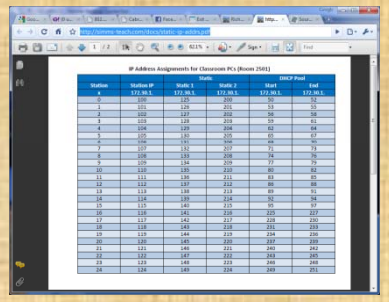


http://simms-teach.com/docs/static-ip-addrs.pdf

do-lab3-elrond script

```
[root@frodo bin]# cat do-lab3-frodo
echo "Setting up Lab 3 on Frodo"

/root/bin/init-network-ubuntu

/root/bin/set-route-ubuntu 192.168.2.0 24 eth0 172.30.n.1xx
/root/bin/set-route-ubuntu 192.168.3.0 24 eth0 172.30.n.1xx

sed -i '/arwen/ d' /etc/hosts
sed -i '$a192.168.2.2 arwen' /etc/hosts
sed -i '/sauron/ d' /etc/hosts
sed -i '$a192.168.3.200 sauron' /etc/hosts

/root/bin/restart-network-ubuntu

echo Use init 6 if hostname changed
```

Modify to your unique static IP address for Elrond eth0 using



http://simms-teach.com/docs/static-ip-addrs.pdf

Other scripts

- We just did a walkthrough of the scripts on Elrond
- Similar scripts can be found in the /root/bin directories on Frodo, Legolas and Sauron
- Each do-lab3-*name* script contains the overall commands to configure that specific VM for Lab 3.
- Each *ubuntu script corresponds to each *centos script except that they are customized for Ubuntu's configuration files and ways of configuring network settings

Activity – Peer Walkthrough

Bridged – eth0 
Frodo

The power of a second set of eyes is invaluable!

VMnet4 – eth0 
Sauron

1. Pair up with another student

2. Verify **Frodo**, **Elrond**, **Legolas** and **Sauron** VMs:
   - ❑ Logged on as root
   - ❑ Scripts are in root's bin directory
   - ❑ The "do-lab3-name" scripts match the VM's name
   - ❑ The other scripts match VM's distro (CentOS or Ubuntu)
   - ❑ Execute permission has been set on all scripts
   - ❑ Cabling is correct
3. Verify the do-lab3-elrond script on **Elrond** has the correct eth0 IP address

Bridged – eth0  eth1 **- VMnet3**
Elrond

VMnet3 – eth0  eth1 **– VMnet4**
Legolas

Activity – Do Lab 3

1. On Frodo, in /root/bin, use:

   ./do-lab3-frodo (type y to confirm network restart)

2. On Elrond, in /root/bin, use:

   ./do-lab3-elrond (type y to confirm network restart)

3. On Legolas, in /root/bin, use:

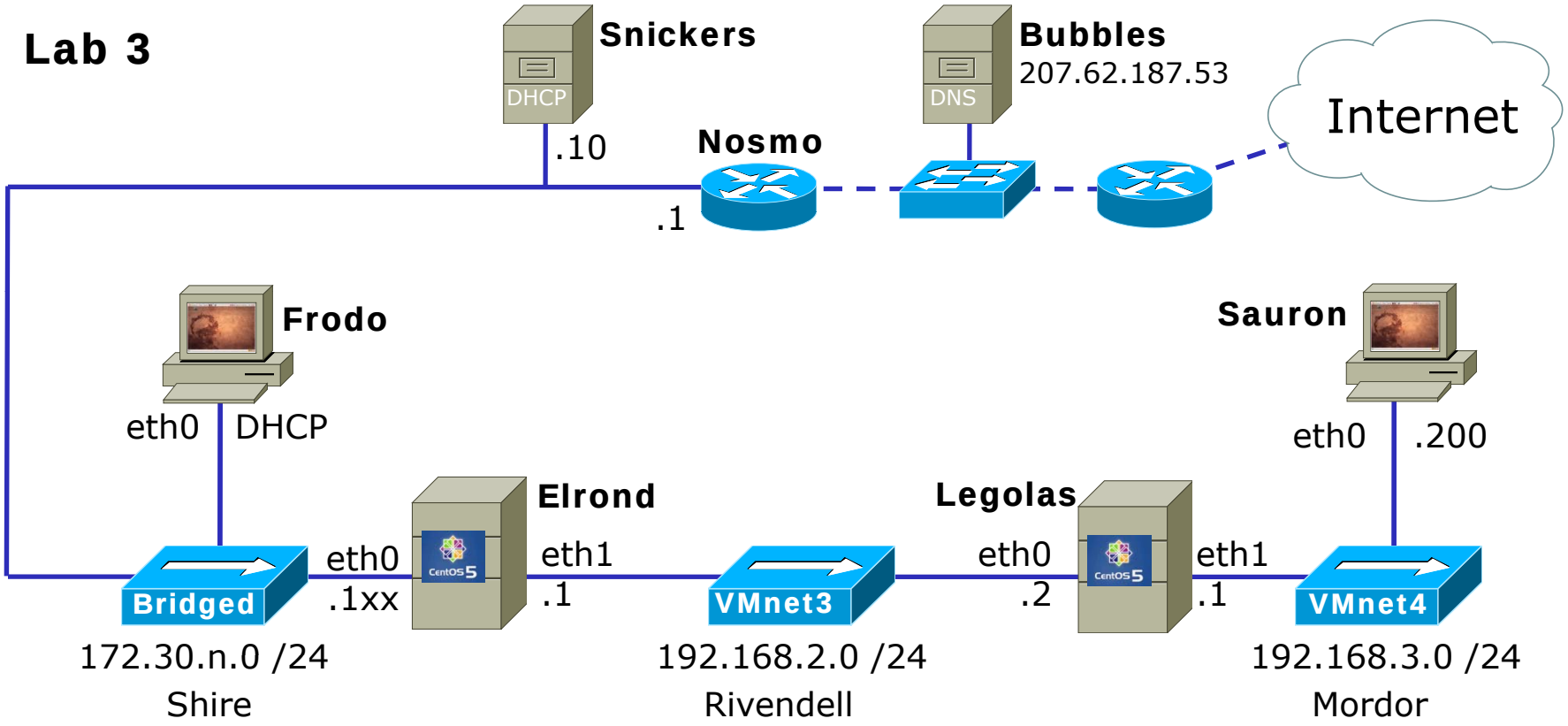
   ./do-lab3-legolas (type y to confirm network restart)

4. On Sauron, in /root/bin, use:

   ./do-lab3-sauron (type y to confirm network restart)

5. Can Frodo **ping sauron** ? If so you just completed Lab 3!

How fast can we implement this on everyone's station?



Frodo
Default GW > 172.30.n.1
192.168.2.0/24> 172.30.n.1xx
192.168.3.0/24 > 172.30.n.1xx

Elrond
Default GW > 172.30.n.1
192.168.3.0/24 > 192.168.2.2
forwarding on

Legolas
Default GW > 192.168.2.1
forwarding on

Sauron
Default GW > 192.168.3.1

Selective Review for Test 2

The Next Test

Same procedure as before:

- Practice test available one week prior.
- Students may work together and use the forum to work out answers to practice test questions.
- Actual test will be very similar and changes highlighted.
- Actual test is open book, open computers, and open VMs.
- During the actual test, students may not ask for or give assistance to others.

A Pizza Bribe for Next Test

	T1	T2
	30	30
	33	
	30	
	27	
	25	
	17	
	18	
	22	
	32	
	34	
	25	
	32	
	29	
	27	
	32	
	30	
	29	
	25	
	31	
	30	
	28	

T1 average score = 27.80

The Pizza Bribe is as follows:

If T2 average > 27.80 then **PIZZA for the CLASS**

The Next Test

Tips:

- Know **how** to work out the answers to all practice test questions.
- Use the forum to collaborate during the week prior to the test.
- Verify your answers on live VMs whenever possible.

Test 2 is cumulative

New topics since the last test include:

- Debian/Ubuntu configuration
- TCP - open and close connections
- TCP - tunable kernel parameters
- TCP - security issues
- Security Issues
- Application Layer
- telnet
- vsftpd
- sshd
- Super daemons
- TCP Wrappers
- SSH Port Forwarding
- Netfilter (firewalls and NAT)
- Firewalls and FTP
- DHCP
- PPP

Lessons 5, 6, 7, 8

Debian / Ubuntu NIC Config (permanent)

hostname

/etc/hostname
```
frodo
```

/etc/hosts
```
< snipped >
127.0.1.1 frodo
< snipped>
```

Restart network service with:
/etc/init.d/networking restart

Static

/etc/network/interfaces
```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
address 172.30.4.133
netmask 255.255.255.0

gateway 172.30.4.1

up route add -net 192.168.2.0/24 gw 172.30.4.107
up route add -net 192.168.3.0/24 gw 172.30.4.107
```

Name server

/etc/resolv.conf
```
nameserver 207.62.187.53
```

DHCP

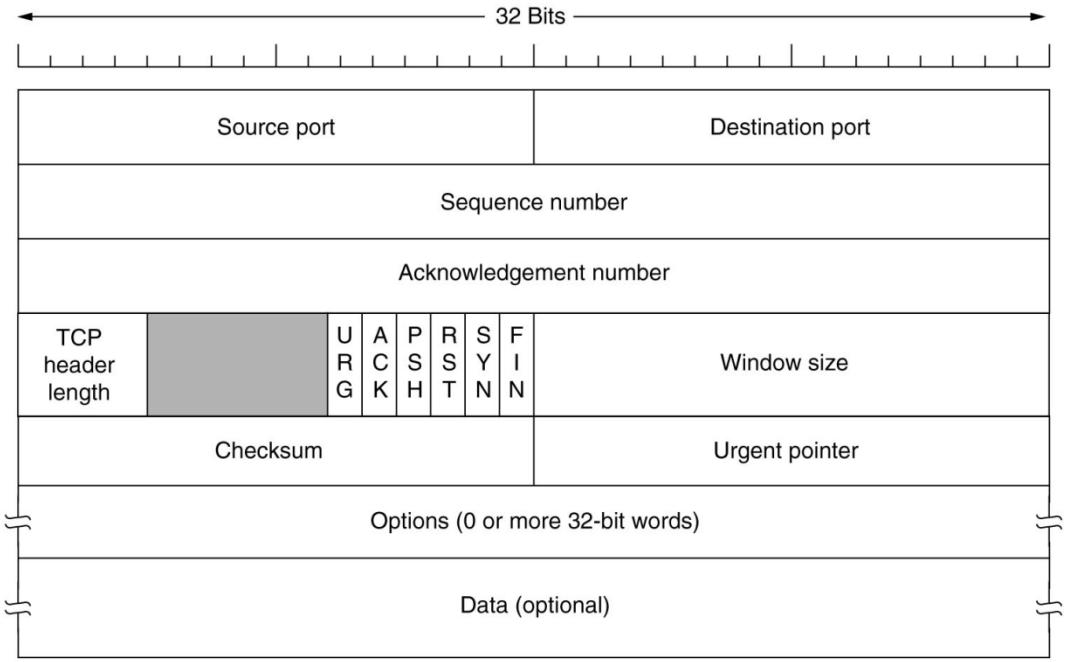
/etc/network/interfaces
```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet dhcp

up route add -net 192.168.2.0/24 gw 172.30.4.107
up route add -net 192.168.3.0/24 gw 172.30.4.107
```

static routes

Transport Layer

The Transmission Control Protocol

TCP Header



Ports *are used to identify application*

Sequence *and* ***acknowledgement numbers*** *are used for flow control.*

ACK, ***SYN*** *and* ***FIN*** *flags are used for initiating connections, acknowledging data received and terminating connections*

Window size *is use to communicate buffer size of recipient.*

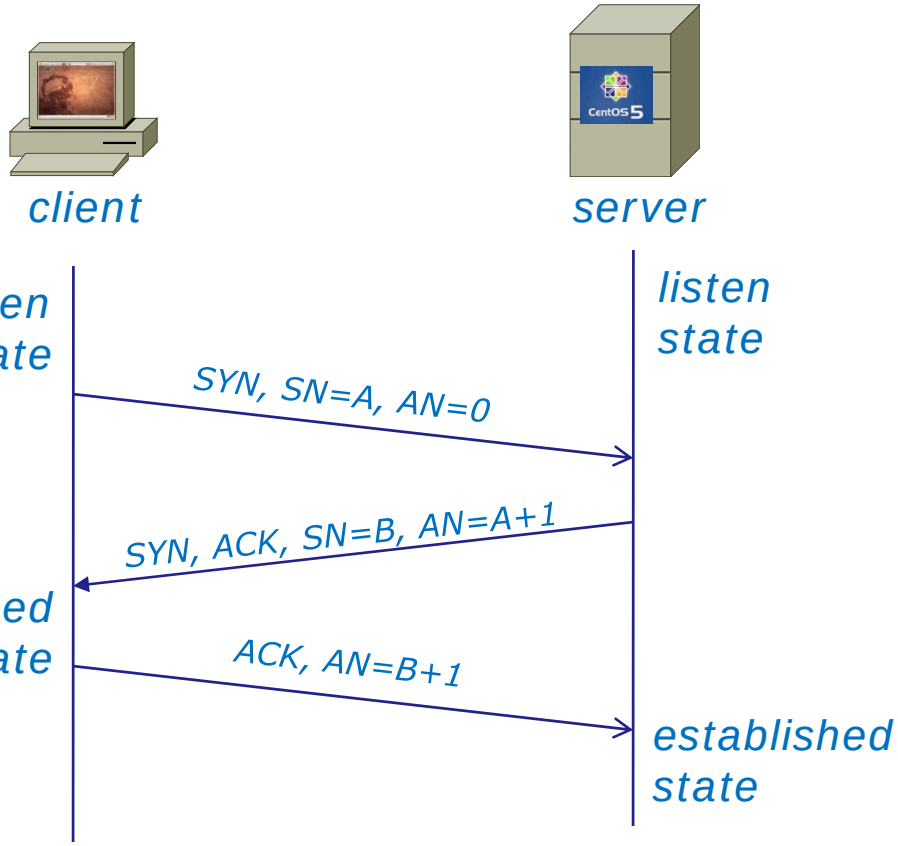
Options *like SACK permit selective acknowledgement*

Data *contains application specific information*

Transport Layer
Transmission Control Protocol

Initial Connection using a **Three-Way Handshake**

1. SYN
2. SYN-ACK
3. ACK



AN=Acknowledgment Number
SN=Sequence Number
ACK=ACK flag set

FTP

Active mode

- Client sends PORT command to indicate port it will listen on
- Server initiates new connection for data transfer to that port

PORT command to listen on port 166, 75
166 decimal = A6 hex
75 decimal = 4b hex
A64B hex = 42571 (decimal)

SIP	SP	DIP	DP	Protocol	Info
172.30.4.83	42855	192.168.2.150	21	FTP	Request: PORT 172,30,4,83,166,75
192.168.2.150	21	172.30.4.83	42855	FTP	Response: 200 PORT command successful. Consider using PAS
172.30.4.83	42855	192.168.2.150	21	FTP	Request: RETR legolas
192.168.2.150	20	172.30.4.83	42571	TCP	ftp-data > 42571 [SYN] Seq=0 Wi
172.30.4.83	42571	192.168.2.150	20	TCP	42571 > ftp-data [SYN, ACK] Seq
192.168.2.150	20	172.30.4.83	42571	TCP	ftp-data > 42571 [ACK] Seq=1 Ack
192.168.2.150	21	172.30.4.83	42855	FTP	Response: 150 Opening BINARY mode data connection for leg
192.168.2.150	20	172.30.4.83	42571	FTP-DATA	FTP Data: 18 bytes
192.168.2.150	20	172.30.4.83	42571	TCP	ftp-data > 42571 [FIN, ACK] Seq=19 Ack=1 Win=5888 Len=0
172.30.4.83	42571	192.168.2.150	20	TCP	42571 > ftp-data [ACK] Se
172.30.4.83	42571	192.168.2.150	20	TCP	42571 > ftp-data [FIN, AC
192.168.2.150	20	172.30.4.83	42571	TCP	ftp-data > 42571 [ACK] Seq=20 Ack=2 Win=5888 Len=0
192.168.2.150	21	172.30.4.83	42855	FTP	Response: 226 File send OK.
172.30.4.83	42855	192.168.2.150	21	TCP	42855 > ftp [ACK] Seq=82 Ack=263 Win=5856 Len=0

Retrieve legolas file

3 way handshake initiated by server

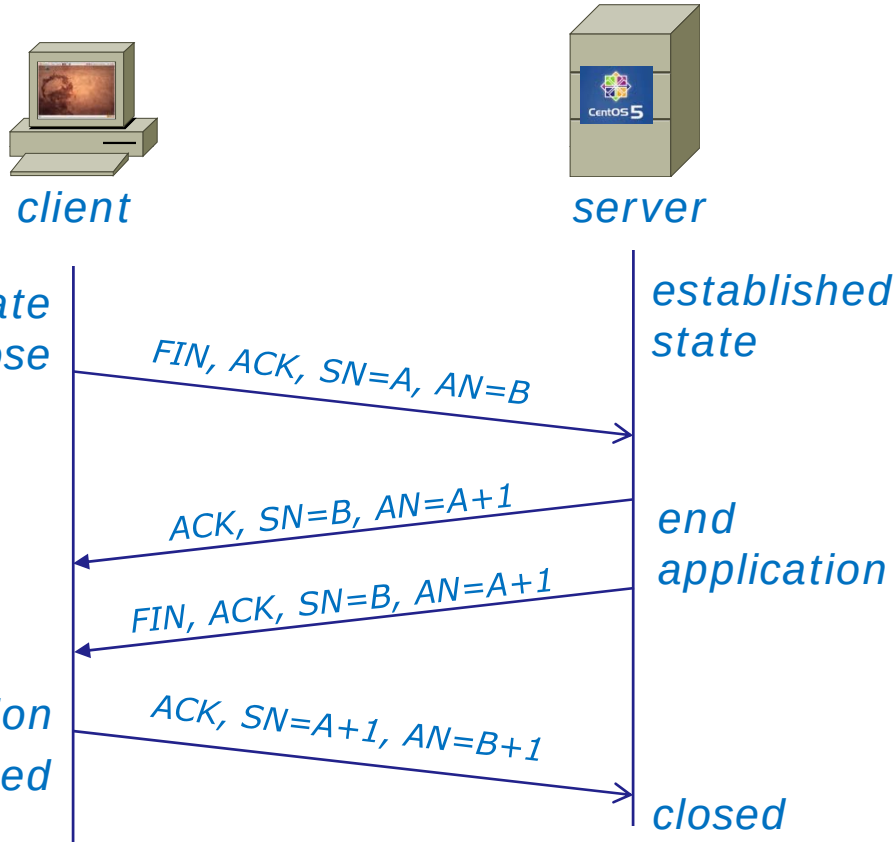
File transfer

4 way handshake to close connection

Transport Layer
Transmission Control Protocol

Closing a Connection with a **Four-Way Handshake**

1. FIN, ACK
2. ACK
3. FIN, ACK
4. ACK

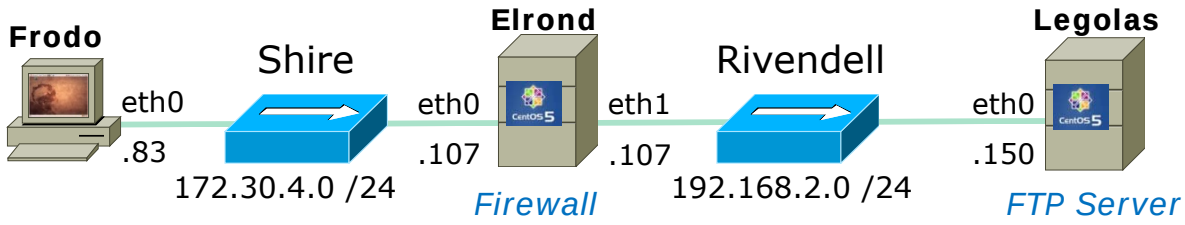


AN=Acknowledgment Number
SN=Sequence Number
ACK=ACK flag set
FIN=FIN flag set

Hidden Pictures

Snowy Feast

mitten
ice-cream bar
pencil
needle
apple
ladle
mushroom
bell
baseball bat
tack
sock
fishhook
mallet
fish
spoon
candle



