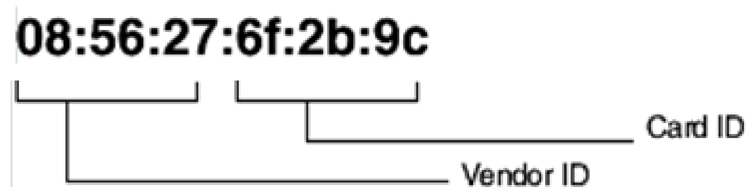
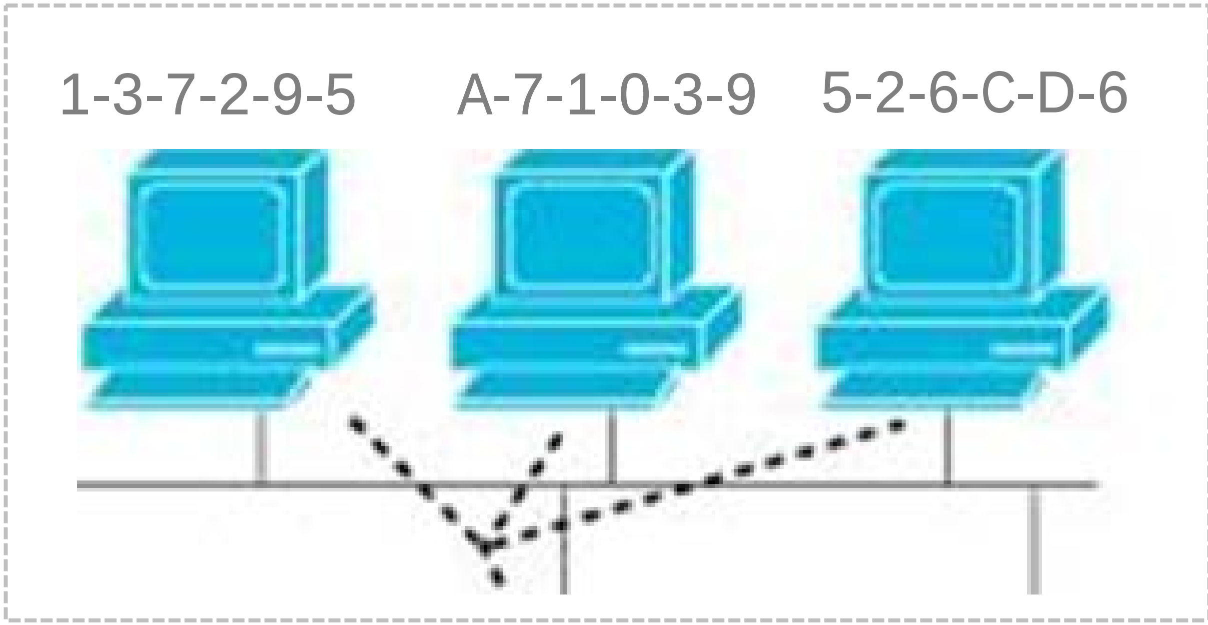
MAC Address - 6-byte sequence assigned to NIC by the manufacturer
('name given at birth' that never changes)



