COMP20200 Unix Programming

Lecture 14

Ravi Reddy Manumachu (ravi.manumachu@ucd.ie)

School of Computer Science, University College Dublin, Ireland

27/03/2023

Lecture overview

- Networking basics.
- Interprocess communication (IPC) using sockets.

Networking basics

- Networking is the practice of connecting computers and sending data between them.
- To understand networking, you must know
  - How does a sender know **where** to send its data?
  - When the receiver receives the data, how does it know **what** it has received?

Simple network

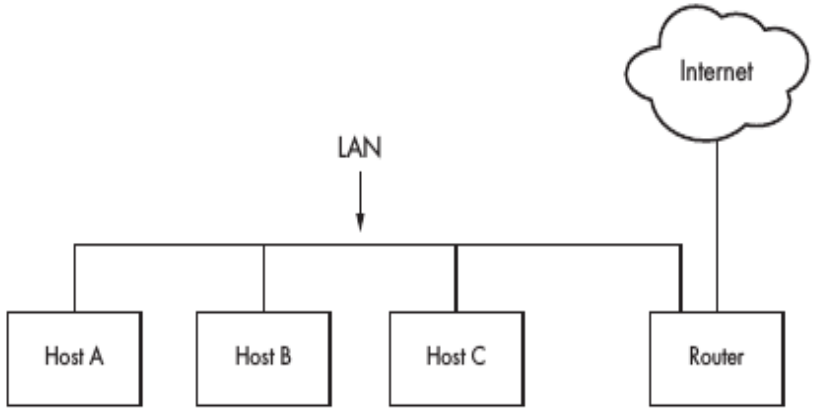

Figure: A typical local area network with a router that provides internet access. (Source: *How Linux Works : What Every Superuser Should Know by Brian Ward*)

- Each computer connected to the network is called a *host*.
- The hosts are connected to a *router*, which is a host that can transfer data from one network to another.
- Hosts A, B, and C and the router complete the local area network (LAN).

Simple network - cont'd

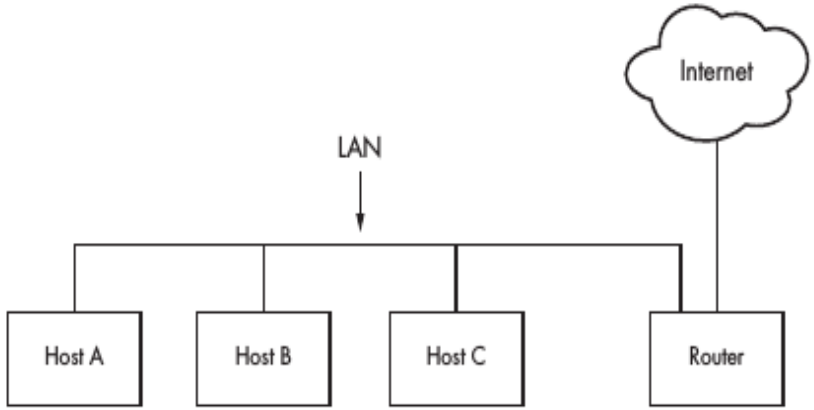

Figure: A typical local area network with a router that provides internet access. (Source: *How Linux Works : What Every Superuser Should Know by Brian Ward*)

- The LAN connections can be wired or wireless.
- The hosts have access to the Internet since the router is also connected to the Internet.

Where to send the data?

