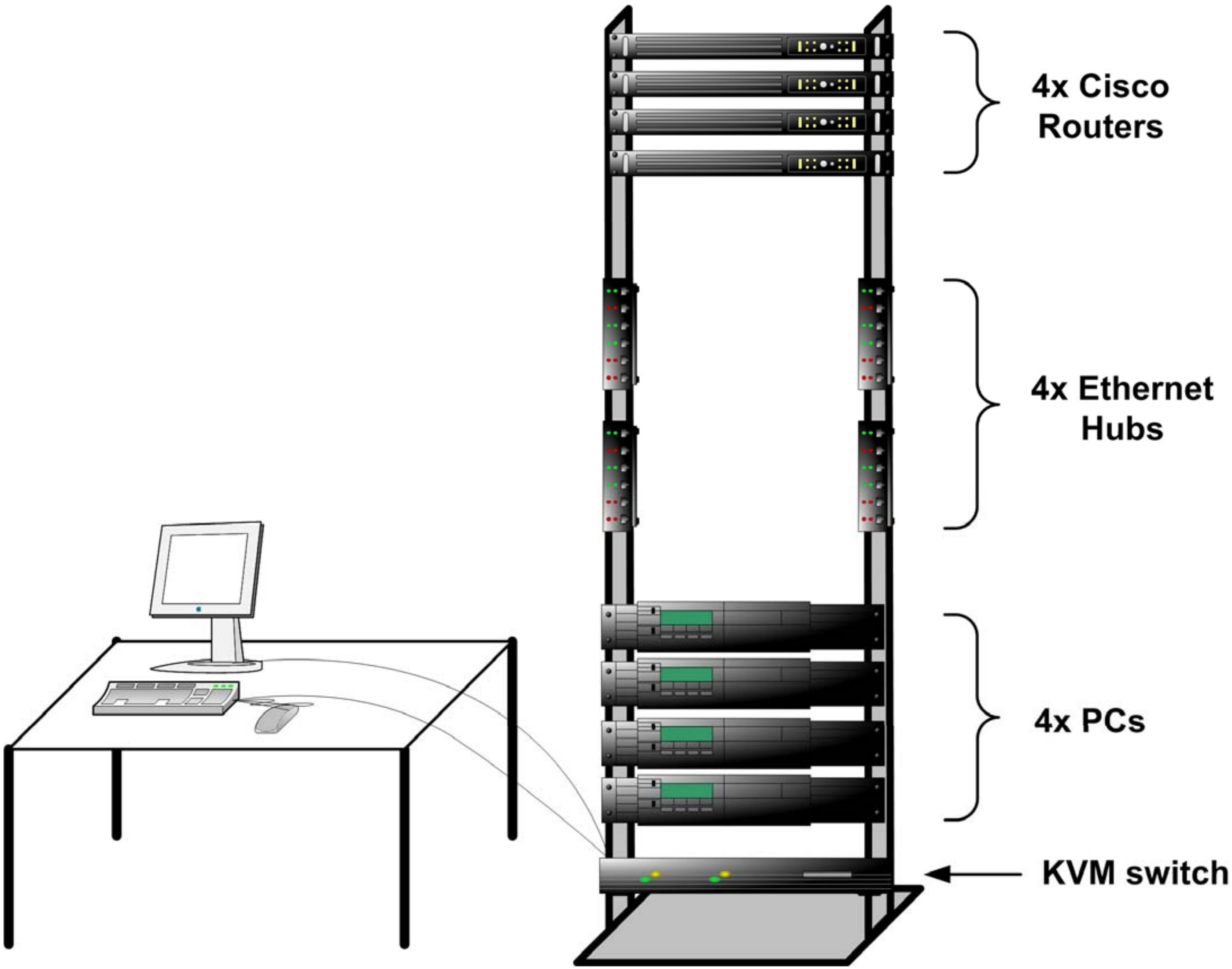
Introduction to the Lab

Lab Equipment & Organization

Internet Lab Equipment

- 4 Cisco 2600 Routers
- 4 Linux PCs
  (Intel Celeron 400MHz, 256MB Ram, 40GB disk, cdrom, floppy)
- 4 Ethernet hubs
  2x 5-port Hub 3Com OfficeConnect Dual Speed (10/100)
  2x 8-port Hub NETGEAR DS108
- 1 monitor, 1 keyboard, 1 mouse
- 1 KVM switch
- Cables

