

CS518

# Packet Handling in Linux

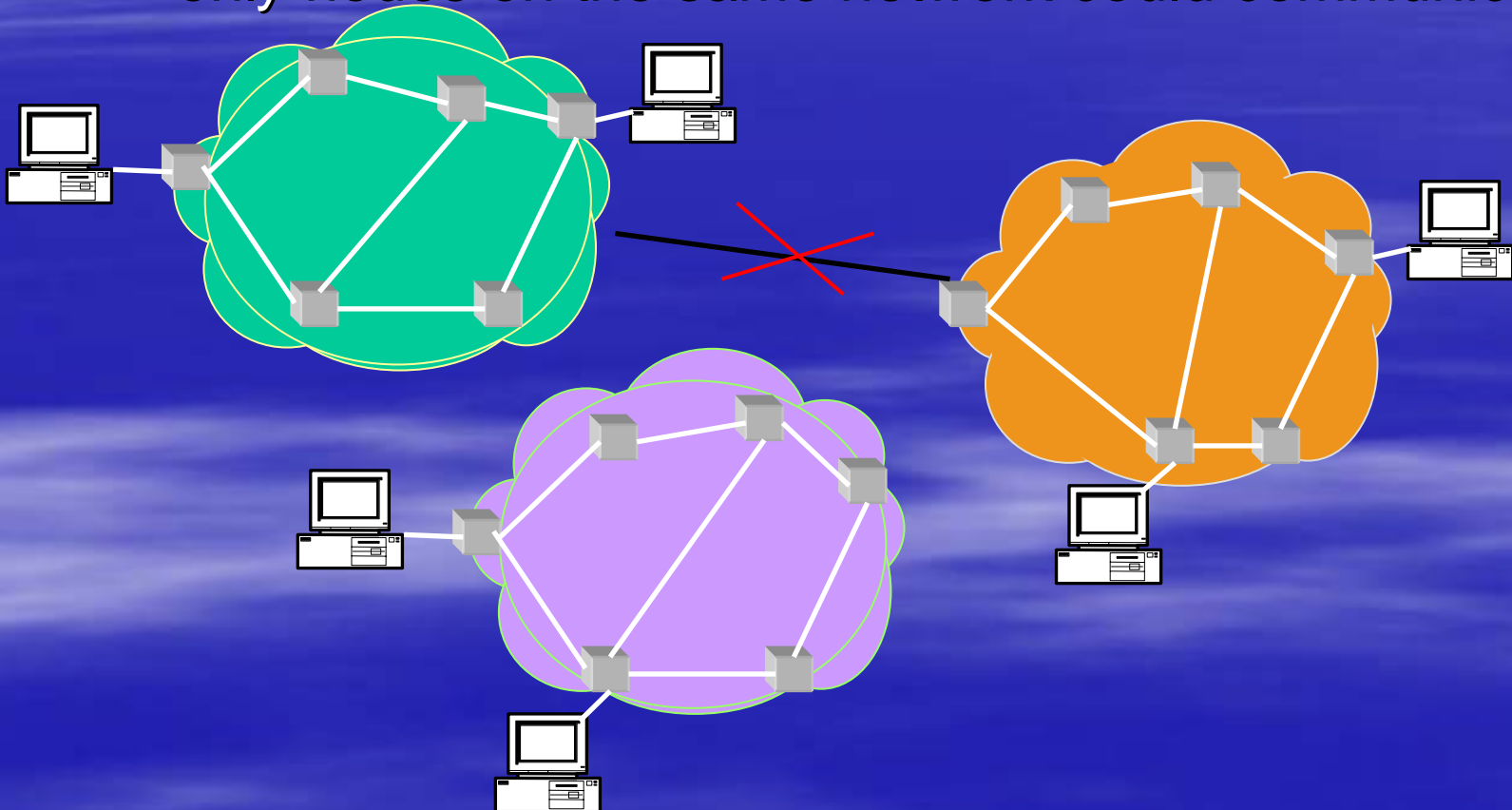
--Gaurav Dawra



- **Internetworking: Past and Present Overview**
- TCP/IP and OSI Model
- Sockets Overview
- Internetworking: Past and Present Overview
- IP solutions:
  - Routing
  - Packet Forwarding
  - Routing Vs Forwarding
  - Address resolution Protocol (ARP)
- Wireless Networking : Brief Overview

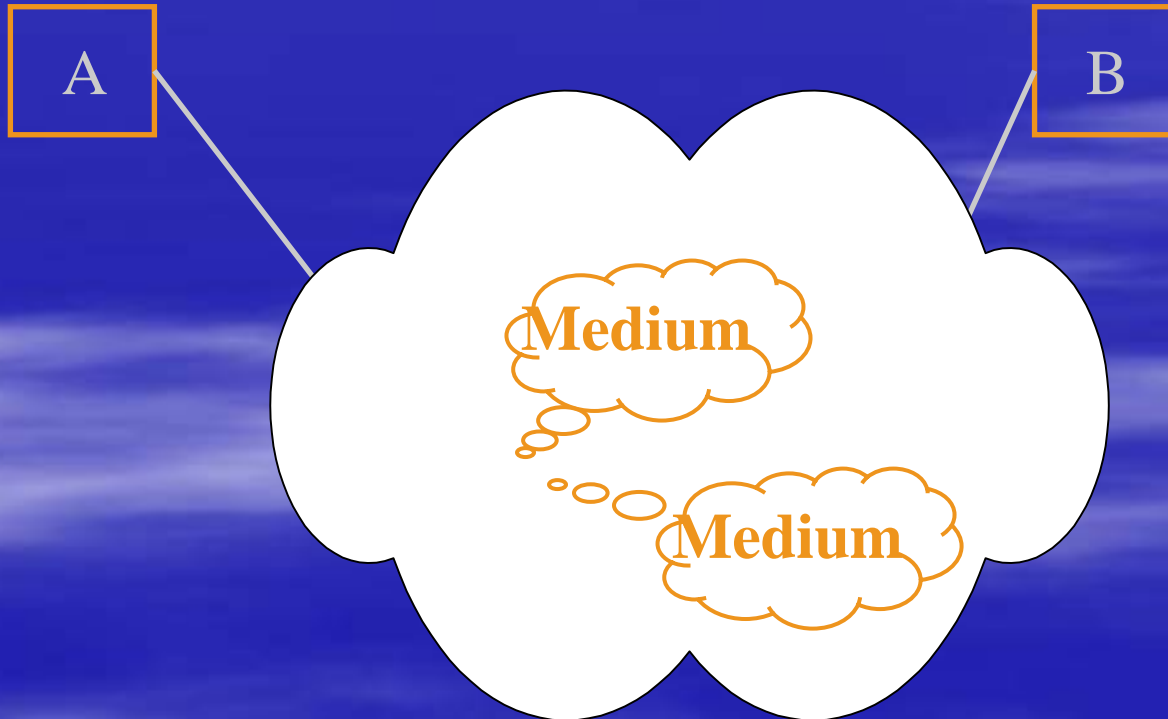
# The Problem

- Before Internet: different packet-switching networks (e.g., ARPANET)
  - only nodes on the same network could communicate

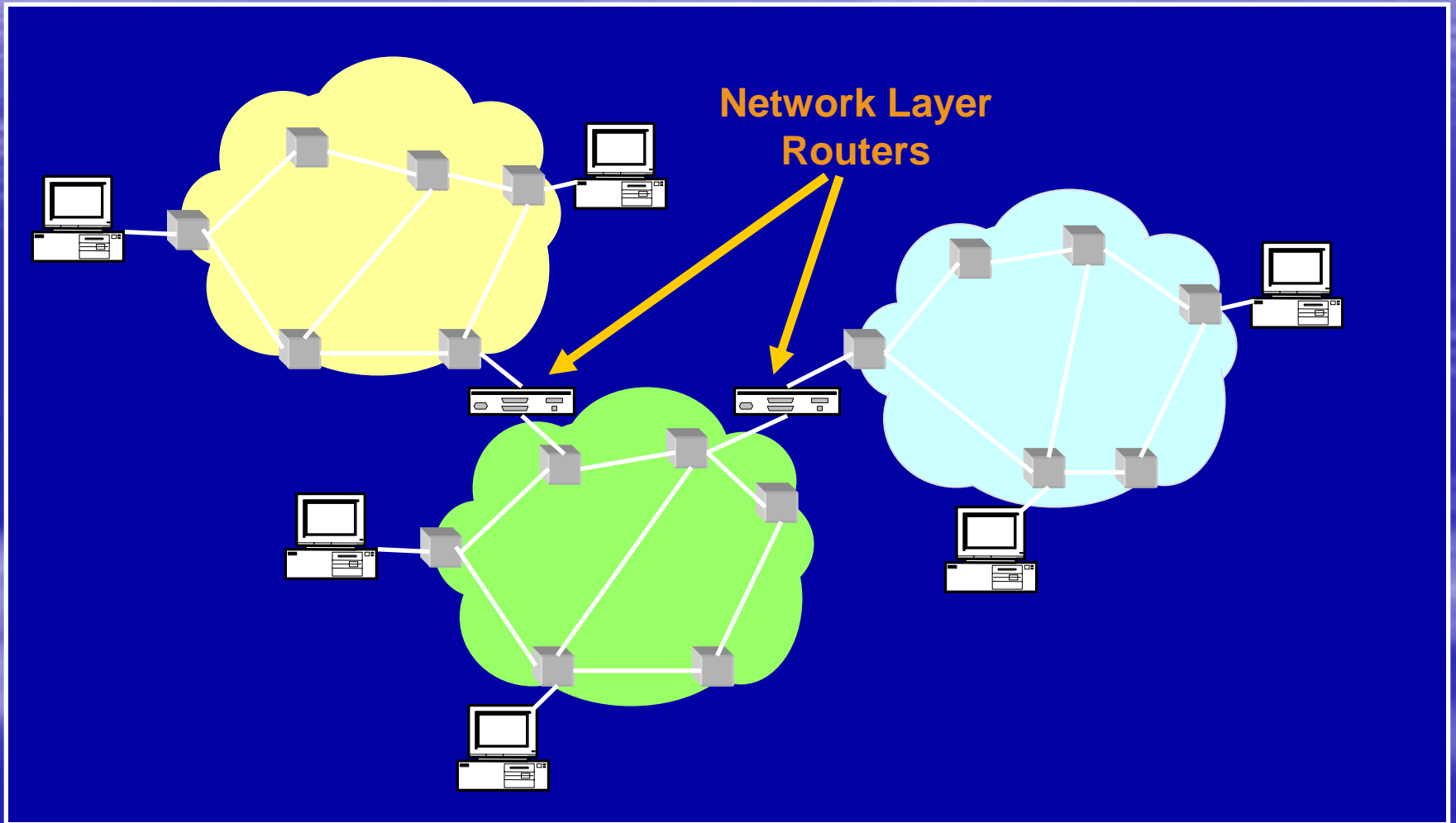


# The Internetworking Problem

- Two nodes communicating across a “**network of networks**” ...
  - How to send packets through this strange medium ?