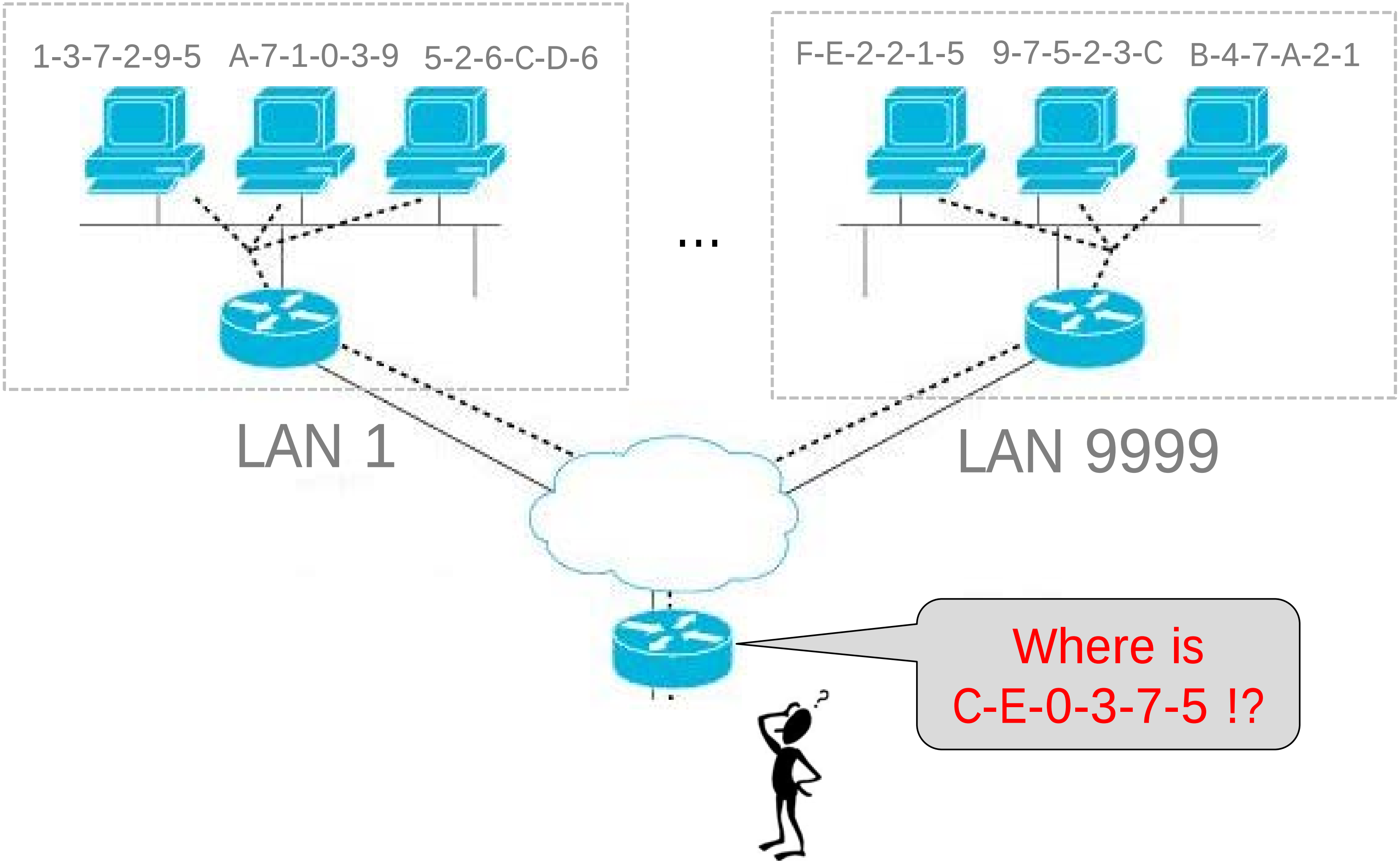


LAN 1

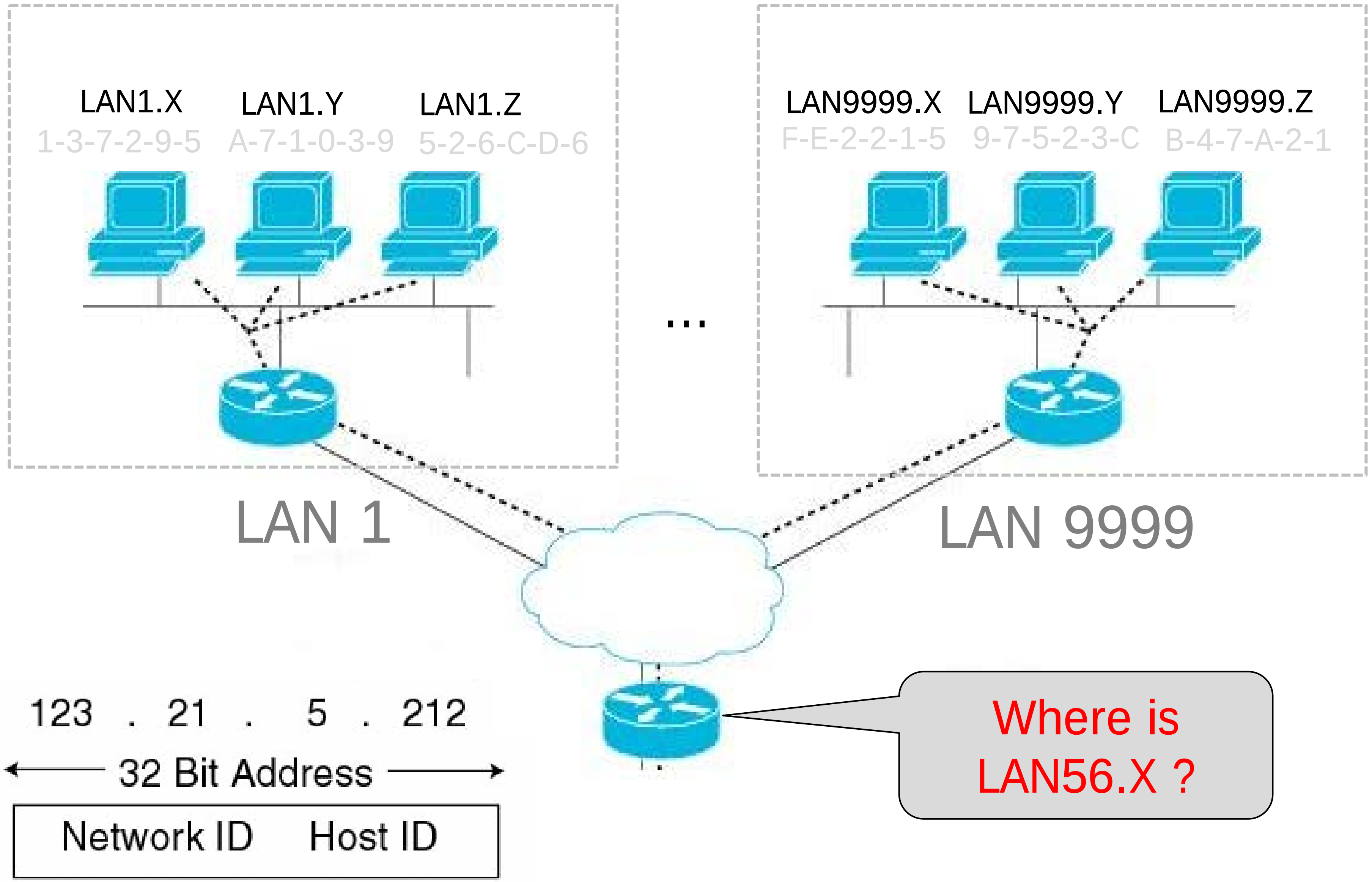
<u>IMPORTANT</u>: MAC addr. of neighboring stations (may) have nothing in common !!!

MAC addresses are convenient for station addressing (packet/frame delivery) in small LAN environments.

What if we try to use MAC Address to address stations (deliver packets) in a WAN !?



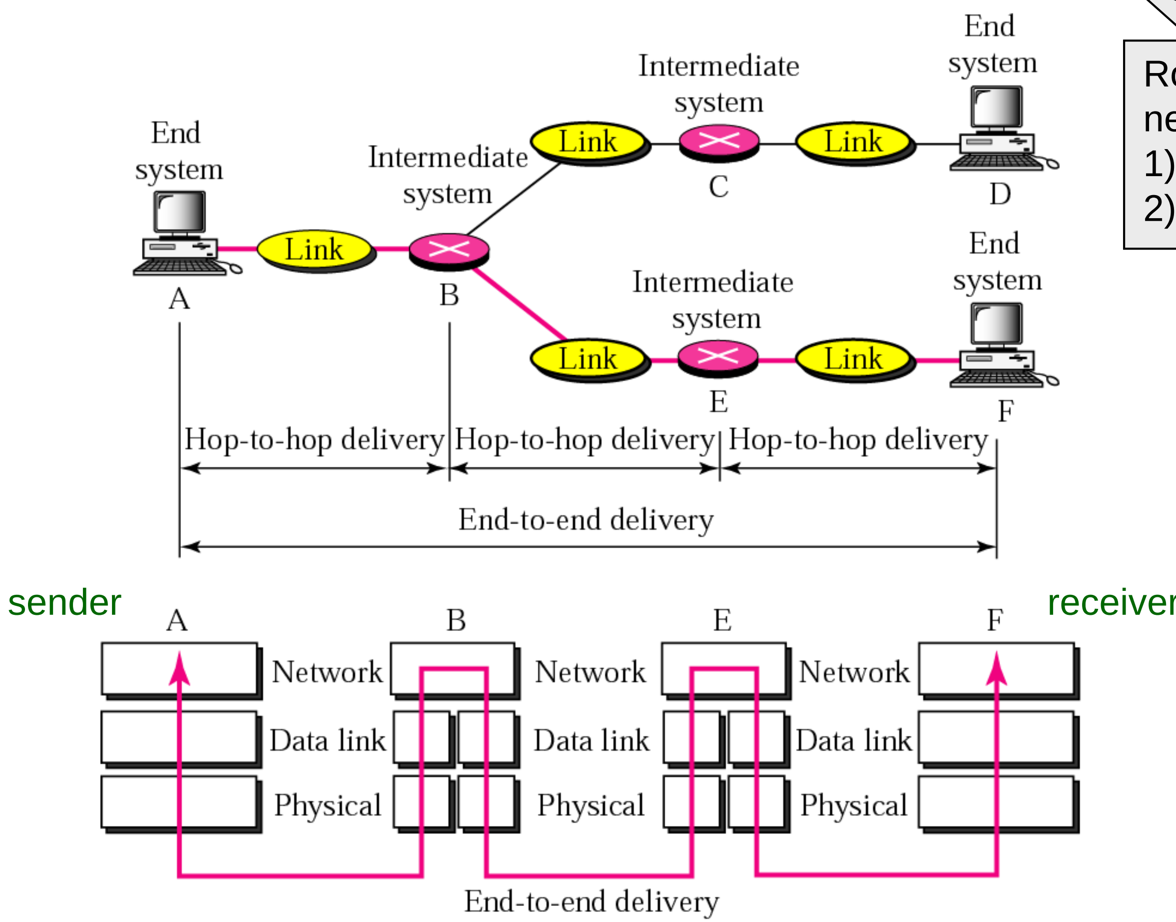
OSI Model: Network Layer



3. Network Layer

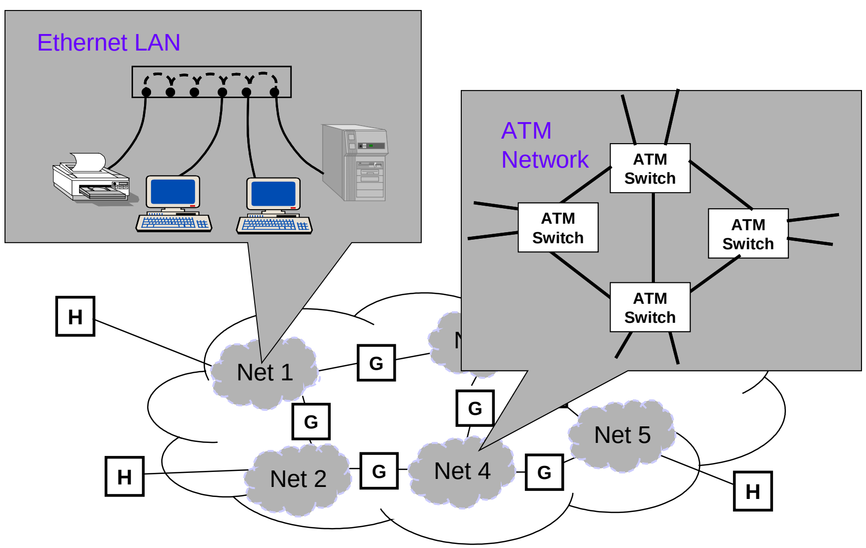
While the data link layer oversees the <u>delivery of packets</u> between two devices on the same network, the network layer is responsible for the <u>source-to-destination delivery</u> of <u>packet</u> across multiple networks / links.