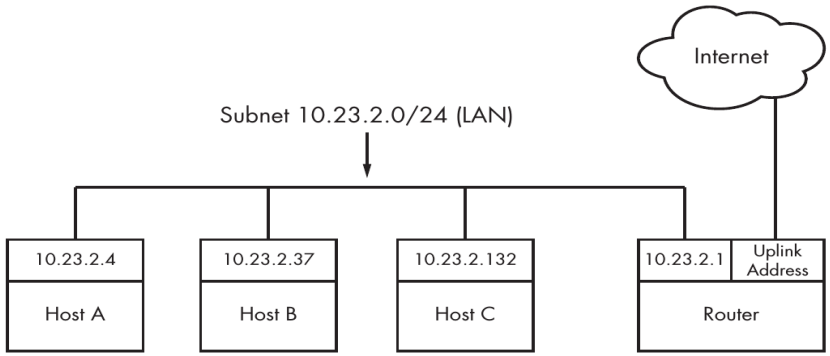

Figure 9-2: Network with IP addresses

Figure: A network with IP addresses. (Source: *How Linux Works : What Every Superuser Should Know by Brian Ward*)

- To communicate with other host, your machine must know that other host's IP address.
- Each host has at least one numeric IP address in the form of a.b.c.d (for example, 10.23.2.4).

Where to send the data?

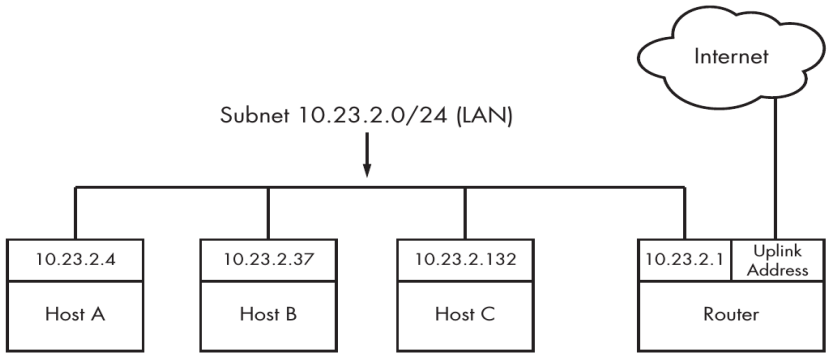

Figure 9-2: Network with IP addresses

Figure: A network with IP addresses. (Source: *How Linux Works : What Every Superuser Should Know by Brian Ward*)

- The Internet is decentralized and contains smaller networks called *subnets*.
- The LAN shown here is a subnet.
- The hosts and the router have IP addresses allowing them to be uniquely identified within a subnet.

Where to send the data?

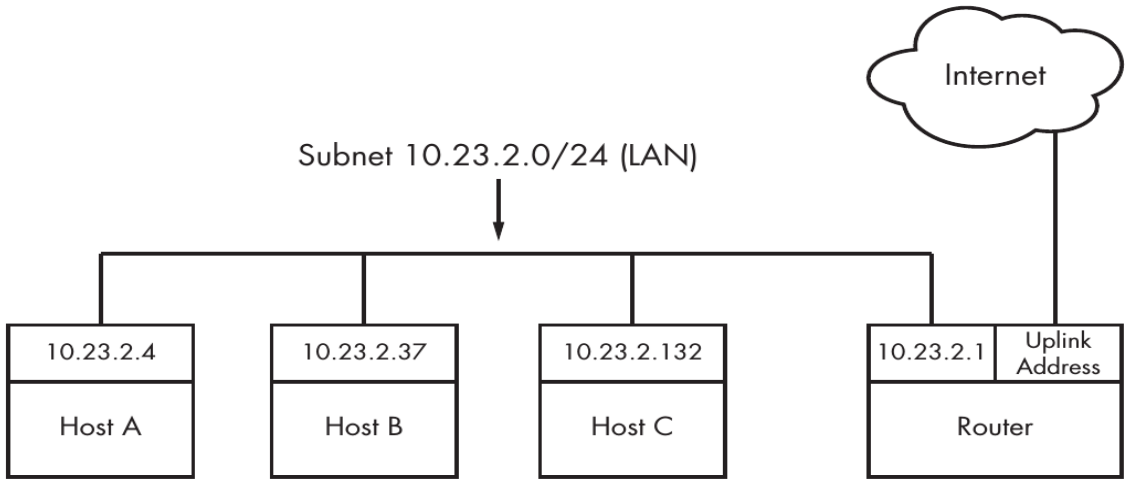

Figure 9-2: Network with IP addresses

Figure: A network with IP addresses. (Source: *How Linux Works : What Every Superuser Should Know by Brian Ward*)

- An IP address consists of 4 bytes, abcd.The dotted-quad address, a.d.c.d, is a human-readable form.
- For Host A to communicate with Host B, it must know the IP address of Host B (10.23.2.37).