# Solution





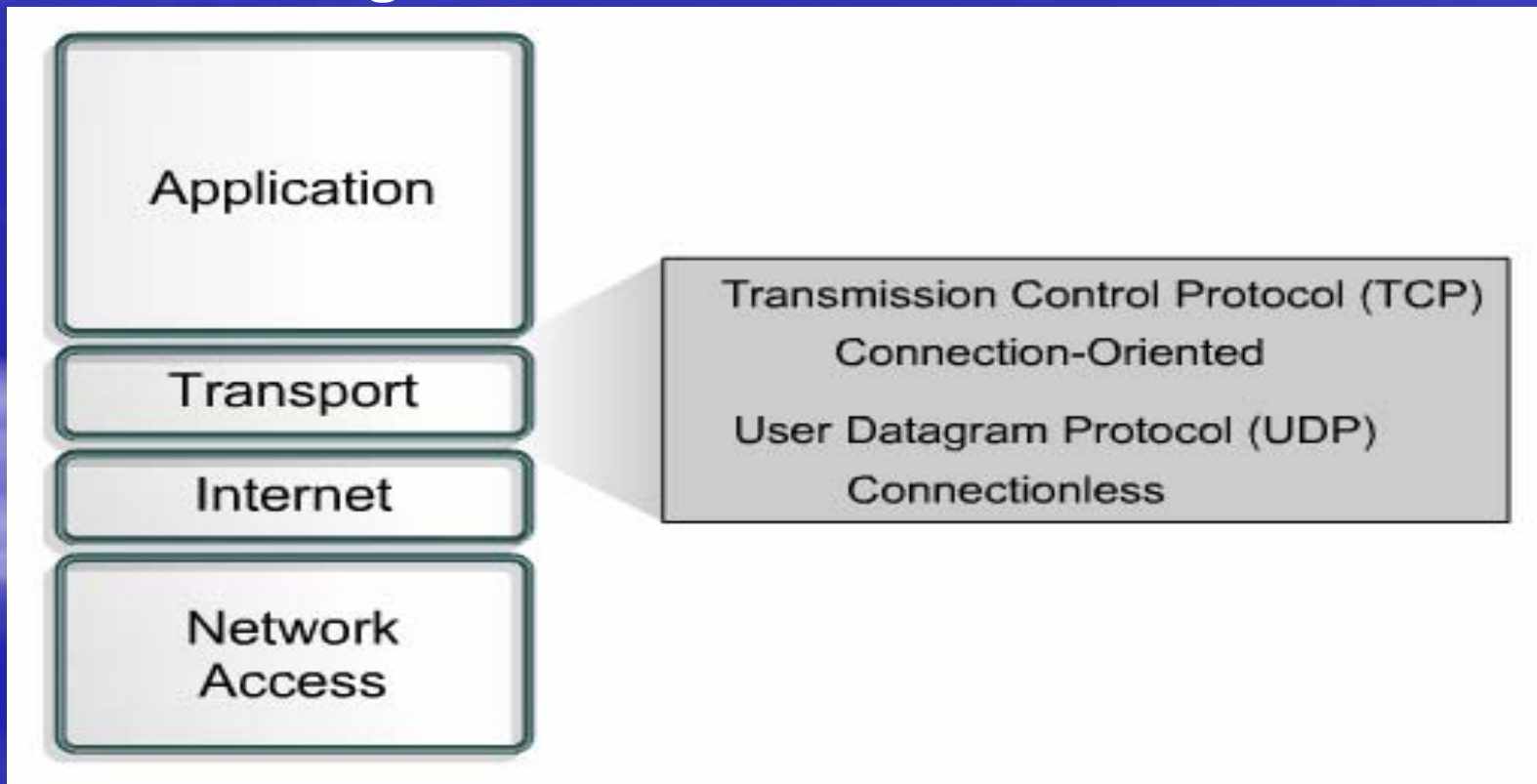
# Where are we?

- Internetworking: Past and Present Overview
- **TCP/IP and OSI Model**
- Sockets Overview
- Internetworking: Past and Present Overview
- IP solutions:
  - Routing
  - Packet Forwarding
  - Routing Vs Forwarding
  - Address resolution Protocol (ARP)
- Wireless Networking : Brief Overview

# TCP/IP (Transport Layer)

## Functions of Transport Layer

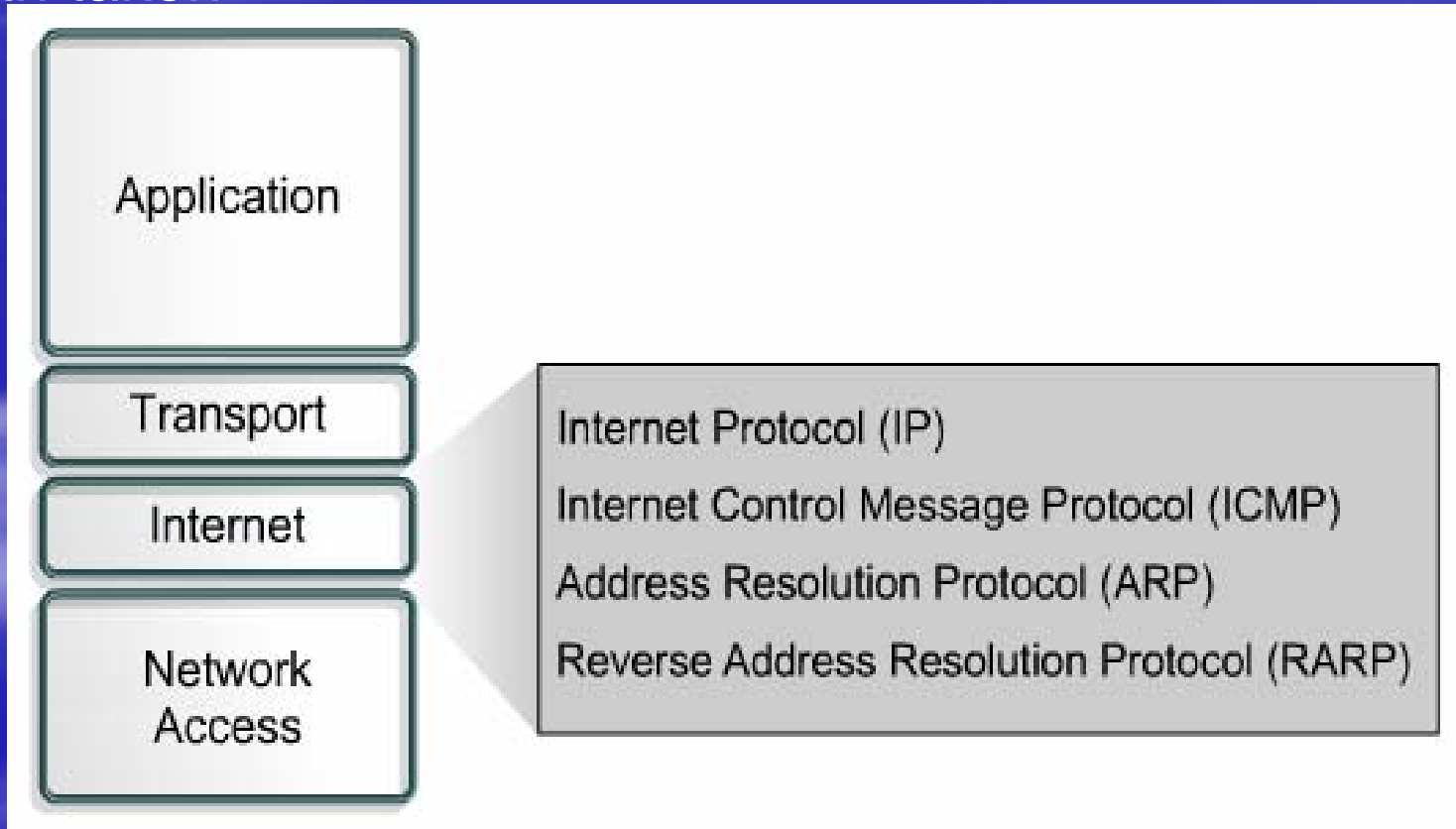
- Establishing End to End Connections
- Ensuring Data Reliability
- Providing Flow Control



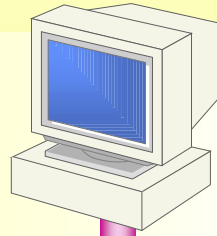
# TCP/IP (Network Layer)

## Internet layer

-- Send packets from a **network node** and have them arrive at the destination node independent of the path taken.