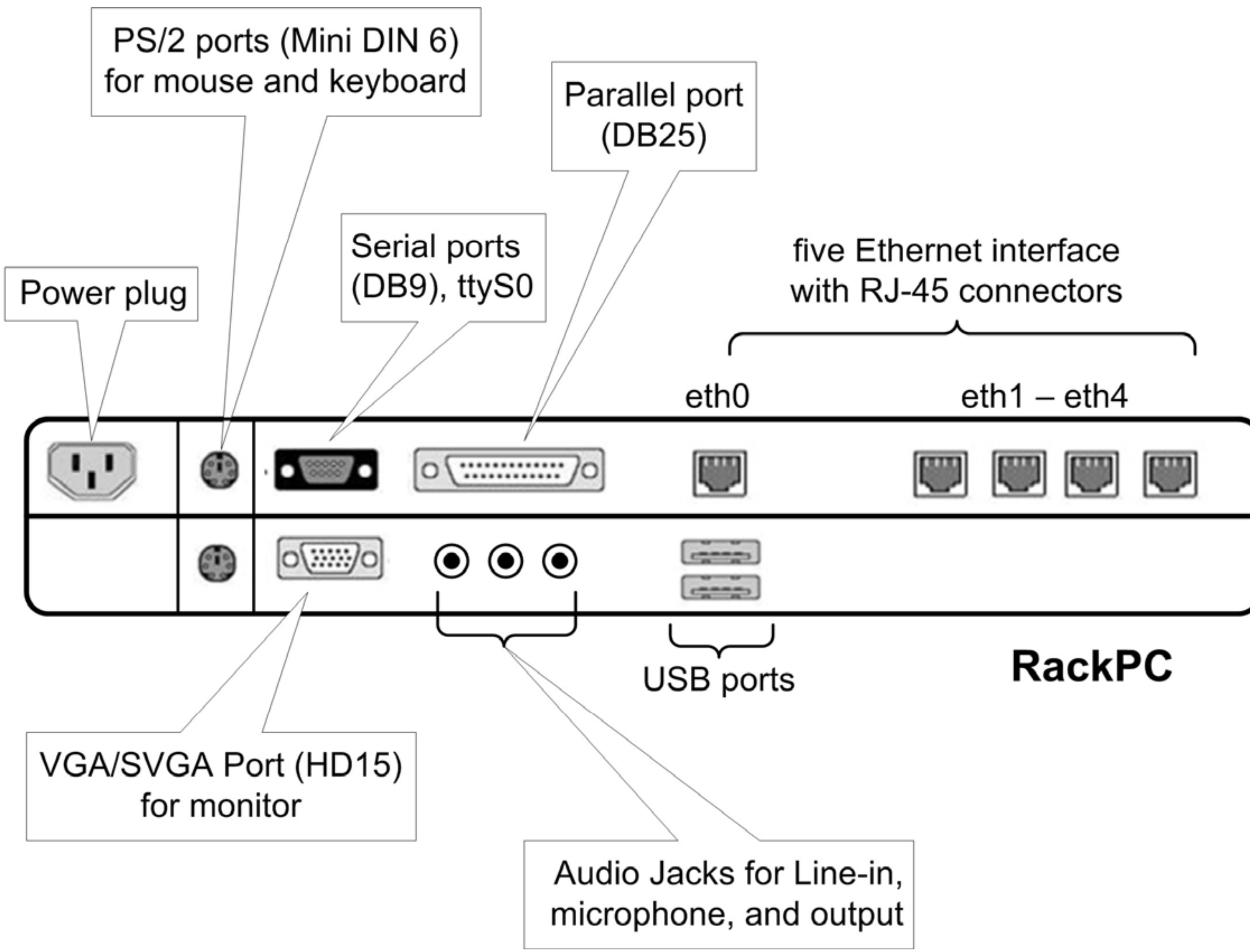
Internet Lab Equipment



Linux PCs

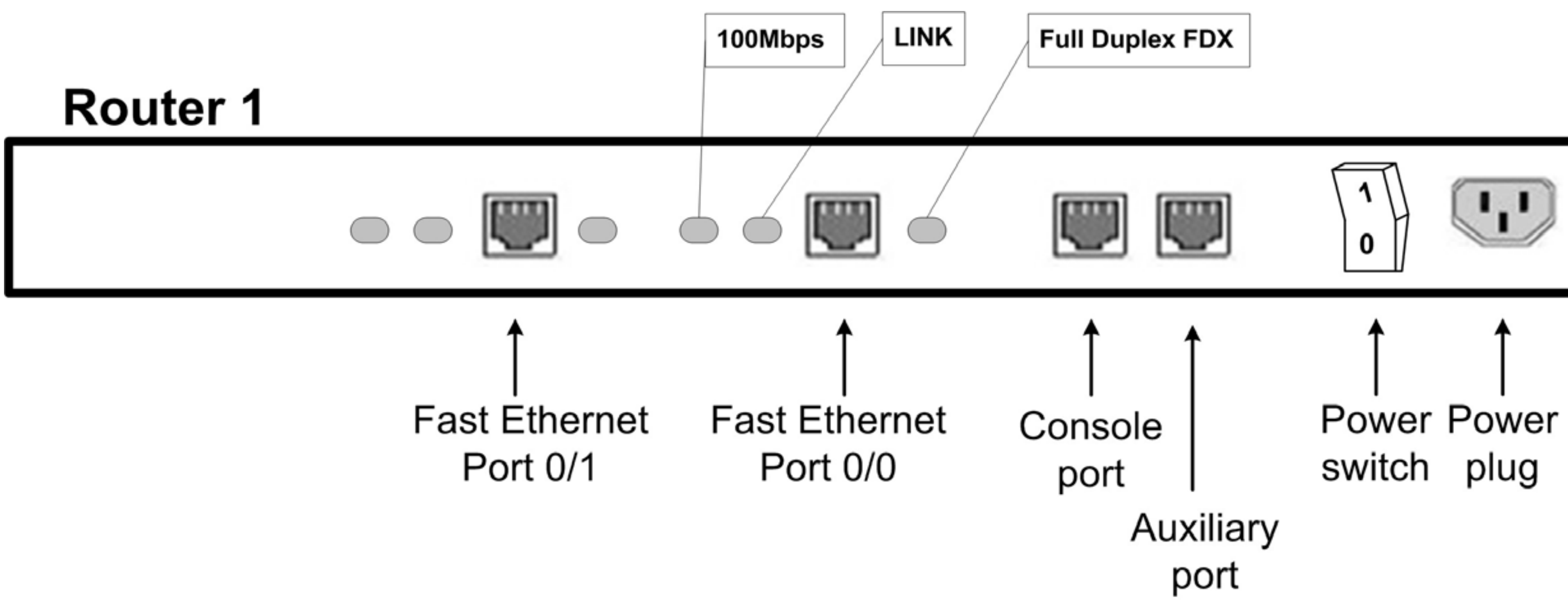
- PCs are labeled as:
  RackPC1, RackPC2, etc.
- PCs run Linux Debian 2.6.12
- Each PC has:
  - a floppy drive,
  - a cdrom drive,
  - a serial port,
  - 5x 10/100 Mbps Ethernet
    interface cards (*NICs*) named eth0 – eth4.

Linux PC


PS/2 ports (Mini DIN 6)
for mouse and keyboard
Parallel port
(DB25)
Serial ports
(DB9), ttyS0
five Ethernet interface
with RJ-45 connectors
Power plug
eth0
eth1 – eth4
USB ports
RackPC
VGA/SVGA Port (HD15)
for monitor
Audio Jacks for Line-in,
microphone, and output

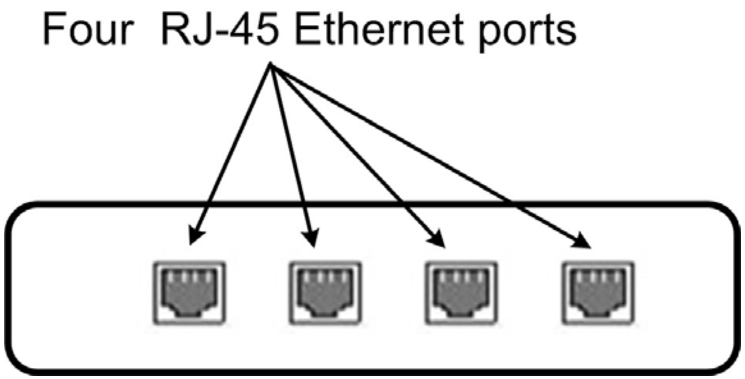
Cisco Routers

- Routers are labeled: Router1, Router2, Router3, Router4.
- Routers run Cisco IOS 12.0 or a later version
- Each router has:
  - a console port
  - an auxiliary port
  - two 10/100 Mbps Fast Ethernet interfaces