Routing over multiple networks:
1) in min time, AND
2) with min overhead.

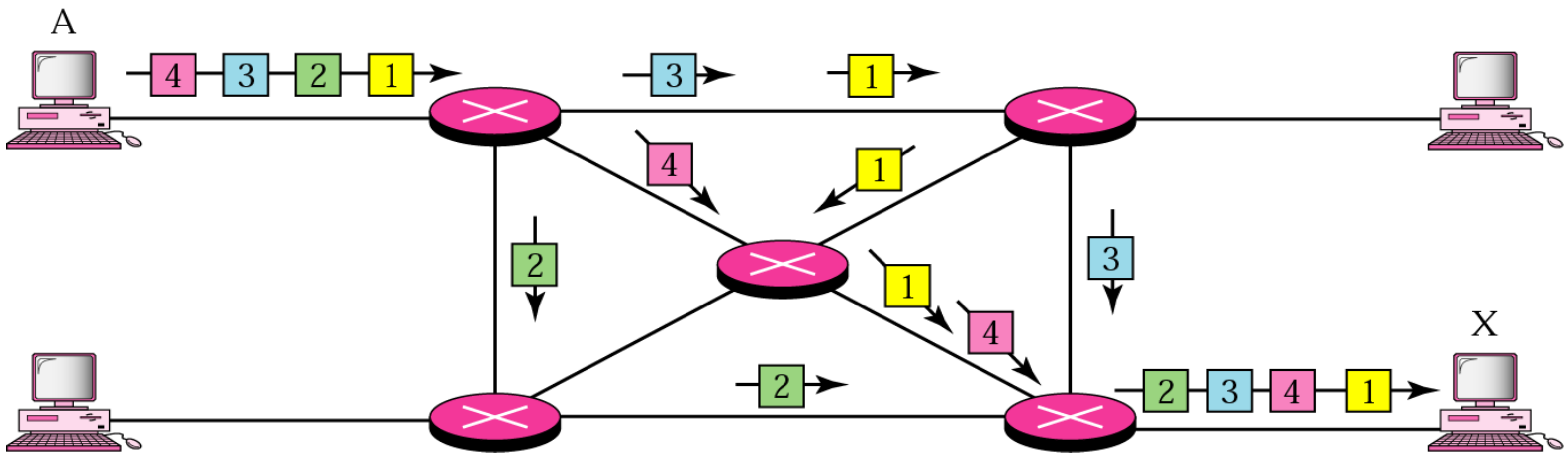


OSI Model: Network Layer

- **logical addressing**: **The physical addressing implemented by the data link layer handles the addressing / delivery problem locally – over a single wire. If a packet passes the network boundary another addressing system is needed to help distinguish between the source and destination network.**
- **routing**: **The N.L. provides the mechanism for routing/switching packets to their final destination, along the optimal path – across a large internetwork.**
- **fragmentation and reassembly**: **The N.L. sends messages down to the D.L.L. for transmission. Some D.L.L. technologies have limits on the length of messages that can be sent. If the packet that the N.L. wants to send is too large, the N.L. must split the packet up, send each piece to the D.L.L, and then have pieces reassembled once they arrive at the N.L. on the destination machine.**



OSI Model: Transport Layer

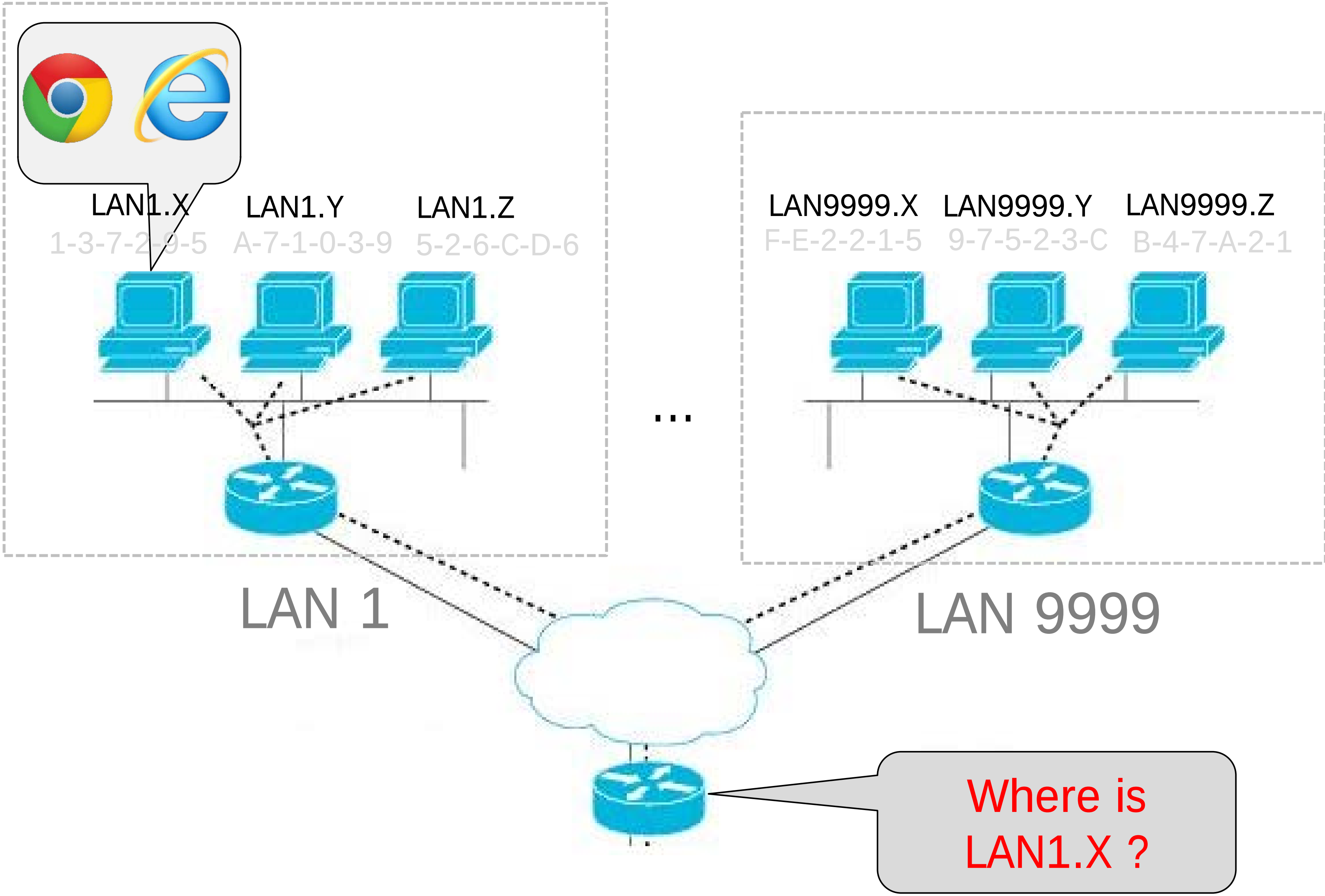


Routing individual packets well (end-to-end) is useful. But what if:

1) Packets come out of order?

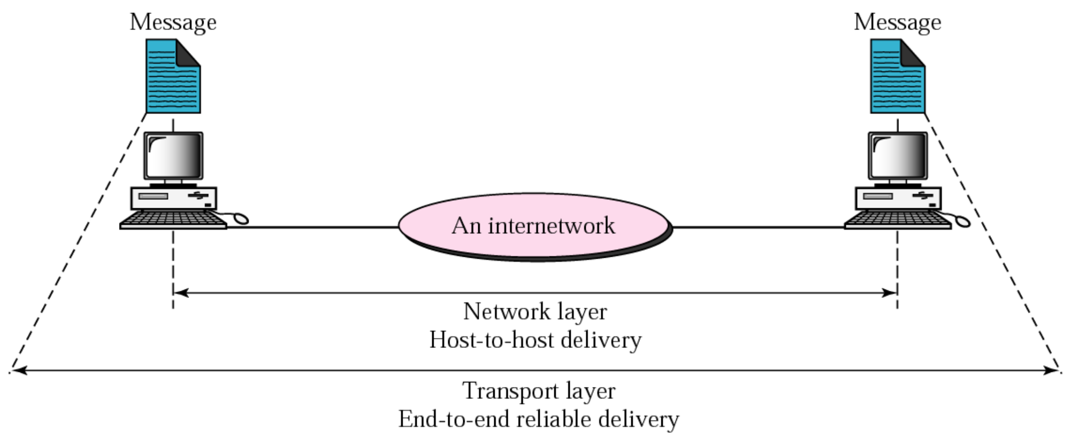
2) Packets get lost?