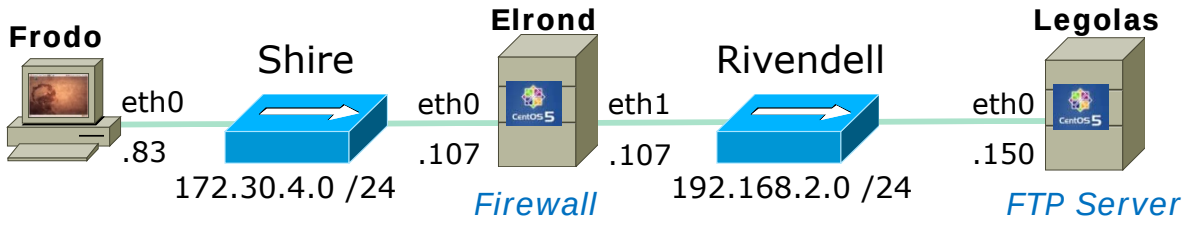
Active mode FTP data transfer from FTP server to client

Three way handshake
(to establish the connection)

Socket used for the transfer
(SIP, SP, DIP, DP)

Four way handshake
(to close the connection)

SIP	SP	DIP	DP	Protocol	Info
172.30.4.83	42855	192.168.2.150	21	FTP	Request: PORT 172,30,4,83,166,75
192.168.2.150	21	172.30.4.83	42855	FTP	Response: 200 PORT command successful. Consider using PAS
172.30.4.83	42855	192.168.2.150	21	FTP	Request: RETR legolas
192.168.2.150	20	172.30.4.83	42571	TCP	ftp-data > 42571 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 TSV=
172.30.4.83	42571	192.168.2.150	20	TCP	42571 > ftp-data [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MS
192.168.2.150	20	172.30.4.83	42571	TCP	ftp-data > 42571 [ACK] Seq=1 Ack=1 Win=5888 Len=0
192.168.2.150	21	172.30.4.83	42855	FTP	Response: 150 Opening BINARY mode data connection for leg
192.168.2.150	20	172.30.4.83	42571	FTP-DATA	FTP Data: 18 bytes
192.168.2.150	20	172.30.4.83	42571	TCP	ftp-data > 42571 [FIN, ACK] Seq=19 Ack=1 Win=5888 Len=0
172.30.4.83	42571	192.168.2.150	20	TCP	42571 > ftp-data [ACK] Seq=1 Ack=19 Win=5856 Len=0
172.30.4.83	42571	192.168.2.150	20	TCP	42571 > ftp-data [FIN, ACK] Seq=1 Ack=20 Win=5856 Len=0
192.168.2.150	20	172.30.4.83	42571	TCP	ftp-data > 42571 [ACK] Seq=20 Ack=2 Win=5888 Len=0
192.168.2.150	21	172.30.4.83	42855	FTP	Response: 226 File send OK.
172.30.4.83	42855	192.168.2.150	21	TCP	42855 > ftp [ACK] Seq=82 Ack=263 Win=5856 Len=0



Active mode FTP data transfer from FTP server to client

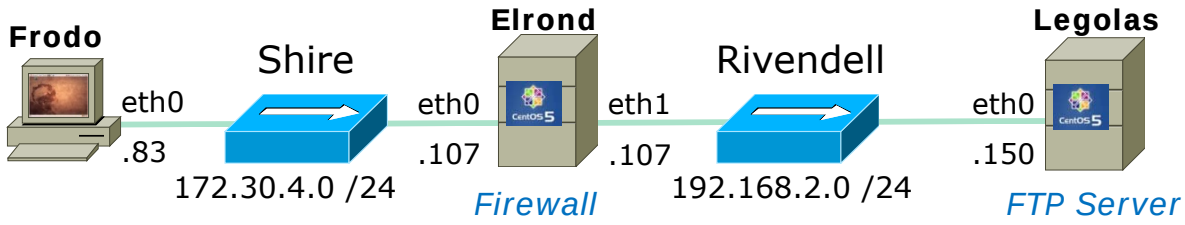
Three way handshake (to establish the connection)

Socket used for the transfer (SIP, SP, DIP, DP)

Four way handshake (to close the connection)

SIP	SP	DIP	DP	Protocol	Info
172.30.4.83	42855	192.168.2.150	21	FTP	Request: PORT 172,30,4,83,166,75
192.168.2.150	21	172.30.4.83	42855	FTP	Response: 200 PORT command successful. Consider using PAS
172.30.4.83	42855	192.168.2.150	21	FTP	Request: RETR legolas
192.168.2.150	20	172.30.4.83	42571	TCP	ftp-data > 42571 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 TSV=
172.30.4.83	42571	192.168.2.150	20	TCP	42571 > ftp-data [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MS
192.168.2.150	20	172.30.4.83	42571	TCP	ftp-data > 42571 [ACK] Seq=1 Ack=1 Win=5888 Len=0
192.168.2.150	21	172.30.4.83	42855	FTP	Response: 150 Opening BINARY mode data connection for leg
192.168.2.150	20	172.30.4.83	42571	FTP-DATA	FTP Data: 18 bytes
192.168.2.150	20	172.30.4.83	42571	TCP	ftp-data > 42571 [FIN, ACK] Seq
172.30.4.83	42571	192.168.2.150	20	TCP	42571 > ftp-data [ACK] Seq=1 Ac
172.30.4.83	42571	192.168.2.150	20	TCP	42571 > ftp-data [FIN, ACK] Seq
192.168.2.150	20	172.30.4.83	42571	TCP	ftp-data > 42571 [ACK] Seq=20 Ack=2 Win=5888 Len=0
192.168.2.150	21	172.30.4.83	42855	FTP	Response: 226 File send OK.
172.30.4.83	42855	192.168.2.150	21	TCP	42855 > ftp [ACK] Seq=82 Ack=263 Win=5856 Len=0

3 way handshake initiated by server



Active mode FTP data transfer from FTP server to client

Three way handshake
(to establish the connection)

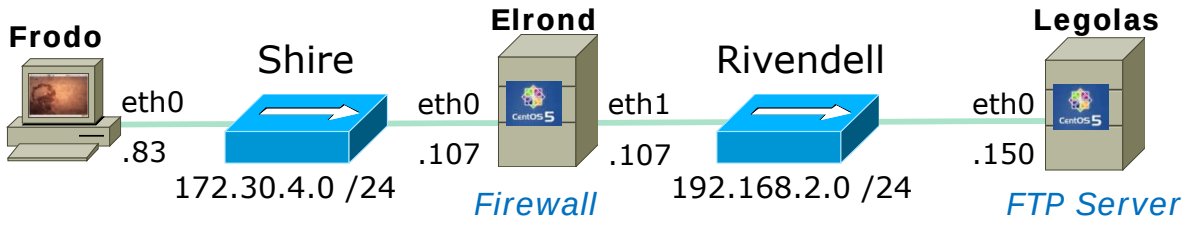
Socket used for the transfer
(SIP, SP, DIP, DP)

Four way handshake
(to close the connection)

SIP	SP	DIP	DP	Protocol	Info
172.30.4.83	42855	192.168.2.150	21	FTP	Request: PORT 172,30,4,83,166,75
192.168.2.150	21	172.30.4.83	42855	FTP	Response: 200 PORT command successful. Consider using PAS
172.30.4.83	42855	192.168.2.150	21	FTP	Request: RETR legolas
192.168.2.150	20	172.30.4.83	42571	TCP	ftp-data > 42571 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 TSV=
172.30.4.83	42571	192.168.2.150	20	TCP	42571 > ftp-data [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MS
192.168.2.150	**20**	**172.30.4.83**	**42571**	**TCP**	**ftp-data > 42571 [ACK] Seq=1 Ack=1 Win=5888 Len=0**
192.168.2.150	21	172.30.4.83	42855	FTP	Response: 150 Opening BINARY mode data connection for leg
192.168.2.150	20	172.30.4.83	42571	FTP-DATA	FTP Data: 18 bytes
192.168.2.150	20	172.30.4.83	42571	TCP	ftp-data > 42571 [FIN, ACK] Seq=19 Ack=1 Win=5888 Len=0
172.30.4.83	42571	192.168.			Seq=1 Ack=19 Win=5856 Len=0
172.30.4.83	42571	192.168.			ACK] Seq=1 Ack=20 Win=5856 Len=0
192.168.2.150	20	172.30.4			Seq=20 Ack=2 Win=5888 Len=0
192.168.2.150	21	172.30.4			d OK.
172.30.4.83	42855	192.168.			82 Ack=263 Win=5856 Len=0

Socket for data

Client	Server
172.30.4.83	192.168.2.150
42571	20



Active mode FTP data transfer from FTP server to client

Three way handshake
(to establish the connection)

Socket used for the transfer
(SIP, SP, DIP, DP)

Four way handshake
(to close the connection)

SIP	SP	DIP	DP	Protocol	Info
172.30.4.83	42855	192.168.2.150	21	FTP	Request: PORT 172,30,4,83,166,75
192.168.2.150	21	172.30.4.83	42855	FTP	Response: 200 PORT command successful. Consider using PAS
172.30.4.83	42855	192.168.2.150	21	FTP	Request: RETR legolas
192.168.2.150	20	172.30.4.83	42571	TCP	ftp-data > 42571 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 TSV=
172.30.4.83	42571	192.168.2.150	20	TCP	42571 > ftp-data [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MS
192.168.2.150	20	172.30.4.83	42571	TCP	ftp-data > 42571 [ACK] Seq=1 Ack=1 Win=5888 Len=0
192.168.2.150	21	172.30.4.83	42855	FTP	Response: 150 Opening BINARY mode data connection for leg
192.168.2.150	20	172.30.4.83	42571	FTP-DATA	FTP Data: 18 bytes
192.168.2.150	20	172.30.4.83	42571	TCP	ftp-data > 42571 [FIN, ACK] Seq=19 Ack=1 Win=5888 Len=0
172.30.4.83	42571	192.168.2.150	20	TCP	42571 > ftp-data [ACK] Seq=1 Ack=19 Win=5856 Len=0
172.30.4.83	42571	192.168.2.150	20	TCP	42571 > ftp-data [FIN, ACK] Seq=1 Ack=20 Win=5856 Len=0
192.168.2.150	20	172.30.4.83	42571	TCP	ftp-data > 42571 [ACK] Seq=20 Ack=2 Win=5888 Len=0
192.168.2.150	21	172.30.4.83	42855	FTP	Response: 226 File send OK.
172.30.4.83	42855	192.168.2.150	21	TCP	42855 > ftp [ACK] Seq=82

4 way handshake to close connection

Transport Layer

TCP Tunable Kernel Parameters

tcp_fin_timeout	*how long to keep in FIN-WAIT-2 state*
tcp_keepalive_time	*how long to keep an unused connection alive*
tcp_sack	*enable/disable selective acknowledgments*
tcp_timestamps	*enable RFC 1323 definition for round-trip measurement*
tcp_window_scaling	*enable RFC 1323 window scaling*
tcp_retries1	*how many times to retry before reporting an error*
tcp_retries2	*how many times to retry before killing connection*
tcp_syn_retries	*how many times to retransmit the SYN, ACK reply*

In the same directory:

ip_forward	*enable/disable selective acknowledgments*

Found in the **/proc/sys/net/ipv4** directory

TCP Tunable Kernel Parameters

```
[cis192@arwen ~]$ cat /etc/sysctl.conf
# Kernel sysctl configuration file for Red Hat Linux
#
# For binary values, 0 is disabled, 1 is enabled.  See sysctl(8) and
# sysctl.conf(5) for more details.

# Controls IP packet forwarding
net.ipv4.ip_forward = 0

# Controls source route verification
net.ipv4.conf.default.rp_filter = 1

# Do not accept source routing
net.ipv4.conf.default.accept_source_route = 0

< snipped >

[cis192@arwen ~]$

[cis192@arwen ~]$ cat /proc/sys/net/ipv4/conf/default/accept_source_route
0
[cis192@arwen ~]$ cat /proc/sys/net/ipv4/conf/default/rp_filter
1
[cis192@arwen ~]$ cat /proc/sys/net/ipv4/ip_forward
0
```

*Note how each entry in **/etc/sysctl.conf** corresponds to a **/proc/sys/net** file*

*Note: Use **sysctl -p** to put in effect any changes made to /etc/sysctl.conf*

Telnet Service and the xinetd super daemon

1. Install: **yum install telnet-server**
2. Configure: **/etc/xinetd.d/telnet (set disable = no)**
3. Firewall: **open TCP 23**
4. SELinux: **no change**
5. Start: **service xinetd start**
6. Automate: **chkconfig xinetd on**
7. Use and Monitor:
   - **ps -ef | grep telnetd**
   - **service xinetd status**
8. Troubleshoot:
   - cabling, interfaces
   - routing and forwarding
   - config file sytax and content
   - /var/log/messages
   - wireshark
   - firewall and selinux
   - universal fix (reboot)
9. Log files: **/var/log/messages**
10. Additional security **(firewall, tcp_wrappers, built-in)**

Access controls

- Configuration files
- TCP Wrappers
- Firewalls

Installing and Configuring Telnet

Edit the configuration file

```
[root@arwen ~]# cat /etc/xinetd.d/telnet
# default: on
# description: The telnet server serves telnet sessions; it uses \
#       unencrypted username/password pairs for authentication.
service telnet
{
        flags           = REUSE
        socket_type     = stream
        wait            = no
        user            = root
        only_from       = 192.168.0.23
        server          = /usr/sbin/in.telnetd
        log_on_failure  += USERID
        disable         = no
}
[root@arwen ~]#
```

Use only_from to restrict clients that can access the Telnet service

Installing and Configuring Telnet

```
only_from = arwen
```
hostname

```
only_from = arwen legolas
```
multiple hostnames

```
only_from = 192.168.3.12 192.168.3.14
```
or IP addresses

```
only_from = 192.168.3.{12, 14}
```
same as above

```
only_from = 192.168.0.0
```
0's are wildcards

```
only_from = sauron 172.30.4.0 10.10.10.{1, 200}
```
mixes

TCP Wrappers

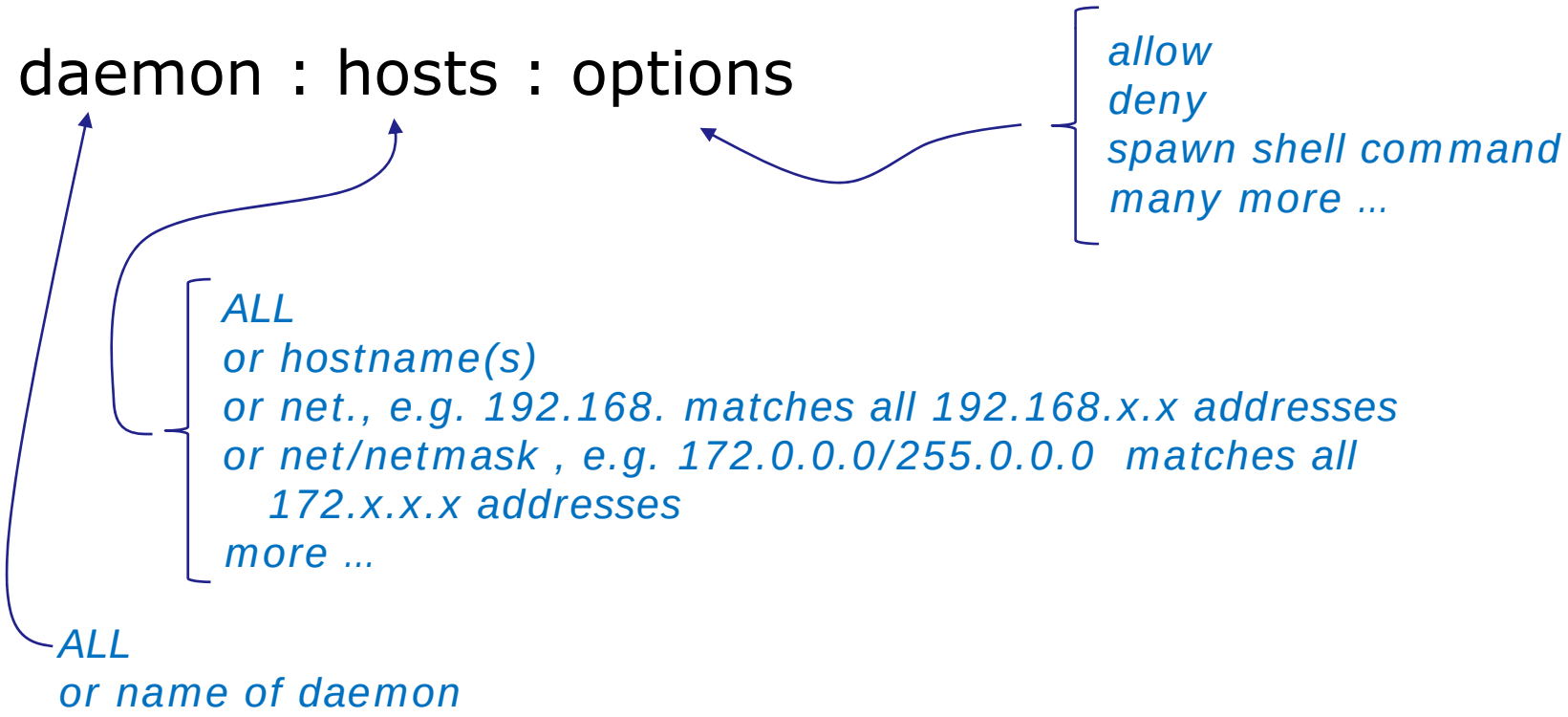
Access controls

- Implemented by the tcpd daemon
- **/etc/hosts.allow** – to specify hosts that may access services
- **/etc/hosts.deny** – to specify hosts that may not access services

Use ldd command on to see if daemon supports TCP Wrappers (i.e. libwrap has been compiled in)

TCP Wrappers

/etc/hosts.allow and **/etc/hosts.deny** syntax



TCP Wrapper Examples

```
[root@arwen ~]# cat /etc/hosts.allow
#
# hosts.allow   This file describes the names of the hosts which are
#               allowed to use the local INET services, as decided
#               by the '/usr/sbin/tcpd' server.
#
sshd: frodo
vsftpd: 172.30.
in.telnetd:  192.168.2.10  127.0.0.1
```

daemons *hosts*

```
[root@arwen ~]# cat /etc/hosts.deny
#
# hosts.deny    This file describes the names of the hosts which are
#               *not* allowed to use the local INET services, as decided
#               by the '/usr/sbin/tcpd' server.
#
# The portmap line is redundant, but it is left to remind you that
# the new secure portmap uses hosts.deny and hosts.allow.  In particular
# you should know that NFS uses portmap!

#deny everything
ALL: ALL
```