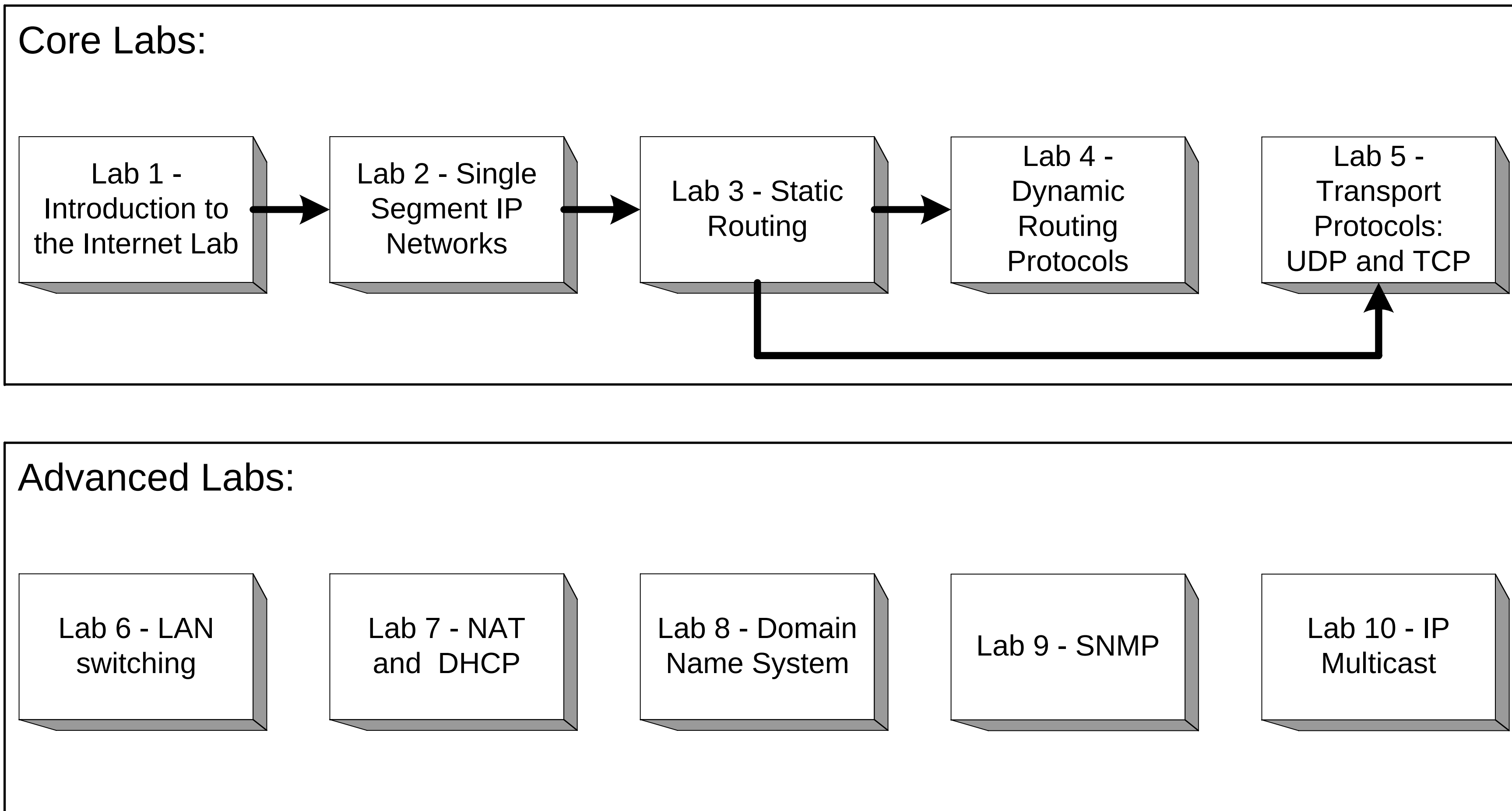


Ethernet Hubs

- Each hub has 4 or more RJ-45 ports
- Ports can operate at 10 Mbps or 100 Mbps



Lab Sequence



Core Labs

- **Lab 1 – Introduction to the Internet Lab**

  Overview of the Internet Lab equipment; introduction to ***ethereal*** **and** ***tcpdump*****.**

- **Lab 2 – Single Segment IP Networks**

  Configuring a network interface for IP networking; address resolution with ARP;

  security problems of common Internet applications.

Core Labs (cont.)

- **Lab 3 – Static routing**

  IP forwarding and routing between IP networks; setup a Linux PC and a Cisco router as an IP router; manual configuration of routing tables.

- **Lab 4 – Dynamic Routing Protocols**

  Routing protocols RIP, OSPF and BGP.

- **Lab 5 – Transport Protocols: UDP and TCP**

  Data transmissions with TCP and UDP; TCP connection management; TCP flow control; retransmissions in TCP; TCP congestion control.

Advanced Labs

- **Lab 6 - LAN switching**

  LAN switching in Ethernet networks; forwarding of Ethernet frames between LAN switches/bridges; spanning tree protocol for loop free routing between interconnected LANs.

- **Lab 7 - NAT and DHCP**

  Setup of a private network; dynamic assignment of IP addresses with DHCP.

- **Lab 8 – Domain Name System**

  Domain name resolution with DNS; name server hierarchy; setup of a DNS root server.

Structure of the Labs

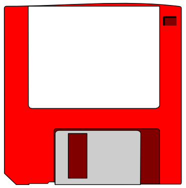
- Each lab has three phases:
  - Pre-laboratory Assignment (Prelab)
  - Lab Session
  - Lab Reports

Structure of the Labs (cont.)

- **Pre-laboratory Assignment (Pre-lab)**
  - Exercises to be completed in advance of the associated lab session.
  - The pre-labs ask you to acquire background knowledge that is needed during the lab exercises.
  - Each pre-lab has a question sheet that must be completed before the corresponding lab session.
  - The answers to the prelab questions are graded.

Structure of the Labs (cont.)

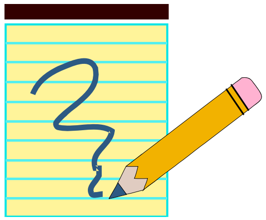
- **Lab Session.**
  - Lab exercises that are performed on the equipment of the Internet lab. All lab exercises can be completed without supervision. The time to complete a lab session should be three hours on the average, but may vary. Complete the laboratory activities to the extent that you can. The activities during the lab session are not graded, however, data collected during the lab session are needed to complete a lab report.
- Floppy disk symbol in the lab manual indicates when you have to collect data.



Floppy disk symbol

Structure of the Labs (cont.)

- **Lab Reports.**
  - After each lab session, you prepare a lab report that summarizes and analyzes the findings from the lab session. A notepad symbol indicates an assignment for the lab report. The lab reports should be submitted as a typewritten document.



Notepad symbol

- The lab report is generally due 1 week after the lab session. The lab report is graded.

- **Note:**
  - Lab reports should not include irrelevant data

In the Lab:

1. Bring formatted floppy disks, the lab manual and the solutions to prelab
2. Reboot Linux PCs
3. Complete exercises as described in the lab manual
4. Take measurements as instructed
5. Save data to floppy disk

Additional notes

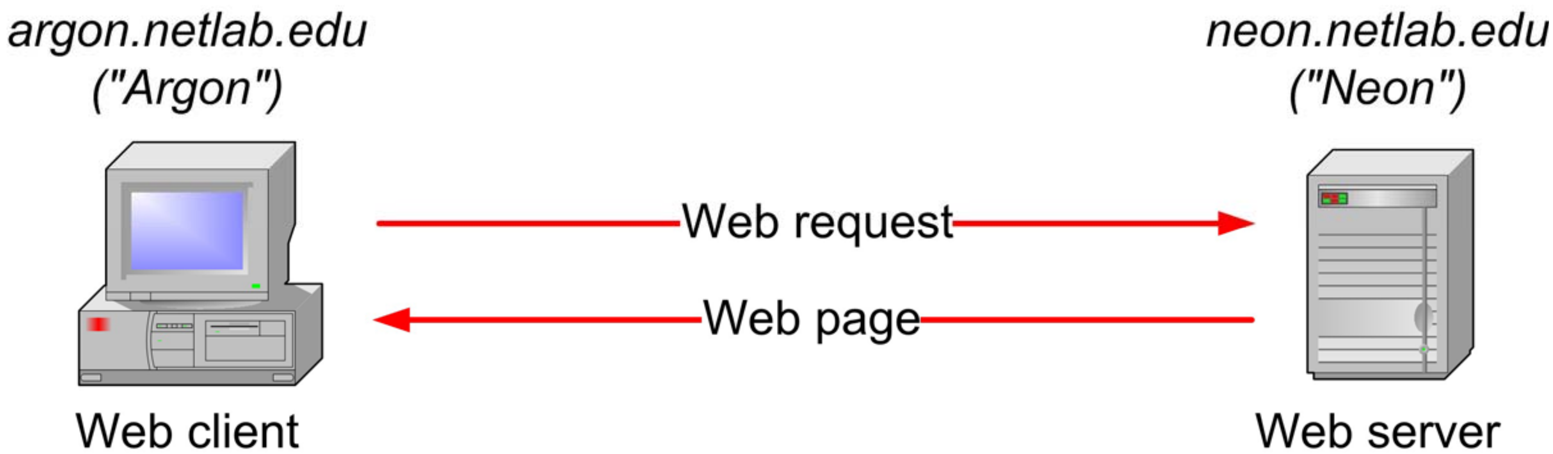
- The equipment of the Internet Lab is not connected to the Internet.
- Each lab has an anonymous feedback sheet. The feedback is used to improve the setup and organization of the labs.
- Since you have administrative (root) privileges on the Internet Lab equipment, exercise caution when modifying the configuration of the Internet Lab equipment.