# TCP/IP Layer



HTTP

FTP

Application Layer

TCP

UDP

Transport Layer

IP

ICMP

IGMP

ARP

Internet Layer

ATM

Ethernet

Network Interface Layer

# OSI Model

Application Layer

Presentation Layer

Session Layer

Transport Layer

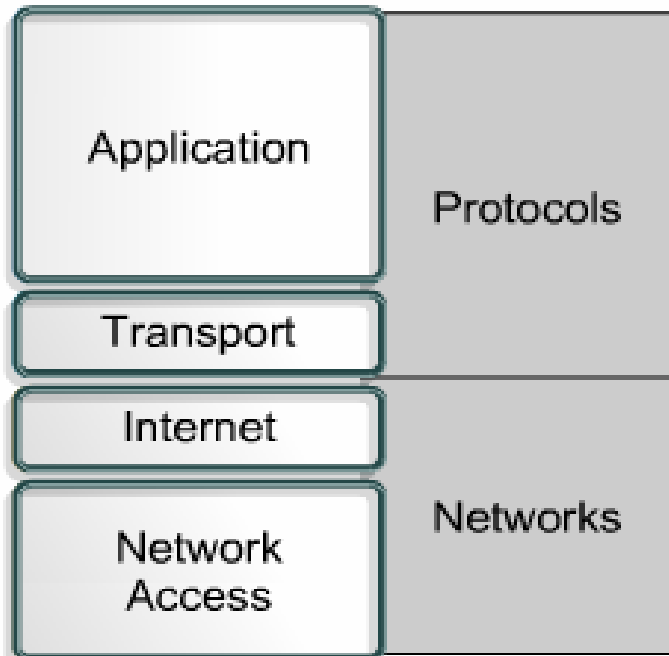
Network Layer

Data Link Layer

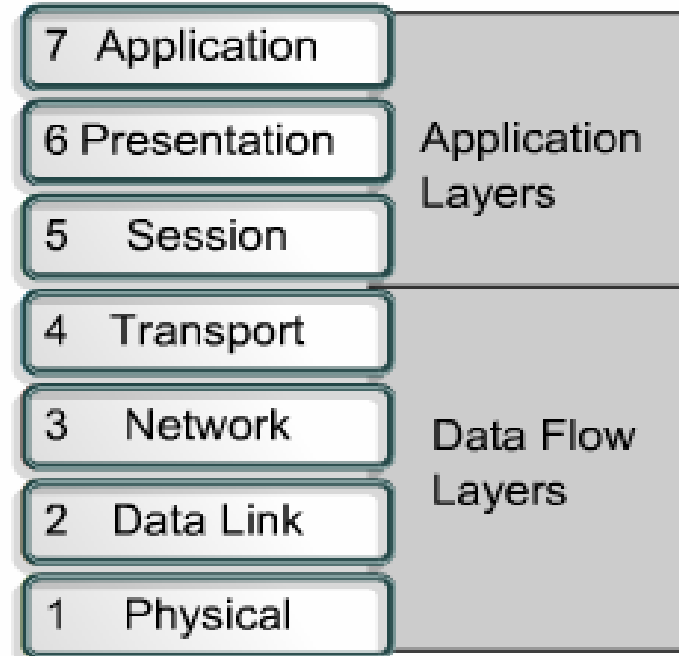
Physical Layer

# Comparing the OSI Model and TCP/IP Model

**TCP/IP Model**



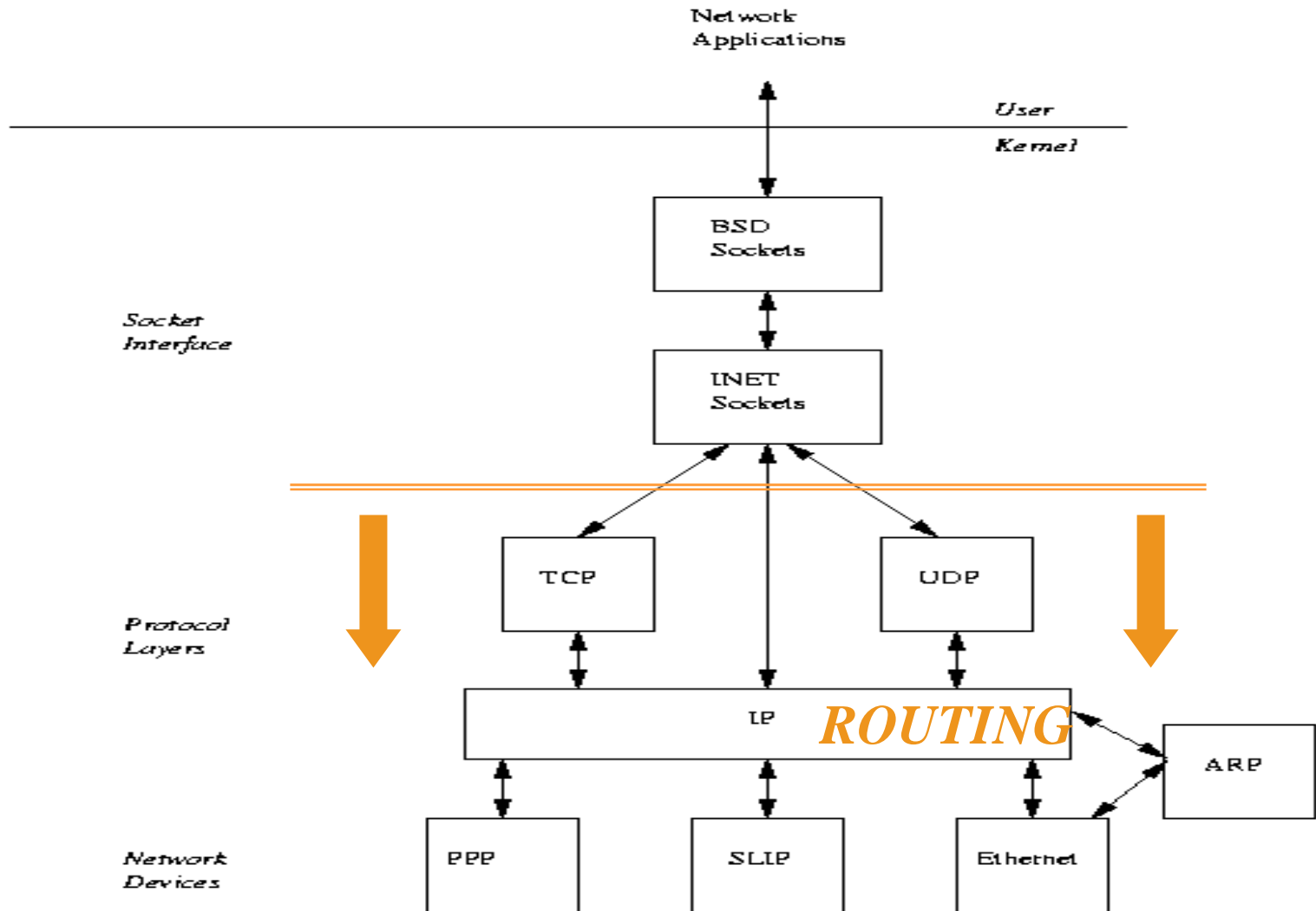
**OSI Model**



# Where are we?

- Internetworking: Past and Present Overview
- TCP/IP and OSI Model
- **Sockets Overview**
- IP solutions:
  - Routing
  - Packet Forwarding
  - Routing Vs Forwarding
  - Address resolution Protocol (ARP)
- Wireless Networking: Brief Overview

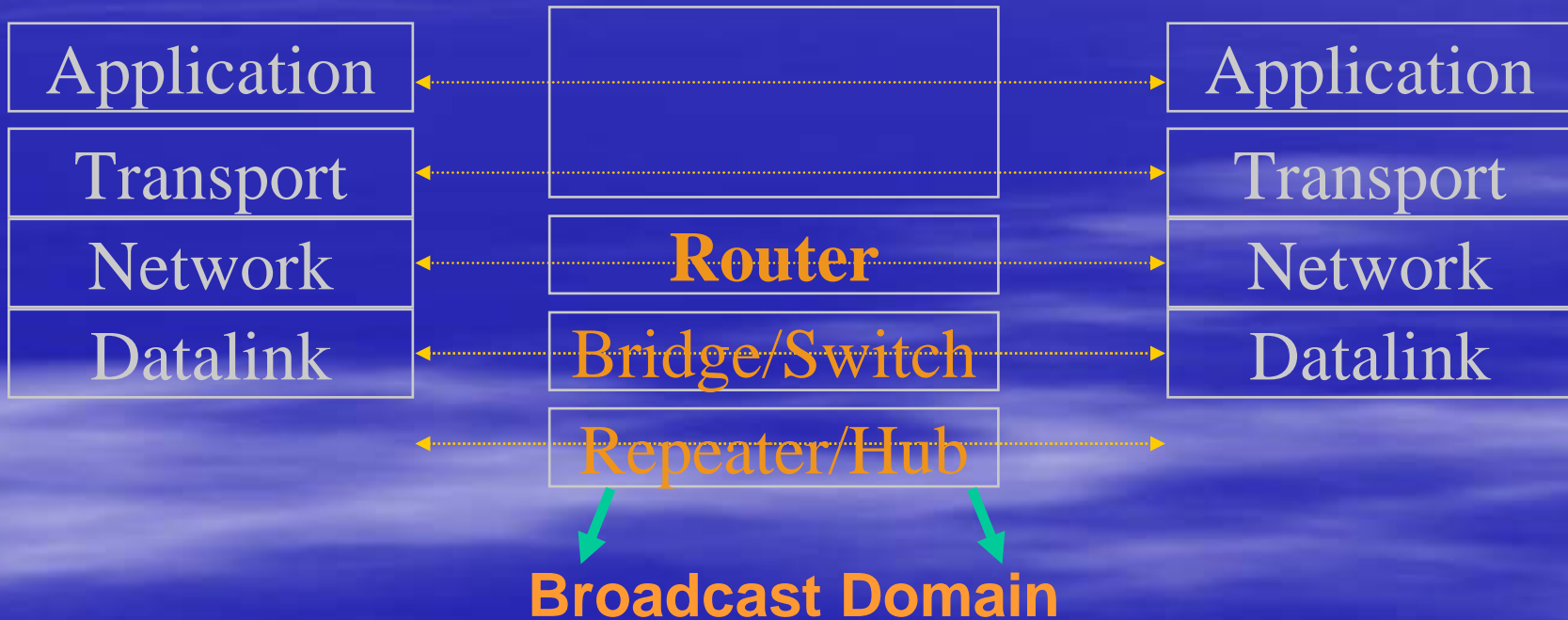
# Overall Picture (One View)



