

Networking Lab Class #3

V. Arun
Spring 2015
UMass Amherst

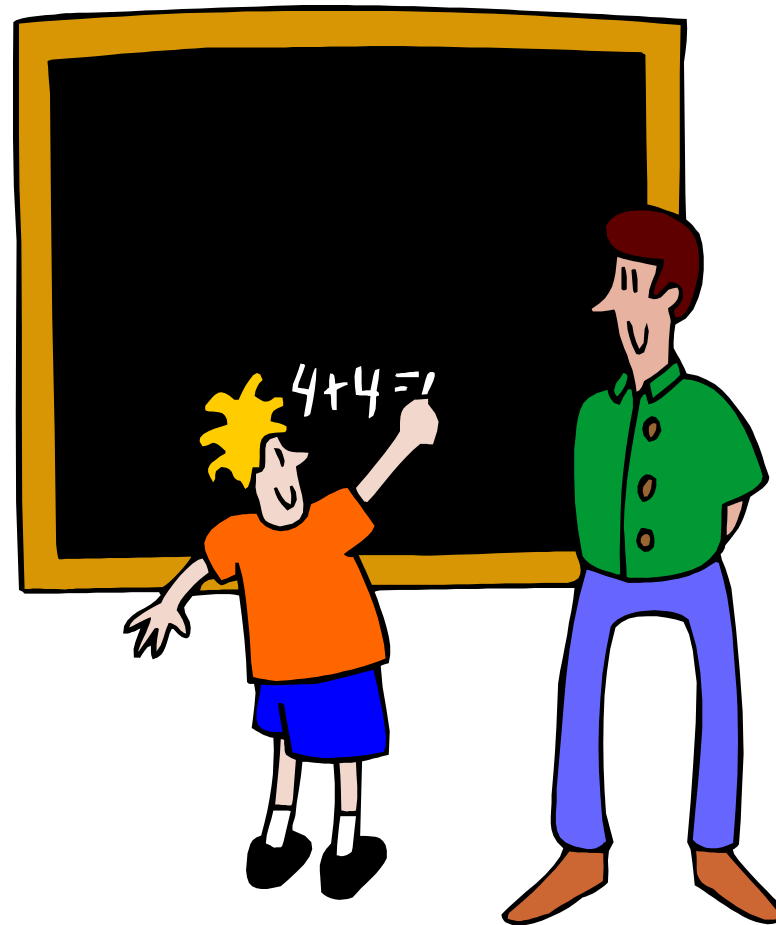
(slides adapted from E. Rosenweig)



This Class

- **Review of Lab 2**
- Review of Question 7
- Preparing for Lab 3

Lab 2: Questions? Comments?





This Class

- Review of Lab 2
- **Review of Question 7**
- Preparing for Lab 3

Exercise 7: Topology & Addresses

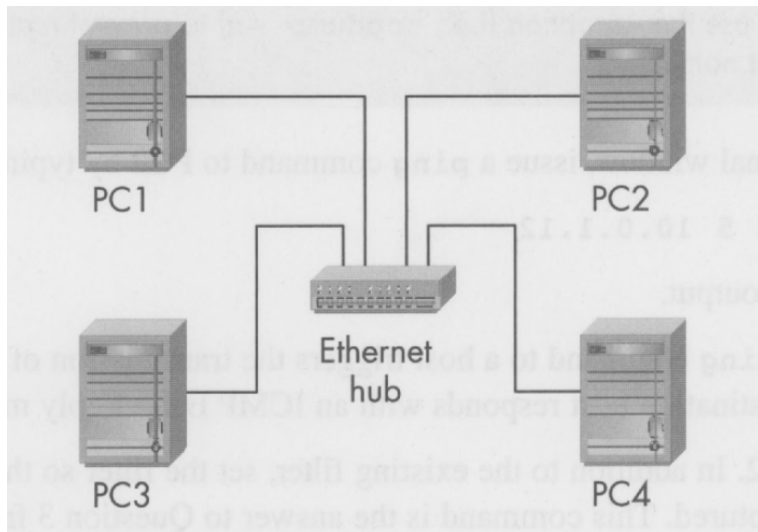


TABLE 2.4. IP addresses for Part 7.