TCP/IP Networking
An Example

Introductory material.

This module illustrates the interactions of the protocols of the TCP/IP protocol suite with the help of an example. The example intents to motivate the study of the TCP/IP protocols.

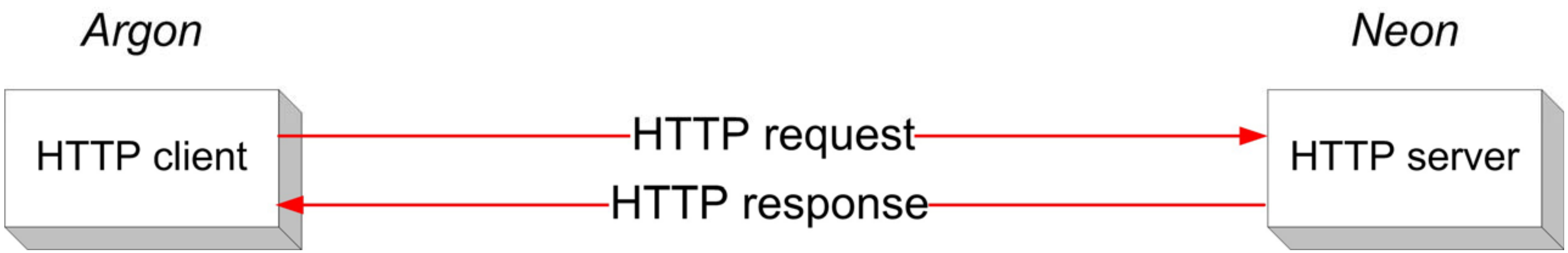
A simple TCP/IP Example



- A user on host *argon.netlab.edu* (“*Argon*”) makes web access to URL *http://neon.netlab.edu/index.html.*

- What actually happens in the network?

HTTP Request and HTTP response



- Web server runs an HTTP server program
- HTTP client Web browser runs an HTTP client program
- sends an HTTP request to HTTP server
- HTTP server responds with HTTP response

HTTP Request

```
GET /example.html HTTP/1.1
Accept: image/gif, */*
Accept-Language: en-us
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0
Host: 192.168.123.144
Connection: Keep-Alive
```

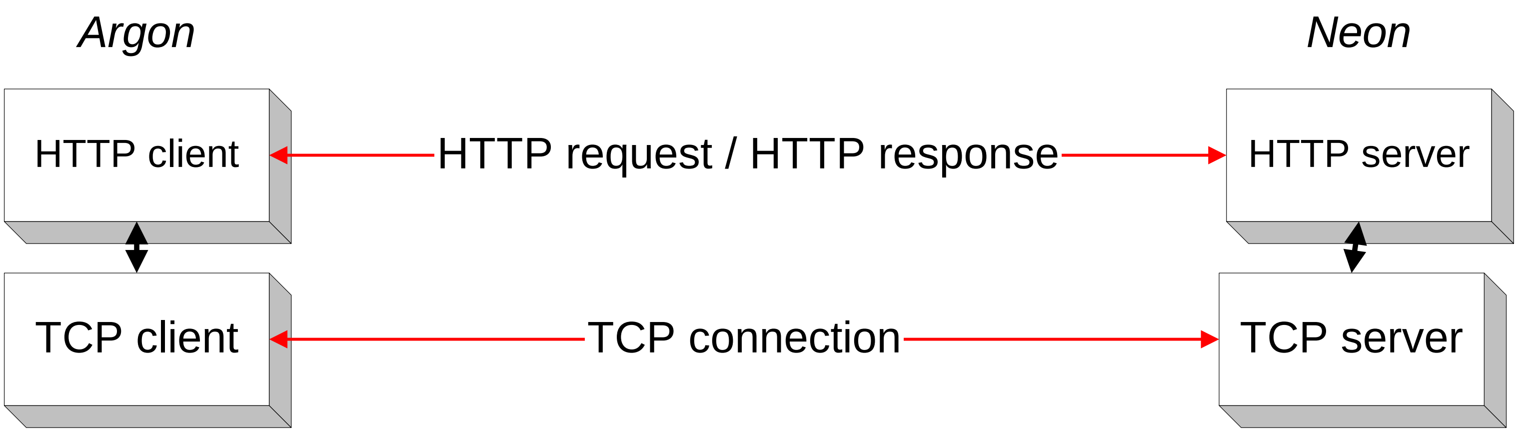
HTTP Response

```
HTTP/1.1 200 OK
Date: Sat, 25 May 2002 21:10:32 GMT
Server: Apache/1.3.19 (Unix)
Last-Modified: Sat, 25 May 2002 20:51:33 GMT
ETag: "56497-51-3ceff955"
Accept-Ranges: bytes
Content-Length: 81
Keep-Alive: timeout=15, max=100
Connection: Keep-Alive
Content-Type: text/html

<HTML>
<BODY>
<H1>Internet Lab</H1>
Click <a href="http://www.netlab.net/index.html">here</a> for the
Internet Lab webpage.
</BODY>
</HTML>
```

- How does the HTTP request get from Argon to Neon ?

From HTTP to TCP



- To send request, HTTP client program establishes an TCP connection to the HTTP server Neon.
- The HTTP server at Neon has a TCP server running

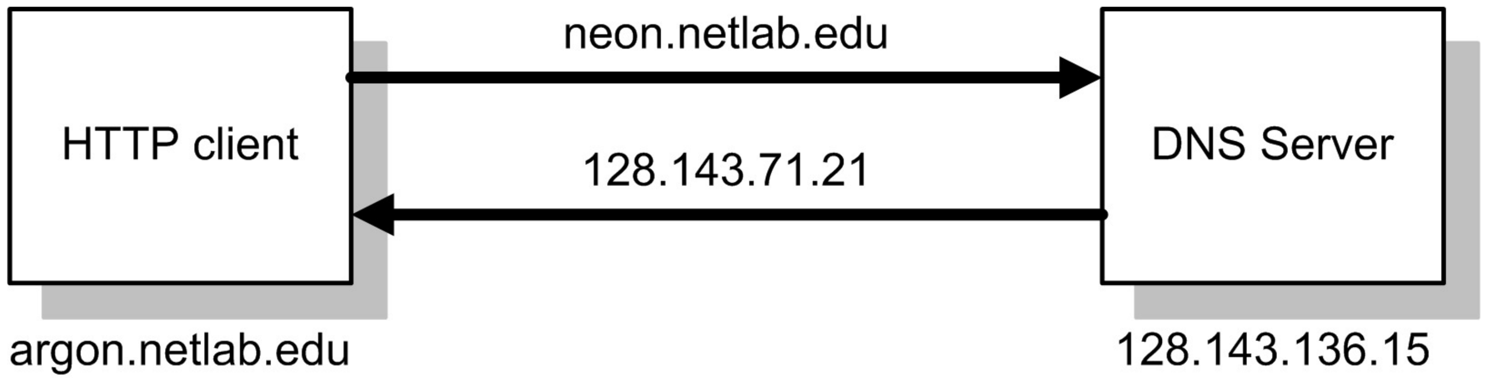
Resolving hostnames and port numbers

- Since TCP does not work with hostnames and also would not know how to find the HTTP server program at Neon, two things must happen:

  1. The name “neon.netlab.edu” must be translated into a **32-bit IP address.**

  2. The HTTP server at Neon must be identified by a 16-bit **port number**.

Translating a hostname into an IP address



- The translation of the hostname *neon.netlab.edu* into an IP address is done via a database lookup
- The distributed database used is called the ***Domain Name System (DNS)***
- All machines on the Internet have an IP address:

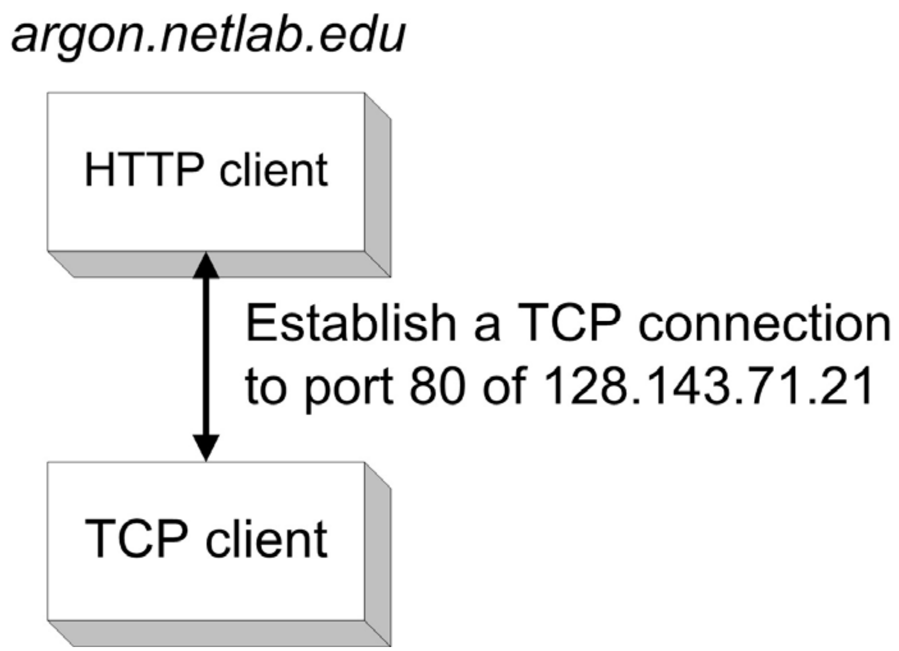
argon.netlab.edu	*128.143.137.144*
neon.netlab.edu	*128.143.71.21*

Finding the port number

- **Note:** Most services on the Internet are reachable via well-known ports.
  - E.g. HTTP servers on the Internet can be reached at port number “80”.
- **So:** Argon simply <u>knows</u> the port number of the HTTP server at a remote machine.
- On most Unix systems, the well-known ports are listed in a file with name /etc/services. The well-known port numbers of some of the most popular services are:

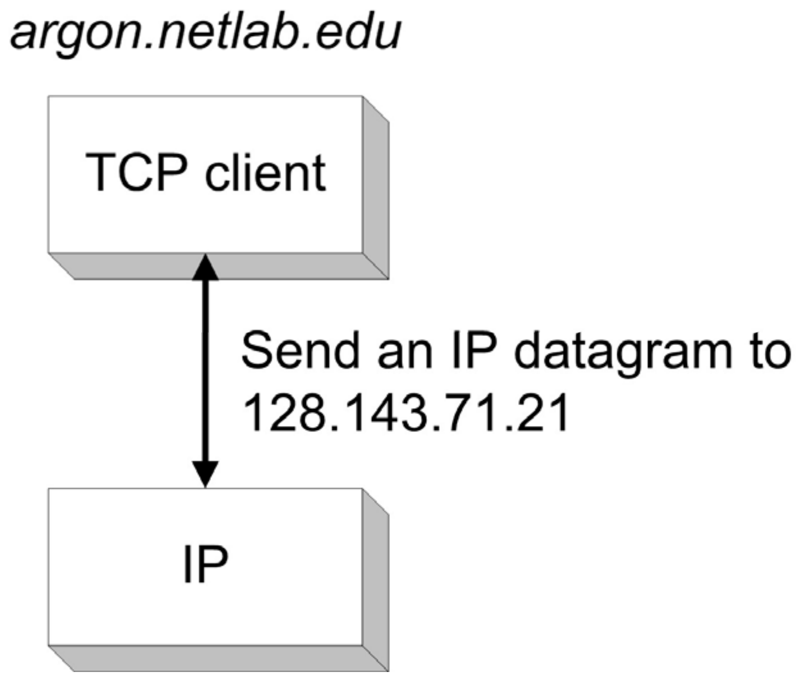
ftp21	finger	79	
telnet	23	http	80
smtp	25	nntp	119

Requesting a TCP Connection



- The HTTP client at *argon.netlab.edu* requests the TCP client to establish a connection to port 80 of the machine with address 128.141.71.21