All daemons and all hosts

Firewall for Telnet

*Telnet port is **not** open*

CentOS

```
[root@arwen ~]# iptables -L  RH-Firewall-1-INPUT --line-numbers
Chain RH-Firewall-1-INPUT (2 references)
num  target     prot opt source               destination
1    ACCEPT     all  --  anywhere             anywhere
2    ACCEPT     icmp --  anywhere             anywhere            icmp any
3    ACCEPT     esp  --  anywhere             anywhere
4    ACCEPT     ah   --  anywhere             anywhere
5    ACCEPT     udp  --  anywhere             224.0.0.251         udp dpt:mdns
6    ACCEPT     udp  --  anywhere             anywhere            udp dpt:ipp
7    ACCEPT     tcp  --  anywhere             anywhere            tcp dpt:ipp
8    ACCEPT     all  --  anywhere             anywhere            state RELATED,ESTABLISHED
9    ACCEPT     tcp  --  anywhere             anywhere            state NEW tcp dpt:ssh
10   REJECT     all  --  anywhere             anywhere            reject-with icmp-host-
prohibited
[root@arwen ~]#
```

Firewall for Telnet

Open the telnet port by inserting at rule 9

```
[root@arwen ~]# iptables -I RH-Firewall-1-INPUT 9  -m state --
state NEW -m tcp -p tcp --dport 23 -j ACCEPT
[root@arwen ~]#
```

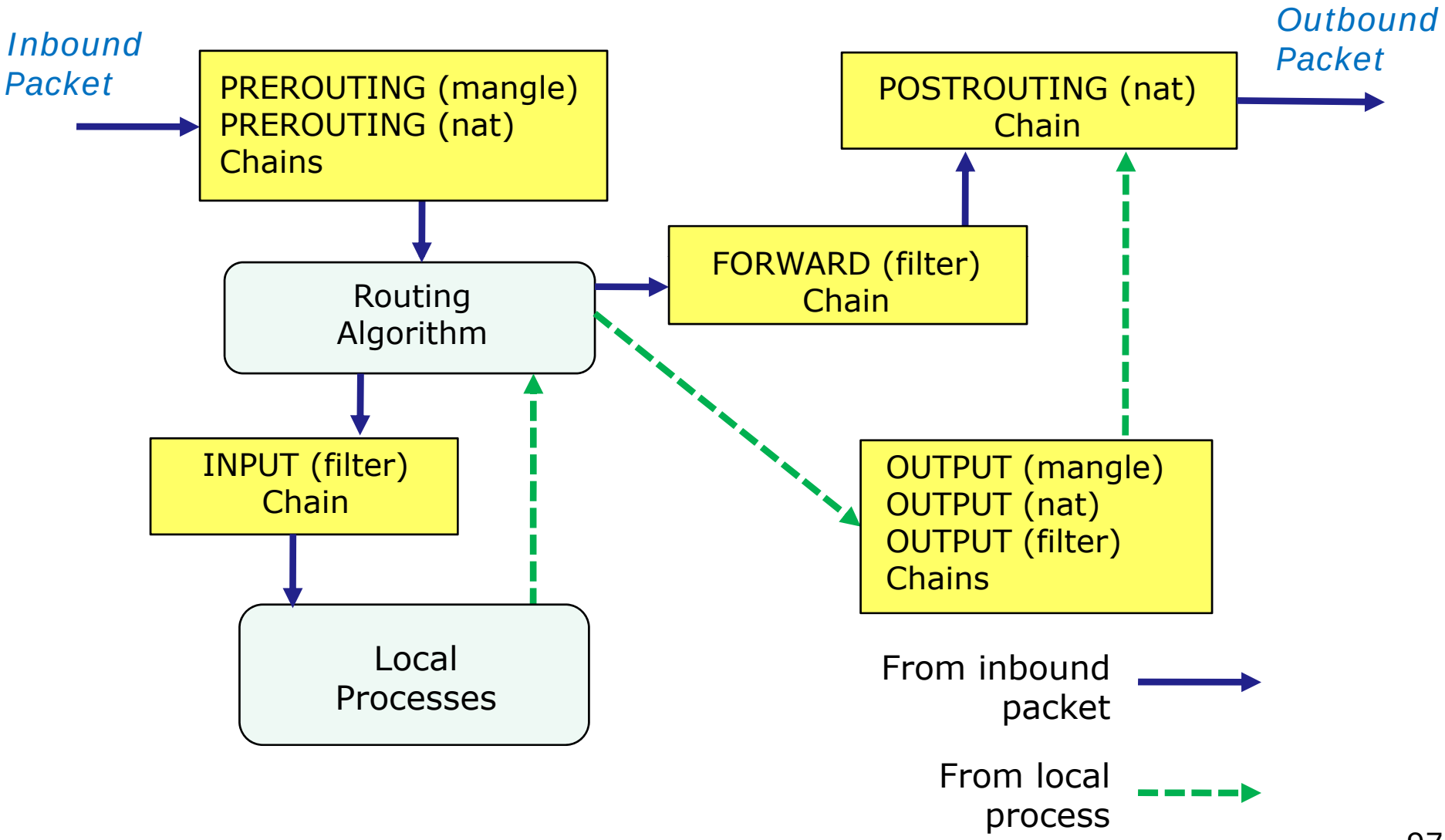
telnet=23

Firewall for Telnet

Telnet port is open

```
[root@arwen ~]# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
RH-Firewall-1-INPUT  all  --  anywhere             anywhere

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
RH-Firewall-1-INPUT  all  --  anywhere             anywhere

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain RH-Firewall-1-INPUT (2 references)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere
ACCEPT     icmp --  anywhere             anywhere            icmp any
ACCEPT     esp  --  anywhere             anywhere
ACCEPT     ah   --  anywhere             anywhere
ACCEPT     udp  --  anywhere             224.0.0.251         udp dpt:mdns
ACCEPT     udp  --  anywhere             anywhere            udp dpt:ipp
ACCEPT     tcp  --  anywhere             anywhere            tcp dpt:ipp
ACCEPT     all  --  anywhere             anywhere            state RELATED,ESTABLISHED
ACCEPT     tcp  --  anywhere             anywhere            state NEW tcp dpt:telnet
ACCEPT     tcp  --  anywhere             anywhere            state NEW tcp dpt:ssh
REJECT     all  --  anywhere             anywhere            reject-with icmp-host-prohibited
[root@arwen ~]#
```

Netfilter – all tables and chains



Netfilter – examples

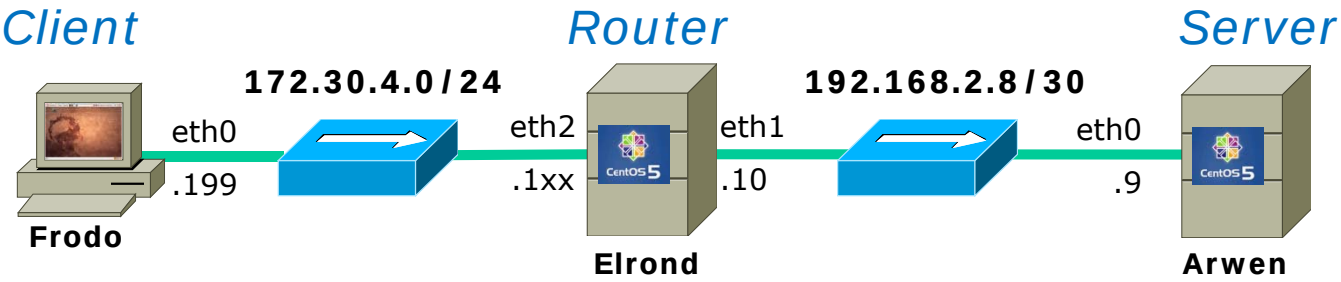
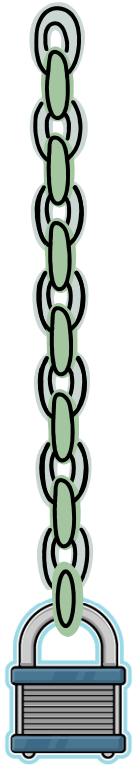


Table: filter
Chain: INPUT



Chain Rules:

-s 172.30.4.199/32 -j REJECT

Reject anything from Frodo

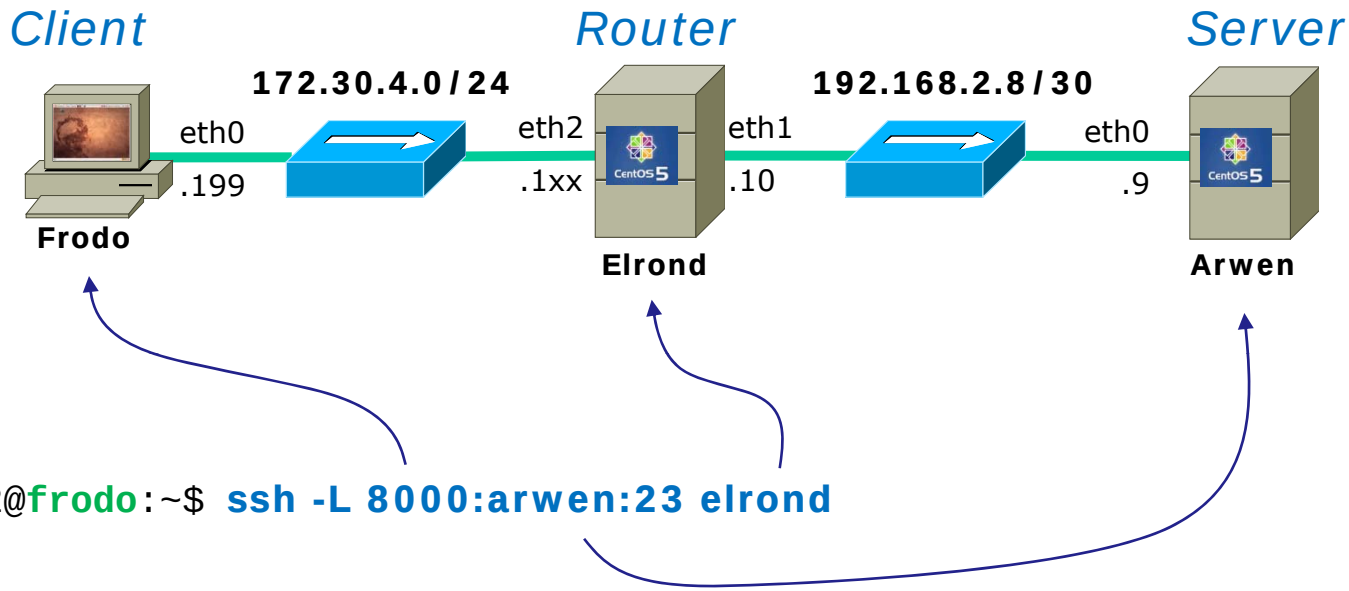
-s 192.168.0.0/16 –j ACCEPT

Accept all packets from 192.168.x.x

Chain Policy: DROP

DROP everything else

SSH Port Forwarding




```
cis192@frodo:~$ ssh -L 8000:arwen:23 elrond
```

Any connection made to port 8000 on Frodo will get forwarded to port 23 on Arwen via Elrond.

The portion of the connection between Frodo and Elrond will be encrypted

SSH Port Forwarding

Enable port forwarding in first terminal

```
cis192@elrond:~
File  Edit  View  Terminal  Tabs  Help
cis192@frodo:~$ ssh -L 8000:arwen:23 elrond
cis192@elrond's password:
Last login: Sun Mar 15 03:11:14 2009 from frodo
[cis192@elrond ~]$
```

Use port forwarding in second terminal

```
cis192@frodo: ~
File  Edit  View  Terminal  Tabs  Help
cis192@frodo:~$ telnet localhost 8000
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
CentOS release 5.2 (Final)
Kernel 2.6.18-92.1.22.el5 on an i686
login: cis192
Password:
Last login: Sun Mar 15 03:48:58 from elrond
[cis192@arwen ~]$ echo This is a secret!
This is a secret!
[cis192@arwen ~]$ exit
logout


Connection closed by foreign host.
cis192@frodo:~$
```

DHCP

DHCP Architecture

DHCP Servers

- Scopes and exclusions
- Reservations
- Leases
- Options
  - IP Address and Netmask
  - Gateway
  - DNS Server
  - Domain name
  - others

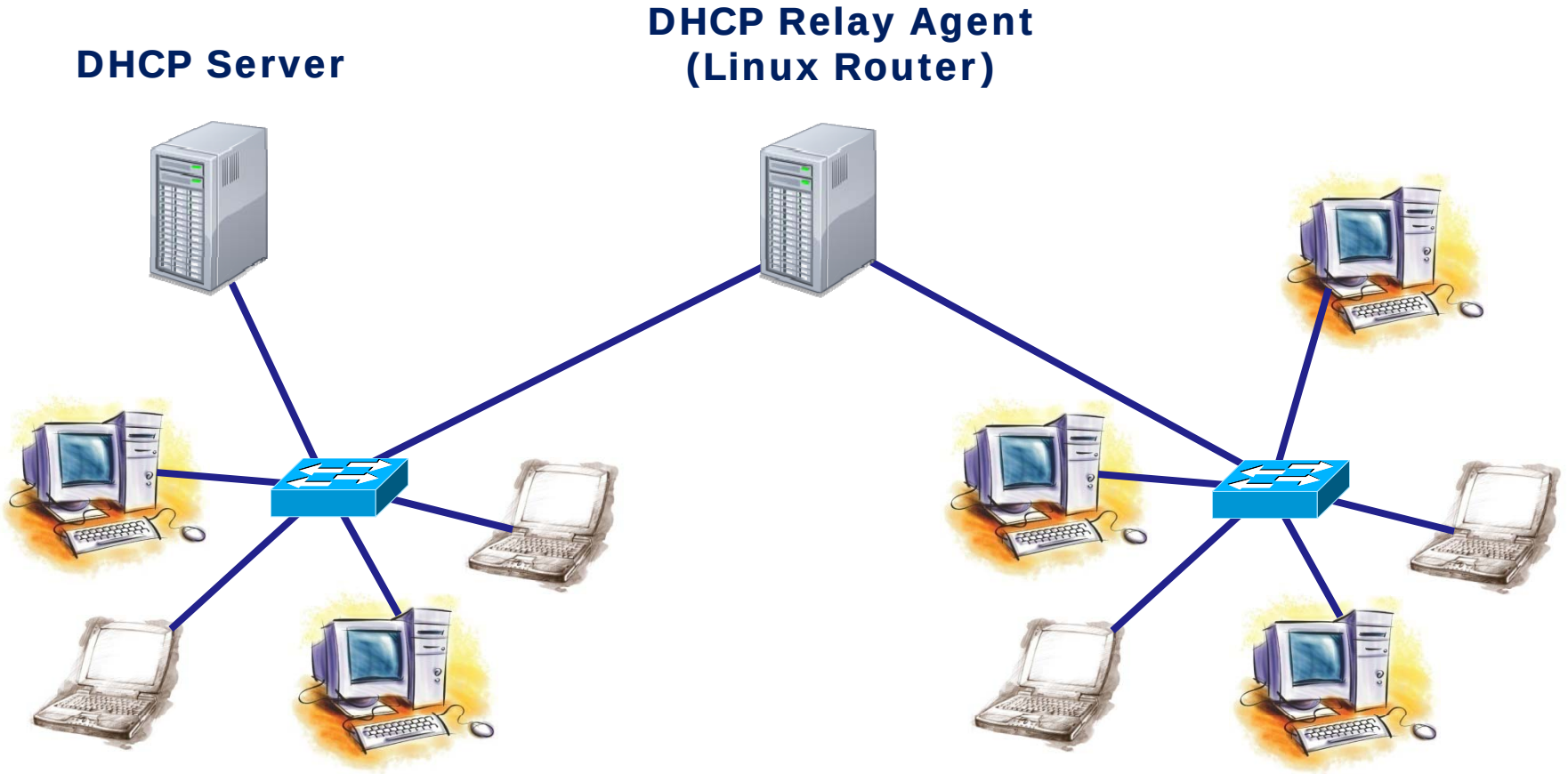
DHCP Relay Agents

DHCP Clients

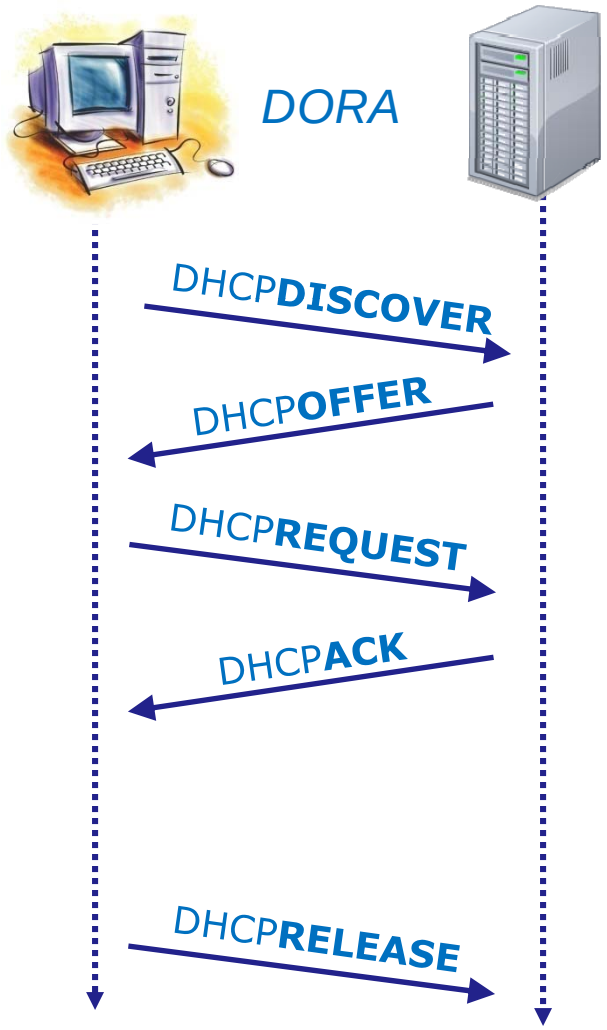
DHCP Clients lease IP addresses from DHCP Servers.