Your computer's IP address

```
$ /sbin/ifconfig
wlp4s0    Link encap:Ethernet  HWaddr f4:96:34:e7:a9:64
          inet addr:192.168.0.13  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: 2a02:8084:6123:580:7ea3:fa88:7afe:4241/64 Scope:Global
...
```

- The host has the IPv4 address, 192.168.0.13.
  - Approximately 4.3 billion addresses.
  - We have already exhausted this number.
- The host has the IPv6 address, 2a02:8084:6123:580:7ea3:fa88:7afe:4241.
- IPv6 address is 128-bit and allows for more unique IP addresses.

ifconfig command

Sections of the man pages

Linux	Solaris	HP-UX	AIX	Contents
1	1	1	1	User-level commands and applications
2	2	2	2	System calls and kernel error codes
3	3	3	3	Library calls
4	7	7	4	Device drivers and network protocols
5	4	4	5	Standard file formats
6	6	–	6	Games and demonstrations
7	5	5	7	Miscellaneous files and documents
8	1m	1m	8	System administration commands
9	9	–	–	Obscure kernel specs and interfaces
–	–	9	–	HP-UX general information

Figure: Sections for man pages in various UNIX platforms (Source: UNIX and Linux System Administration Handbook.)

```
shell$ man ifconfig
IFCONFIG(8)                Linux Programmer's Manual                IFCONFIG(8)

NAME
       ifconfig - configure a network interface

SYNOPSIS
...
```

Your computer's IP address

```
shell$ ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP>...
   ...
2: enp3s0:
   ...
3: wlp0s20f3:
    inet 192.168.0.19 ...
    inet6 2a02:8084:611e:2a00:e849:717:70a1:291b...
```

- IPV4 address: 192.168.0.19
- IPV6 address: 2a02:8084:611e:2a00:e849:717:70a1:291b

Your computer's hostname

```
$ hostname
hclserver01.ucd.ie
```

- A hostname is a symbolic identifier for a system that is connected to a network.
- Domain Name System (DNS) is a distributed database that maps hostnames to IP addresses and vice versa.

How processes communicate with each other?

Sockets: Overview

- Sockets allow data to be exchanged between applications either on the same host or different hosts connected by a network.
- In a typical client-server scenario, client and server applications communicate using a socket.
  - Each application creates a socket.
  - The server binds its socket to a well-known address so that clients can locate it.