Invoking the IP Protocol

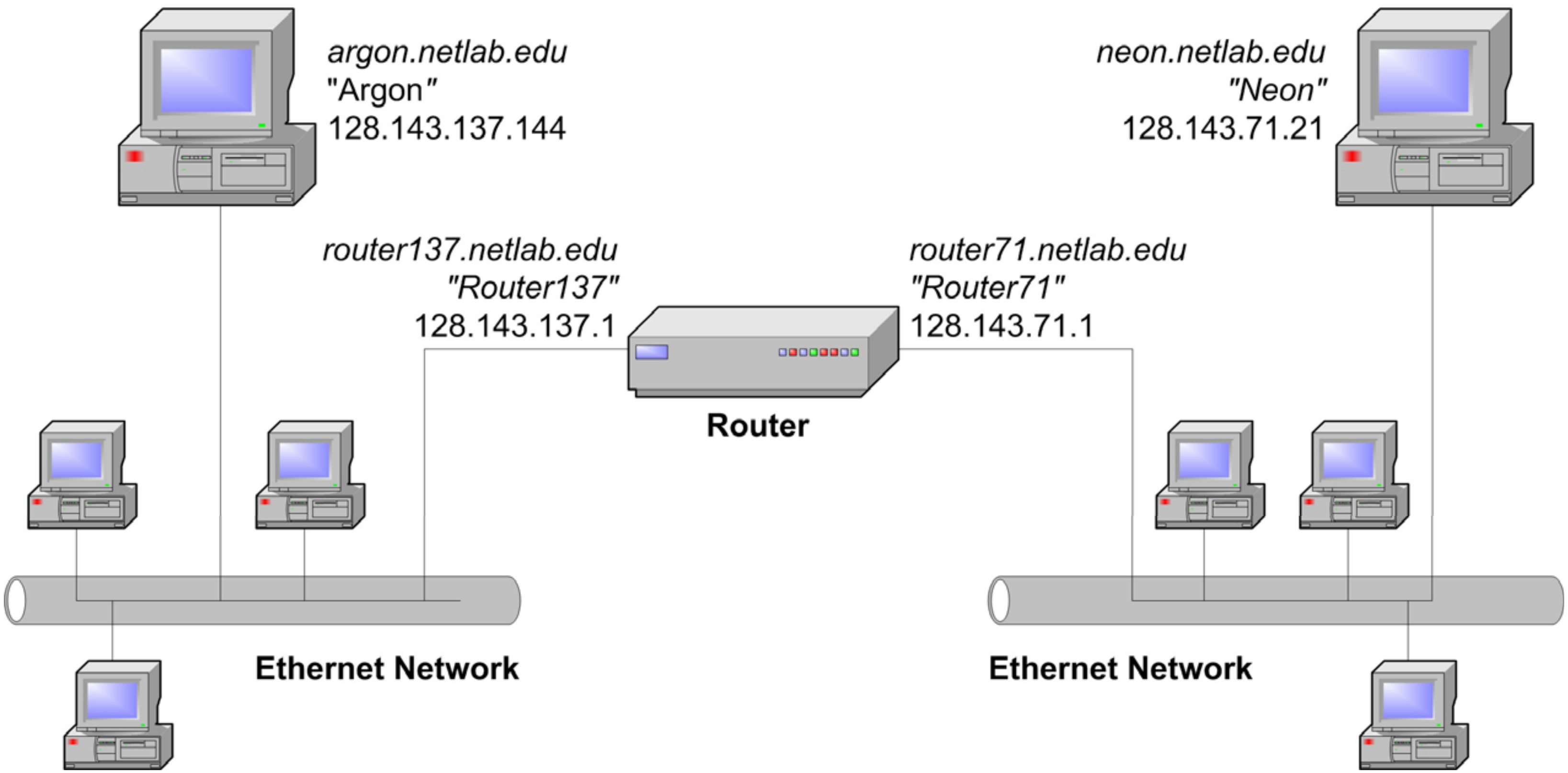


- The TCP client at *Argon* sends a request to establish a connection to port 80 at *Neon*
- This is done by asking its local IP module to send an IP datagram to *128.143.71.21*
- *(The data portion of the IP datagram contains the request to open a connection)*

Sending the IP datagram to an IP router

- *Argon (128.143.137.144)* can deliver the IP datagram directly to *Neon (128.143.71.21*), only if it is on the same local network ("subnet")
- But *Argon* and *Neon* are <u>not</u> on the same local network
  (**Q**: How does *Argon* know this?)
- So, *Argon* sends the IP datagram to its default gateway
- The default gateway is an IP router
- The default gateway for *Argon* is *Router137.netlab.edu* (128.143.137.1).

The route from ***Argon*** to ***Neon***

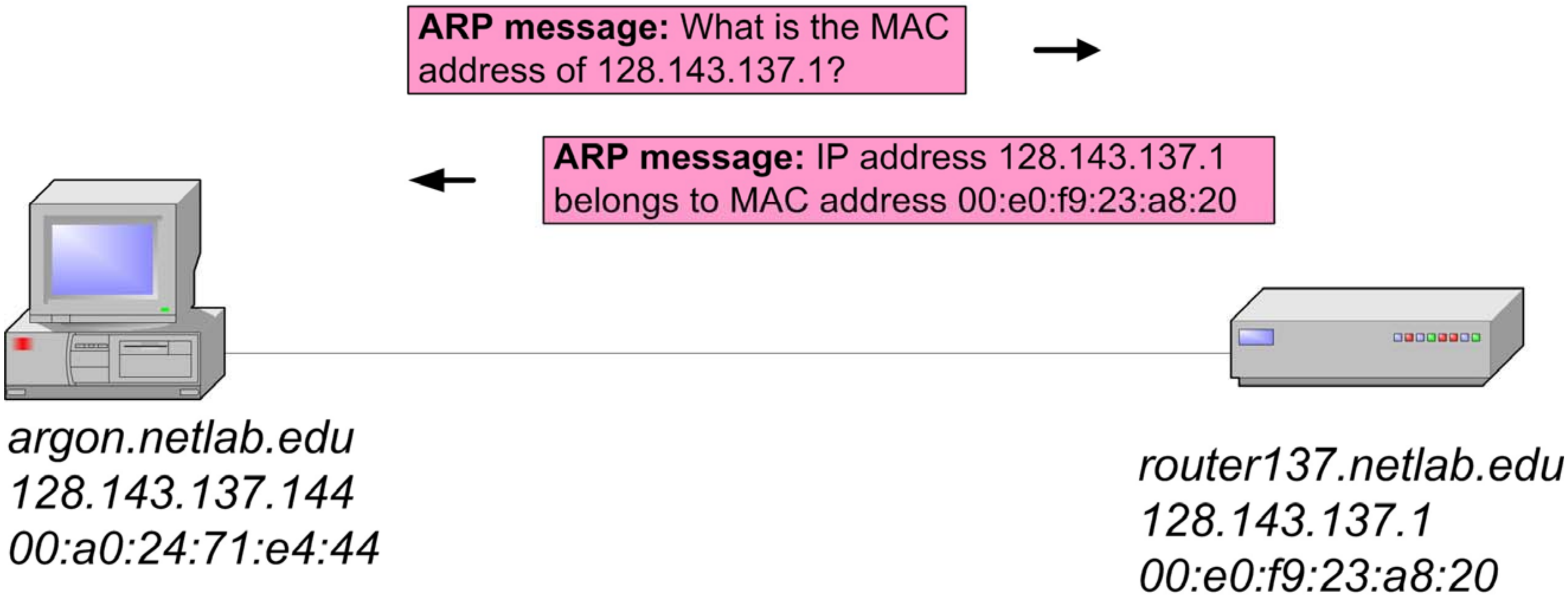


- Note that the gateway has a different name for each of its interfaces.

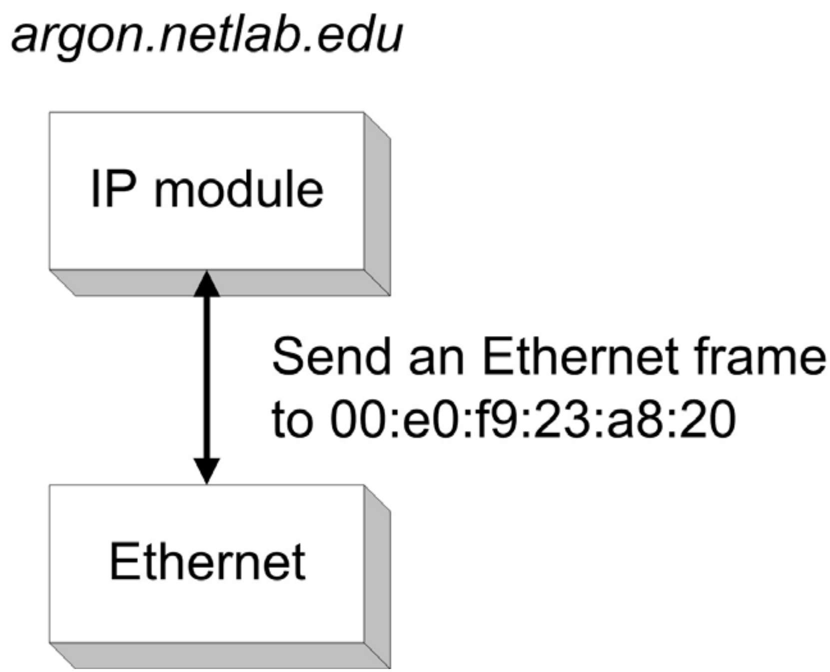
Finding the MAC address of the gateway

- To send an IP datagram to Router137, *Argon* puts the IP datagram in an Ethernet frame, and transmits the frame.
- However, Ethernet uses different addresses, so-called Media Access Control (MAC) addresses (also called: physical address, hardware address).
- Therefore, *Argon* must first translate the IP address 128.143.137.1 into a MAC address.
- The translation of addressed is performed via the Address Resolution Protocol (ARP)