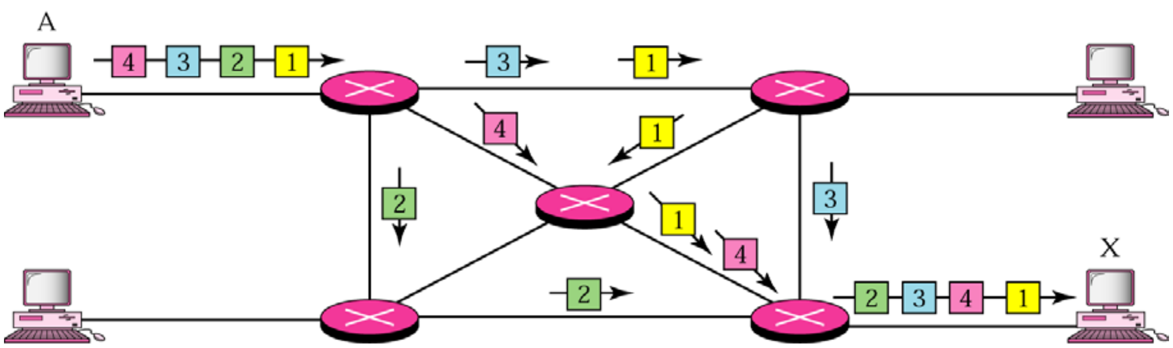
3) Multiple 'programs' (applications) run at the same time on the same sending/receiving machine?


LAN1.X
1-3-7-2-9-5
LAN1.Y
A-7-1-0-3-9
LAN1.Z
5-2-6-C-D-6
LAN9999.X
F-E-2-2-1-5
LAN9999.Y
9-7-5-2-3-C
LAN9999.Z
B-4-7-A-2-1
…
LAN 1
LAN 9999
Where is LAN1.X ?

4. Transport Layer

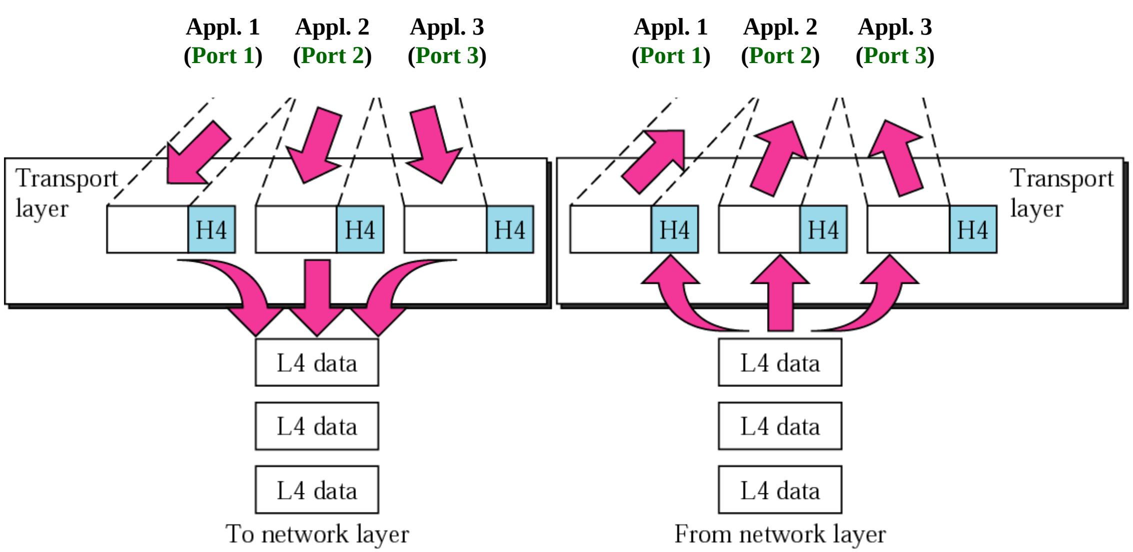
The transport layer is responsible for process-to-process delivery of entire message. While the network layer gets each packet to the correct computer, the transport layer gets the entire message to the correct process on that computer.





OSI Model: Transport Layer

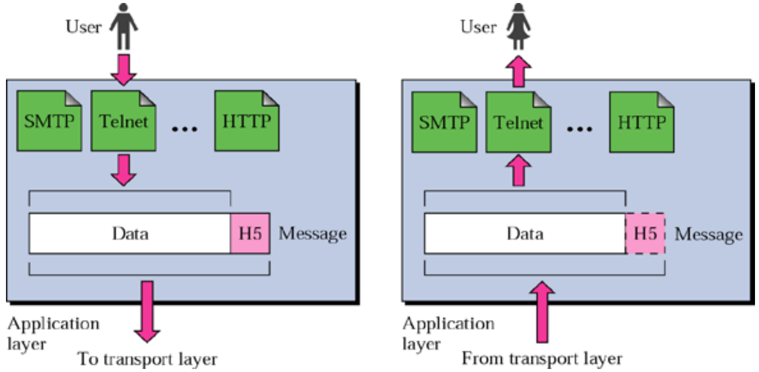
- **port addressing**: Computers often run several processes at the same time. For this reason, process-to-process delivery means delivery not only from one computer to the other but also from a specific process on one computer to a specific process on the other. The transport layer header therefore must include a type of address called a **port address**.
- **segmentation and reassembly**: A message is divided into segments, each segment containing a sequence number. These numbers enable the transport layer to reassemble the message correctly upon arrival at the destination, and to identify and replace packets that were lost in the transmission.
- **flow & error control**: Flow & error control at this layer are performed end-to-end rather than across a single link.



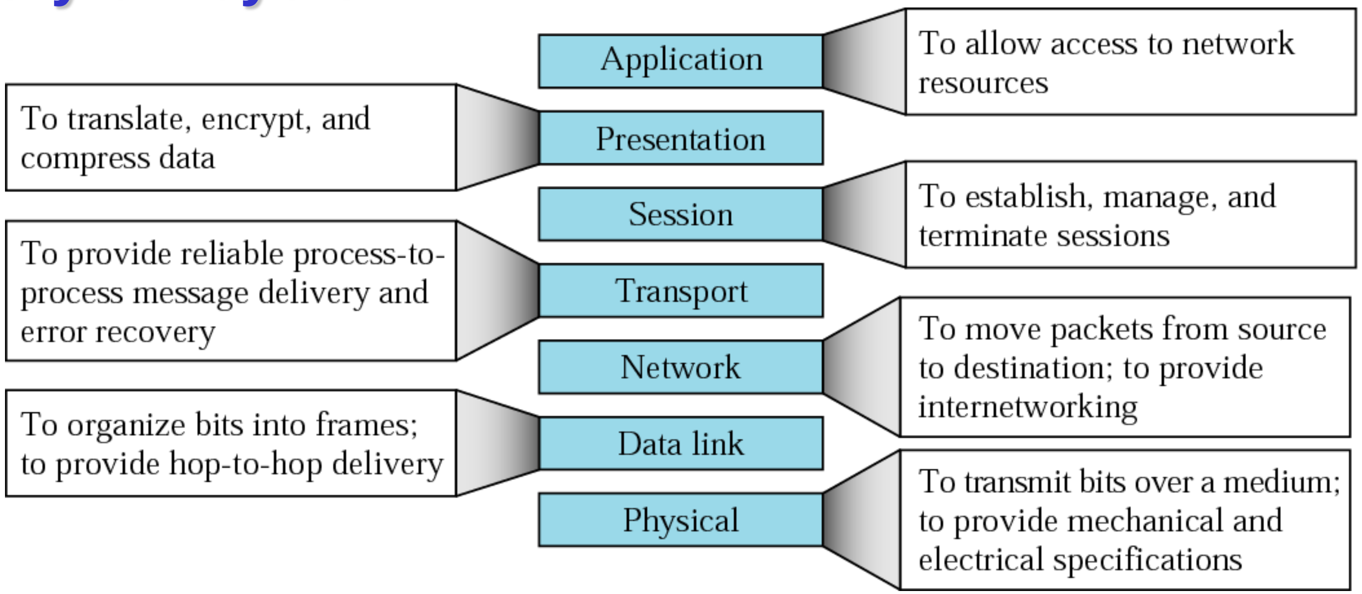
Application Layer (i.e. OSI Session + Presentation + Application Layer)

The application layer is responsible for providing the actual good-quality service to the user.