Sockets: History

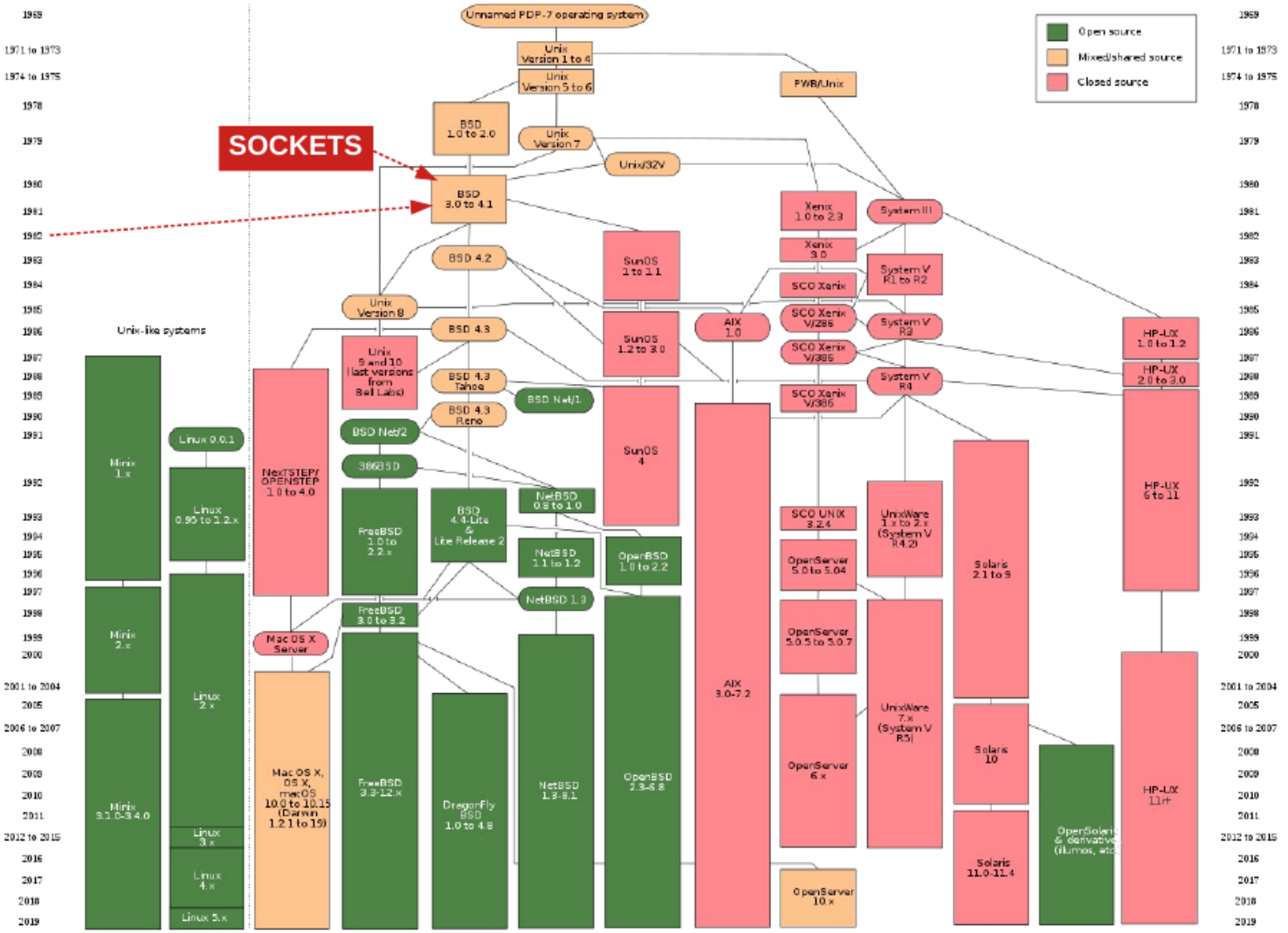

Figure: Sockets introduced in 1982 in 4.1aBSD OS.

Sockets: Creation

```
#include <sys/socket.h>

int fd = socket(domain, type, protocol);
```

- A socket is created using the *socket()* call.
- The call returns a file descriptor.
- Since in Unix "everything is a file", so are sockets.

socket() call: System call or library function?

Sections of the man pages

Linux	Solaris	HP-UX	AIX	Contents
1	1	1	1	User-level commands and applications
2	2	2	2	System calls and kernel error codes
3	3	3	3	Library calls
4	7	7	4	Device drivers and network protocols
5	4	4	5	Standard file formats
6	6	–	6	Games and demonstrations
7	5	5	7	Miscellaneous files and documents
8	1m	1m	8	System administration commands
9	9	–	–	Obscure kernel specs and interfaces
–	–	9	–	HP-UX general information

Figure: Sections for man pages in various UNIX platforms (Source: UNIX and Linux System Administration Handbook.)

```
shell$ man socket
SOCKET(2)      Linux Programmer's Manual        SOCKET(2)

NAME
       socket - create an endpoint for communication
...
```

socket() first argument: Communication domains

int fd = socket(**domain**, type, protocol);

A communication domain specifies

- the method of identifying a socket;
- the format of a socket address;
- the range of communication (between applications on the same computer or different computers connected via a network).