Address resolution with ARP

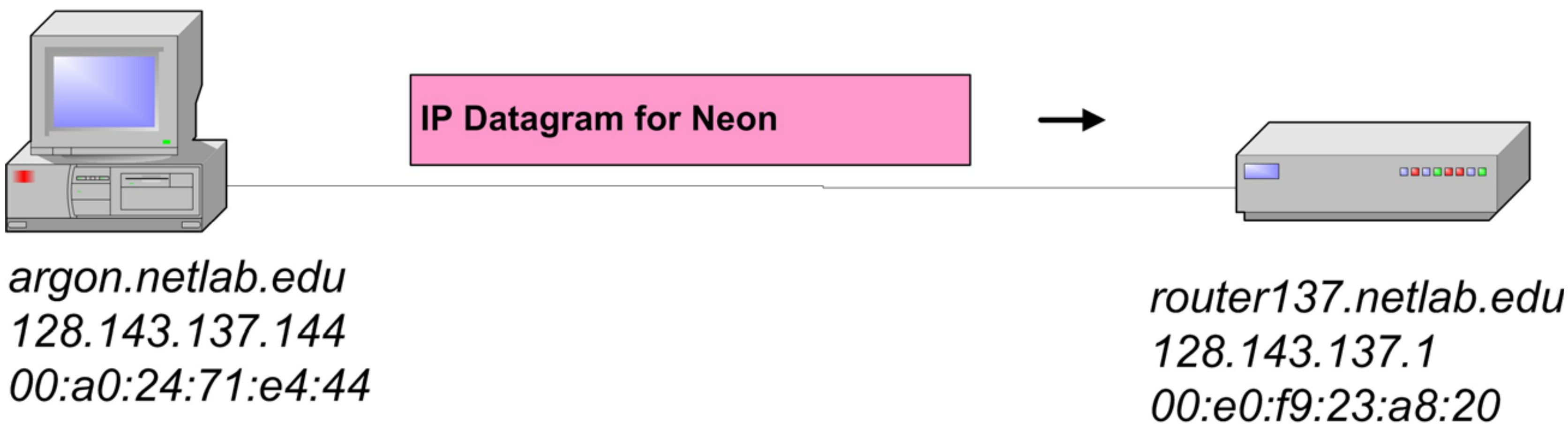


Invoking the device driver



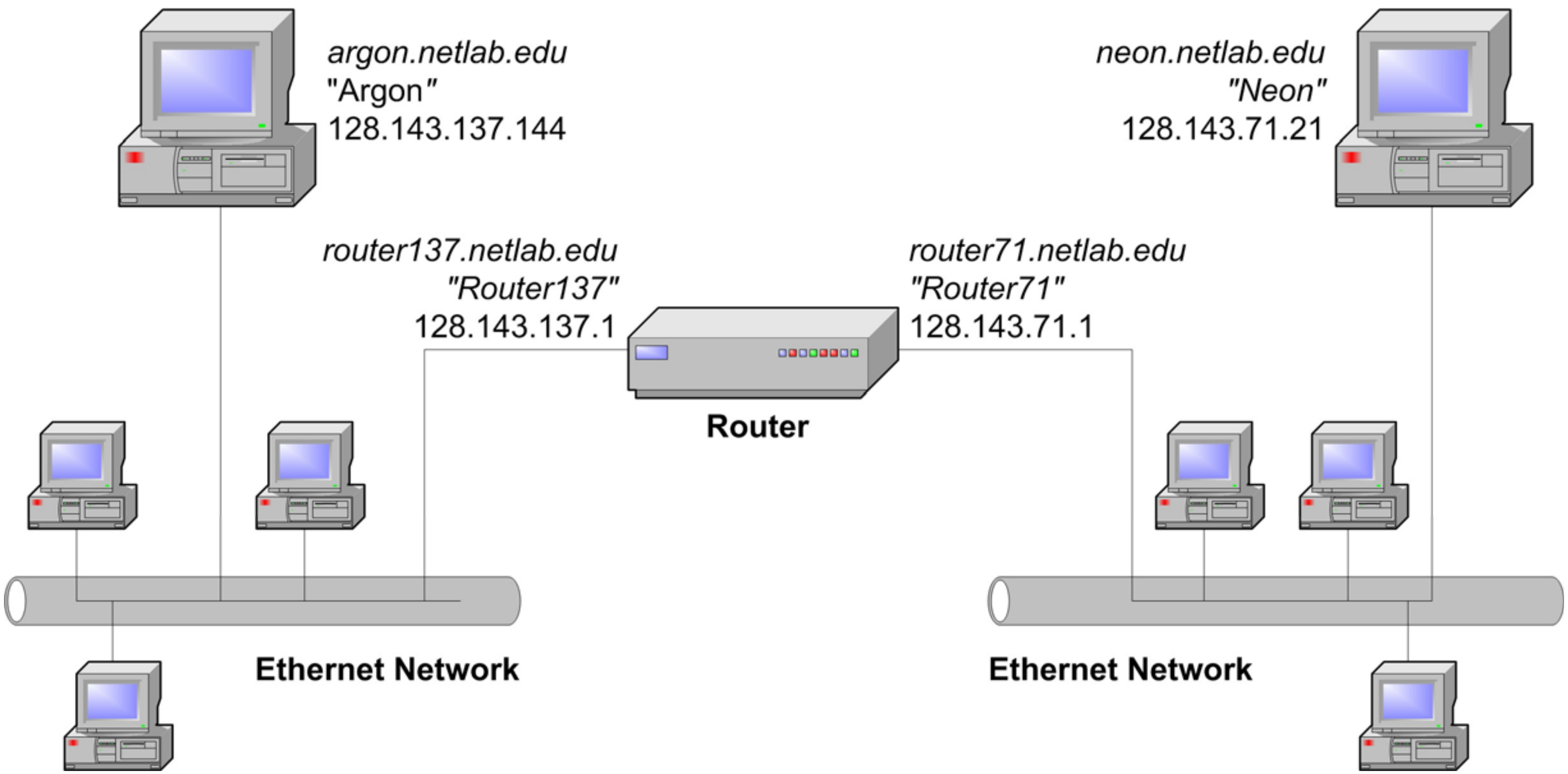
- The IP module at *Argon*, tells its Ethernet device driver to send an Ethernet frame to address *00:e0:f9:23:a8:20*

