Lecture 9

Network interface.

Department of Computer Systems FIT, Czech Technical University in Prague

©Jan Trdlička, 2010

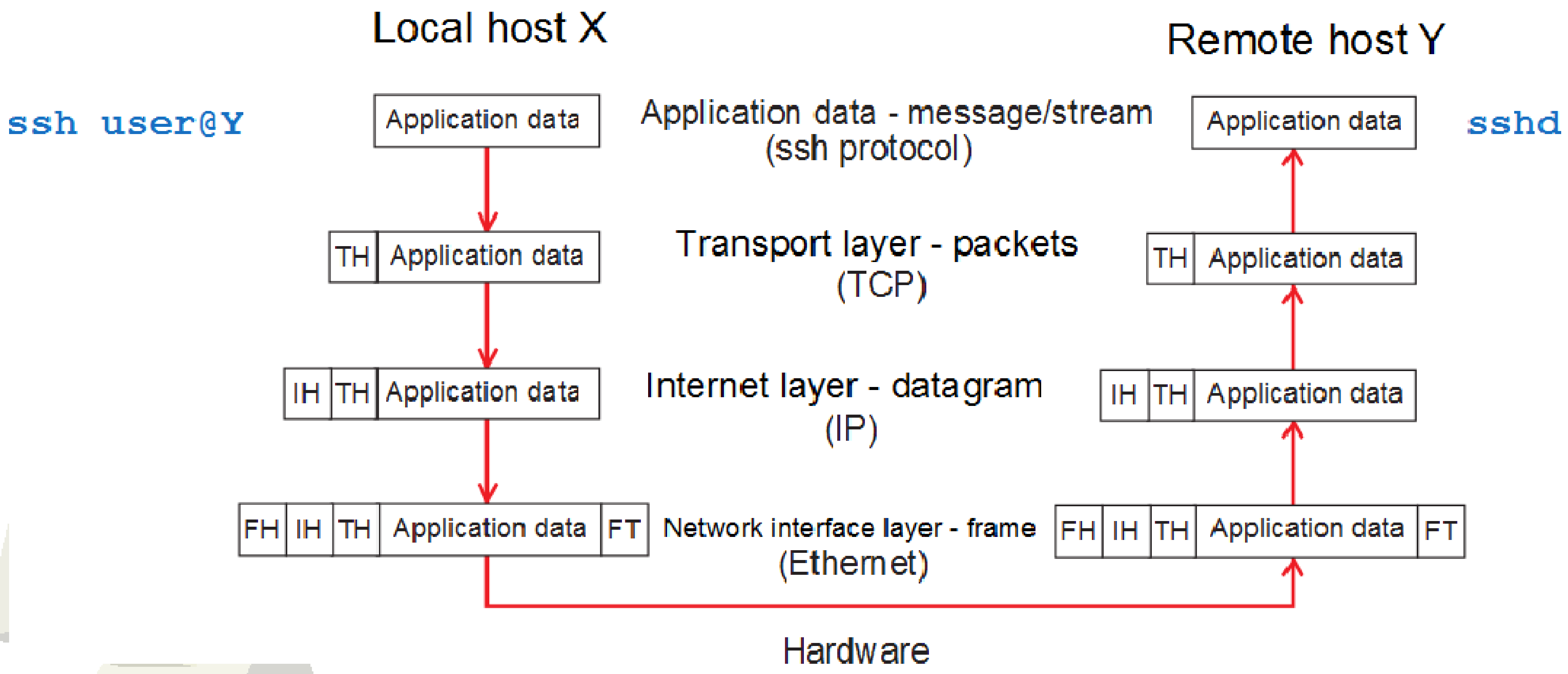
Network models

- **ISO/OSI model**
  - from theory to practice
  - 7 layers
  - not very popular
- **TCP/IP model**
  - form practice to theory
  - 5 layers
  - Model = structure
  - Layer = function
  - Protocol = rules
- **Protocols**
  - sets of rules
  - for TCP/IP model are defined by RFC (**R**equest **F**or **C**omments)