- **synchronization**: If a system is sending a large file, insert checkpoints every 100 pages to ensure that each 100-page unit is received and acknowledged independently. Thus, if a crash happens during the transmission of page 523, the only pages that need to be resend are pages 501 to 523.
- **encryption**: To carry sensitive information, a system must be able to ensure privacy. Encryption transforms the original information to another form, while decryption reverses the received message back to its original form.
- **compression**: Data compression reduces the number of bits contained in the information – it is particularly important in the transmission of multimedia.



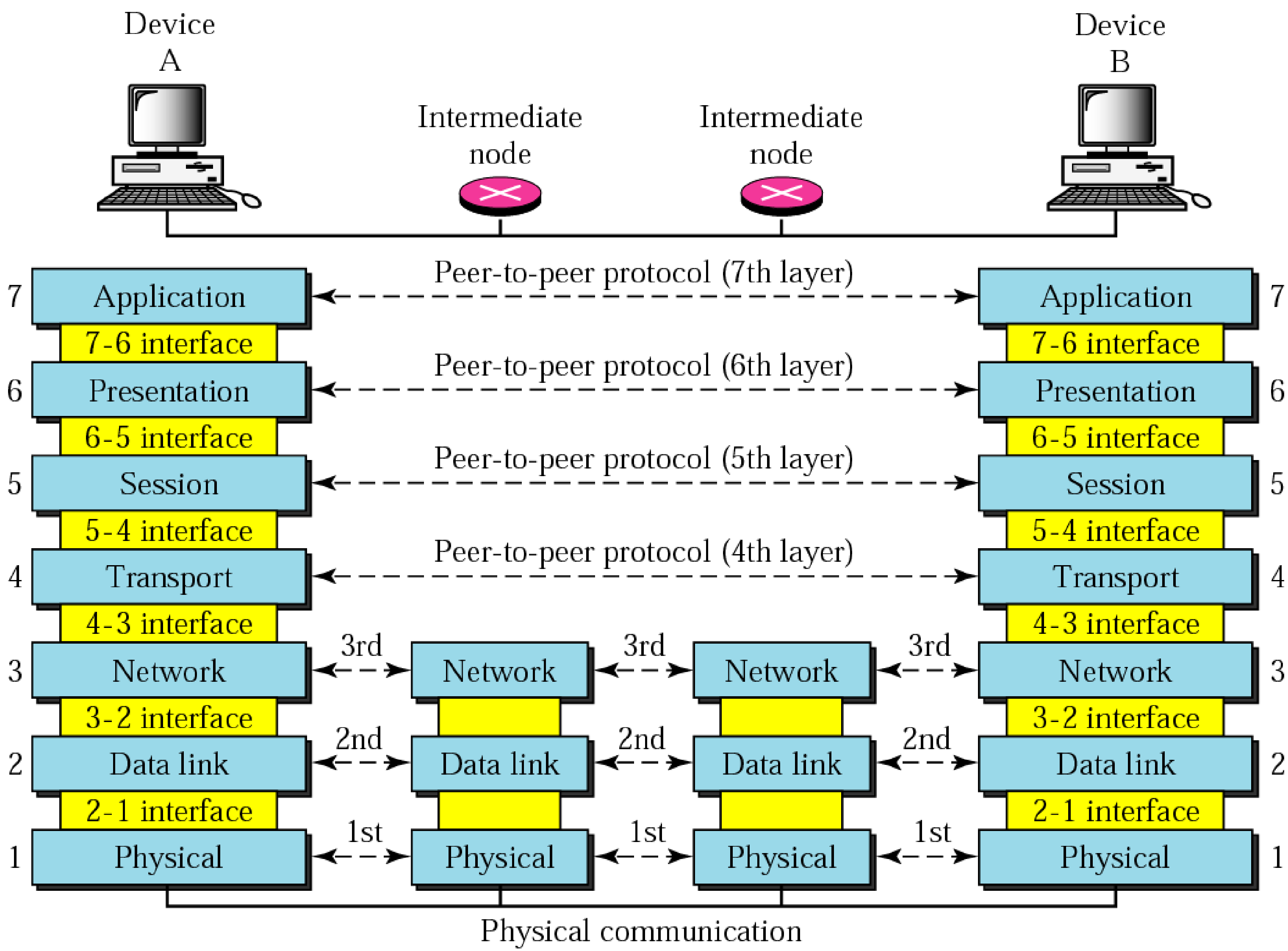
Summary of Layers



Why 7 Layers?

- **physical and application layer = bottom and top**
- **data link layer – bundles all link-dependent details**
- **network layer – responsible for hop-to-hop routing**
- **transport layer – responsible for end-to-end flow control**
- **session and presentation layer – provide some useful features; these can be easily provided in application layer**

OSI Model: Summary



OSI Model: Summary

Why did OSI Model Fail in Practice?

(1) Bad Timing

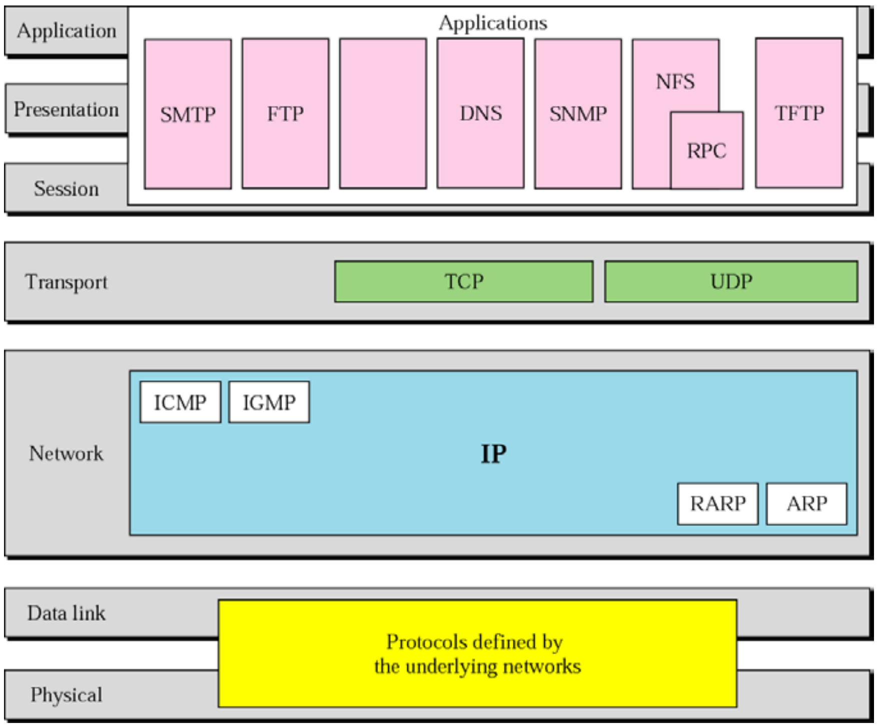
- although essential elements of OSI model were in place quickly, final standard (model + protocols) was not published until 1984
- by the time it took to develop OSI protocol standards, TCP/IP network architecture emerged as an alternative for open system interconnection
- free distribution of TCP/IP as part of Berkeley UNIX system ensured widespread use and development of numerous applications at various academic institutions