Sending an Ethernet frame



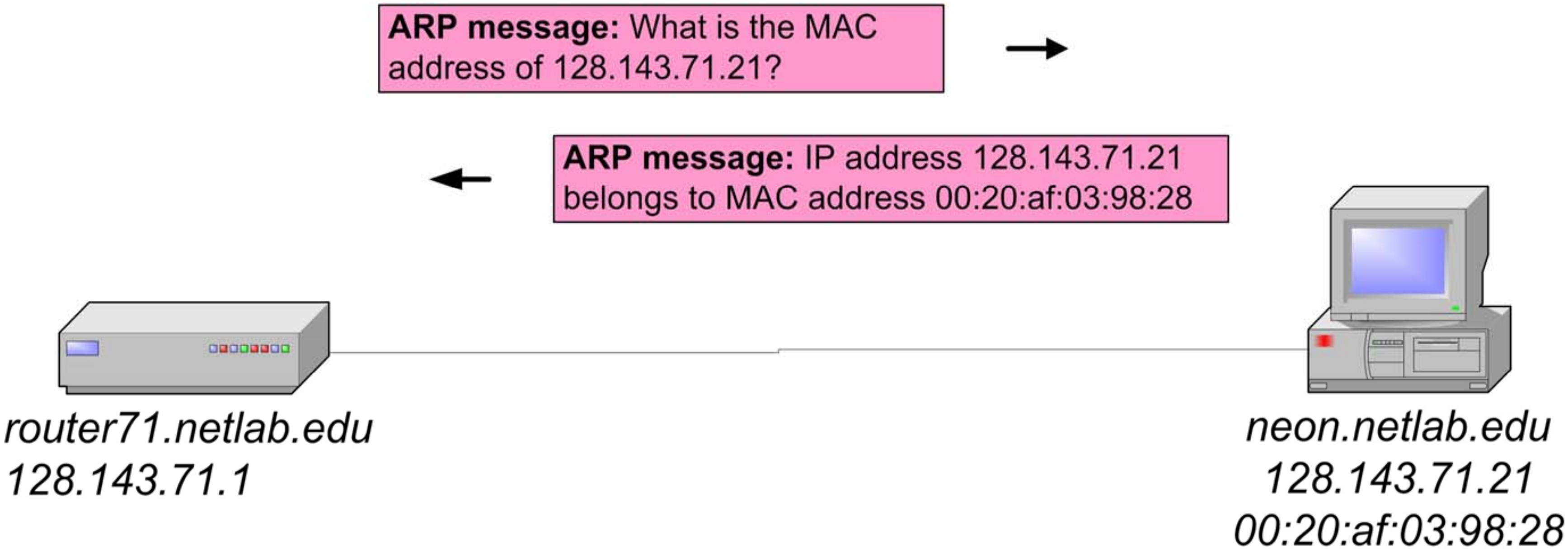
- The Ethernet device driver of *Argon* sends the Ethernet frame to the Ethernet network interface card (NIC)
- The NIC sends the frame onto the wire

Forwarding the IP datagram



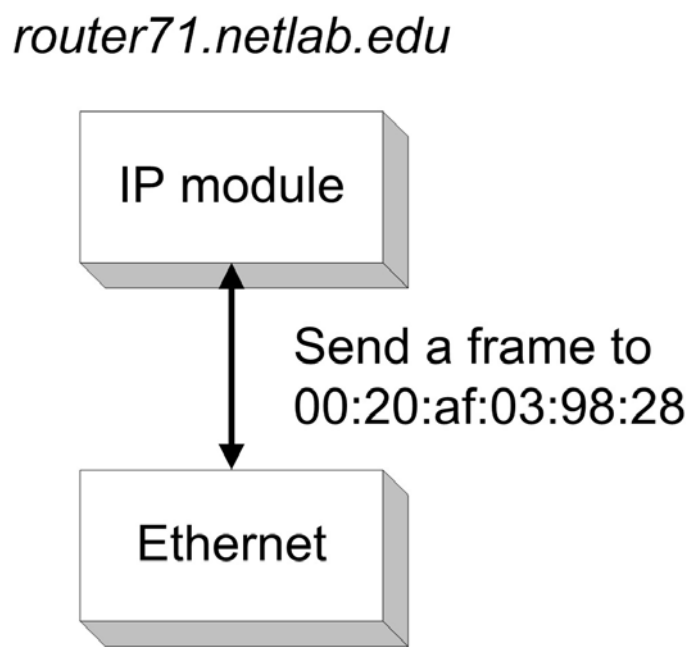
- The IP router receives the Ethernet frame at interface 128.143.137.1, recovers the IP datagram and determines that the IP datagram should be forwarded to the interface with name 128.143.71.1
- The IP router determines that it can deliver the IP datagram directly

Another lookup of a MAC address



- The router needs to find the MAC address of *Neon*.
- Again, ARP is invoked, to translate the IP address of *Neon* (128.143.71.21) into the MAC address of neon (00:20:af:03:98:28).

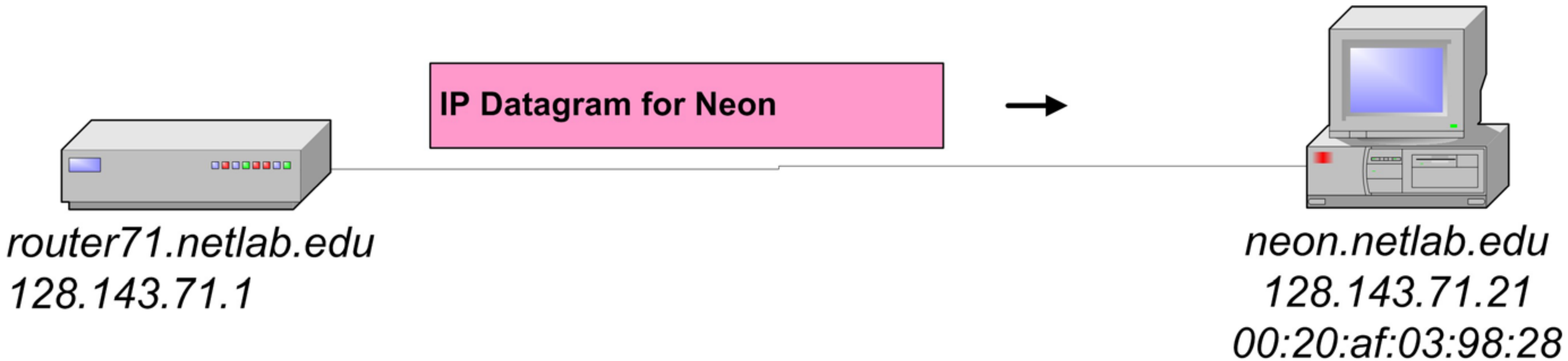
Invoking the Device Driver at the Router



- The IP protocol at *Router71*, tells its Ethernet device driver to send an Ethernet frame to address *00:20:af:03:98:28*

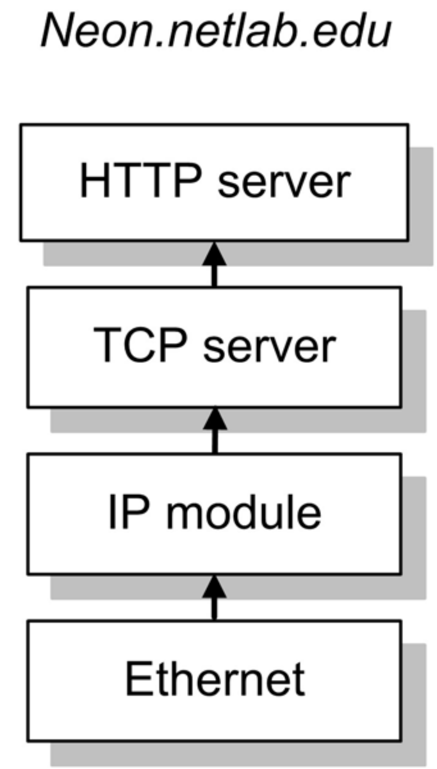
Sending another Ethernet frame

- The Ethernet device driver of *Router71* sends the Ethernet frame to the Ethernet NIC, which transmits the frame onto the wire.



Data has arrived at Neon

- *Neon* receives the Ethernet frame
- The payload of the Ethernet frame is an IP datagram which is passed to the IP protocol.
- The payload of the IP datagram is a TCP segment, which is passed to the TCP server



Wrapping up the example

- Data traverses a sequence of layers
- Each layer has protocols to handle the packets
- Next Lecture (Lab 2)
  - Layered architecture of the Internet
  - Protocols at each layer