# Where are we Now?

- Internetworking: Past and Present Overview
- TCP/IP and OSI Model
- IP addressing
- Sockets Overview
- **IP solutions:**
  - **IP Packet Forwarding**
  - **Routing**
  - **Routing Vs Forwarding**
  - Address resolution Protocol (ARP)
- Wireless Networking : Brief Overview

# TCP/IP Model



# *Routing vs. Forwarding*

- ❑ **Forwarding**: select an *output port* based on destination address and **routing table**
- ❑ **Routing**: process by which **routing table** is *built..*
  - ❑ ... so that the series of *local forwarding decisions* takes the packet to the destination with very high probability
  - ❑ ... the path chosen/allotted to the packet is *efficient* in some sense... Filtering of packets

# IP Forwarding



**Direct Connectivity**  
-- Same Network



**Indirect Connectivity**  
--Not on the Same Network

# IP Forwarding

## *(direct connectivity)*

- *Source & Destination in same network*  
*(direct connectivity)*
  - Recognize that destination IP address is on *same network*.
  - Find the **destination** LAN address.
  - Send IP *packet encapsulated* in LAN frame directly to the destination LAN address
  - IP in MAC/LAN FRAME: Encapsulation => source/destination IP addresses don't change

# IP Forwarding

## *(indirect connectivity)*

- B) *Source & Destination in different networks (indirect connectivity)*
  - Recognize that destination IP address is **NOT** on same network.
  - Look up destination IP address in a (forwarding) table to find a match, **called the next hop router IP address.**
  - Send packet encapsulated in a LAN frame to the LAN address corresponding to the IP address of the next-hop router.

# Forwarding Table

## Command: 'route'

- Kernel IP routing table

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
153.90.192.0	*	255.255.248.0	U	0	0	0	eth0
5.0.0.0	*	255.0.0.0	U	0	0	0	srcr
6.0.0.0	*	255.0.0.0	U	0	0	0	wlan0
169.254.0.0	*	255.255.0.0	U	0	0	0	eth0
127.0.0.0	*	255.0.0.0	U	0	0	0	lo
default	epgwy192.msu.mo	0.0.0.0	UG	0	0	0	eth0

# Addressing

- *[1] How to find if destination is in the same network ?*

– IP address = **network ID** + **host ID**.

-- *If the Source and Destination ID's Match (Same Network)*

--*If not, We need Router*



*Boundary*

192.168.0.10    Netmask 255.255.255.0

10.1.2.1        Netmask 255.0.0.0

# Address Resolution

- *How to find the LAN address corresponding to an IP address ?*
  - *Address Resolution Problem.*
  - Solution: ARP, RARP (*I will come back to this*)

# IP Forwarding: Example Scenario

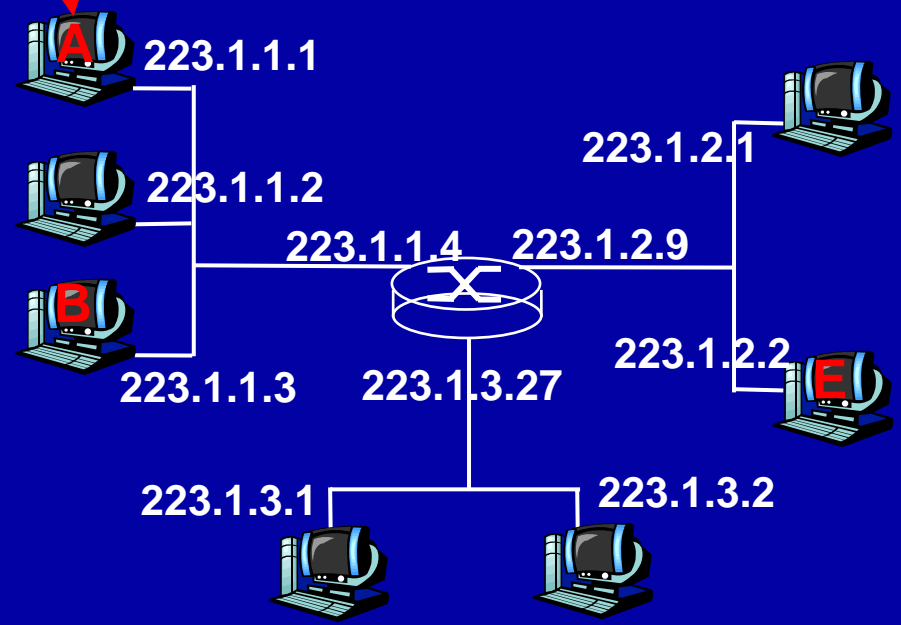
## routing table in A

Dest. Net.	next router	Nhops
223.1.1		1
223.1.2	223.1.1.4	2
223.1.3	223.1.1.4	2

IP datagram:

misc fields	source IP addr	dest IP addr	data

datagram remains unchanged, as it travels source to destination  
addr fields of interest here



# IP Forwarding (Direct)

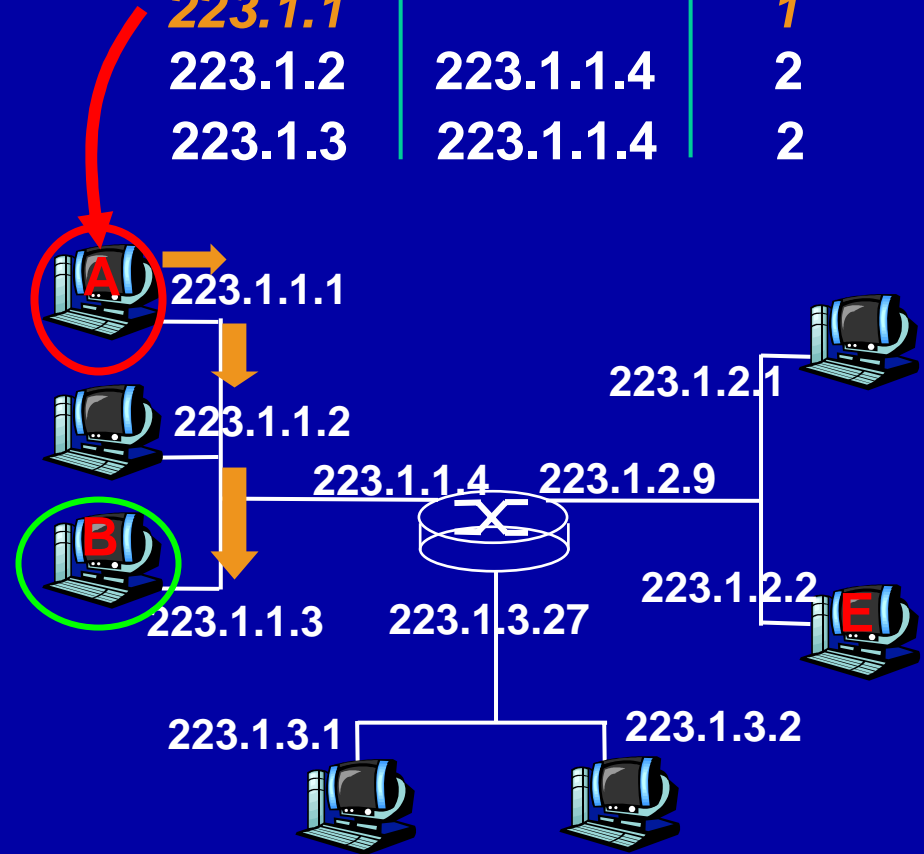
misc fields	223.1.1.1	223.1.1.3	data
-------------	-----------	-----------	------

Starting at A, given IP datagram addressed to B:

look up net. address of B  
find B is on same net. as A  
link layer will send datagram directly to B inside link-layer frame

B and A are directly connected

Dest. Net.	next router	Nhops
223.1.1		1
223.1.2	223.1.1.4	2
223.1.3	223.1.1.4	2



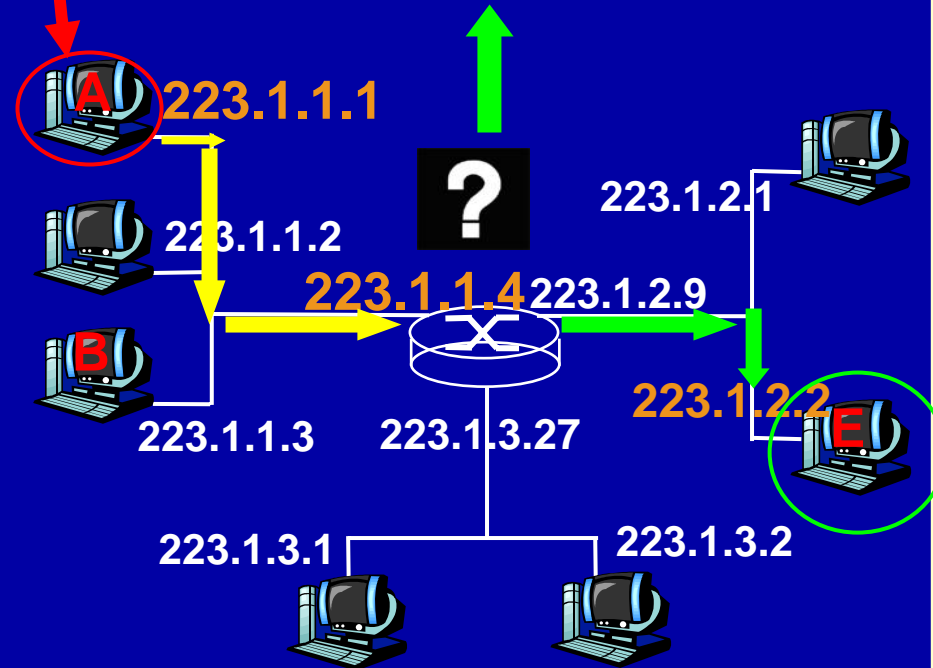
# IP Forwarding (Indirect): Step 1

misc fields	223.1.1.1	223.1.2.2	data
-------------	-----------	-----------	------

Starting at A, dest. E:  
 look up network address of E  
 E on *different* network  
 A, E not directly attached