DHCP Relay agents lets one DHCP server service non-connected subnets

DHCP


DHCP Server
DHCP Relay Agent
(Linux Router)
DHCP Clients
DHCP Clients

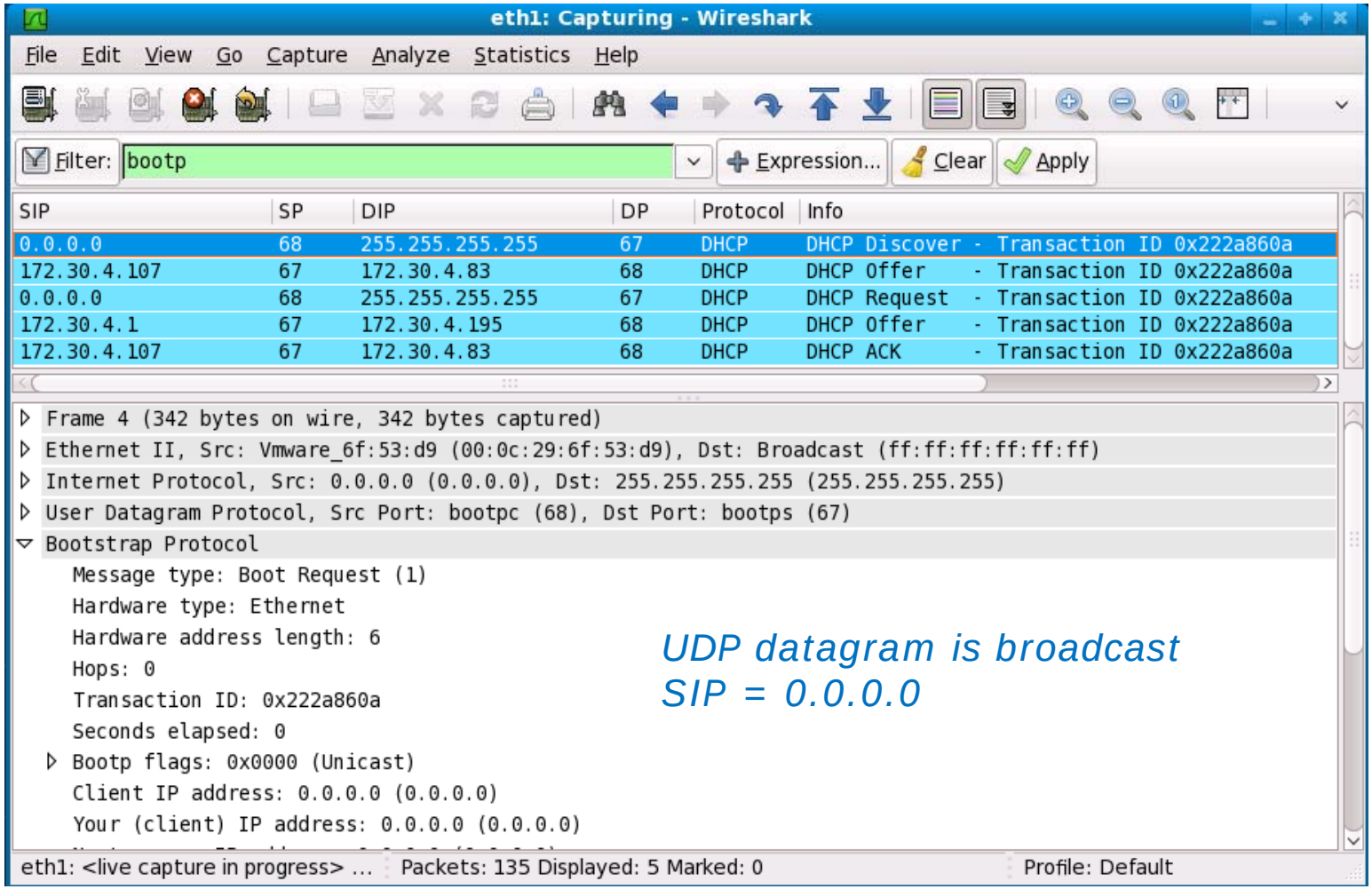
DHCP


DORA
DHCPDISCOVER
DHCPOFFER
DHCPREQUEST
DHCPACK
DHCPRELEASE

frodo

DHCPDISCOVER (broadcast)

Help, I need an IP address!



UDP datagram is broadcast
SIP = 0.0.0.0

elrond

DHCP

Global and specific settings for DHCP Lab Rivendell subnet

```
[root@elrond ~]# cat /etc/dhcpd.conf
ddns-update-style interim;
ignore client-updates;
option time-offset              -25200;  # Pacific Daylight Time (-7 HR)

#
#   R I V E N D E L L
#
subnet 192.168.2.0 netmask 255.255.255.0 {
        option routers                  192.168.2.1XX; # Default GW
        option subnet-mask              255.255.255.0;
        option domain-name              "rivendell";
        option domain-name-servers      207.62.187.53;

        range dynamic-bootp             192.168.2.50 192.168.2.99;
        default-lease-time              21600; # 6 hours
        max-lease-time                  43200; # 12 hours

        # reservations
        host legolas {
                hardware ethernet       00:0C:29:7C:18:F5;
                fixed-address           192.168.2.150;
        }
}
```

Will be the eth1 interface on your station's Elrond

DHCP

elrond

Settings for DHCP Lab Mordor subnet in /etc/dhcpd.conf

```
#
#   M O R D O R
#
subnet 192.168.3.0 netmask 255.255.255.0 {
        option routers                  192.168.3.150; # Default GW
        option subnet-mask              255.255.255.0;
        option domain-name              "mordor";
        option domain-name-servers      207.62.187.53;

        range dynamic-bootp             192.168.3.50 192.168.3.99;
        default-lease-time              21600; # 6 hours
        max-lease-time                  43200; # 12 hours
}
```

DHCP

elrond

Settings for DHCP Lab Shire subnet in /etc/dhcpd.conf

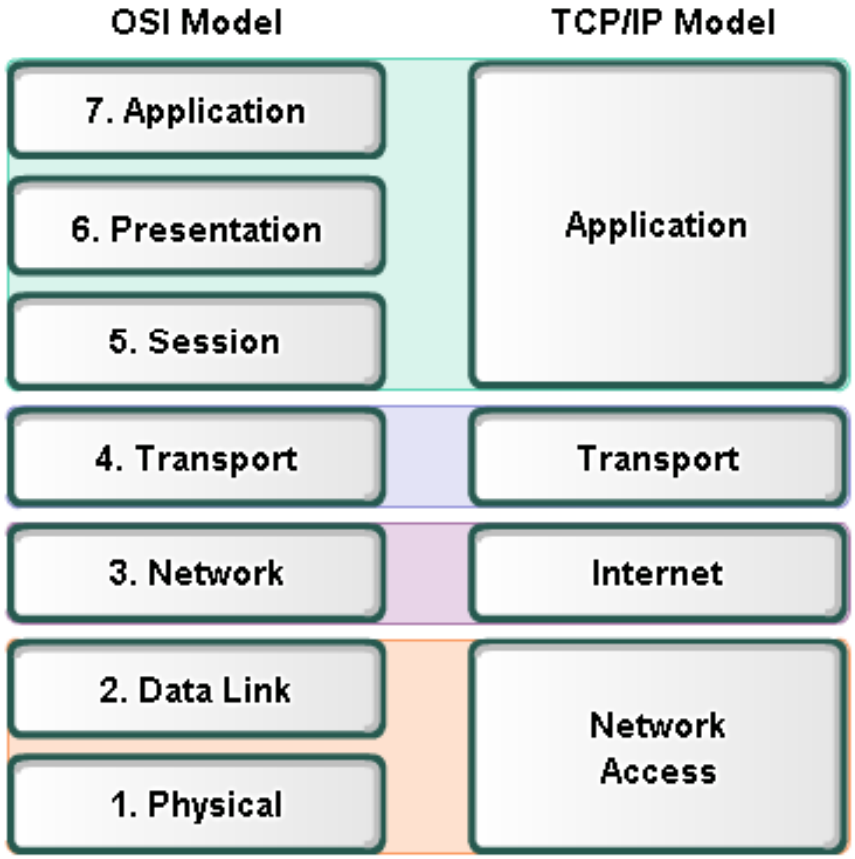
```
#
#   S H I R E
#
subnet 172.30.4.0 netmask 255.255.255.0 {
        option routers                  172.30.N.1;
        option subnet-mask              255.255.255.0;
        option domain-name              "shire";
        option domain-name-servers      207.62.187.53;

        range dynamic-bootp             172.30.N.80 172.30.N.84;
        default-lease-time              21600;
        max-lease-time                  43200;
}
[root@elrond ~]#
```

N=1 for the classroom and N=4 for the lab

Use the pool of addresses based on your station number to avoid conflicts!

PPP

Layer 2 Technologies



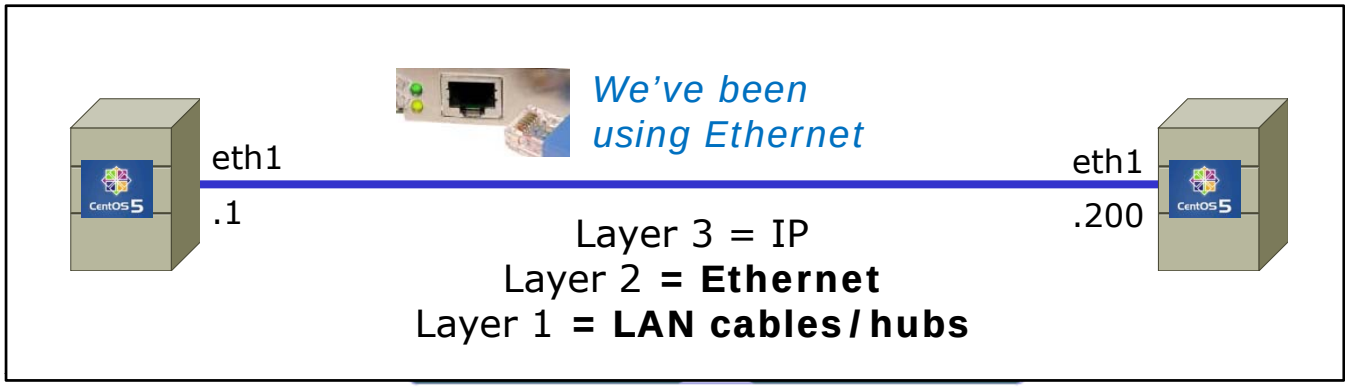
Layer 2 technologies

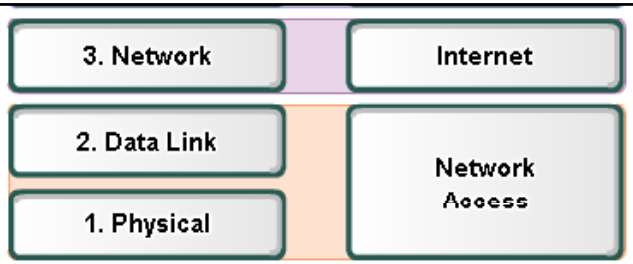
- X.25
- HIPPI
- Ethernet/IEEE 802.3
- Token Ring
- FDDI/CDDI
- Fibre Channel
- ATM
- PPP

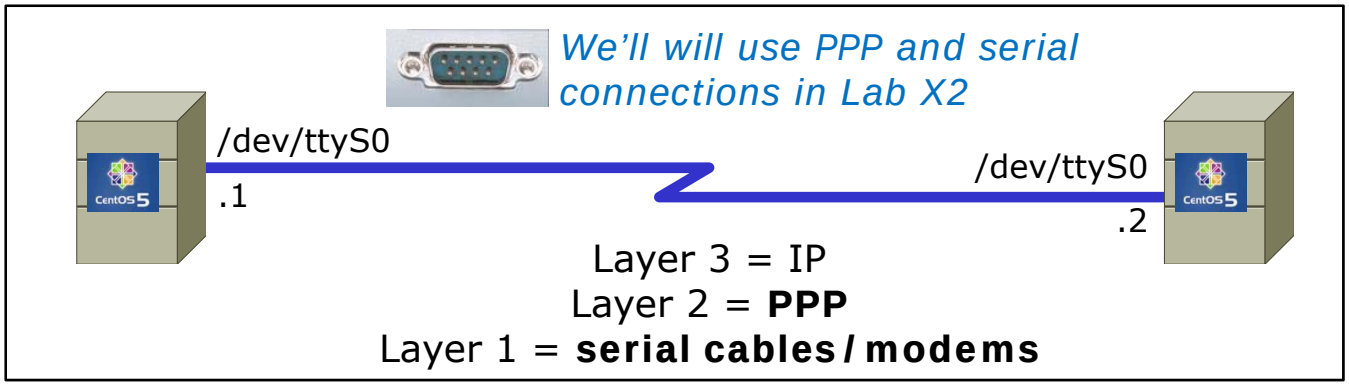
*Up to now we have been using **Ethernet** for Layer 2.*

*In LabX2 we will implement **PPP** over a serial connection.*

Layer 2 Technologies



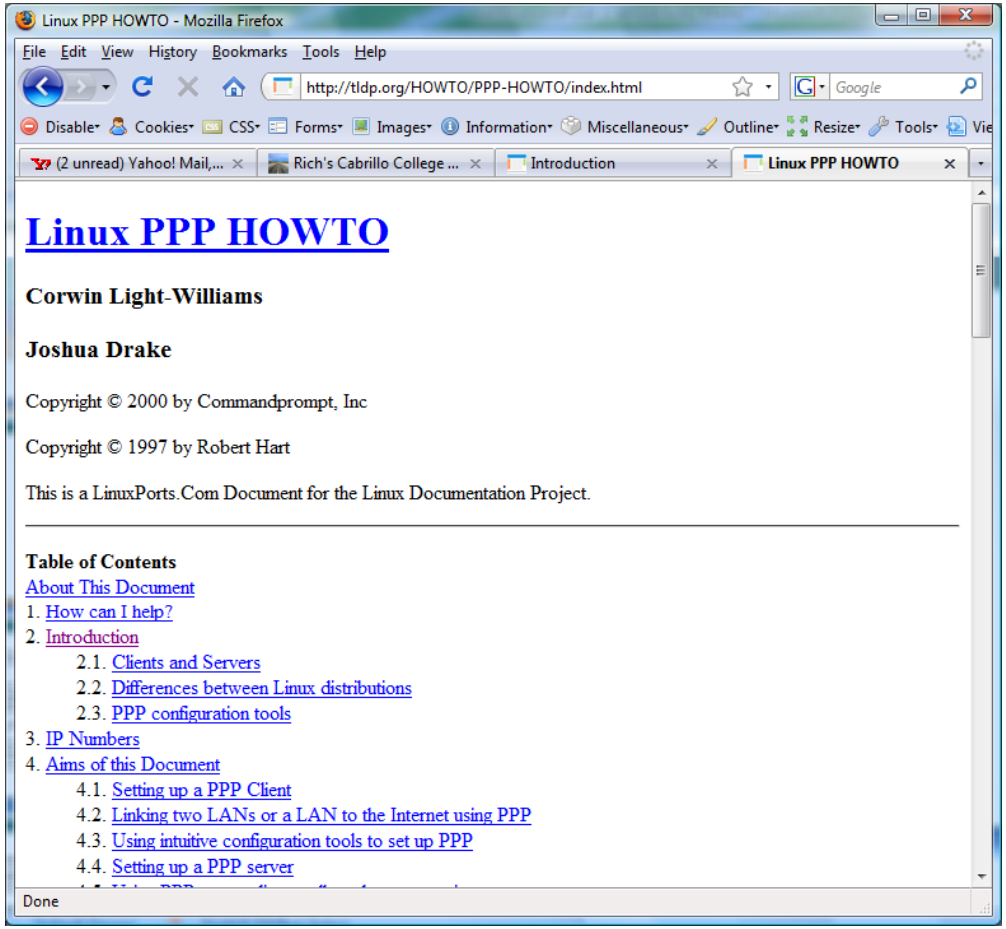




PPP is used rather than Ethernet for serial lines

PPP

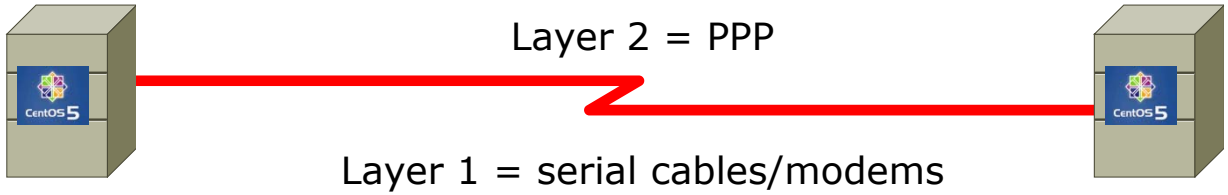
http://tldp.org/HOWTO/PPP-HOWTO/index.html



Lots of good information on PPP here!

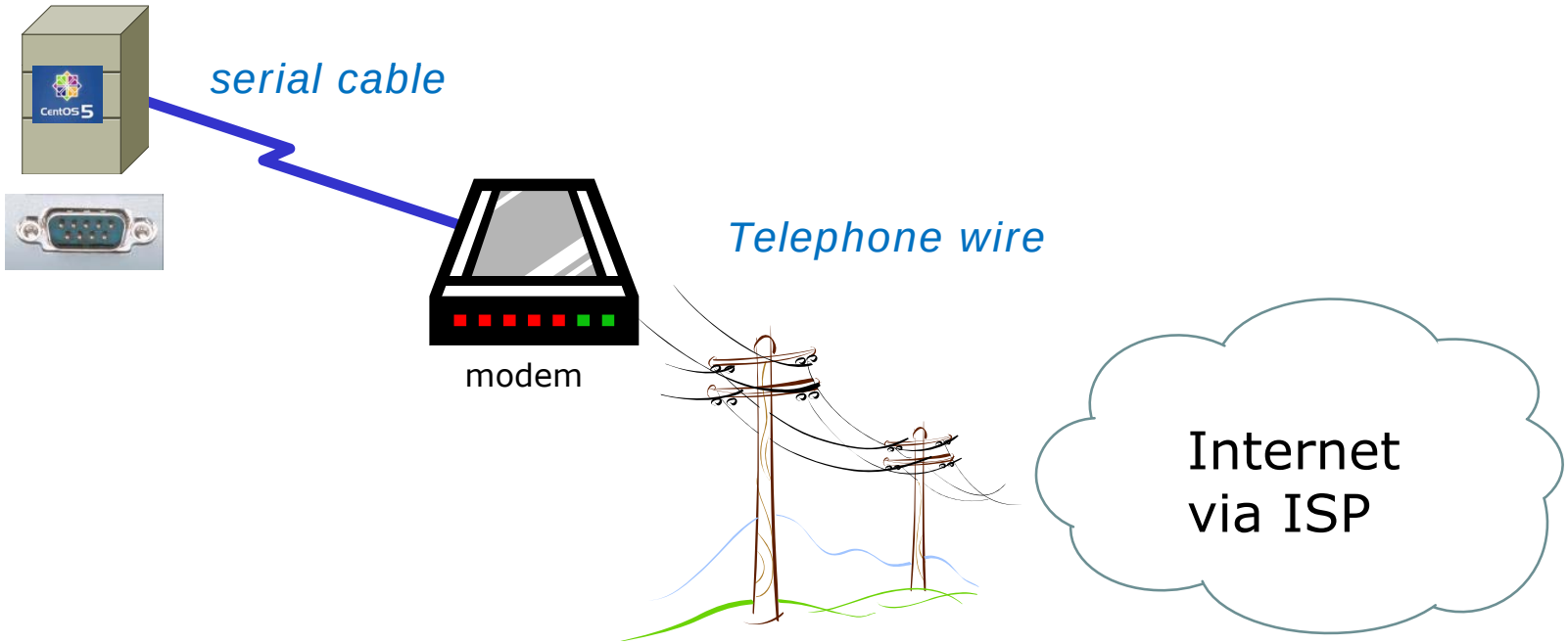
PPP

- PPP = Point to Point protocol (RFC 1331)
- A point to point network has only two hosts (at each end of the serial connection)
- PPP allows running IP and other network protocols over a serial link
- Serial links can be:
  - Direct connections using a null-modem cable
  - Using modems and telephones lines



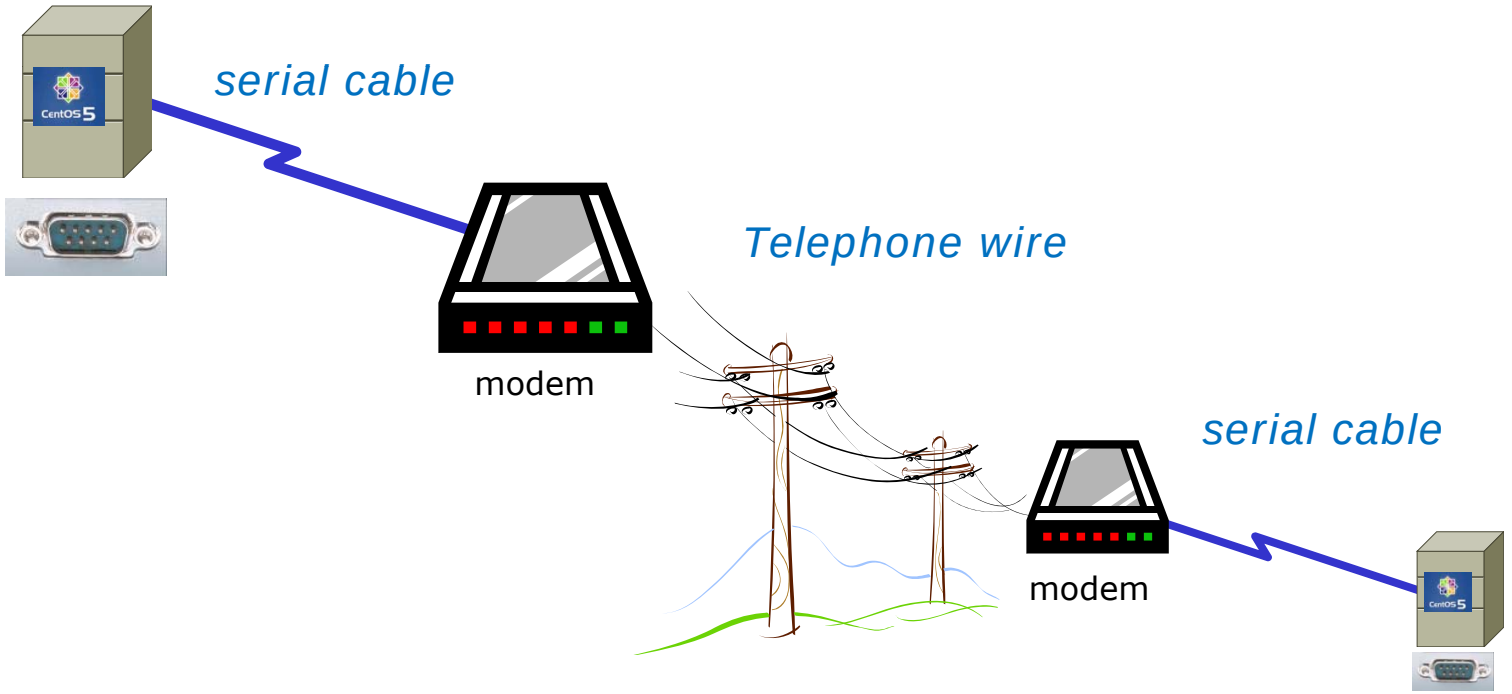
PPP

- PPP can be used as a dial-up connection to the Internet via your ISP



PPP

- PPP can be used as a WAN technology to connect LANs together



Features of PPP and SLIP

Both protocols offer the ability to send datagrams over a serial-line connection.