Sockets: Communication domains

Table: Socket domains

Domain	Communication performed	Communication between applications	Addr. format
AF_UNIX	within kernel	on same host	pathname
AF_INET	via IPv4	on hosts connected via an IPv4 network	32-bit IPv4 address + 16-bit port number
AF_INET6	via IPv6	on hosts connected via an IPv6 network	128-bit IPv6 address + 16-bit port number

- The UNIX (**AF_UNIX**) domain allows communication between applications on the same host.
- The IPv4 (**AF_INET**) domain allows communication between applications running on hosts connected via an IPv4 network.
- The IPv6 (**AF_INET6**) domain allows communication between applications running on hosts connected via an IPv6 network.

socket() second argument: socket types

int fd = socket(domain, **type**, protocol);

- There are at least two types of sockets:
  - stream
  - datagram
- Both socket types are available in the UNIX (AF_UNIX) and Internet socket domains (AF_INET, AF_INET6).

Sockets: Types

Table: Socket types

Property	**Stream socket**	**Datagram socket**
Reliable delivery?	**Y**	**N**
Message boundaries preserved?	N	Y
Connection-oriented?	Y	N

- Stream sockets (SOCK_STREAM) provide a reliable, bidirectional, byte-stream communication channel.
- **Reliable** means that
  - We are guaranteed that the transmitted data will arrive intact at the receiving application.
  - Or we will receive notification of a probable failure in transmission.

Stream socket (reliability)