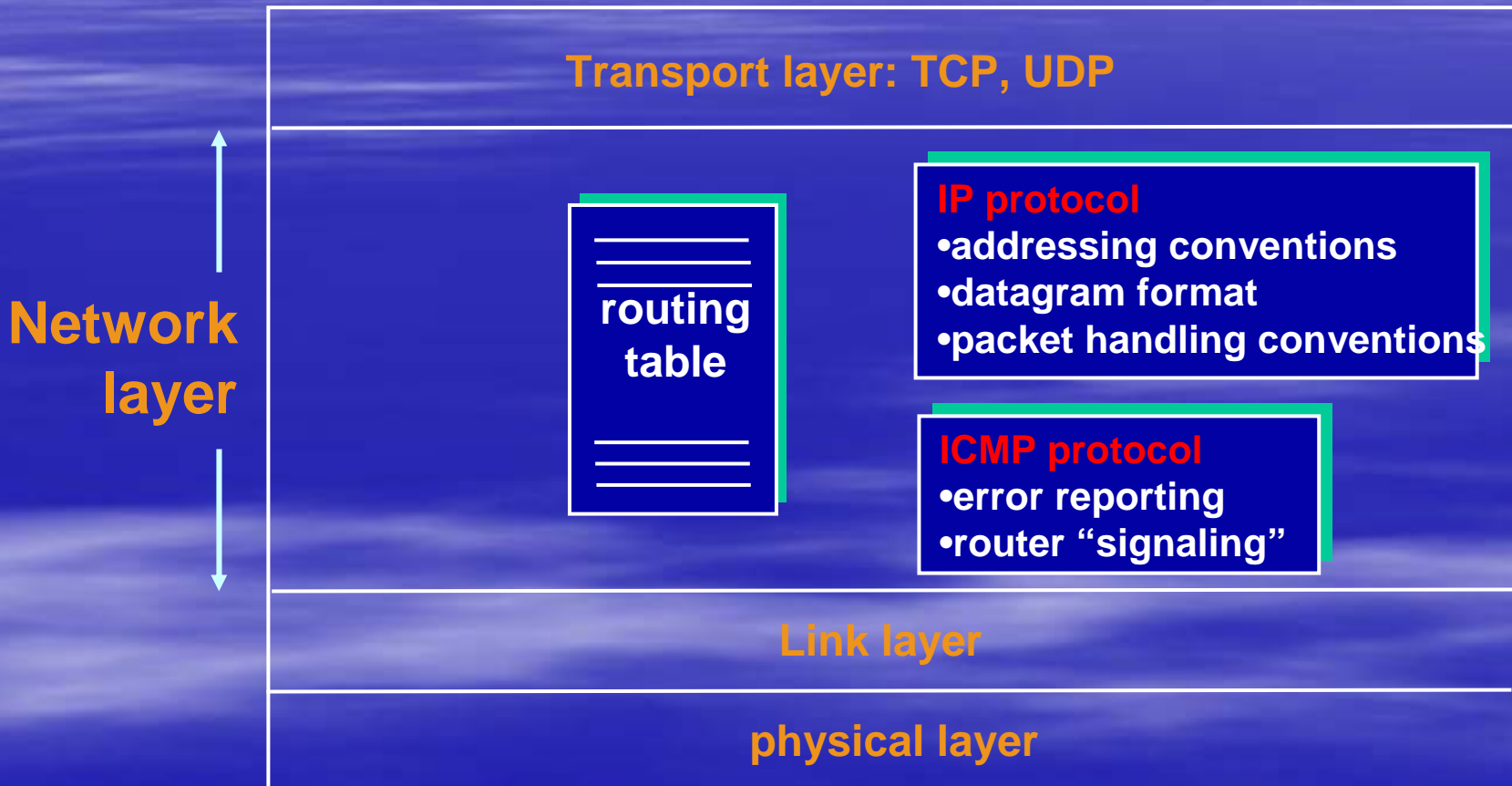
routing table: next hop router to E is 223.1.1.4  
 link layer sends datagram to router 223.1.1.4 inside link-layer frame  
 datagram arrives at 223.1.1.4  
 continued.....

Dest. Net.	next router	Nhops
223.1.1		1
223.1.2	223.1.1.4	2
223.1.3	223.1.1.4	2

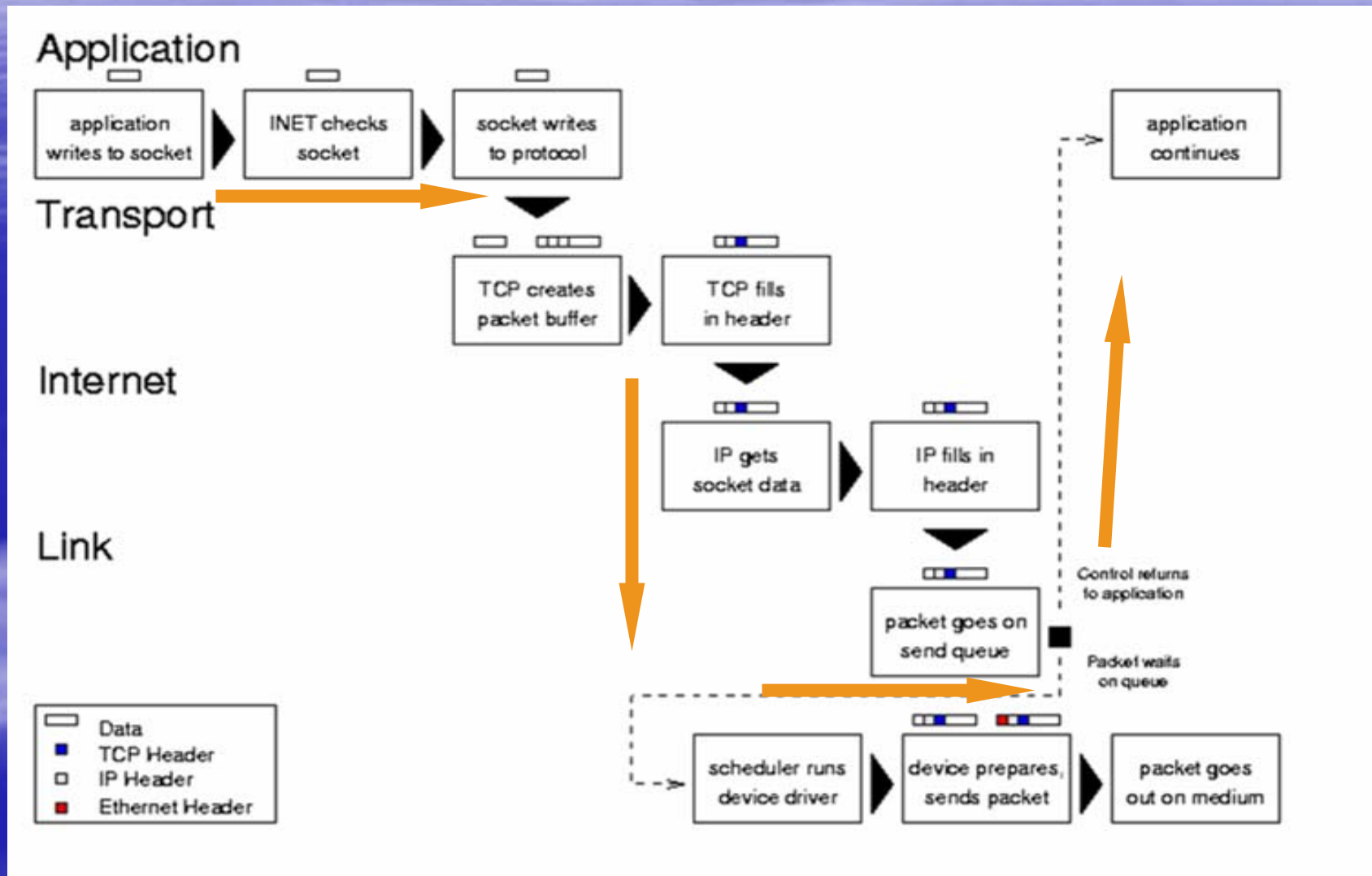


# The Internet Network layer

Host, router network layer functions:

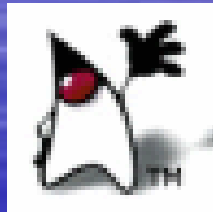


# Peek into source Code (Sending Out)



# Peek into source Code

## ipv4/net/ip\_output.c



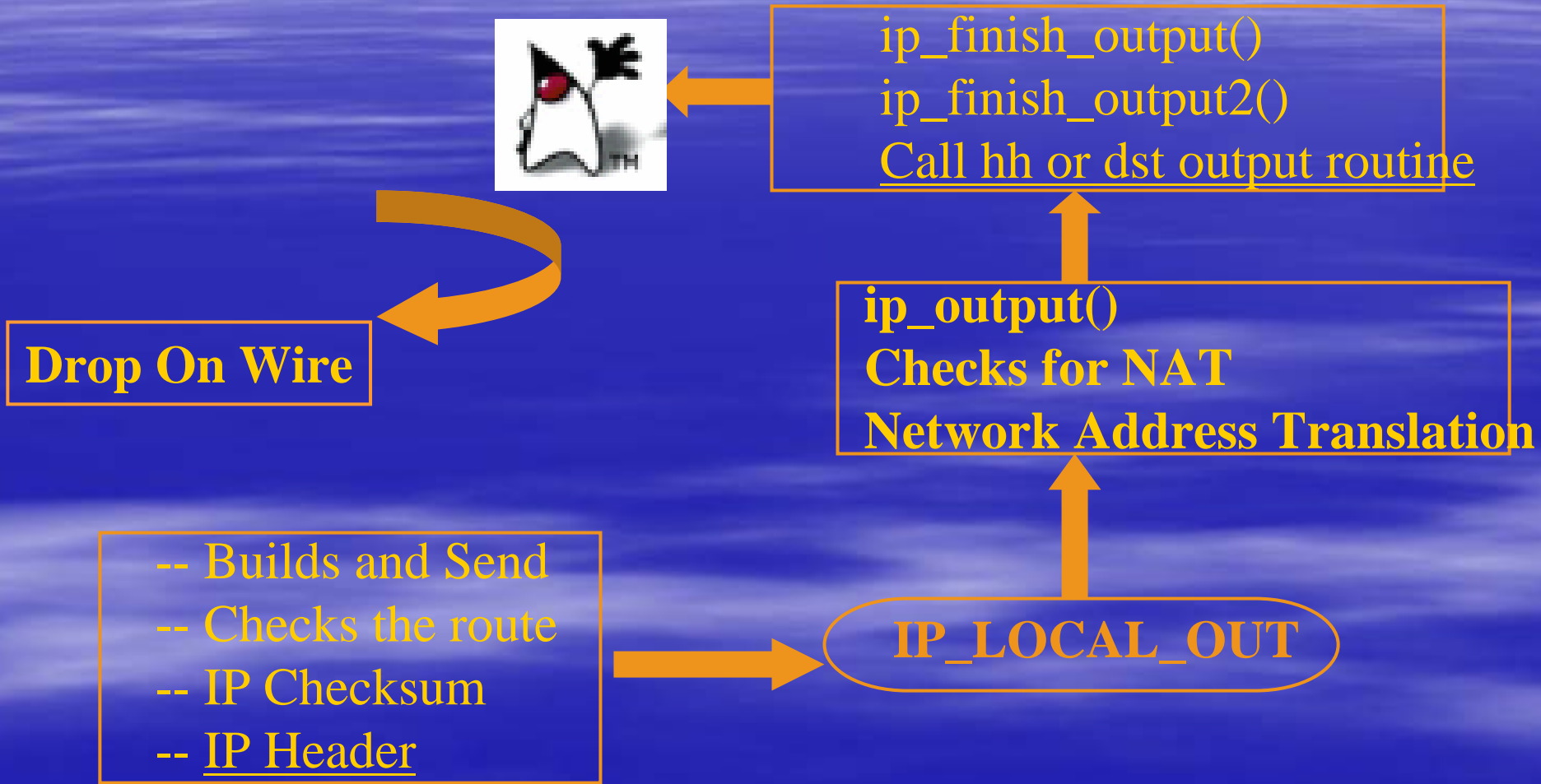
`ip_finish_output()`  
`ip_finish_output2()`  
Call hh or dst output routine

`ip_output()`  
Checks for NAT  
Network Address Translation

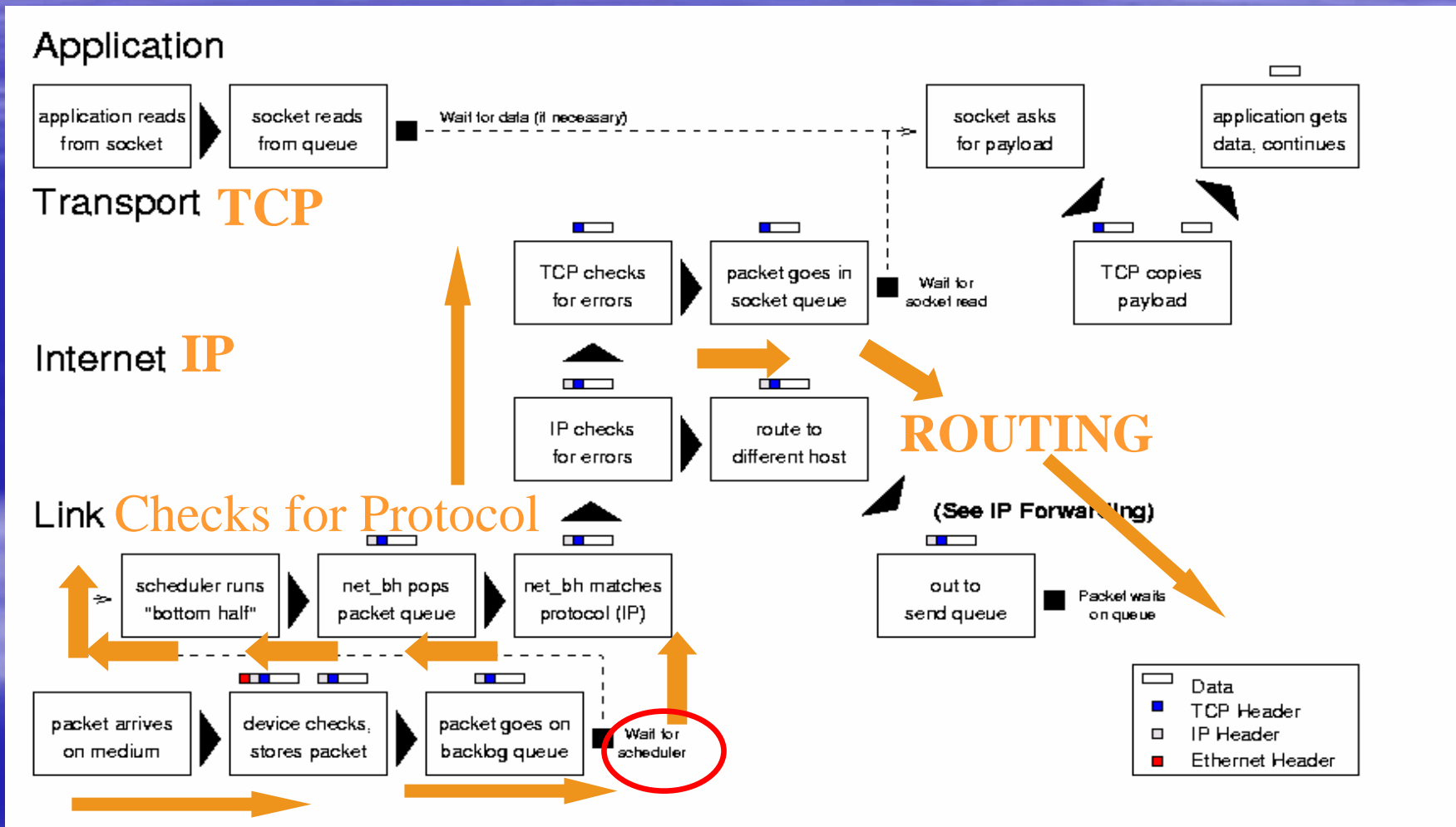
-- Builds and Send  
-- Checks the route  
-- IP Checksum  
-- IP Header

`IP_LOCAL_OUT`

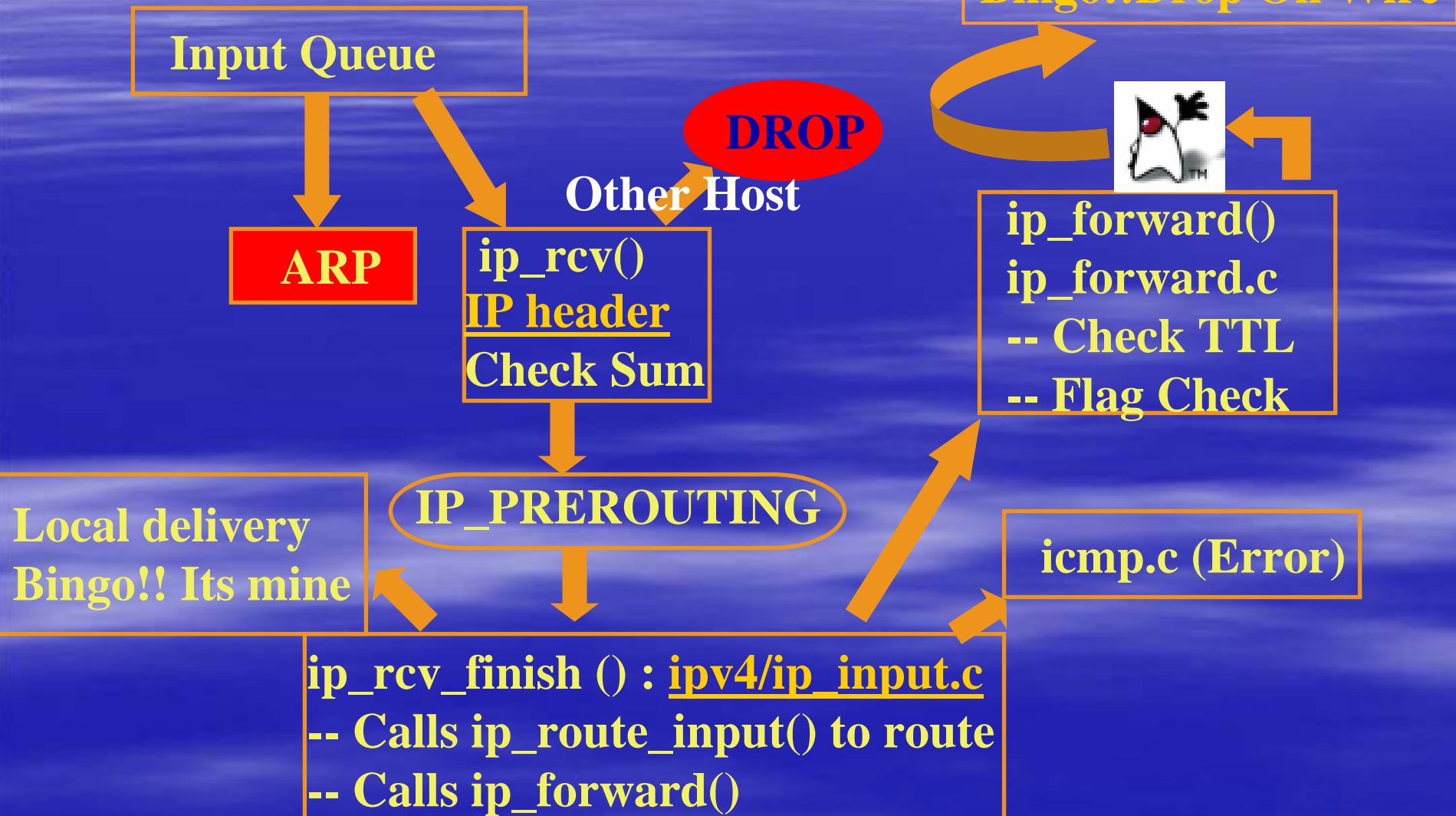
Drop On Wire



# Peek into source Code: ( Packet Receiving: ip\_input.c)



# Peek into source Code: ( Packet Receiving: `ip_input.c` )



# Where are we?

- Internetworking: Past and Present Overview
- TCP/IP and OSI Model
- IP addressing
- Sockets Overview
- Internetworking: Past and Present Overview
- IP solutions:
  - IP Packet Forwarding
  - Routing
  - Routing Vs Forwarding
  - **Address resolution Protocol (ARP)**
- Wireless Networking: Brief Overview