SLIP

- Works only with TCP/IP
- No error detection unless SLIP headers become corrupted
- Supports header compression only
- Supports only *clear-text* authentication

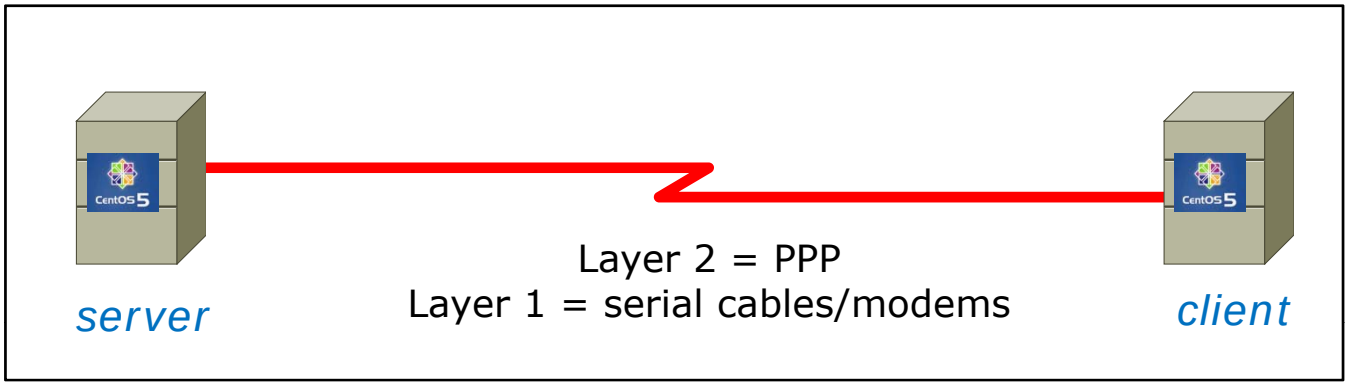
PPP

- Supports TCP/IP as well as UDP/IP, IPX/SPX, and Appletalk
- Built-in error detection
- Supports built-in data compression using the Van Jacobson compression algorithm
- Supports various authentication mechanisms e.g. PAP and CHAP

Password Authentication Protocol

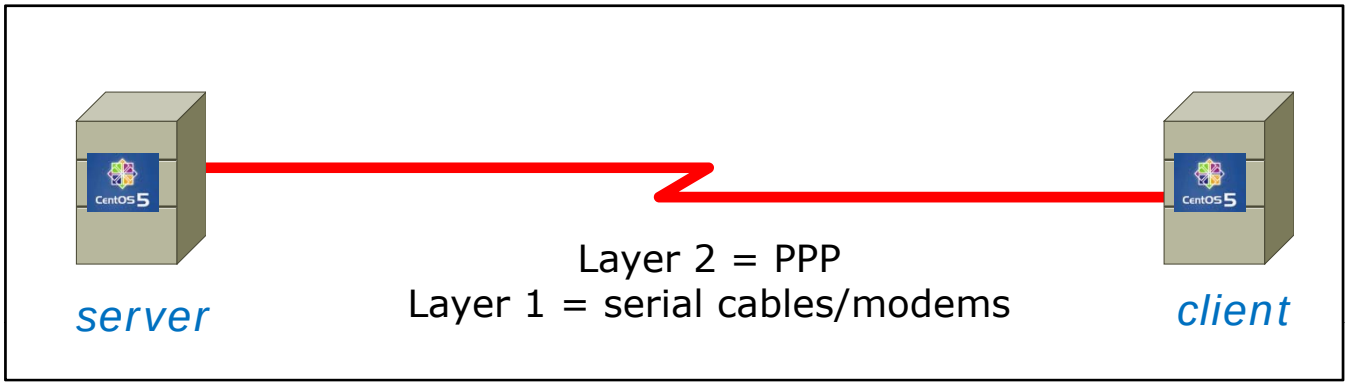
Challenge Handshake Authentication Protocol

PPP Architecture



- PPP is also called a *Peer-to-Peer* protocol because there is fundamentally no difference between the server and the client.
- The ppp daemons (services) must be running on both sides of the connection.
- The computer that initiates the call is called the client, the one who answers the call is the server.

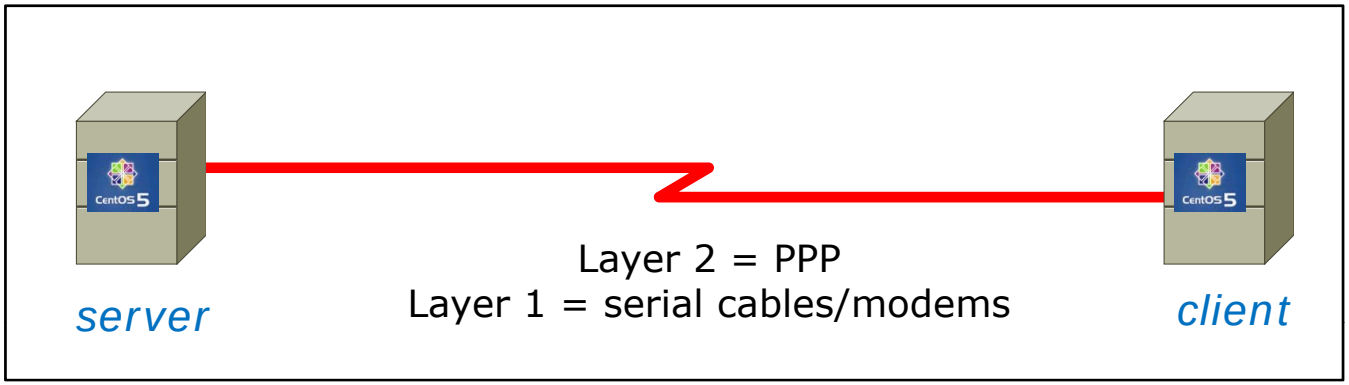
PPP Architecture (continued)



- Network Control Protocol (NCP) provides PPP with a means of differentiating between the different stacks it can transport, such as using IPCP for delivering TCP/IP packets.

- Authorization Protocol Provides a built-in authentication mechanism for PPP connections using either:
  - Password Authentication Protocol (PAP) or
  - Challenge Handshake Authentication Protocol (CHAP)

PPP Architecture (continued)



- Link Control Protocol (LCP) negotiates important link establishment options such as the maximum datagram size. Also helps to facilitate automated link establishment setup.

- High-level Data Link Control Protocol (HDLC) Provides frame boundary information and an added checksum for built-in error detection.

PPP Architecture

PPP runs as two major components:

1. Kernel portion - consists of and manages low-level protocols

```
[root@arwen ~]# lsmod | grep  "^ppp"
ppp_deflate            9793  2
ppp_async             15169  1
ppp_generic           30037  6 ppp_deflate,ppp_async
```

2. User portion - consists of and manages the authentication protocols
    - **pppd** - runs the various protocols
    - **chat** - provides automated dialing management for modem connections

Both of these programs rely on command line options and/or shell scripts to configure how they operate

Setting Up PPP

- Install the software if necessary which may require building and adding kernel modules:
    - Red Hat, CentOS and Ubuntu already have PPP kernel support out of the box.
    - Make sure the pppd service has been installed:

```
[root@arwen ~]# rpm -qa | grep ppp
ppp-2.4.4-2.el5
rp-pppoe-3.5-32.1
```

- Check your serial port
    - **setserial /dev/ttyS0** to look for modern, higher speed 16450A/16550A UART chip
    - **stty –a** to look for baud rate, parity and stop bits
- Configure your modem

setserial and stty commands

```
[root@arwen ~]# setserial /dev/ttyS0
/dev/ttyS0, UART: 16450, Port: 0x03f8, IRQ: 4
[root@arwen ~]#

[root@arwen ~]# stty -a
speed 38400 baud; rows 24; columns 80; line = 0;
intr = ^C; quit = ^\; erase = ^?; kill = ^U; eof = ^D; eol = M-^?; eol2 = M-^?;
swtch = M-^?; start = ^Q; stop = ^S; susp = ^Z; rprnt = ^R; werase = ^W;
lnext = ^V; flush = ^O; min = 1; time = 0;
-parenb -parodd cs8 hupcl -cstopb cread -clocal -crtscts -cdtrdsr
-ignbrk brkint -ignpar -parmrk -inpck -istrip -inlcr -igncr icrnl ixon -ixoff
-iuclc ixany imaxbel iutf8
opost -olcuc -ocrnl onlcr -onocr -onlret -ofill -ofdel nl0 cr0 tab0 bs0 vt0 ff0
isig icanon iexten echo echoe echok -echonl -noflsh -xcase -tostop -echoprt
echoctl echoke
[root@arwen ~]#
```

Has modern UART chip

38400 baud, no parity, data 8 bits, one stop bit, XON/XOFF flow control
*(use **man stty** for complete details)*

Lab X2

Lab X2 – Extra Credit Lab

Part 1 – Cable Arwen and Sauron using a virtual serial cable between their serial ports. Enable Arwen to accept login connections.

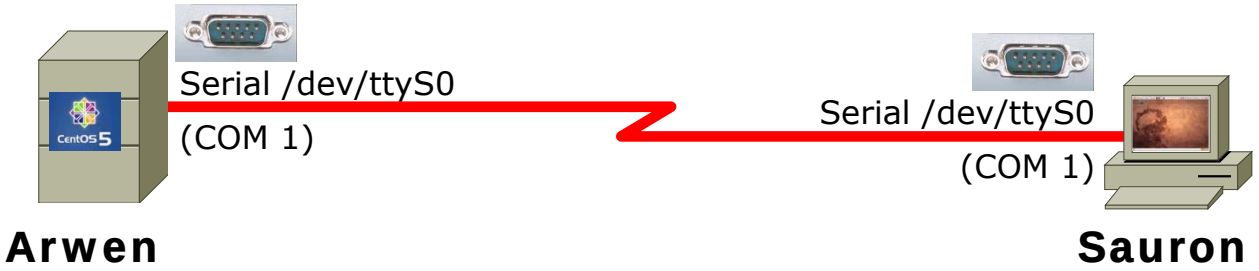
Part 2 – Manually login to Arwen from Sauron using the minicom terminal emulator over the serial connection.

Part 3 – Create a guest user account on Arwen and have it automatically run pppd at login (via a pppd command added to the end of it's .bash_profile script). Manually login from Arwen using minicom then manually run pppd on Sauron to establish a PPP level connection.

Part 4 – Automate making the connection with a script on Sauron that run pppd and uses the chat program to respond to the login request.

Exploring Serial Connections

Console port example with **minicom**



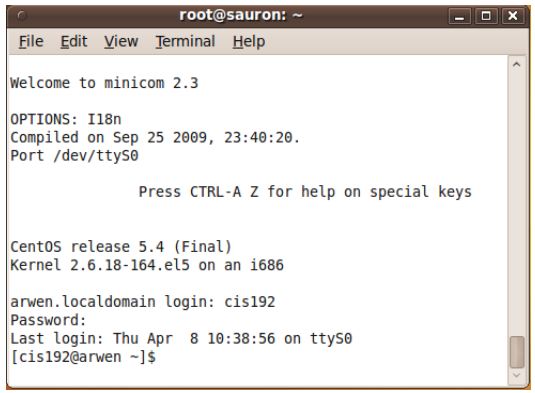
On Arwen, add this line to /etc/inittab:
`s1:35:respawn:/sbin/agetty 38400 ttyS0`

This enables the login process for any connections to the serial port /dev/ttyS0

On Sauron, configure minicom (a terminal emulator) to use:

- /dev/ttyS0
- 38400 baud
- 8 bits data
- no parity
- 1 stop bit
- hardware flow control

Note: PPP is not used for this, just using the serial connection for console access



Login to Arwen using ***minicom -o***

Exploring Serial Connections

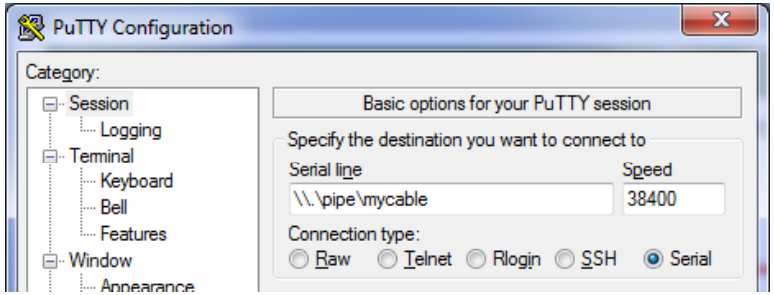
Console port example using **Putty**

Server

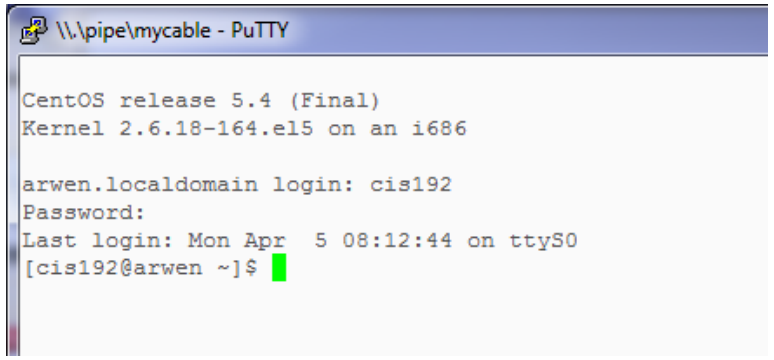
On windows station, configure Putty to use com port or pipe

On Arwen, add this line to /etc/inittab:

```
s1:35:respawn:/sbin/agetty 38400 ttyS0
```

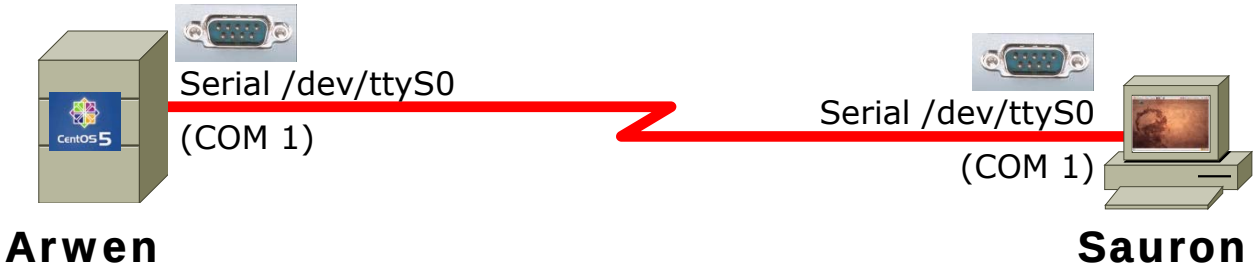


Note: PPP is not used for this, just using the serial connection for console access



Exploring Serial Connections

PPP example with bash_profile script on server, minicom on client (part 1)



On Arwen,

Add this line to /etc/inittab:

```
s1:35:respawn:/sbin/agetty 38400 ttyS0
```

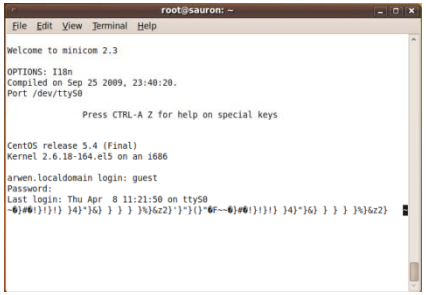
Add a user guest that runs this command at login (added to bash_profile):

```
/usr/sbin/pppd -detach crtscts
proxyarp 10.0.0.1:10.0.0.2 /dev/ttyS0
38400
```

(all on one line)

pppd must be run on both ends to establish the connection

On Sauron,



Login as guest on Arwen using ***minicom -o***

Exit minicom and run this command quickly:

```
pppd -detach crtscts
/dev/ttyS0 38400 &
```

(all on one line)

Exploring Serial Connections

PPP example with bash_profile script on server, minicom on client (part 2)

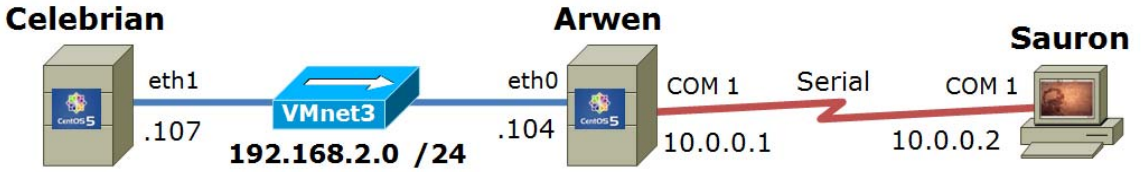
On Sauron,

```
root@sauron:~# pppd -detach crtscts /dev/ttyS0 38400 &
[1] 1675
root@sauron:~# Using interface ppp0
Connect: ppp0 <--> /dev/ttyS0
Deflate (15) compression enabled
Cannot determine ethernet address for proxy ARP
local  IP address 10.0.0.2
remote IP address 10.0.0.1

root@sauron:~# ifconfig
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:4 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:240 (240.0 B)  TX bytes:240 (240.0 B)

ppp0      Link encap:Point-to-Point Protocol
          inet addr:10.0.0.2  P-t-P:10.0.0.1  Mask:255.255.255.255
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:5 errors:0 dropped:0 overruns:0 frame:0
          TX packets:5 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:3
          RX bytes:69 (69.0 B)  TX bytes:75 (75.0 B)

root@sauron:~#
```