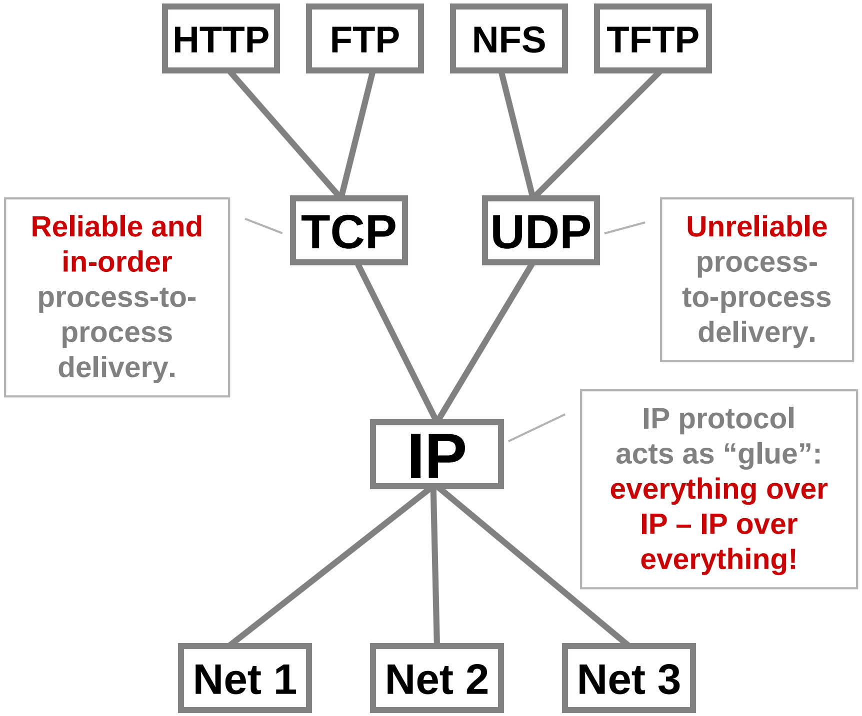
(2) Complexity and Inefficiency

- 7-layer OSI model was specified before there was much experience in designing large-scale OSI networks – several design choices were made in absence of concrete evidence of their effectiveness
- some functions, e.g. error control, appear in several layers (data link, transport, application) $\Rightarrow$ overall efficiency reduced

Internet Model

Internet Model and Hourglass Protocol Stack

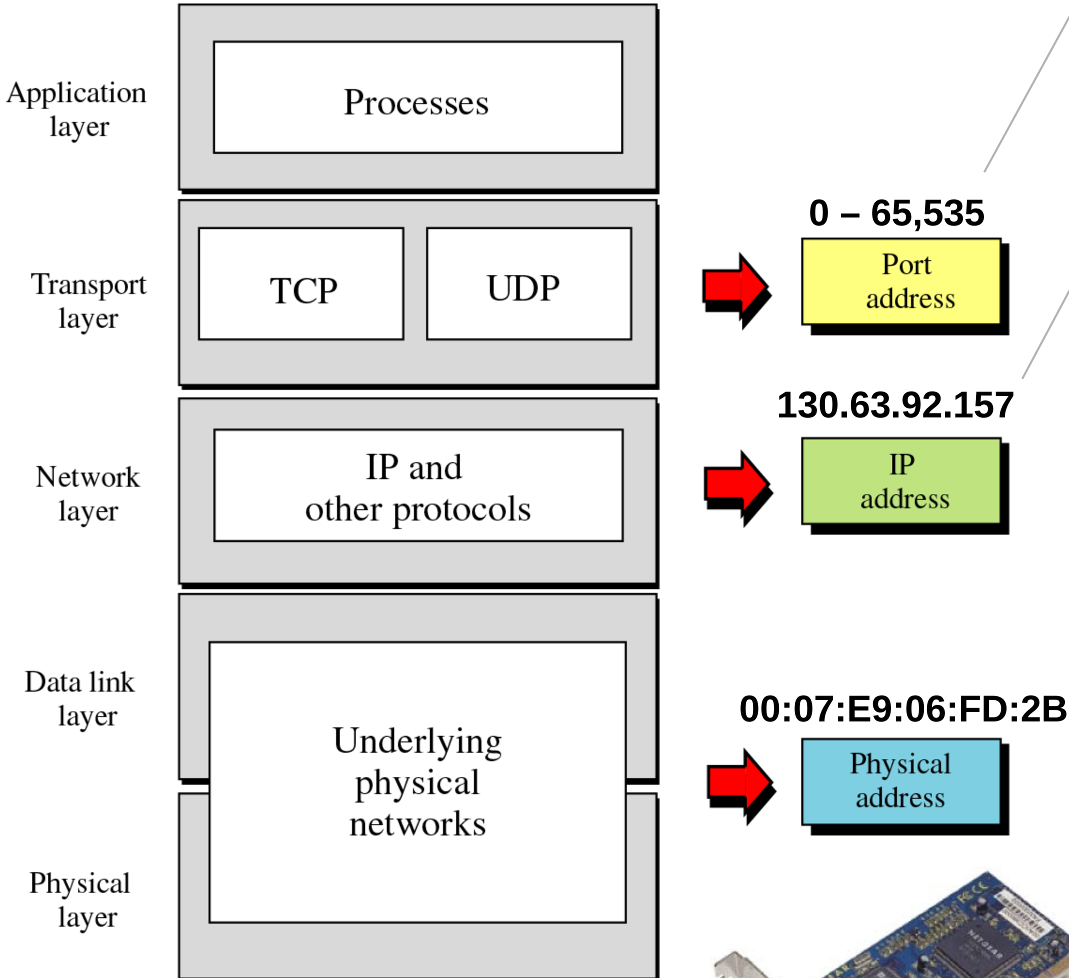




The operation of one single protocol at the network layer (IP protocol) over various networks provides independence from the underlying network technologies.
IP over anything, anything over IP!

Internet Model (cont.)

Addresses in TCP/IP Model



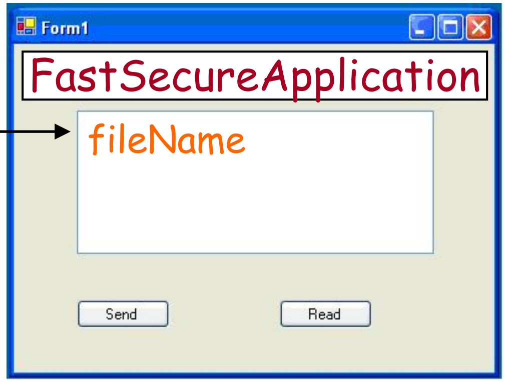
- **locally unique logical address** used to differentiate between applications sharing the same IP address

- **globally unique logical address** used to locate corresponding node in the entire Internet
- **hierarchical addresses that can be easily aggregated in routing tables** ⇒ **fast routing !**

- **globally unique NIC address** used to located corresponding node on a LAN
- **each NIC on a subnetwork may have different manufactures** ⇒

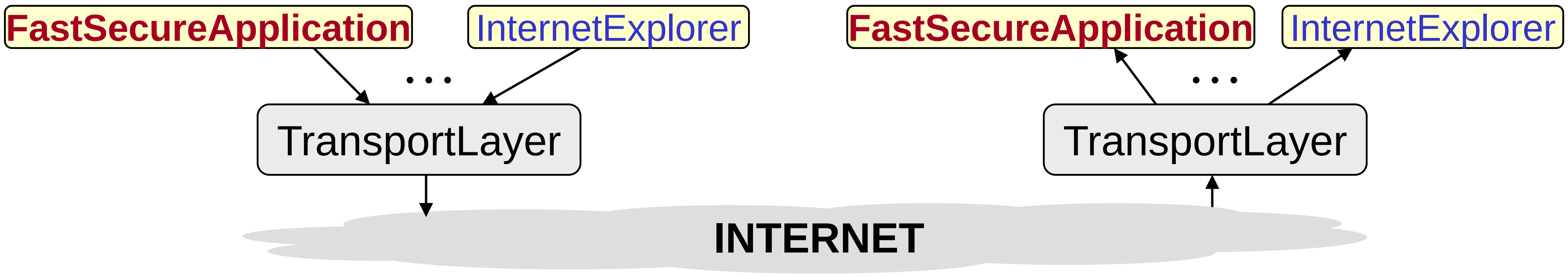
  we cannot aggregate physical addresses in routing tables ⇒

  large networks cannot use these addresses to identify hosts !