Network models

ISO/OSI Model	TCP/IP Model
Application layer	Application layer
Presentation layer	
Session layer	
Transport layer	Transport layer
Network layer	Internet layer
Data link layer	Network interface layer
Physical layer	Hardware layer

Encapsulation



Hardware

- **Topologies**
  - **bus** – Ethernet using coaxial cable
  - **star** – Ethernet using twisted pair
  - **ring** – Token ring

- **Network communication**
  - **circuit switch data**
    - connection is made across the network between the two different points
    - all the network resources are available for the exclusive use of these two parties whether they are sending data or not.
    - e.g. ATM
  - **packet switch data**
    - improve the efficiency of the transfer of bursts of data, by sharing the one communications channel with other similar users
    - e.g. Ethernet, Token Ring, FDDI

Hardware

- Types of networks
  - LAN (Local Area Network)
  - MAN (Metropolitan Area Network)
  - WAN (Wide Area Network)

LAN system components

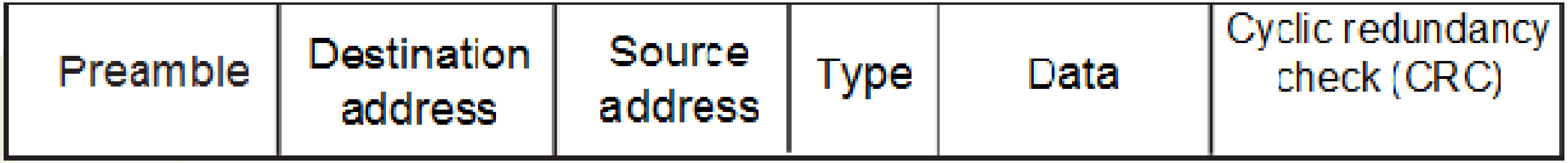
- **Repeater, hub**
  - operates at the hardware layer
  - simply retransmits incoming electrical signals
- **Switch, bridge**
  - operates at the network interface layer (using MAC addresses)
  - connects two separate networks to form a single large continuous LAN
  - The bridge only divides the network up into two segments, each with its own collision domain and each retaining its full bandwidth
- **Router**
  - operates at the internet layer (using IP addresses)

Network interface layer

- providing communication between nodes on the same network using **frames**

- using **MAC (Media Access Control)** addresses

- Structure of a frame:



Preamble	Destination address	Source address	Type	Data	Cyclic redundancy check (CRC)

Network interface layer

- **Example of protocols**
  - IEEE 802.3 Ethernet
  - IEEE 802.11 – WiFi

    …

  - SLIP (Serial Line IP)
  - PPP (Point to Point Protocol)

  - ARP (Address Resolution Protocol)
  - RARP (Reverse Address Resolution Protocol)

Network interface layer

- ## How to display MAC address

```
$ ifconfig –a
lo0: flags=2001000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4,VIRTUAL>
        mtu 8232 index 1  inet 127.0.0.1 netmask ff000000

bge0: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4>
        mtu 1500 index 2  inet 147.32.192.154 netmask fffff000
        broadcast 147.32.207.255  ether 00:04:76:A4:DF:9B

bge3: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4>
         mtu 1500 index 3  inet 172.16.16.17 netmask ffff0000
          broadcast 172.16.255.255  ether 00:04:76:A4:AF:02
```

Network interface layer

- **How to display statistics**

```
$ netstat –i
```

```
Name    Mtu   Net/Dest      Address       Ipkts   Ierrs Opkts   Oerrs Collis Queue
lo0     8232 loopback      localhost      1981597 0     1981597 0     0      0
e1000g0 1500 fray1         fray1          7308678 0     7711202 0     0      0
e1000g3 1500 fray1-e1000g3 fray1-e1000g3 7320842 0     5915331 0     0      0
```

Internet layer

- Primary functions:
  - Routing data between networks
  - Fragmenting and reassembly of data