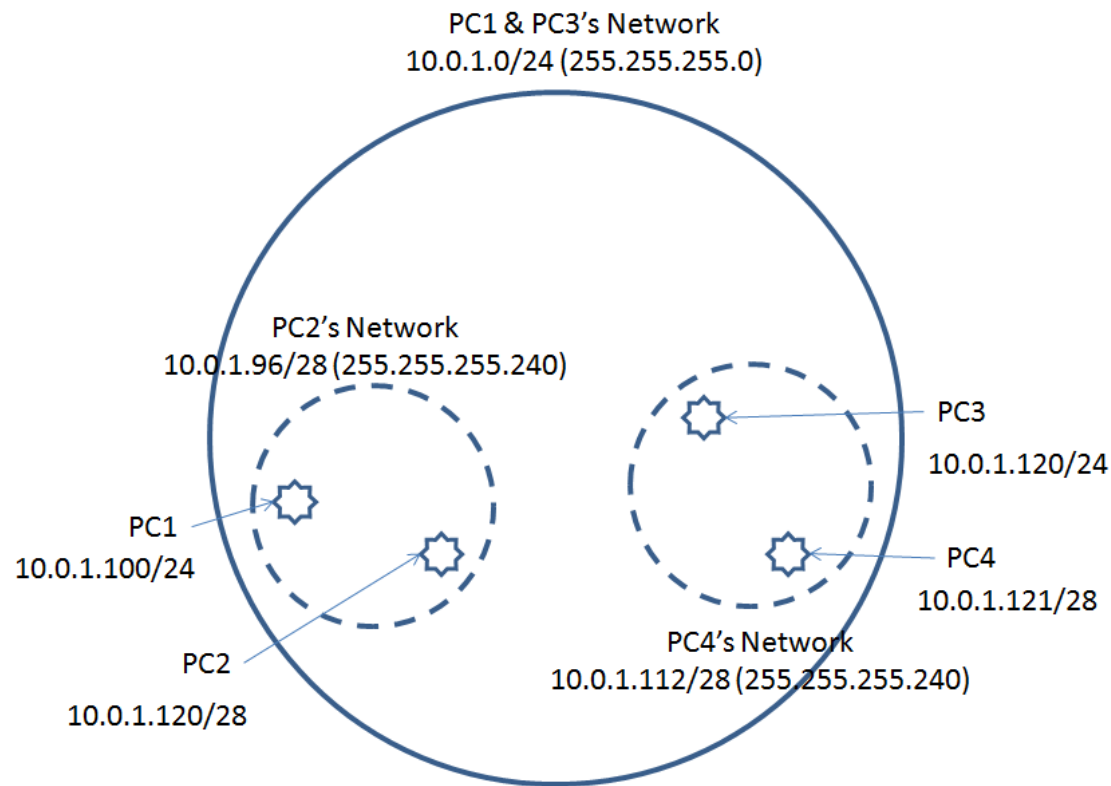
Linux PC	IP Address of Ethernet Interface <i>eth0</i>	Network Mask
PC1	10.0.1.100/24	255.255.255.0
PC2	10.0.1.101/28	255.255.255.240
PC3	10.0.1.120/24	255.255.255.0
PC4	10.0.1.121/28	255.255.255.240




Address ranges

- PC1:
 - 10.0.1.100/24, in network 10.0.1.0/24 (mask 255.255.255.0)
 - Lowest/Highest IP addresses: 10.0.1.1/24 to 10.0.1.254/24
- PC2:
 - 10.0.1.101/28, in network 10.0.1.96/28 (mask 255.255.255.240)
 - Lowest/Highest IP addresses: 10.0.1.97/28 to 10.0.1.110/28.
- PC3:
 - 10.0.1.120/24, in network 10.0.1.0/24 (mask 255.255.255.0)
 - Lowest/Highest IP addresses: 10.0.1.1/24 to 10.0.1.254/24 (same as PC1).
- PC4:
 - 10.0.1.121/28, in network 10.0.1.112/28 (mask 255.255.255.240)
 - Lowest/Highest IP addresses: 10.0.1.113/28 to 10.0.1.126/28

Logical View of the Network





Conditions for a successful ping

1. Is the destination reachable?
 - If not in the current subnet, is a gateway defined?
2. Is destination MAC address known?
 - If not cached, use “arp” to resolve
3. Use ICMP to ping the destination
 - Depending which of the above steps fails, you get a different message.
 - Addressability (IP) should be bidirectional.