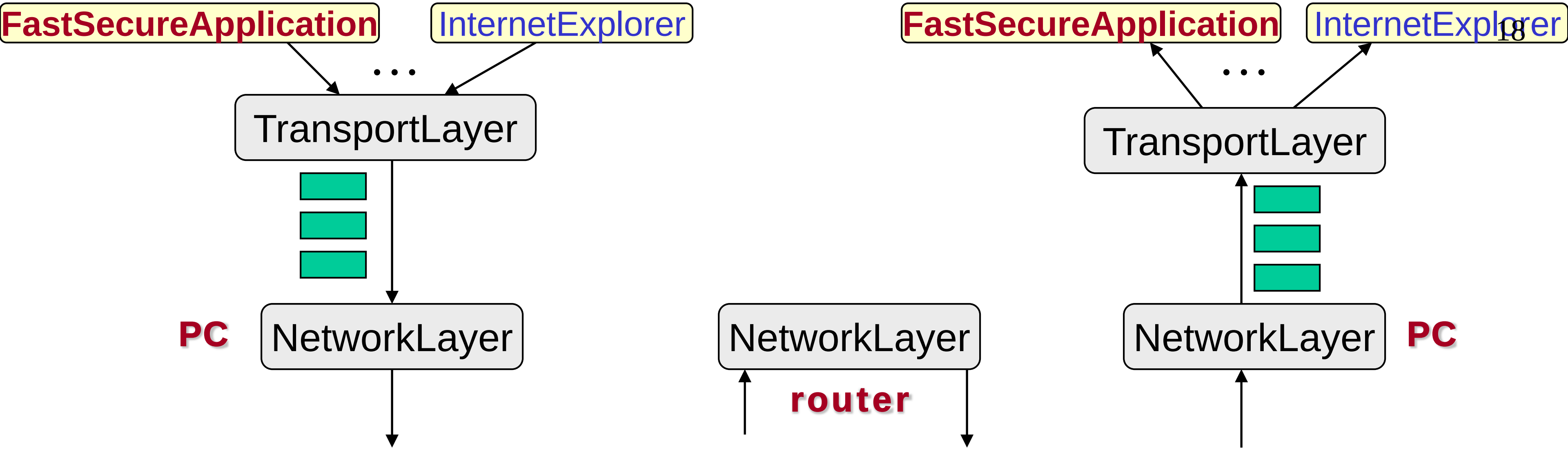



```
public void main(String[] args) {
    ...
    FastSecureApplication.send(file);
    ...
}
```

```
class FastSecureApplication {
    Object compress(Object file);
    Object encrypt(Object file);
    static void send(Object file);
    static Object deCompress();
    static Object deCrypt(Object file);
    static Object receive();
    ...
    static void send(Object file) {
            Object compressedFile = compress(file);
            Object encryptedFile = encrypt(compressedFile);
            TransportLayer.send(encryptedFile);
    ...
}
```




```
class TransportLayer {
    int pickPortNumber();
    Packets[] reassemble(Object file);
    Packets[] addHeaders(Packets[] filePackets, int portNmb);
    static void send(Object applicationLayerFile);
    static Object assemble();
    static Packets[] removeHeaders(Object file);
    ...
    static void send(Object applicationLayerFile) {
      int portNmb = pickPortNumber();
      Packets[] filePackets = reassemble(applicationLayerfile);
      Packets[] packetsWithHeader = addHeaders(filePackets, portNmb);
      for (int i=1; i< packetsWithHeader.length; i++) {
            NetworkLayer.send(packetsWithHeader[i]);
         ...
      }
```

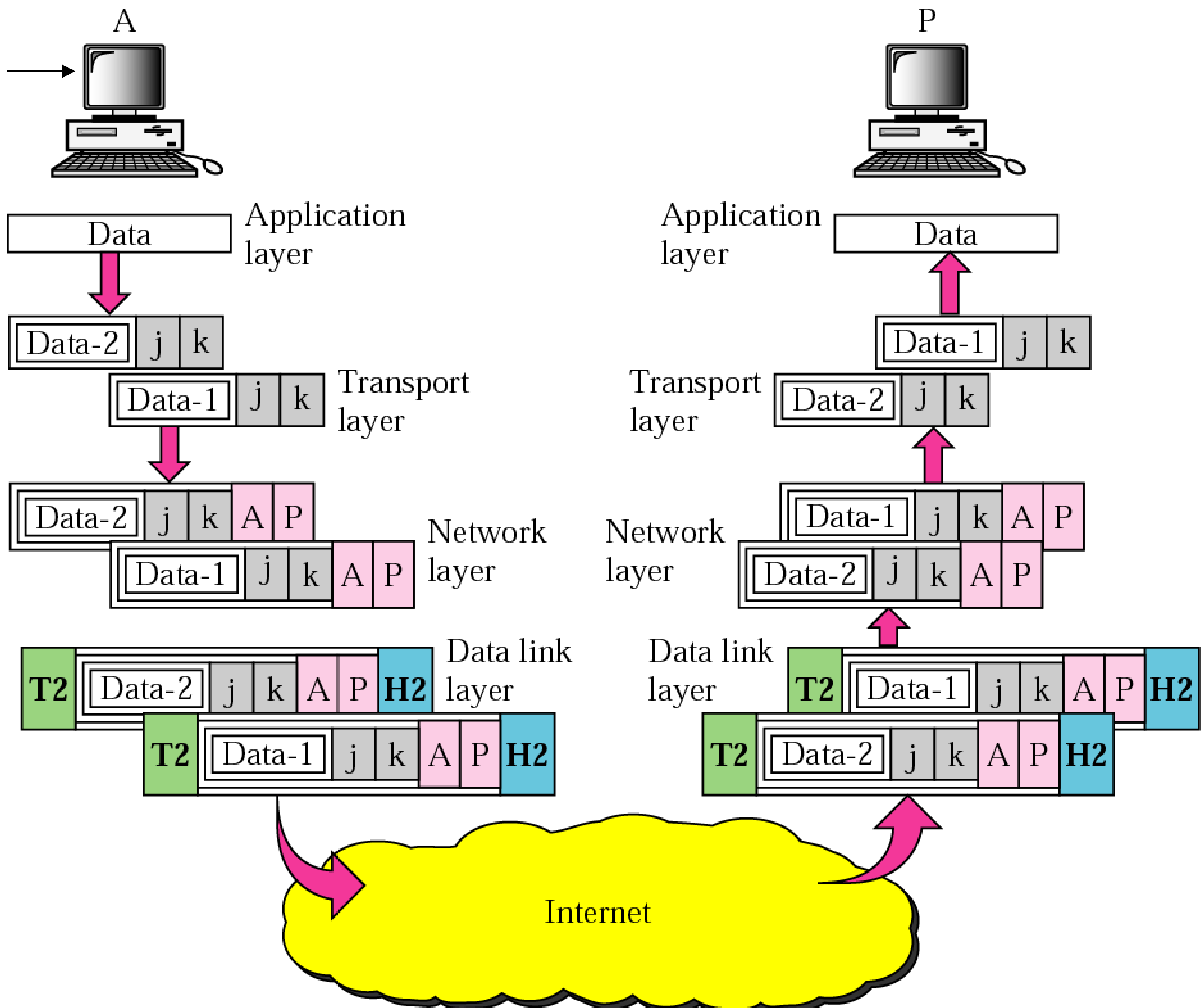



```
class NetworkLayer {
    ...
    static void send(Object trsportLayerPaket) {
        Packet IPPacket = addHeader(trsportLayerPacket, IPAddress, etc.);
        DataLinkLayer.send(IPPacket);
        ....      }

    static void sendRouter(Object IPPacket) {
        ... findNextHop(IPPacket);
        DataLinkLayer.send(IPPacket);
        ...       }
```