# Question: Why ARP?

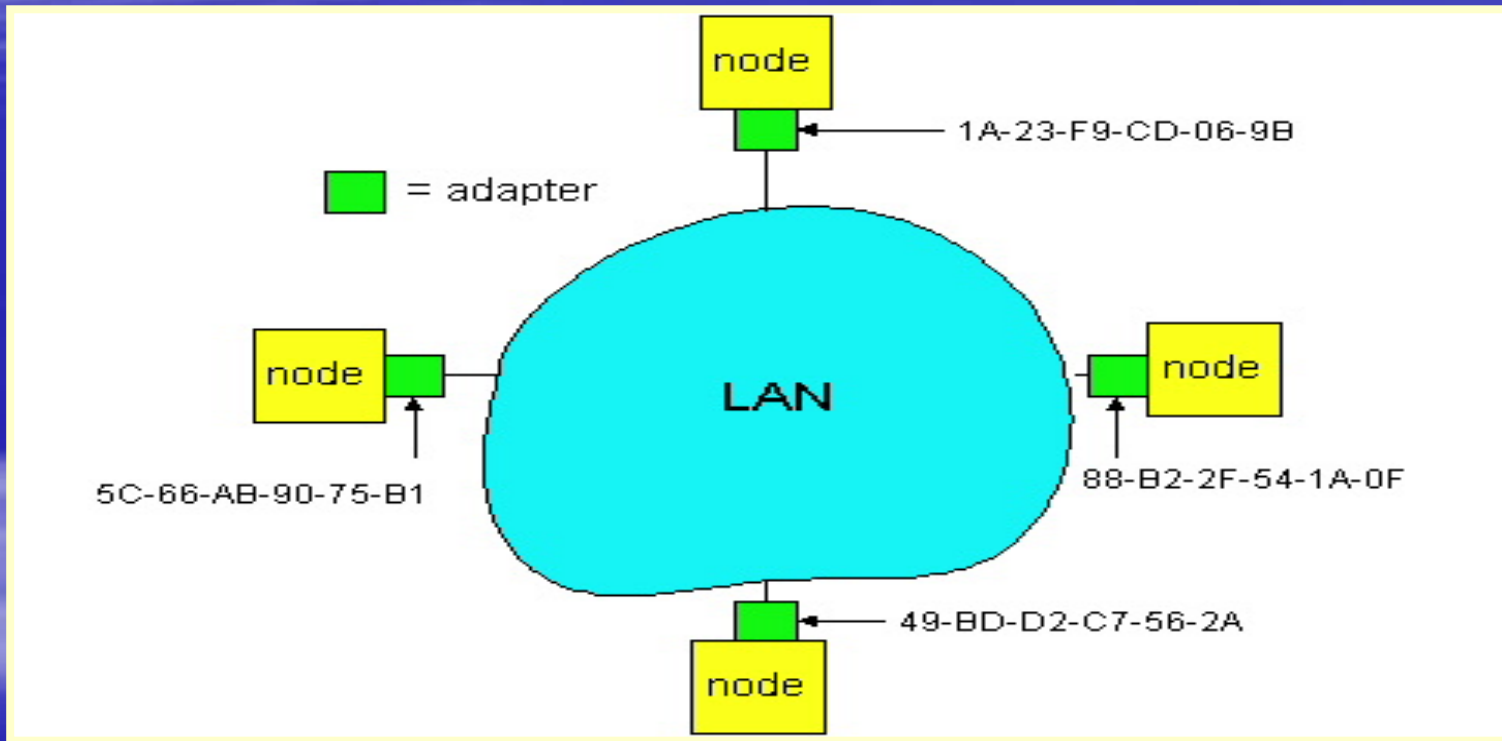
- Do **Binding**:

--Association between a IP address and a hardware address (one for every device) is called a **binding**.

# ARP Techniques

Table Lookup

Dynamic Binding :ARP



Pictorial View of ARP

# ARP techniques (cont.)

- **1: Table Lookup:**

Searching or indexing to get MAC addresses

- Similar to lookup in /etc/hosts for names
- Problem: change Ethernet card => change table

IP Address	MAC Address
192.168.0.1	0A:4B:00:00:07:08
192.168.0.2	0B:4B:00:00:07:00
192.168.0.3	0A:5B:00:01:01:03

# ARP techniques (Continued)

- **2. Dynamic Binding: ARP**
  - The host *broadcasts* a request:  
“What is the MAC address of 127.123.115.08?”
  - The host whose IP address is 127.123.115.08 replies back: “The MAC address for 127.123.115.08 is 8A-5F-3C-23-45-56<sub>16</sub>”
  - ARP responses cached; LRU + Entry Timeout
- These two methods are allowed in TCP/IP networks.

# Source code



**arp\_send(): arp.c**

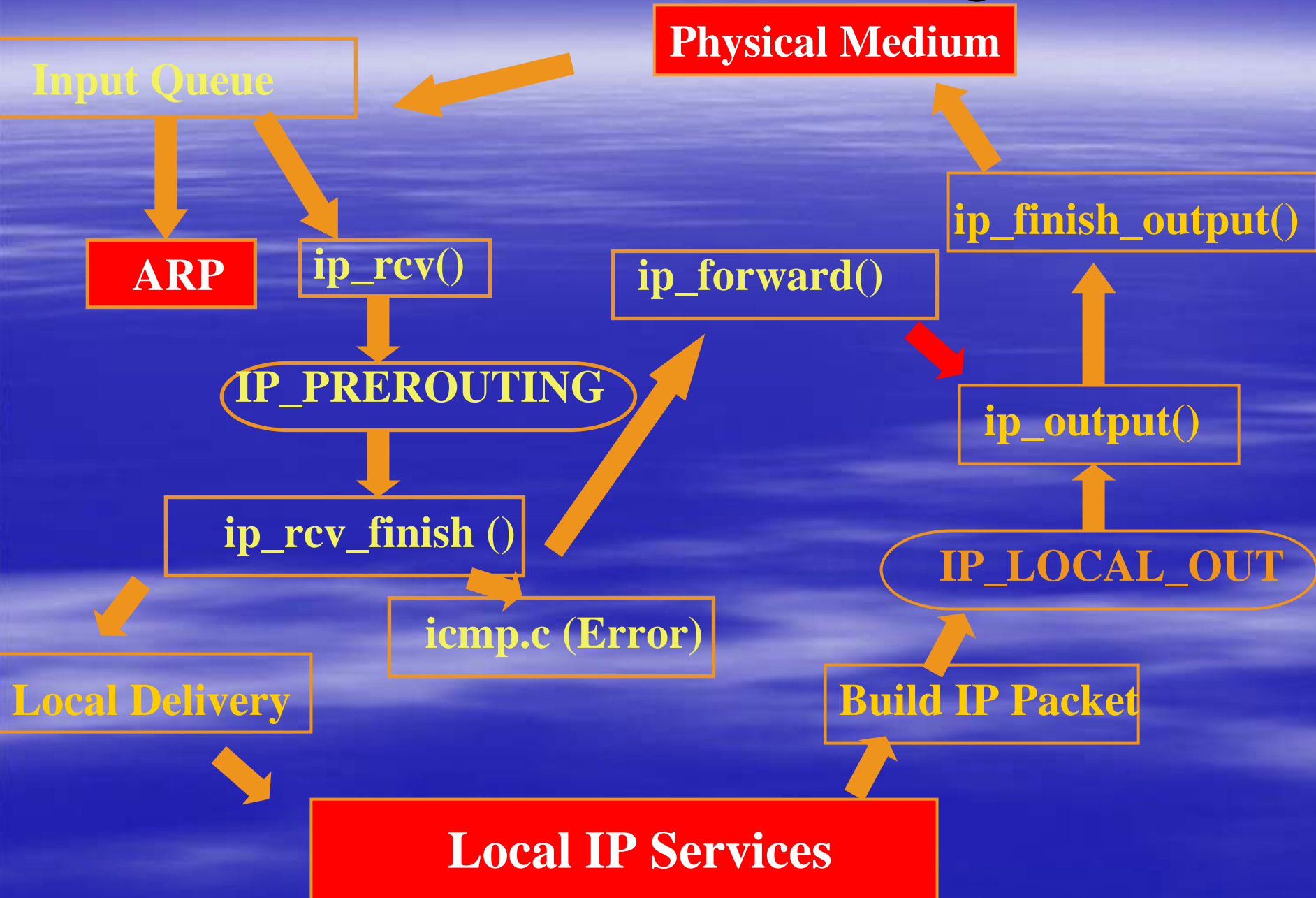
**-- Check Device Supports ARP**

**-- Allocate Buffer**

**-- fills Header/ARP Information**

**-- calls dev\_queue\_xmit() to send**

# How Does all of this Fit together?



# Where are we?

- Internetworking: Past and Present Overview
- TCP/IP and OSI Model
- IP addressing
- Sockets Overview
- Internetworking: Past and Present Overview
- IP solutions:
  - Routing
  - Packet Forwarding
  - Routing Vs Forwarding
  - Address resolution Protocol (ARP)
- **Wireless Networking: Brief Overview**

# Wireless Networks

-- The use of infra-red or radio frequency signals to share information and resources between devices

## Wireless State of Art

**IEEE 802.11b**

(wi-fi)