PPP connection established, not both the local IP address and remote IP address are shown in ifconfig output

Lab X2

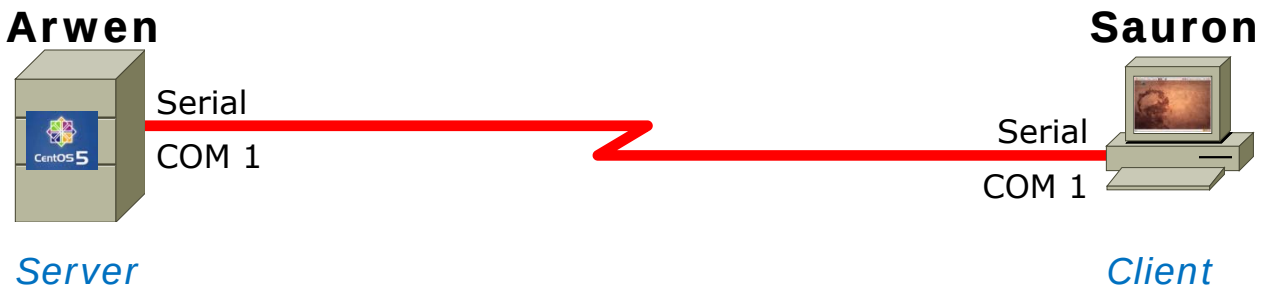
Using a named pipe for the virtual null modem cable between the two serial COM ports



Using Ethernet as the LAN layer 2 protocol over the hub and LAN cables

Using PPP as the WAN layer 2 protocol over the serial connection

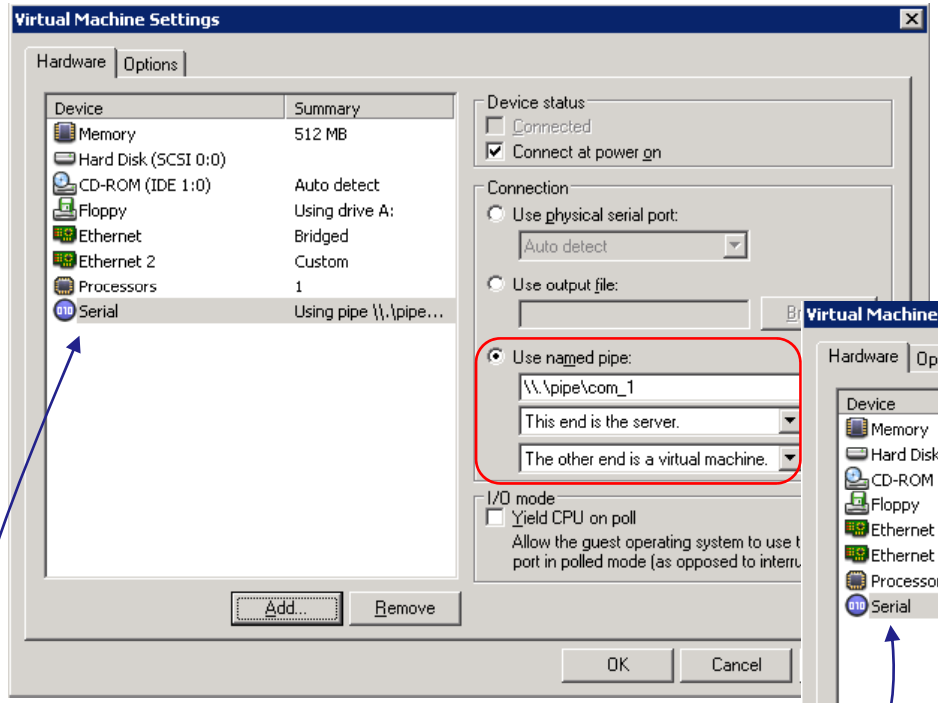
Lab X2 – Serial connections



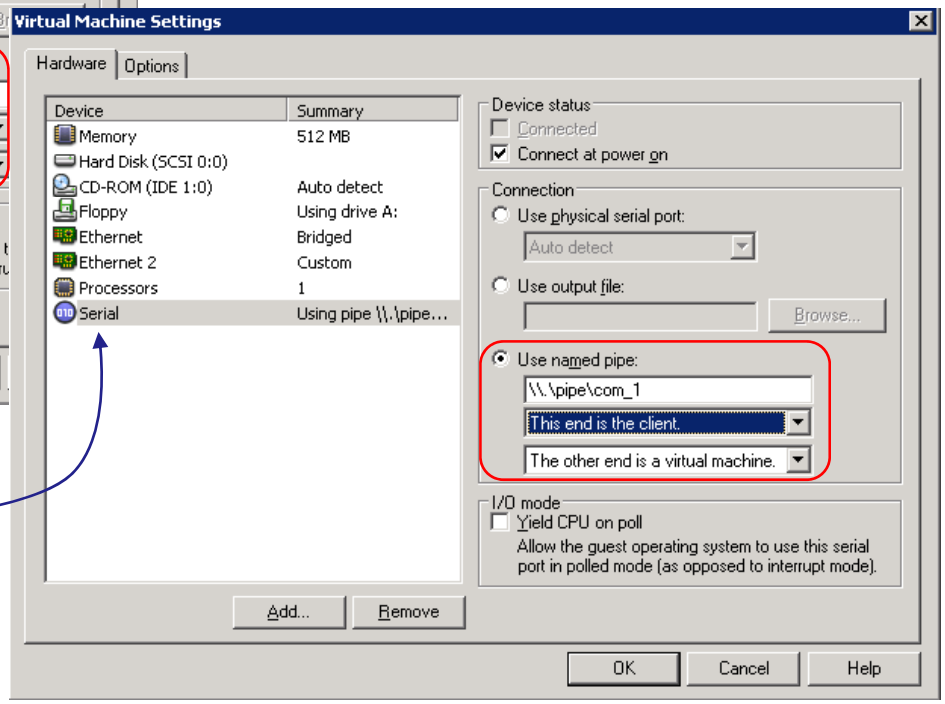
- *If you use real computers to do Lab X2, then you would connect the COM ports using a **null modem cable***
- *If you use VMware or VirtualBox VMs, then you would make a virtual serial connection using **OS pipes***

Lab X2 – Serial connections with VMware Server

Arwen (the server end)



Sauron (the client end)

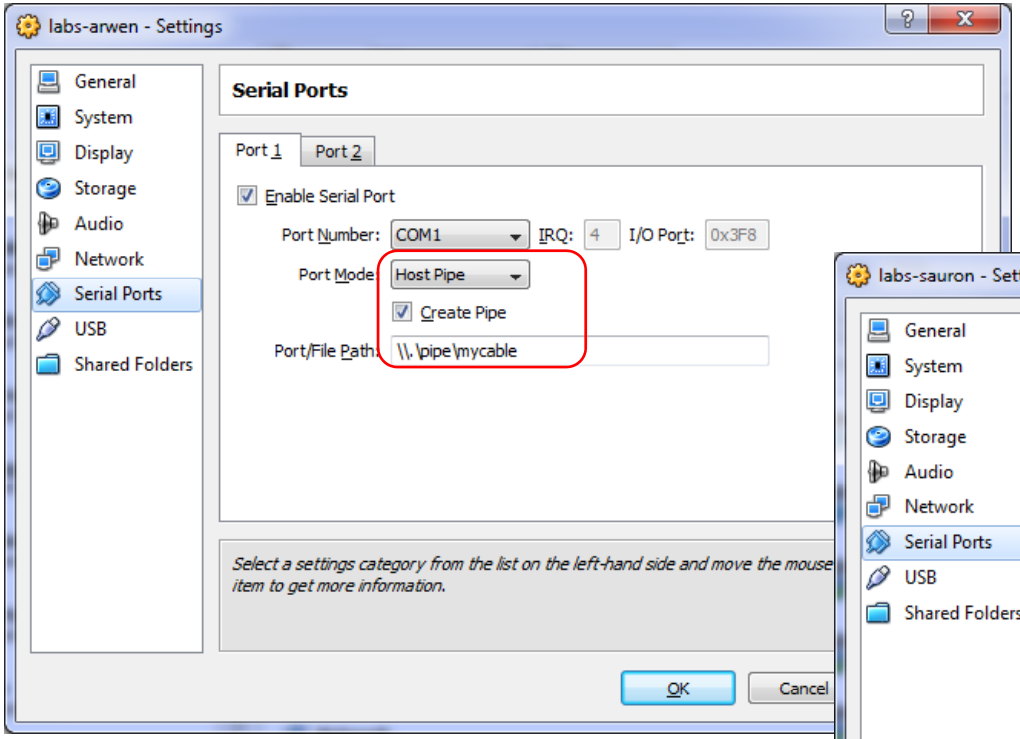


Use the Hardware Wizard to add serial ports

Lab X2 – Serial connections with VirtualBox on Windows

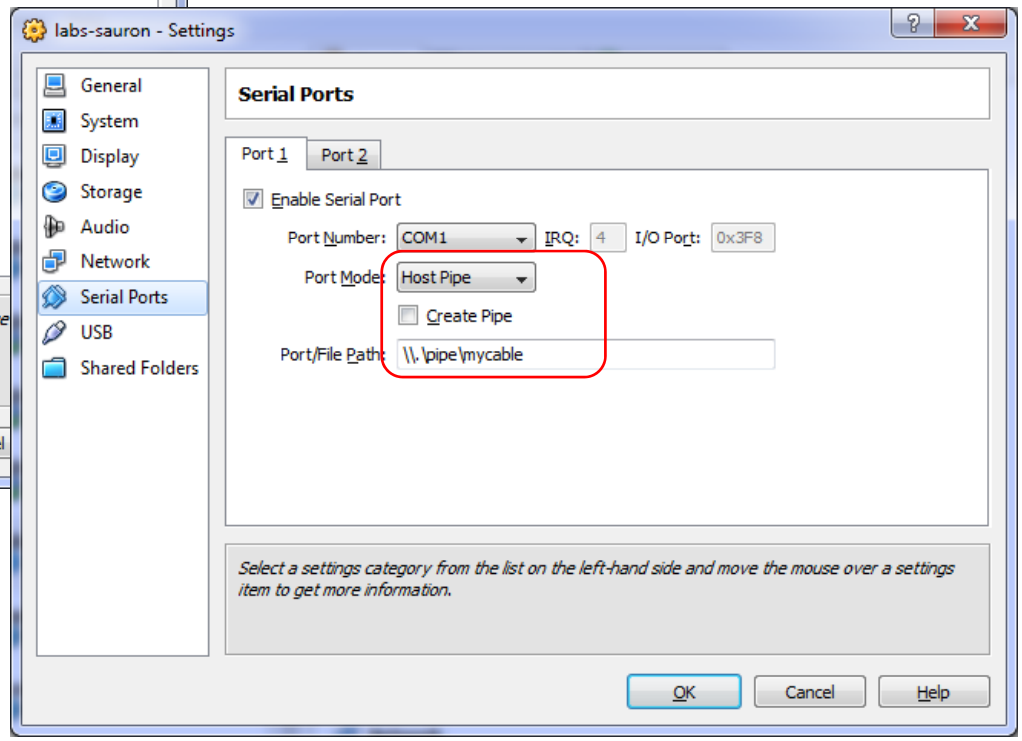
Arwen

(the "server" end of the connection so Create Pipe is checked)



Sauron

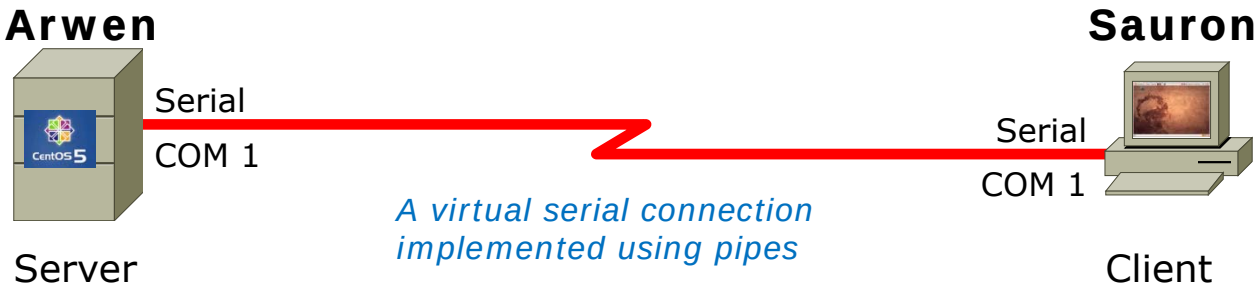
(the "client" end so Create Pipe is NOT checked)

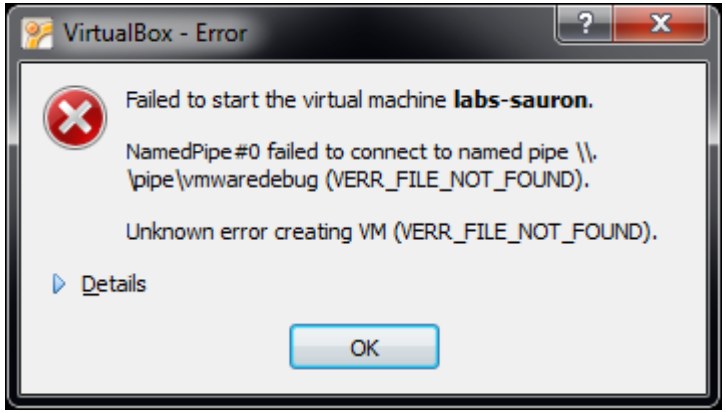


Use the Serial Ports Settings to configure COM1 to use the named pipe ***\\.\pipe\mycable*** *on the host*

Note: pipes are used by an operating system to enable inter-process communication

Lab X2 – Serial connections with VirtualBox on Windows





Note: Sauron does not create the pipe used to create the virtual serial connection. This error message will be displayed if Arwen is not running. So startup Arwen before Sauron when using VirtualBox

Lab X2

In the DOS/Windows world serial ports are called COM 1, COM 2, etc.

```
[root@arwen ~]# ls -l /dev/ttyS?
crw--w---- 1 ppp  tty  4, 64 Mar 25 06:56 /dev/ttyS0
crw-rw---- 1 root uucp 4, 65 Mar 24 16:39 /dev/ttyS1
crw-rw---- 1 root uucp 4, 66 Mar 24 16:39 /dev/ttyS2
crw-rw---- 1 root uucp 4, 67 Mar 24 16:39 /dev/ttyS3
[root@arwen ~]#
```

Each serial port is considered by UNIX to be a device. In the past these serial ports were used to connect terminals. Teletypes were terminals without a screen (had a keyboard and printer).

Note: DOS COM1 = Linux /dev/ttyS0

Lab X2 Commands

Lab X2

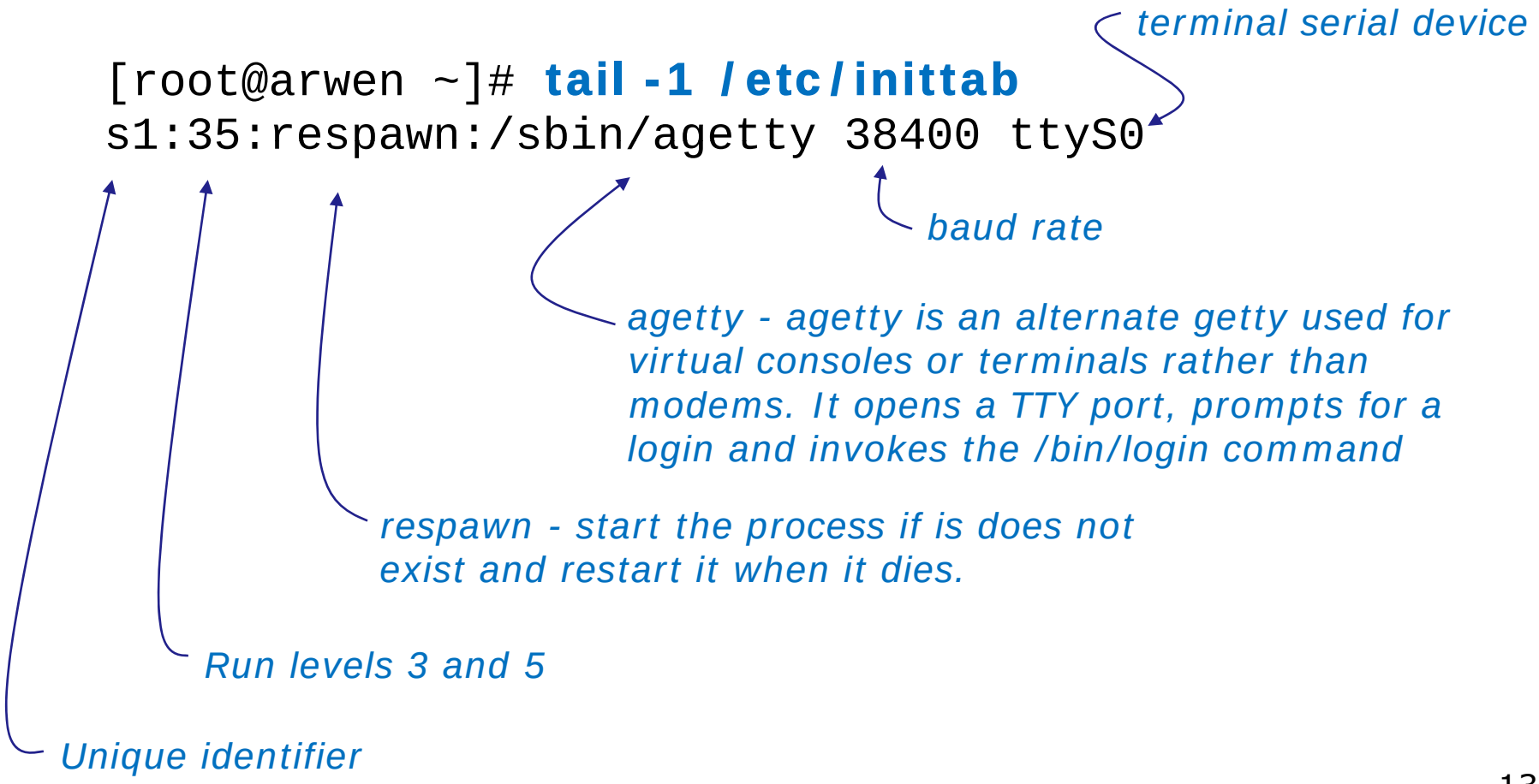
This is COM 1 on Linux

```
[root@arwen ~]# setserial /dev/ttyS0
/dev/ttyS0, UART: 16550A, Port: 0x03f8, IRQ: 4
[root@arwen ~]#
```

The setserial command sets or reports on serial port configuration.