- **Using logical addresses** (IPv4 and IPv6):
  - define **address of subnet**
  - define **address of node**

- **Example of protocols:**
  - **IP (Internet Protocol)**
  - **ICMP (Internet Control Message Protocol)**

Internet layer

- How to display IP address: `ifconfig -a`

- Test the reachability of a host on IP network `ping`

```
$ ping www.google.com
www.google.com is alive
$ ping -s www.google.com
PING www.google.com: 56 data bytes
64 bytes from nf-in-f103.google.com (64.233.183.103): icmp_seq=0. time=24.8 ms
64 bytes from nf-in-f103.google.com (64.233.183.103): icmp_seq=1. time=24.7 ms
^C
----www.google.com PING Statistics----
2 packets transmitted, 2 packets received, 0% packet loss
round-trip (ms)  min/avg/max/stddev = 24.7/24.7/24.8/0.045
```

Internet layer

- **Logical address – host name**
  - simple host name (e.g. fray1)
  - fully qualified domain name (FQDN) (e.g. fray1.fit.cvut.cz)

  - How to translate?
    - local database **`/etc/hosts`**
    - name services: DNS, NIS, NIS+, LDAP

  - file **`/etc/nsswitch.conf`** defines which database will be used

Internet layer

- **Routing**
  - process of selecting paths in a network along which to send network traffic
  - routing directs packet forwarding, the transit of logically addressed packets from their source toward their ultimate destination through intermediate nodes
  - routing is implemented by routing table in the Unix kernel
  - routing table is created
    - statically by command **`route`**
    - dynamically by protocols: RIP, OSPF, BGP,...

- **Netmask**
  - define splitting of IP address into subnet address and host

Internet layer

- **Listing of routing table:** `netstat –r` (Solaris) or `route` (Linux)

```
$ netstat -r
 Routing Table: IPv4
 Destination           Gateway           Flags  Ref     Use     Interface
 -------------------- -------------------- ----- ----- ---------- ---------
 default              gw129.fit.cvut.cz    UG        1       2082
 147.32.232.128       fray1                U         1        319 e1000g0
 172.16.0.0           fray1-e1000g3        U         1       1129 e1000g3
 localhost            localhost            UH       10      36269 lo0
```

- **Test of routing path:** `traceroute`

```
$ traceroute sunray1.felk.cvut.cz
traceroute: Warning: Multiple interfaces found; using 147.32.192.154 @ bge0
traceroute to sunray1.felk.cvut.cz (147.32.80.36), 30 hops max, 40 byte packets
1  147.32.192.1 (147.32.192.1)  0.946 ms  0.650 ms  0.696 ms
2  r1de-fel.net.cvut.cz (147.32.252.29)  0.557 ms  0.726 ms  0.711 ms
...
```

Transport layer

- Using ports to identify services running on the given host

- **Protocols**
  - **TCP (Transmission Control Protocol)**, RFC 793,…
    - provides reliable, ordered delivery of a stream of bytes from a program on one computer to another program on another computer
  - **UDP (User Datagram Protocol)**, RFC 768
    - provides not reliable delivery of a stream of bytes from a program on one computer to another program on another computer
- **Ports**
  - 2^{16} ports for TCP, 2^{16} ports for UDP
  - ports 0-1023 are privileged ports (only processes with EUID=0 can use them)
  - **well-known ports** **`/etc/services`** are assigned to widely-used types of network services j (IANA = Internet Assigned Numbers Authority)
  - **dynamically assigned ports** are registered by port mapper server (e.g. **`rpcbind`**, **`portmap`** process)

Application layer

- **Application startup**
  - manually
  - automatically during system startup
  - at request of service by process **`inetd`**

- **Program `inetd`**
  - started during system startup
  - listening on designated ports (e.g. **`/etc/inetd.conf`**)
  - when a TCP packet or UDP packet arrives with a particular destination port number, inetd launches the appropriate server program to handle the connection

Application layer

- Listing of active network connections: `netstat -a`