- 2.4GHz
- 11mbps

**IEEE 802.11g**

- 2.4GHz
- 54mbps
- Compatible with 802.11b

**IEEE 802.11a**

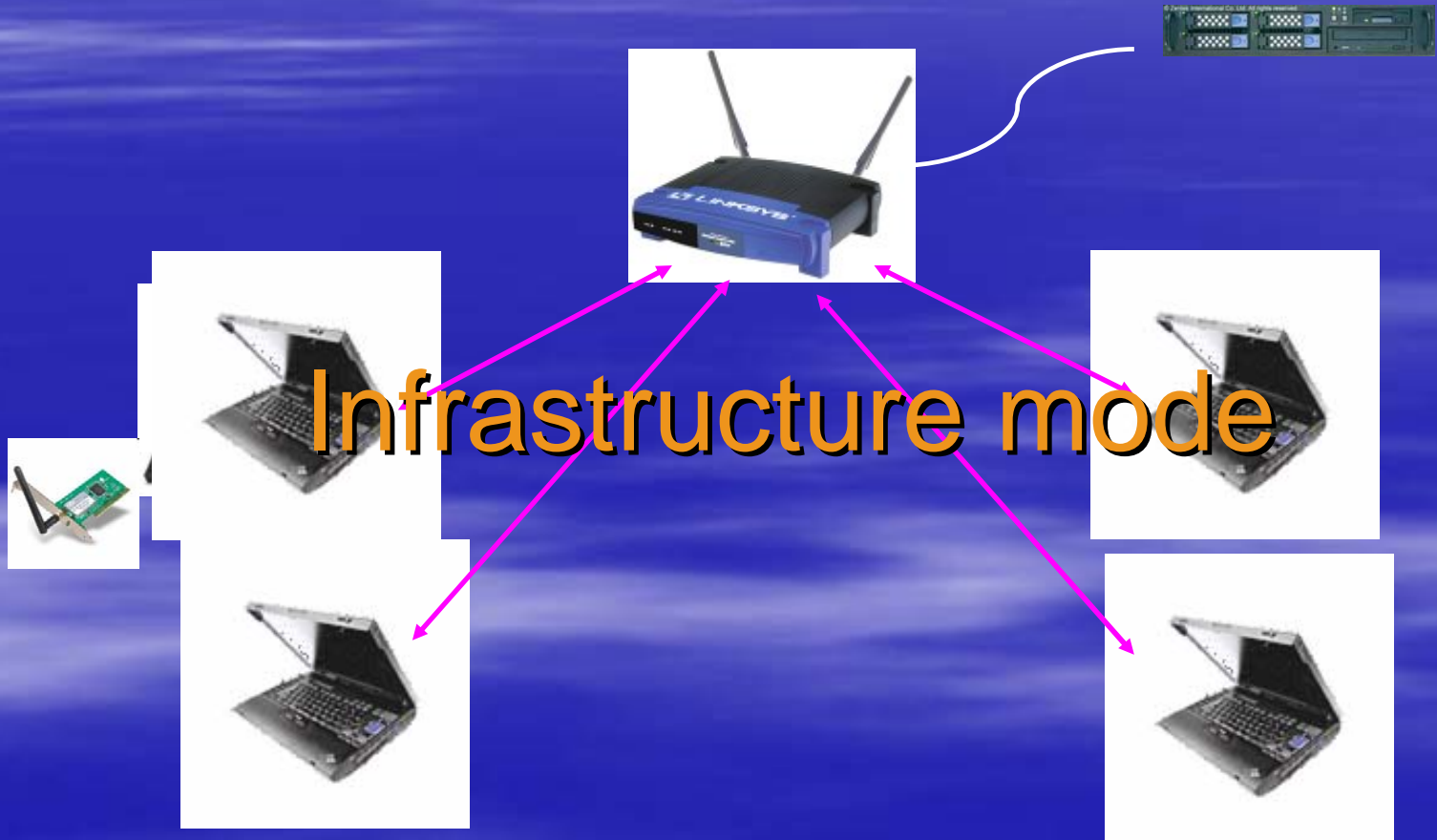
- 5GHz
- 54mbps

**IEEE 802.16a**

**WiMaX**

- 30 Mbps
- 30 Miles

# Types of WLAN (Wireless LAN)



*Each NODE is HOST and ROUTER*



**Ad-hoc Mode**

**We Set up a FIVE NODE Multi Hop Ad-Hoc Network**

**Linux 2.4.20  
HOST, ROUTER, NAT  
DHCP Server.**

**Wireless Card is configured using  
-- HostAP driver and pcmcia-cs package**

# What is 'Software Router'

--Each Node in our Network is:

-- **A Router in a Computer (Our Case : Linux 2.4.20 and 2.4.26 Machines )**

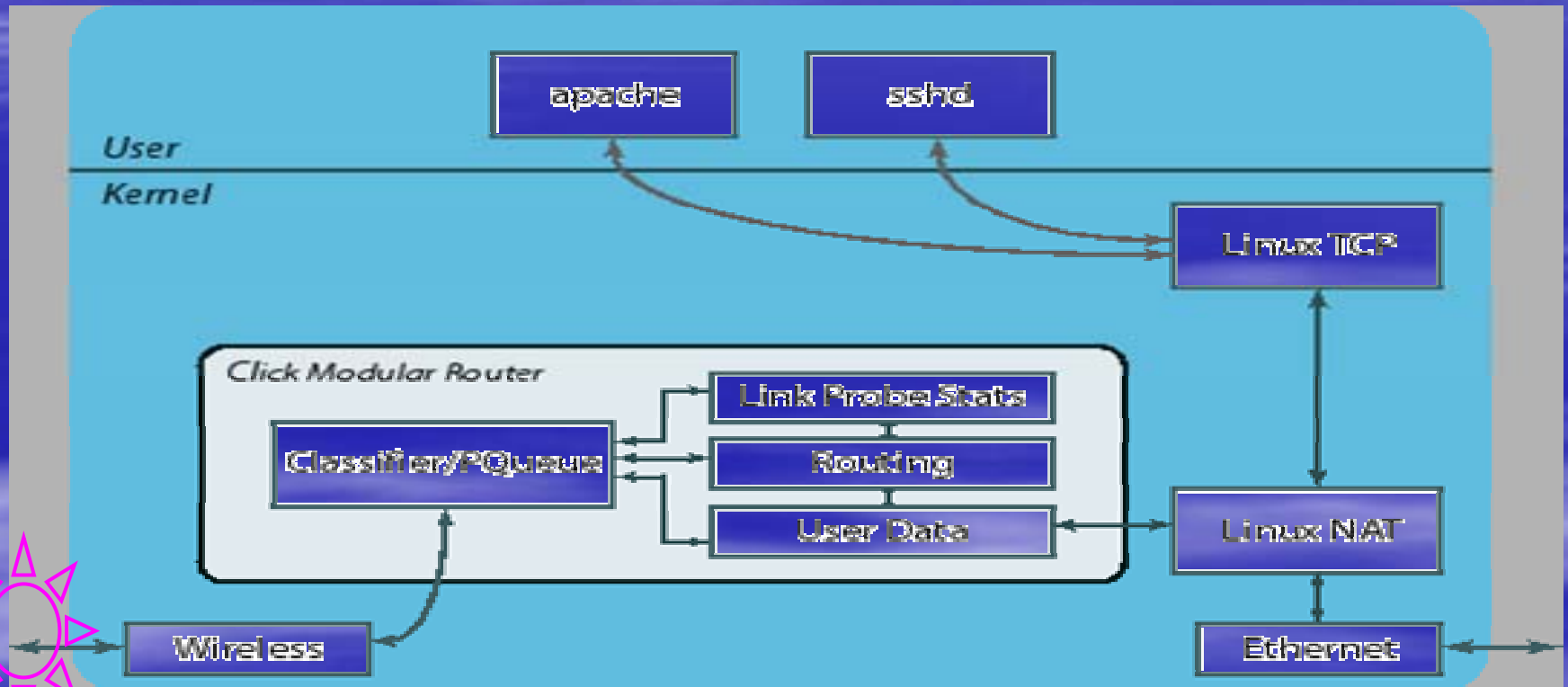
-- **Program that receives, processes and forwards packets to the next node**

-- **We are using *CLICK MODULAR ROUTER* (developed at MIT)**

-- **Features of Click:**

- **High level *programming interface* to the developer**
- **Hides *Kernel Complexities***
- **Easily installed and configured**
- **Provide *flexibility* to writes own *Elements* and eventually **Combines these elements to write any *Ad Hoc Routing Protocol*****

# Pictorial View of Each Click Router Node



5.0.0.0 `srcr` Interface

6.0.0.0 `wlan0` Interface

Private LAN attached to HOST

`Eth0` Interface: 192.168.0.1

153.90.196.95

# Click Config File Example

*FromDevice(eth0) -> Print -> ToDevice(wlan0);*

- 1. FromDevice(eth0): Get a packet from eth0 interface*
- 2. Print: Print the packet ( can see using 'dmesg')*
- 3. ToDevice(wlan0): Send the packet to eth1 interface*

=====

Start CLICK by loading Click.config file to /click/config

Command:

**Demonstration**

dawra@roofnet\$ insmod click.o (Kernel Side Module made for Click)

dawra@roofnet\$ cp click.config /click/config

Check out:

<http://www.pdos.lcs.mit.edu/click>

# Summary

- Internetworking: Past and Present Overview
- TCP/IP and OSI Model
- Sockets Overview
- Internetworking: Past and Present Overview
- IP solutions:
  - Routing
  - Packet Forwarding
  - Routing Vs Forwarding
  - Address resolution Protocol (ARP)
- Wireless Networking : An Overview

# What are the Resources?

- MIT Click Modular Router : <http://www.pdos.lcs.mit.edu/click>
- Roofnet MIT testbed: <http://www.pdos.lcs.mit.edu/roofnet/design>
- Computer Networking: Andrew S. Tanenbaum
- Internet Protocols Coursework : RPI institute
- Linux Source Code : <http://lxr.linux.no/source/>
- Linux Networking Documentation : <http://www.kernelnewbies.org/>
- Linux Documentation Project : <http://www.tldp.org/>
- Linux Kernel : [www.kernel.org](http://www.kernel.org)