Lab X2

Handling the login process on the pppd server



Lab X2

Handling the login process on the pppd server

```
[root@arwen ~]# telinit q
```

*Tells init to reread the **/etc/inittab** file after making changes*

Lab X2

```
[root@arwen ~]# chmod u+s /usr/sbin/pppd
[root@arwen ~]# ls -l /usr/sbin/pppd
-r-sr-xr-x 1 root root 312236 Mar 14  2007 /usr/sbin/pppd
```

*This sets a special permission called the **setuid** bit. This allows users to run an executable with the permissions of the executable's owner.*

```
[root@arwen ~]# stat /usr/sbin/pppd
  File: `/usr/sbin/pppd'
  Size: 312172          Blocks: 632        IO Block: 4096
regular file
Device: fd00h/64768d    Inode: 308263      Links: 1
Access: (4555/-r-sr-xr-x)  Uid: (    0/    root)   Gid: (
0/    root)
Access: 2010-04-04 03:20:12.000000000 -0700
Modify: 2009-01-20 20:27:13.000000000 -0800
Change: 2010-04-04 19:45:23.000000000 -0700
```

*FYI, the **stat** command provides additional inode information about a file than a long listing (ls –l) does.*

Lab X2

minicom
is a small terminal emulator with a dialing capability

```
[root@arwen ~]# minicom -s
                -o
```

-s option is used to setup defaults which are saved in ***/etc/minicom/minirc.dfl***

-o option prevents initialization. Useful for restarting a session

Use ***apt-get install minicom*** *to install on Ubuntu*

Lab X2

minicom

is a small terminal emulator with a dialing capability

```
root@sauron:~# minicom -s
```

Select choice and hit Enter

```
+-----[configuration]------+
| Filenames and paths      |
| File transfer protocols  |
| Serial port setup        |
| Modem and dialing        |
| Screen and keyboard      |
| Save setup as dfl        |
| Save setup as..          |
| Exit                     |
| Exit from Minicom        |
+--------------------------+
```

Use Escape to go back up one level
Use Enter to make sections
Use Letters to make choices

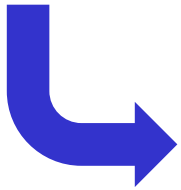
```
+-----------------------------------------------------------------------+
| A -    Serial Device      : /dev/tty8                                 |
| B - Lockfile Location     : /var/lock                                 |
| C -   Callin Program      :                                           |
| D -  Callout Program      :                                           |
| E -    Bps/Par/Bits       : 115200 8N1                                |
| F - Hardware Flow Control : Yes                                       |
| G - Software Flow Control : No                                        |
|                                                                       |
|    Change which setting?                                              |
+-----------------------------------------------------------------------+
        | Screen and keyboard      |
        | Save setup as dfl        |
        | Save setup as..          |
        | Exit                     |
        | Exit from Minicom        |
        +--------------------------+
```

Select option and type new configuration value

Lab X2

```
+-----------------------------------------------------------------------+
| A -    Serial Device      : /dev/ttyS0                                |
| B - Lockfile Location     : /var/lock                                 |
| C -   Callin Program      :                                           |
| D -  Callout Program      :                                           |
| E -    Bps/Par/Bits       : 38400 8N1                                 |
| F - Hardware Flow Control : Yes                                       |
| G - Software Flow Control : No                                        |
|                                                                       |
|    Change which setting?                                              |
+-----------------------------------------------------------------------+
        | Screen and keyboard      |
        | Save setup as dfl        |
        | Save setup as..          |
        | Exit                     |
        | Exit from Minicom        |
        +--------------------------+
```

When finished use Esc to exit menu



```
+-----[configuration]------+
| Filenames and paths      |
| File transfer protocols  |
| Serial port setup        |
| Modem and dialing        |
| Screen and keyboard      |
| Save setup as dfl        |
| Save setup as..          |
| Exit                     |
| Exit from Minicom        |
+--------------------------+
```

Use Save setup as dfl to save

```
+-----[configuration]------+
| Filenames and paths      |
| File transfer protocols  |
| Serial port setup        |
| Modem and dialing        |
| Screen and keyboard      |
| Save setup as dfl        |
| Save setup as..          |
| Exit                     |
| Exit from Minicom        |
+--------------------------+
```

Use Exit from Minicom to exit

Lab X2

```
root@sauron:~# minicom -o

Welcome to minicom 2.3

OPTIONS: I18n
Compiled on Oct 24 2008, 06:37:44.
Port /dev/ttyS0

               Press CTRL-A Z for help on special keys


CentOS release 5.2 (Final)
Kernel 2.6.18-92.1.22.el5 on an i686

arwen.localdomain login: cis192
Password:
Last login: Tue Mar 24 17:27:32 on ttyS0
[cis192@arwen ~]$ hostname
arwen.localdomain
[cis192@arwen ~]$



CentOS release 5.2 (Final)
Kernel 2.6.18-92.1.22.el5 on an i686

arwen.localdomain login:

                          +----------------------+
                          | Leave without reset? |
                          |     Yes       No     |
                          +----------------------+


 CTRL-A Z for help |115200 8N1 | NOR | Minicom 2.3    | VT102 | Online 00:01
 root@sauron:~#
```

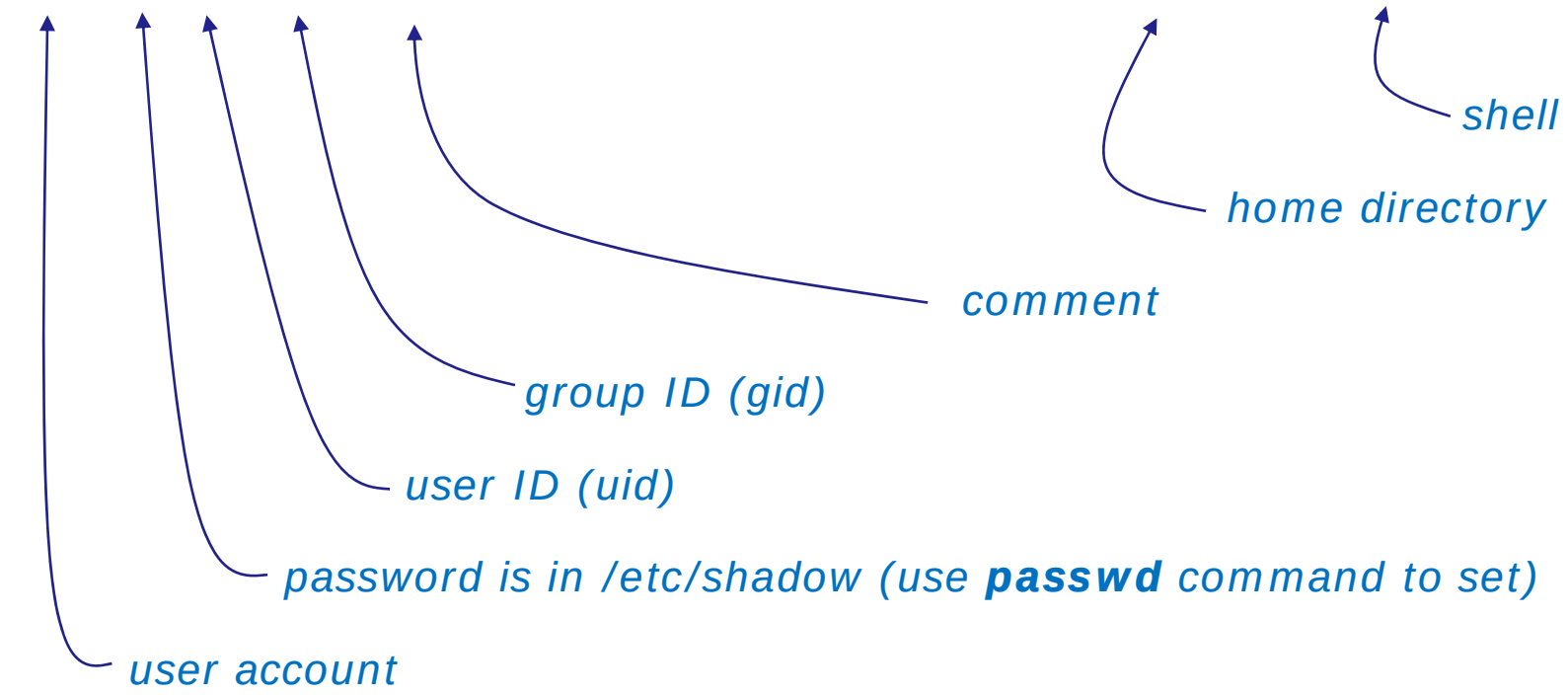
Example session using minim –o to log into Arwen at other end of the serial connection

Ctrl-A z q
(press Ctrl and A keys together, then z then q)

Lab X2

Creating a new user account on the server side with **useradd**

```
[root@arwen ~]# useradd -c "Guest account for serial access" guest
[root@arwen ~]# cat /etc/passwd | grep guest
guest:x:501:501:Guest account for serial access:/home/guest:/bin/bash
```



Lab X2

The .bash_profile file for the guest user

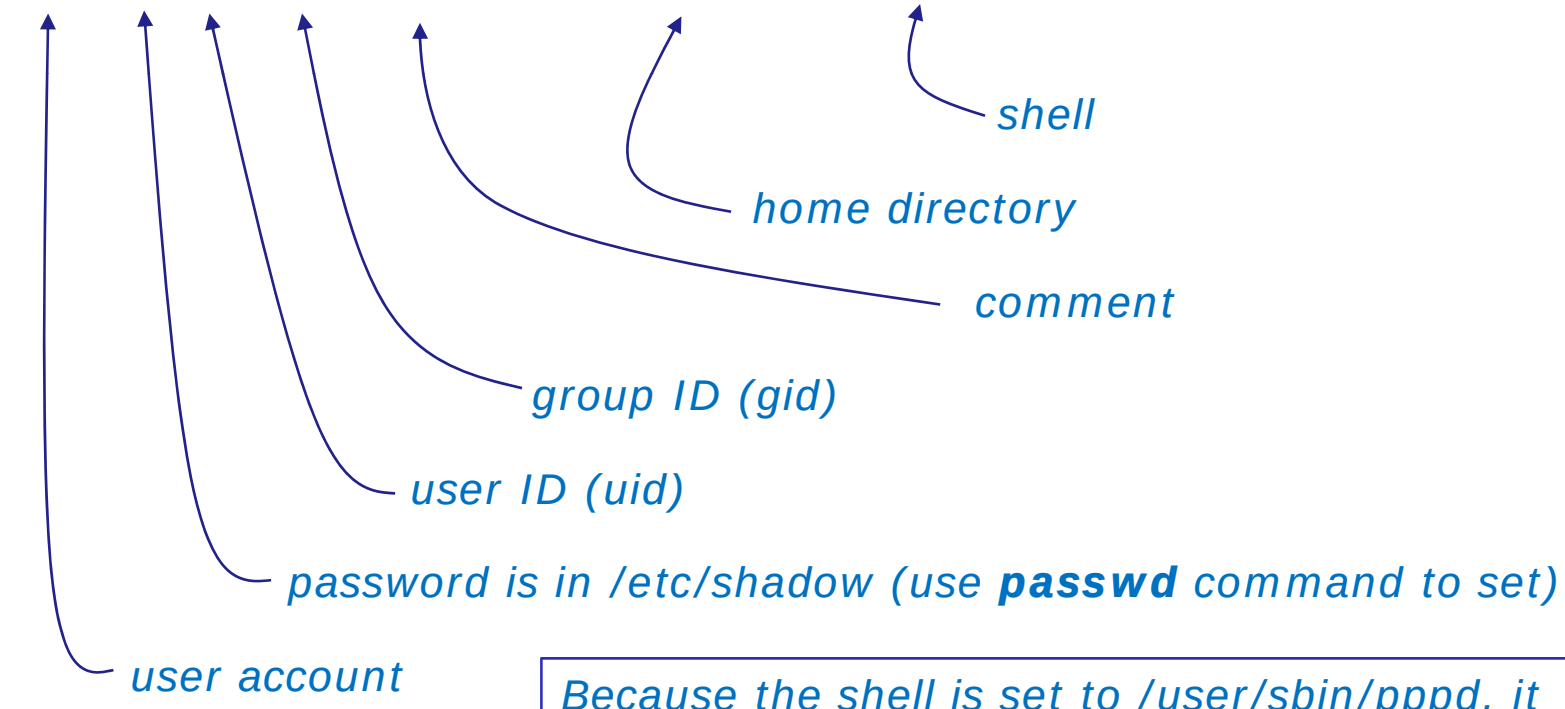
```
[root@arwen ~]# cat /home/guest/.bash_profile
# .bash_profile

# Get the aliases and functions
if [ -f ~/.bashrc ]; then
        . ~/.bashrc
fi

# User specific environment and startup programs

PATH=$PATH:$HOME/bin

export PATH
/usr/sbin/pppd -detach crtscts proxyarp 10.0.0.1:10.0.0.2 /dev/ttyS0 38400
[root@arwen ~]#
```

This is used in Part 3 of Lab X2. As soon as guest logs in, the pppd service is run automatically on the server.

Lab X2

Creating a new user account on the server side with **useradd**

```
[root@arwen ~]#  useradd -c "PPP Account" -d /etc/ppp -s /usr/sbin/pppd ppp
[root@arwen ~]# cat /etc/passwd | grep ppp
ppp:x:502:502:PPP Account:/etc/ppp:/usr/sbin/pppd
```



Because the shell is set to /user/sbin/pppd, it is run as soon as the ppp user logs inusing the option in /etc/ppp/options

Lab X2

The server side options can be put on the command line

/usr/sbin/pppd -detach crtscts proxyarp 10.0.0.1:10.0.0.2 /dev/ttyS0 38400

or in the configuration file

```
[root@arwen ~]# cat /etc/ppp/options
-detach
crtscts
lock
proxyarp
10.0.0.1:10.0.0.2
/dev/ttyS0
38400
```

Don't fork to become a background process (otherwise pppd will do so if a serial device is specified).

Use hardware flow control using RTS and CTS signals to control the flow of data on the serial port.

Specifies that pppd should use a UUCP-style lock on the serial device to ensure exclusive access to the device.

Add an entry to this system's ARP [Address Resolution Protocol] table with the IP address of the peer and the Ethernet address of this system.

IP address for server-end: client-end

Serial device

Desired baud rate

Refer to **pppd** man page for full details

Lab X2

Command line (client side) to make a connection

With this option, pppd will detach (run in the background) once it has successfully established the ppp connection (to the point where the first network control protocol, usually the IP control protocol, has come up).

Add a default route to the system routing tables, using the peer as the gateway, when IPCP negotiation is successfully completed. This entry is removed when the PPP connection is broken.

```
pppd updetach crtscts defaultroute /dev/ttyS0 38400 connect \
"exec chat -v TIMEOUT 3 ogin:--ogin: ppp assword: secret"
```

command line (client side)

Lab X2

Command line (client side) to make a connection

```
root@sauron:~# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
root@sauron:~#
root@sauron:~# pppd updetach crtscts defaultroute /dev/ttyS0 38400 connect
"exec chat -v TIMEOUT 3 ogin:--ogin: ppp assword: secret"
Serial connection established.
Using interface ppp0
Connect: ppp0 <--> /dev/ttyS0
Deflate (15) compression enabled
Cannot determine ethernet address for proxy ARP
local  IP address 10.0.0.2
remote IP address 10.0.0.1
root@sauron:~# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.0.0.1        0.0.0.0         255.255.255.255 UH    0      0        0 ppp0
0.0.0.0         0.0.0.0         0.0.0.0         U     0      0        0 ppp0
root@sauron:~#
```

updetach *option:*
Makes pppd run in the background when link comes up

defaultroute *option:*
Adds a route to the peer for all traffic

Lab X2

Command line (client side) to make a connection

```
pppd updetach crtscts defaultroute /dev/ttyS0 38400 connect \
"exec chat -v TIMEOUT 3 ogin:--ogin: ppp assword: secret"
```

*The **connect option** can be used to run a script which in this case runs the chat command.*

The chat command is used to handle the login automatically.

Lab X2

Command line (client side) to make a connection

```
pppd updetach crtscts defaultroute /dev/ttyS0 38400 connect \
"exec chat -v TIMEOUT 3 ogin:--ogin: ppp assword: secret"
```

Requests verbose mode for logging purposes.

Lab X2

Command line (client side) to make a connection

```
pppd updetach crtscts defaultroute /dev/ttyS0 38400 connect \
"exec chat -v TIMEOUT 3 ogin:--ogin: ppp assword: secret"
```

Set the timeout to 3 seconds

Lab X2

Command line (client side) to make a connection

```
pppd updetach crtscts defaultroute /dev/ttyS0 38400 connect \
"exec chat -v TIMEOUT 3 ogin:--ogin: ppp assword: secret"
```

expect:send *pairs:*
*expect …**ogin** then send **ppp**,*
*expect …**assword** then send **secret***

*Note: the **--ogin** is **sub-expect:sub-send** pair. If the first login is not received, send a single return (empty line) and look again for another login*

Note, because the beginning of the expected word may be garbled due to a flakey modem connection, just look for the end of the word (e.g. login to ogin, password to assword)

Lab X2

Troubleshooting

Tips

- Serial connection can only be used by one pair of computers at a time.
  E.g. Both minicom on Sauron and Putty workstation cannot access serial COM 1 on Arwen at the same time.

- View log file:
  cat var/log/messages | grep pppd

Lab X2

Troubleshooting

```
cis192@sauron:~$ su -
Password:
root@sauron:~# ./ppp-on
Serial connection established.
Using interface ppp0
Connect: ppp0 <--> /dev/ttyS0
LCP: timeout sending Config-Requests
Connection terminated.
Modem hangup
root@sauron:~#
```

Remove default gateway on Arwen

Lab X2

Troubleshooting

```
root@sauron:~# ./ppp-on
Connect script failed
root@sauron:~#
```

Make sure you have logged out from any previously made serial connections. You may need to run minicom –o again to see if you are still logged in as guest.

