CMPSCI 577 Operating Systems Design and Implementation **Spring 2020**

Lecture 19: 04/07

Lecturer: ***Prashant Shenoy*** *Scribe:* ***Shashwat***

19.1 MINIX Networking

This lecture discusses about MINIX and UNIX networking protocols

19.2 Communication Protocols

Protocol: Set of rules of communication agreed to by all parties

Protocol Stack: Each layer provides service to layer N+1 by using own layer N procedures and provides an interface to N-1 layer.

example: ISO

Router: A message is sent in terms of one or more network packets which goes through intermediate nodes called routers, which checks for the source and destination and decides which next node should the packet be forwarded to

19.3 TCP/IP Protocol

TCP is a reliable protocol in which the packets are received in the order they are sent. It provides end to end communication. It is responsible for the detecting losses and retransmission and also responsible for rate control of packets.

User Application Process: ex. browser process using HTTP protocol wherein Web browser and the client are the two application level processes communicating over HTTP protocol

UDP An unreliable and much simpler protocol which does not provide any guarantee for delivery i.e does not provide guarantee of no loss. It implements transport layer function.

IP Protocol: It does hop by hop communication. It adds headers such as source and destination which is the IP address.

Physical layer: Wired means of sending the message like LAN, WAN

Each layer implements a protocol which are agnostic of the protocols that are underneath

19.4 Socket Communication

Socket: Its provides an abstraction similar to file.

Ports: These are addresses on the machines that represent end points. Port number is the end point address for communication.

Send is the equivalent for write and **receive** is the equivalent for read.

Berkley Socket Primitives: The primitives include Socket, Bind, Listen, Accept, Connect, Send, Receive, Close

19.5 Linux Network Architecture

File Access: If writing to a file, then that write request arrives at VFS layer which can handle both network and ext file system.

Socket Access:It has more than 1 protocol families including INET and UNIX. Protocols include UDP, TCP and IP. Network interface card consists of ethernet and 802.11 which is the WIFI family.

19.6 Sockets in Linux Kernel

Contains sys calls like socket, connect and accept. These all implement the POSIX interface which is somewhat independent of the protocol itself.

Handler table tracks all the open sockets.

19.7 Protocol Families

This layer is analogous to the VFS layer in the file system. The two that are implemented are **INET** and **UNIX** families.

Each family can have multiple protocols in itself.

INET has an implementation of **TCP** and **UDP**

Dummy Functions: Functions which are unused in a protocol

Shared: Functions which are same across protocols

19.8 Fragmentation and routing

If message is larger than the max size then this message is partitioned into smaller network packets through the process known as **fragementation**.

Fragmentation is performed inside ip_fragment

IP layer is responsible for reassembly of the message

Routing table defines the next route to send the network packets through

Route cache:Allows to look into the recently cached entries and find a match

19.9 Data Link Layer

Data Link layer is **Ethernet** in our case. **Ethernet driver** has a **circular buffer** having queued packets for transmission. Buffer has the **head** and **tail**. Driver takes the packets from the head of the buffer and starts transmitting. The IP layer keeps appending to the tail of the buffer. Buffer mechanism allows to handle speed mismatches. Circular buffers have a match size and so does this Ethernet driver.

19.10 *NIX Networking Commands

Each **Ethernet card** has a unique **MAC address** which is used for communication over a wired/wireless LAN. Ethernet addresses are hardcoded by the manufacture. **IP addresses** are assigned when the machine boots up. Private addresses can be reused over a LAN for communication.

Ping command allows to send packets to other machine and checks if the other machine is up and also measures the rountrip time

ifconfig is used to assign IP net addresses to the ethernet card.

netstat is used more in the **LINUX** context to set or view routing table

netstat -rn:- is used to view the routing table and **route** is used to modify the routing table

19.11 MINIX INET

INET essentially implements the internet protocol which is the TCP/IP family. It collectively implements Ethernet layer, IP layer and the TCP/IP layer. Ethernet card is a I/O device wherein the driver is in the "drivers" directory.

19.12 Data Link Layer in MINIX

Hardware involves : **Ethernet** and **Modem**. Each of them have their own device drivers. Each has drivers for each vendors interface.

A particular machine can have a both **wired** and **wireless** interface and both can be active. The way this is handled is by using the device **major** and **minor** number. The **major** number identifies class and **minor** number identifies which specific card it is if there are more than 1 card in the machine.

19.13 INET server

inet.c: is the main function for **INET** server and handles various messages types from VFS and DL_ETH

buf.c: Buffering code to allocate data for sending and receiving network packets

mnx_eth.c: code for sending/receiving ethernet frames to/from ethernet driver

inet_config.c: Configure networking devices

mq.c: Message queue structure

sr.c: Code to interface with file system

generic/udp.c: Code for UDP protocol

generic/tcp.c: Code for TCP protocol