Exercise 7 results

- a) (PC1 → PC3) Works.
- b) (PC1 → PC2) Works.
- c) (PC1 → PC4) Does not work!
 - “arp” successful, but PC4 cannot reach PC1!
 - Review Wireshark data




Exercise 7 results

- d) (PC4 → PC1) Does not work!
 - a) “Network is unreachable” error in PC4. No “arp” initiated
- e) (PC2->PC4) Does not work!
 - a) “Network is unreachable” error in PC2. No “arp” initiated
- f) (PC2->PC3) Does not work!
 - e) “Network is unreachable” error in PC2. No “arp” initiated



This Class

- Review of Lab 2
- Review of Question 7
- **Preparing for Lab 3**



Lab 3 - Prep.

- In this lab, you will
 - Configure a router
 - Configure a Linux PC to use as a router
 - Proxy ARP, redirects...
 - Setup static routes




Configuring Routers

- PC_x used to configure $Router_x$
 - Type *kermit* in the console to start session
 - Username and password - same as PCs
 - But not all routers will request it
- Use “?” for possible-command completion
 - Tab as usual
- Follow book step-by-step instructions for setup




Configuring Linux PC as Router

- Simple command
 - `echo "1" > /proc/sys/net/ipv4/ip_forward`
- Sets PC so it forwards packets
- Lab 3 - static routing tables
 - `route` command




Lab 3 - Tip 1

- Prepare your own script files for setup
 - Lab will take a while. If you use the same configuration for large parts, script files will save time next time you have a slot.
- Router setup scripts?
 - Pretty handy once you learn it.
 - See a [short description](#) on the class Web site
 - Under “Lab 3” in “Lab Tips & FAQ”



Lab 3 - Tip 2

- Test each link right after you build it
 - Networks will start to be big and take time to construct
 - Don't work for hour and then start debugging
 - Debug as you go!



Lab 3 – Tip 3

- Router debugging
 - `show ip route`
 - `show interfaces`
 - `show ip interface brief`
 - `show running-config`
- Try to really understand router config
 - Very useful in coming Labs
 - Critical for debugging



show ip route - sample

```
Router# show ip route
```

```
Codes: I - IGRP derived, R - RIP derived, O  
- OSPF derived, C - connected, S - static,  
B - BGP derived..
```

```
C 10.119.254.0 255.255.255.0 is possibly  
down, routing via 0.0.0.0, eth0  
O E2 10.110.0.0 [160/5] via 10.119.254.6,  
0:01:00, eth1
```




Scripting Router Configuration

- Prepare a script file with extension “ksc” and mode “rwxr-xr-x” (use `chmod 0755 <fn>.ksc`)
 - Example follows
- Start kermit

```
kermit
set line /dev/ttys0
set carrier-watch off
```
- Call the script file

```
take <script-file.ksc>
```
- Connect to router (in kermit)

```
connect
```



```
An example of router script
output \13
input 10 >
lineout enable
input 10 #
lineout configure terminal
input 10 (config)#
lineout interface fastEthernet 0/0
input 10 (config-if)#
lineout ip address 10.0.1.11 255.255.255.0
input 10 (config-if)#
lineout shutdown
input 10 (config-if)#
lineout no shutdown
input 10 (config-if)#
lineout int fa0/1
input 10 (config-if)#
lineout ip add 10.0.2.11 255.255.255.0
input 10 (config-if)#
lineout shut
input 10 (config-if)#
lineout no shut
input 10 (config-if)#
lineout end
input 10 #
lineout disable
input 10 >
lineout logout
```



Questions?

Enjoy Lab 3!