Wrap

New commands, daemons:

- pppd
- chat
- minicom
- setserial
- stty

Configuration files

- /etc/ppp/options
- /etc/minicom/minirc.dfl

Next Class

Assignment: Check Calendar Page http://simms-teach.com/cis192calendar.php

- Test next week on lessons 5 - 8 and related labs
  - Example questions:
    - How do you recognize a 3-way handshake in Wireshark?
    - What command on Red Hat family systems would configure the vsftpd service to startup automatically when powering up?
    - For firewall purposes when is a TCP stream considered to be "established" on the server side?
    - What are two different commands on Red Hat family systems that would cause the xinetd daemon to reread its configuration files?
  - Extra credit Lab X2 on PPP available now

Backup

Classroom Static IP addresses for VM's

Station	IP	Static 1
Instructor	172.30.1.100	172.30.1.125
Station-01	172.30.1.101	172.30.1.126
Station-02	172.30.1.102	172.30.1.127
Station-03	172.30.1.103	172.30.1.128
Station-04	172.30.1.104	172.30.1.129
Station-05	172.30.1.105	172.30.1.130
Station-06	172.30.1.106	172.30.1.131
Station-07	172.30.1.107	172.30.1.132
Station-08	172.30.1.108	172.30.1.133
Station-09	172.30.1.109	172.30.1.134
Station-10	172.30.1.110	172.30.1.135
Station-11	172.30.1.111	172.30.1.136
Station-12	172.30.1.112	172.30.1.137

Station	IP	Static 1
Station-13	172.30.1.113	172.30.1.138
Station-14	172.30.1.114	172.30.1.139
Station-15	172.30.1.115	172.30.1.140
Station-16	172.30.1.116	172.30.1.141
Station-17	172.30.1.117	172.30.1.142
Station-18	172.30.1.118	172.30.1.143
Station-19	172.30.1.119	172.30.1.144
Station-20	172.30.1.120	172.30.1.145
Station-21	172.30.1.121	172.30.1.146
Station-22	172.30.1.122	172.30.1.147
Station-23	172.30.1.123	172.30.1.148
Station-24	172.30.1.124	172.30.1.149

Note the static IP address for your station to use in the next class exercise

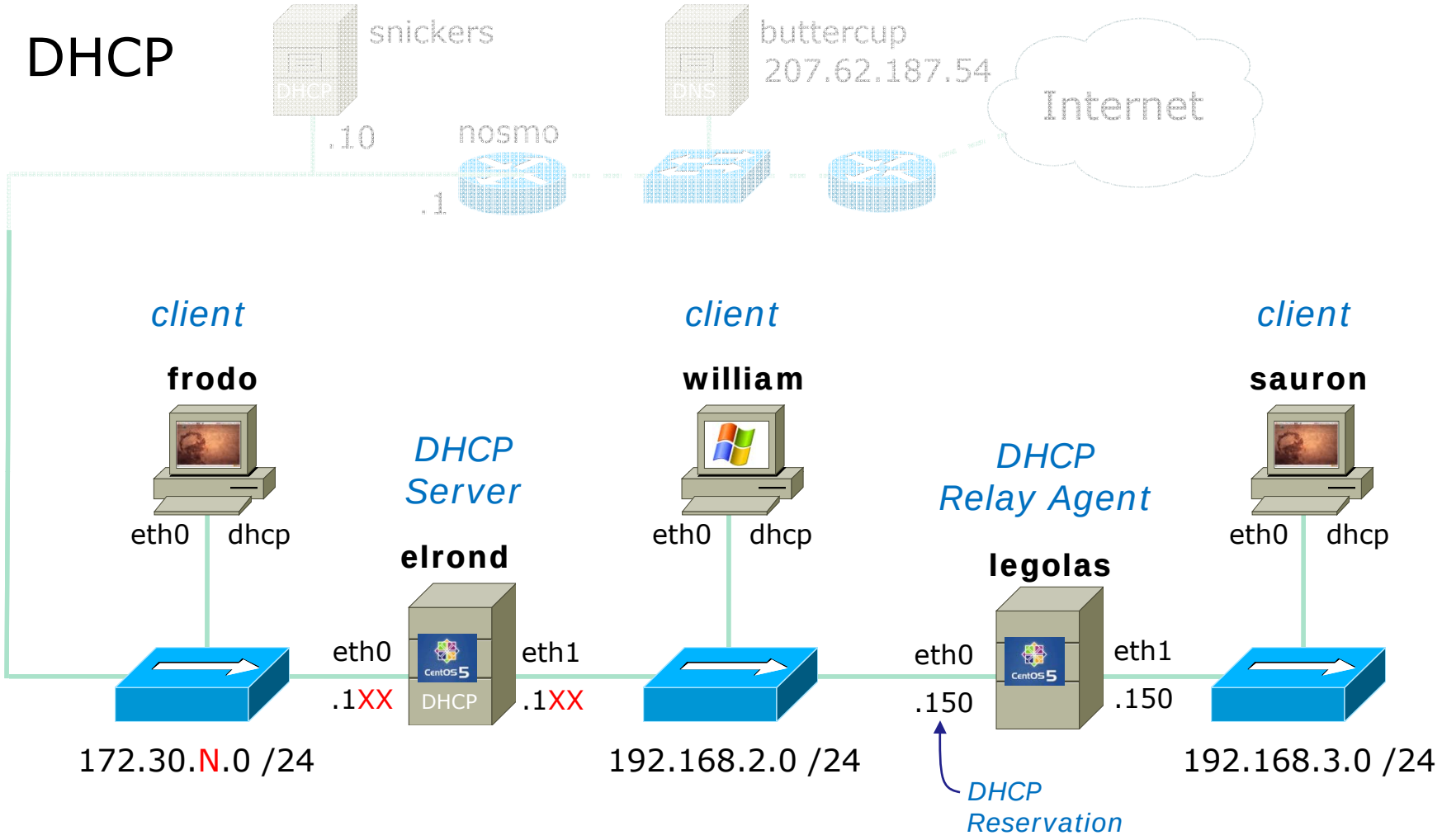
Classroom DHCP IP allocation pools table by station number

Station	IP	Start	End
01	172.30.1.101	172.30.1.50	172.30.1.54
02	172.30.1.102	172.30.1.55	172.30.1.59
03	172.30.1.103	172.30.1.60	172.30.1.64
04	172.30.1.104	172.30.1.65	172.30.1.69
05	172.30.1.105	172.30.1.70	172.30.1.74
06	172.30.1.106	172.30.1.75	172.30.1.79
07	172.30.1.107	172.30.1.80	172.30.1.84
08	172.30.1.108	172.30.1.85	172.30.1.89
09	172.30.1.109	172.30.1.90	172.30.1.94
10	172.30.1.110	172.30.1.95	172.30.1.99
11	172.30.1.111	172.30.1.200	172.30.1.204
12	172.30.1.112	172.30.1.205	172.30.1.209

Station	IP	Start	End
13	172.30.1.101	172.30.1.210	172.30.1.214
14	172.30.1.102	172.30.1.215	172.30.1.219
15	172.30.1.103	172.30.1.220	172.30.1.224
16	172.30.1.104	172.30.1.225	172.30.1.229
17	172.30.1.105	172.30.1.230	172.30.1.234
18	172.30.1.106	172.30.1.235	172.30.1.239
19	172.30.1.107	172.30.1.240	172.30.1.244
20	172.30.1.108	172.30.1.245	172.30.1.249
21	172.30.1.109	172.30.1.250	172.30.1.254
22	172.30.1.110	172.30.1.30	172.30.1.34
23	172.30.1.111	172.30.1.35	172.30.1.39
24	172.30.1.112	172.30.1.20	172.30.1.44
Instruct	172.30.1.100	172.30.1.45	172.30.1.49

Use these pools of addresses based on your station number to avoid conflicts on the classroom network

DHCP


snickers
DHCP
.10
nosmo
.1
buttercup
207.62.187.54
DNS
Internet
client
frodo
eth0
dhcp
DHCP
Server
elrond
eth0
eth1
.1XX
DHCP
.1XX
172.30.N.0 /24
client
william
eth0
dhcp
192.168.2.0 /24
DHCP
Relay Agent
legolas
eth0
.150
eth1
.150
DHCP
Reservation
client
sauron
eth0
dhcp
192.168.3.0 /24
Shire
Rivendell
Mordor

Exercise - Debian / Ubuntu NIC Config (permanent)

```
[root@arwen ~]# ipcalc -npmb 10.10.10.141/22
NETMASK=255.255.252.0
PREFIX=22
BROADCAST=10.10.11.255
NETWORK=10.10.8.0

cis192@sawyer:~$ cat /etc/hostname
sawyer

cis192@sawyer:~$ cat /etc/network/interfaces
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
address 10.10.10.141
broadcast 10.10.11.255
netmask 255.255.252.0
network 10.10.8.0

gateway 10.10.8.1

up route add -net 192.168.3.0/24 gw 10.10.8.10

cis192@sawyer:~$
```

Exercise - Debian / Ubuntu NIC Config (permanent)

```
[root@arwen ~]# ipcalc -npmb 10.10.10.141/22
NETMASK=255.255.252.0
PREFIX=22
BROADCAST=10.10.11.255
NETWORK=10.10.8.0

root@sawyer:~# cat /etc/hosts
127.0.0.1       localhost
127.0.1.1       sawyer

# The following lines are desirable for IPv6 capable
hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
ff02::3 ip6-allhosts
root@sawyer:~#
```

Exercise - Debian / Ubuntu NIC Config (permanent)

```
cis192@sawyer:~$ ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:0c:29:6f:53:d9
          inet addr:10.10.10.141  Bcast:10.10.11.255  Mask:255.255.252.0
          inet6 addr: fe80::20c:29ff:fe6f:53d9/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:209 errors:0 dropped:0 overruns:0 frame:0
          TX packets:27 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:35602 (35.6 KB)  TX bytes:4755 (4.7 KB)
          Interrupt:18 Base address:0x1400

cis192@sawyer:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use
Iface
192.168.3.0     10.10.8.10      255.255.255.0   UG    0      0        0 eth0
10.10.8.0       0.0.0.0         255.255.252.0   U     0      0        0 eth0
169.254.0.0     0.0.0.0         255.255.0.0     U     1000   0        0 eth0
0.0.0.0         10.10.8.1       0.0.0.0         UG    100    0        0 eth0
cis192@sawyer:~$ ping -c2 sawyer
PING sawyer (127.0.1.1) 56(84) bytes of data.
64 bytes from sawyer (127.0.1.1): icmp_seq=1 ttl=64 time=1.26 ms
64 bytes from sawyer (127.0.1.1): icmp_seq=2 ttl=64 time=0.152 ms

--- sawyer ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1007ms
rtt min/avg/max/mdev = 0.152/0.710/1.269/0.559 ms
cis192@sawyer:~$ ping -c2 10.10.10.141
PING 10.10.10.141 (10.10.10.141) 56(84) bytes of data.
64 bytes from 10.10.10.141: icmp_seq=1 ttl=64 time=0.295 ms
64 bytes from 10.10.10.141: icmp_seq=2 ttl=64 time=0.071 ms

--- 10.10.10.141 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 999ms
rtt min/avg/max/mdev = 0.071/0.183/0.295/0.112 ms
cis192@sawyer:~$
```

Exercise - CentOS NIC Config (permanent)

```
[root@arwen ~]# cat /etc/sysconfig/network-scripts/ifcfg-eth0
# Advanced Micro Devices [AMD] 79c970 [PCnet32 LANCE]
DEVICE=eth0
ONBOOT=yes
HWADDR=00:0c:29:70:d5:71
BOOTPROTO=static
IPADDR=10.10.8.100
NETMASK=255.255.252.0
BROADCAST=10.10.11.255
[root@arwen ~]#

[root@arwen ~]# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:0C:29:70:D5:71
          inet addr:10.10.8.100  Bcast:10.10.11.255  Mask:255.255.252.0
          inet6 addr: fe80::20c:29ff:fe70:d571/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1002 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1088 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:761805 (743.9 KiB)  TX bytes:107613 (105.0 KiB)
          Interrupt:177 Base address:0x1400

[root@arwen ~]#
```

TCP connection exercise

SIP	SP	DIP	DP	Protocol	Info	Packet Numbers
172.30.4.83	42855	192.168.2.150	21	FTP	Request: PASV	1
192.168.2.150	21	172.30.4.83	42855	FTP	Response: 227 Entering Passive Mode (192,168,2,150,200,83	2
172.30.4.83	42855	192.168.2.150	21	TCP	42855 > ftp [ACK] Seq=88 Ack=313 Win=5856 Len=0	3
172.30.4.83	41025	192.168.2.150	51283	TCP	41025 > 51283 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 WS=5	4
192.168.2.150	51283	172.30.4.83	41025	TCP	51283 > 41025 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1	5
172.30.4.83	41025	192.168.2.150	51283	TCP	41025 > 51283 [ACK] Seq=1 Ack=1 Win=5856 Len=0	6
172.30.4.83	42855	192.168.2.150	21	FTP	Request: RETR legolas	7
192.168.2.150	21	172.30.4.83	42855	FTP	Response: 150 Opening BINARY mode data connection for leg	8
192.168.2.150	51283	172.30.4.83	41025	FTP-DATA	FTP Data: 18 bytes	9
192.168.2.150	51283	172.30.4.83	41025	TCP	51283 > 41025 [FIN, ACK] Seq=19 Ack=1 Win=5888 Len=0	10
172.30.4.83	41025	192.168.2.150	51283	TCP	41025 > 51283 [ACK] Seq=1 Ack=19 Win=5856 Len=0	11
172.30.4.83	42855	192.168.2.150	21	TCP	42855 > ftp [ACK] Seq=102 Ack=378 Win=5856 Len=0	12
172.30.4.83	41025	192.168.2.150	51283	TCP	41025 > 51283 [FIN, ACK] Seq=1 Ack=20 Win=5856 Len=0	13
192.168.2.150	51283	172.30.4.83	41025	TCP	51283 > 41025 [ACK] Seq=20 Ack=2 Win=5888 Len=0	14
192.168.2.150	21	172.30.4.83	42855	FTP	Response: 226 File send OK.	15
172.30.4.83	42855	192.168.2.150	21	TCP	42855 > ftp [ACK] Seq=102 Ack=397 Win=5856 Len=0	16

What is the socket being used for the FTP data transfer?

Socket for data transfer

Client	Server
172.30.4.83	192.168.2.150
41025	51283

After which packet number does the FTP server regard the data transfer connection as being in the *Established* state?

6

What service makes use of the state of a connection?

firewall (iptables)

TCP Tunable Parameters exercise

- Revert Arwen to snapshot

For Arwen:

How many retries will Arwen do on a tcp connection before killing it?

cat /proc/sys/net/ipv4/tcp_retries2

15

Is TCP Selective acknowledgment enabled or disabled?

cat /proc/sys/net/ipv4/tcp_sack

1

How would you enable IP packet forwarding?

echo 1 > /proc/sys/net/ipv4/ip_forward

How would you enable IP packet forwarding permanently?

Put net.ipv4.ip_forward=1 line in /etc/sysctl.conf, then do sysctl -p

Exercise - Debian / Ubuntu NIC Config (permanent)

Sauron

1. Revert Sauron to snapshot

2. Configure Sauron permanently:
   - Hostname = Sawyer
   - Static IP = 10.10.10.141/22
   - Default gateway = 10.10.8.1
   - Static route to 192.168.3.0/24 via 10.10.8.10

3. Test:
   - ping sawyer
   - ping 10.10.10.141

Hint: Use ipcalc on one of the CentOS systems

Using PPP over a direct null modem connection

Test for connectivity

Start pppd on either side

pppd -detach crtscts lock <local IP>:<remote IP> /dev/ttyS0 38400 &

PPP Configuration Utilities

- WvDial - A command-line pppd driver
- rp3 - RedHat PPP dialer (Graphical)
- Linuxconf - Universal (almost) Linux PPP dialer

Linking two LANS using PPP

- Setting up the IP addresses
- Setting up the routing
- Network security

Setting up a PPP Server

- Getting the software together
- Setting up standard (shell access) dialup.
- Setting up the PPP options files
- Setting pppd up to allow users to (successfully) run it
- Setting up the global alias for pppd

ISP Information

- The phone number to call (don't forget 9 if behind a PABX)
- Dynamic or static IP numbers
- DNS server IP addresses (does not come dynamically at connect time)
- If PAP or CHAP is used, you need an id and "secret"
- What starting command to invoke.

TCP Tunable Parameters Exercise

- Revert Arwen to snapshot

For Arwen:

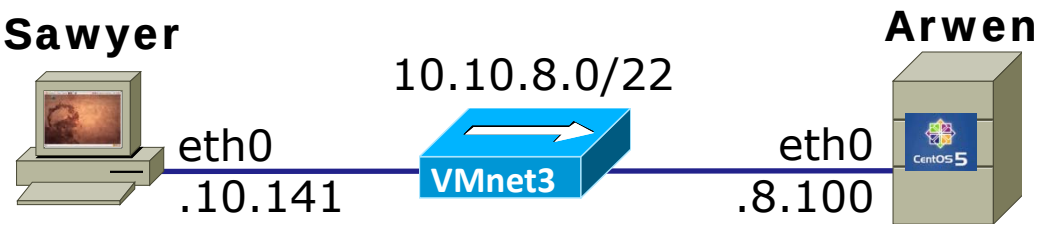
How many retries will Arwen do on a TCP connection before killing it?

Is TCP Selective acknowledgment enabled or disabled?

How would you enable IP packet forwarding temporarily?

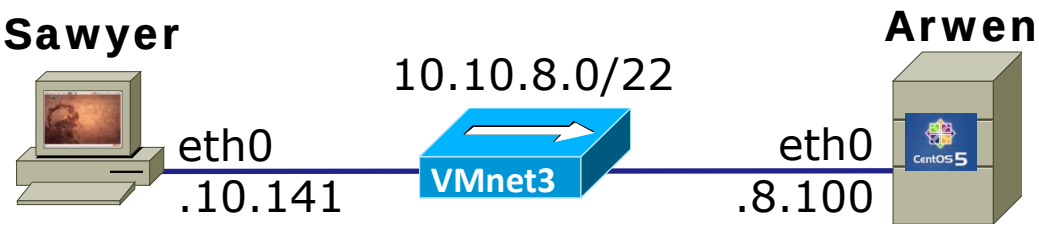
How would you enable IP packet forwarding permanently?

Access controls using xinetd configuration file



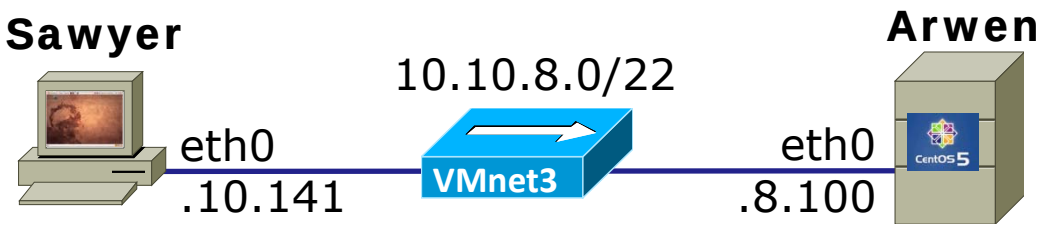
- Join Sawyer and Arwen to the 10.10.8.0/22 network
- Test using pings from both ends
- Disable the firewall on Arwen
  - lokkit
  - or iptables -F and iptables -X
- Telnet from Sawyer to Arwen

Access controls using xinetd configuration file



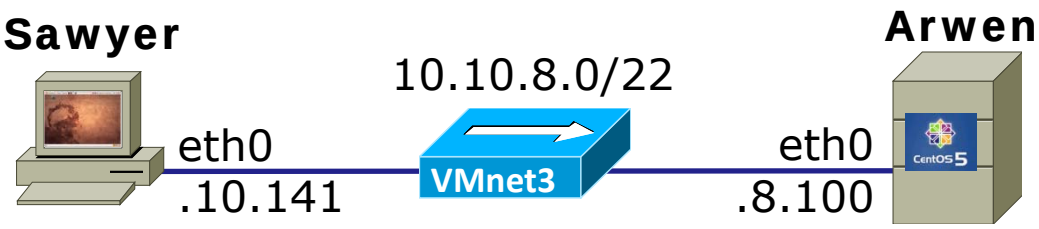
- Configure telnet service configuration file on Arwen to not allow Sawyer.
- Verify Sawyer is blocked and gets "Connection closed by foreign host" error message.
- Now configure telnet service configuration file on Arwen to only allow Sawyer.
- Login using telnet from Sawyer to Arwen to verify.

Access controls using Firewall



- Enable the firewall with lokkit or service iptables restart.
- Verify Sawyer is blocked and gets "Unable to connect to remote host: No route to host" error message.
- Modify Arwen's firewall to allow incoming telnet connections
- Login using telnet from Sawyer to Arwen to verify.

Access controls using TCP Wrappers



- Configure TCP wrappers /etc/hosts.deny on Arwen to not allow any access to any services.
- Verify Sawyer is blocked and gets " Connection closed by foreign host " error message.
- Now configure TCP wrappers on Arwen to only allow Sawyer to use telnet service.
- Login using telnet from Sawyer to Arwen to verify.