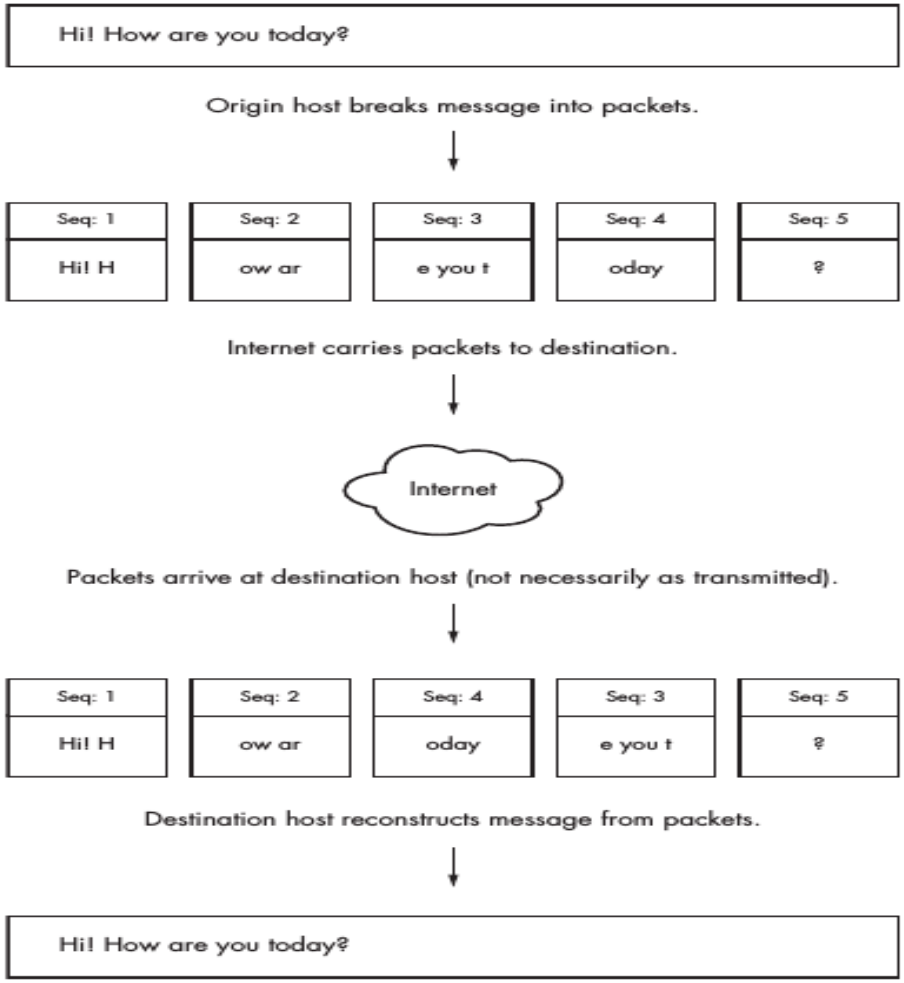

Figure: Sending a message reliably. (Source: *How Linux Works : What Every Superuser Should Know by Brian Ward*)

Sockets: SOCK_STREAM

Table: Socket types

Property	**Stream socket**	**Datagram socket**
Reliable delivery?	Y	N
Message boundaries preserved?	**N**	**Y**
Connection-oriented?	Y	N

- **Bidirectional** means that data may be transmitted in either direction between the two sockets.
- **byte-stream** means that there is no concept of message boundaries.
  - A reading process can read blocks of data of any size, regardless of the size of blocks written by the writing process;
  - Bytes are read from a stream in exactly the order they were written;
  - It is not possible to randomly access the data.

Sockets: SOCK_STREAM

Table: Socket types

Property	**Stream socket**	**Datagram socket**
Reliable delivery?	Y	N
Message boundaries preserved?	N	Y
Connection-oriented?	**Y**	**N**

- Stream sockets operate in connection pairs. Hence, they are described as *connected-oriented*;
- Before communication can commence, a communication channel is established between the two endpoints.

Sockets: SOCK_DGRAM

Table: Socket types

Property	**Stream socket**	**Datagram socket**
Reliable delivery?	Y	**N**
Message boundaries preserved?	N	**Y**
Connection-oriented?	Y	**N**

- With datagram sockets (SOCK_DGRAM) (in the internet domain), data transmission is not reliable, messages may arrive out of order, be duplicated, or not arrive at all.
- Message boundaries are preserved.
- Datagram sockets are called *connectionless* sockets.
- NOTE: datagram sockets in UNIX domain (**AF_UNIX**) are reliable.

Sockets: SOCK_STREAM and SOCK_DGRAM

- Communication via a stream socket is analogous to a telephone call;
- Before communication can take place, one application must connect its socket to another's application socket;
- Communication via a datagram socket is analogous to a postal system;
- Like a postal system, when multiple datagrams (letters) are sent from one address to another, they may not arrive at all or arrive in the order they were sent;
- Unlike a postal system, same datagram could arrive more than once.

socket() third argument: protocol argument

int fd = socket(domain, type, **protocol**);

- *protocol* is always set to 0 for datagram and stream sockets.
- *protocol* is specified as IPPROTO_RAW for raw sockets (SOCK_RAW).
- Raw socket type (SOCK_RAW) allows an application to communicate directly with the IP layer.
- Programming raw sockets requires in-depth understanding of TCP/IP network layers and is out-of-scope of this module.

Sockets: UDP and TCP

- In the Internet domain (AF_INET, AF_INET6), datagram sockets employ the User Datagram Protocol (UDP);
- Stream sockets (usually) employ the Transmission Control Protocol (TCP);
- From now on, we will use the term
  - UDP socket to refer to an Internet domain datagram socket;
  - TCP socket to refer to an Internet domain stream socket.

Lookahead: Lecture 15

In the next lecture,

- Overview of the other socket system calls;
- Develop an iterative server using TCP sockets.

Q & A