Networking Basics 2: Routing & Ports

1. TCP/IP Introduction
2. **Routing and Ports**
   - **routing**
   - **ports**
   - **NAT**
3. TCP/IP Protocols
4. Ethernet
5. DNS

Routing

Routing refers to the process of communicating/forwarding a packet.

A machine must know two pieces of information to route packets:

1. its own IP address
2. the subnet mask for its local (physical) network

Each outgoing packet address is then classified as either:

- *remote address* – destination IP address differs from machine's IP address in the *network part*
- *local address* – destination IP address differs from machine's IP address *only in the host part*

How a packet is sent depends on the packet's address type.

Routing (contd.)

Local addresses:

- packet is sent directly to the (local) destination machine
- get MAC address of machine corresponding to destination IP address using ARP
- send packet via Ethernet (encapsulated in Ethernet frame)

Remote addresses:

- packet is sent to the **gateway** (router) machine
- get MAC address of gateway device via ARP
- send packet via Ethernet to that device
- leave original destination IP address in IP packet header
- gateway machine forwards based on desination IP address

Routing (contd.)

Routing example:

- machine IP: 192.168.1.147
- netmask: 255.255.255.192
- 147_{10} is 10010011_2 so network 2 bits are 10
- thus *local addresses* have form: 192.168.1.(10xxxxxx)
- so range is 192.168.1.128 to 192.168.1.191
- however two of these addresses cannot be used for hosts due to special meanings:
  - 192.168.1.128 means "this machine"
  - 192.168.1.191 means broadcast
- all other addresses are *remote addresses*

Ports

While an IP address identifies a machine/host, additional info will be required to identify a particular *process* (e.g., server or client) that a remote machine wants to communicate with.

This is done by associating a *16 bit integer* called a **port** with the process.

16 bits means ports range from 0 through 65535.

An IP address and port number should together uniquely identify a process to communicate with.

A process associates itself with a port by creating a **socket** and giving it an *address*.

Sockets are the *communication endpoints* for network IPC.

Ports (contd.)

There are 3 classes of ports:

- **well-known**
  - 0 to 1023
  - assigned by Internet authority to particular type of server
  - can be opened only by root on UNIX/Linux systems
- **registered**
  - 1024 to 49151
  - more server ports, no root restrictions
- **dynamic/private**
  - 49152 to 65535
  - used by clients when open socket/connection

(**Ephemeral**: older term for ports 1024+.)

Ports (contd.)

See `/etc/services` for list of well-known ports.

This is the file that is used to map from numeric ports to service names in Linux.

Note that each transport protocol has its own *"port space,"* so TCP port 53 is different from UDP port 53.

Most servers use only TCP or UDP, but a few are capable of using both (e.g., DNS).

Most servers/services use only a single port, but some use multiple (e.g., FTP, Bittorrent, etc.).

Ports (contd.)

Key well-known service ports:

- FTP: 20 & 21
- SSH: 22
- SMTP (email): 25
- DNS: 53
- HTTP (web): 80 and HTTPS (secure web): 443
- POP (email): 109, 110
- NETBIOS/SMB (MS file sharing+): 137, 138, 139
- IPP (printing): 631
- X11: 6000 (and up)

Network Address Translation (NAT)

Network Address Translation (**NAT**) is a technique for using a *single routable IP address* and router/gateway to handle multiple machines:

- LAN machines use non-routable (private) IP addresses
- NAT router has a single (routable) IP address
- outgoing packets:
  - router rewrites packet's local (non-routable) IP address with its own routable IP address
  - router chooses new port and rewrites IP packet's source port to it
  - router stores new port to local IP address-port mapping

continues...

Network Address Translation (contd.)

...continued

- incoming packets:
  - router uses packet's destination port to lookup local machine IP address and true destination port in its mappings
  - router rewrites packet's destination IP address and port based on mapping
  - router fowards packet to correct local machine based on mapping