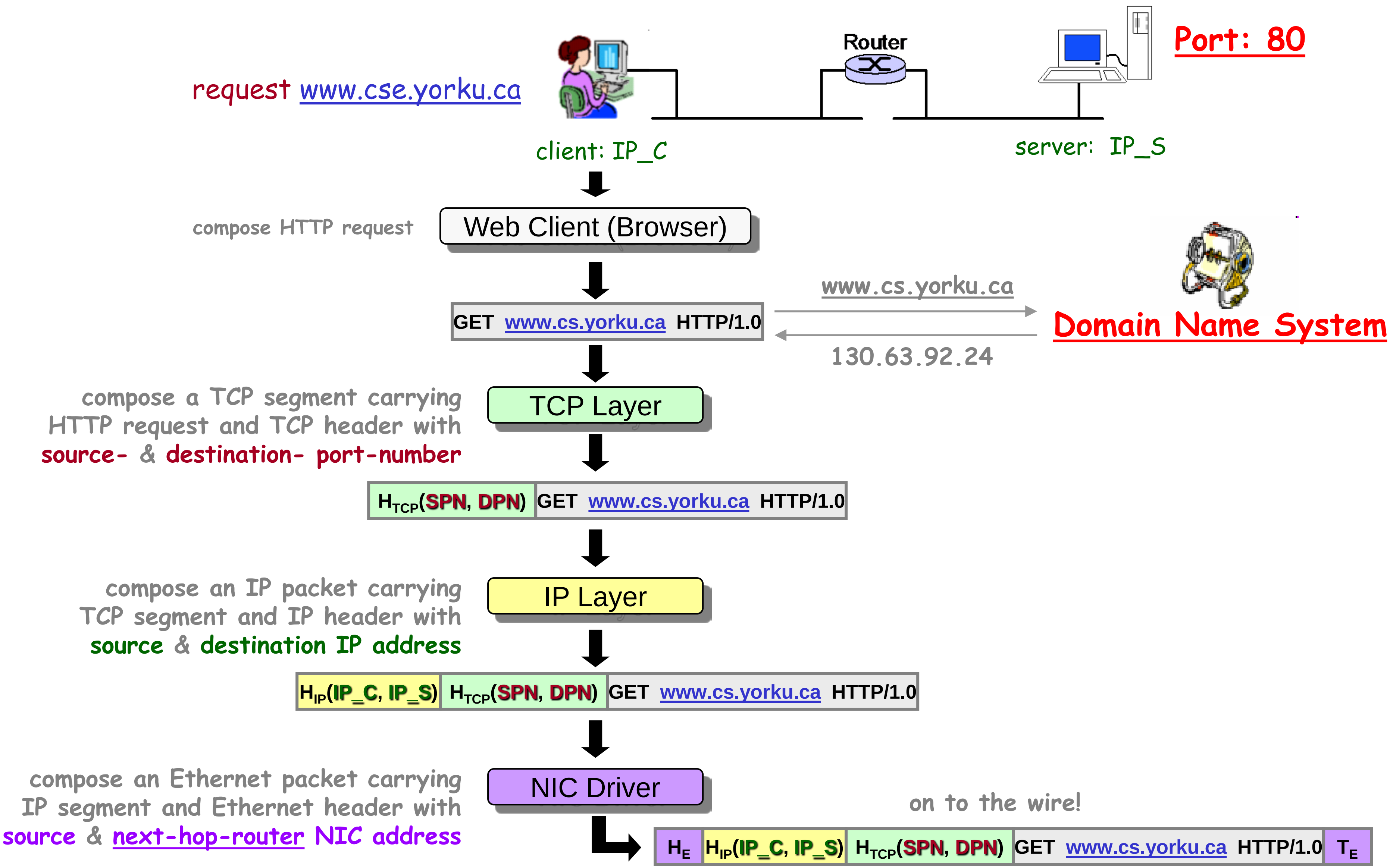
Example

Assume we want to exchange an image between computers A and P. The image, after being compressed, occupies 1000 bytes. The maximum packet size is 500 bytes.

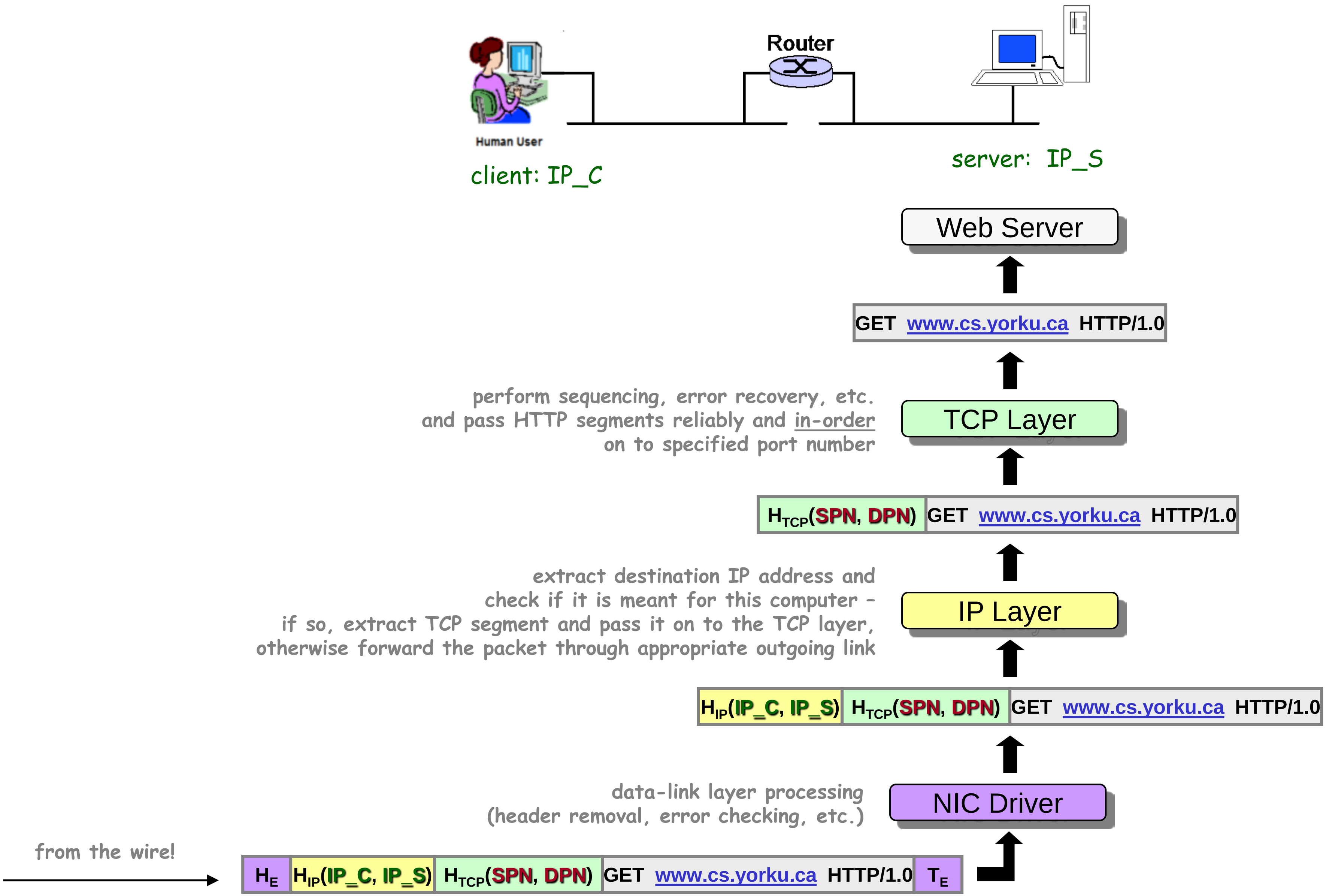


TCP/IP Protocol: How the Layers Work Together

Example [web-page retrieval – assumption: TCP connection established!]



TCP/IP Protocol: How the Layers Work Together (cont.)

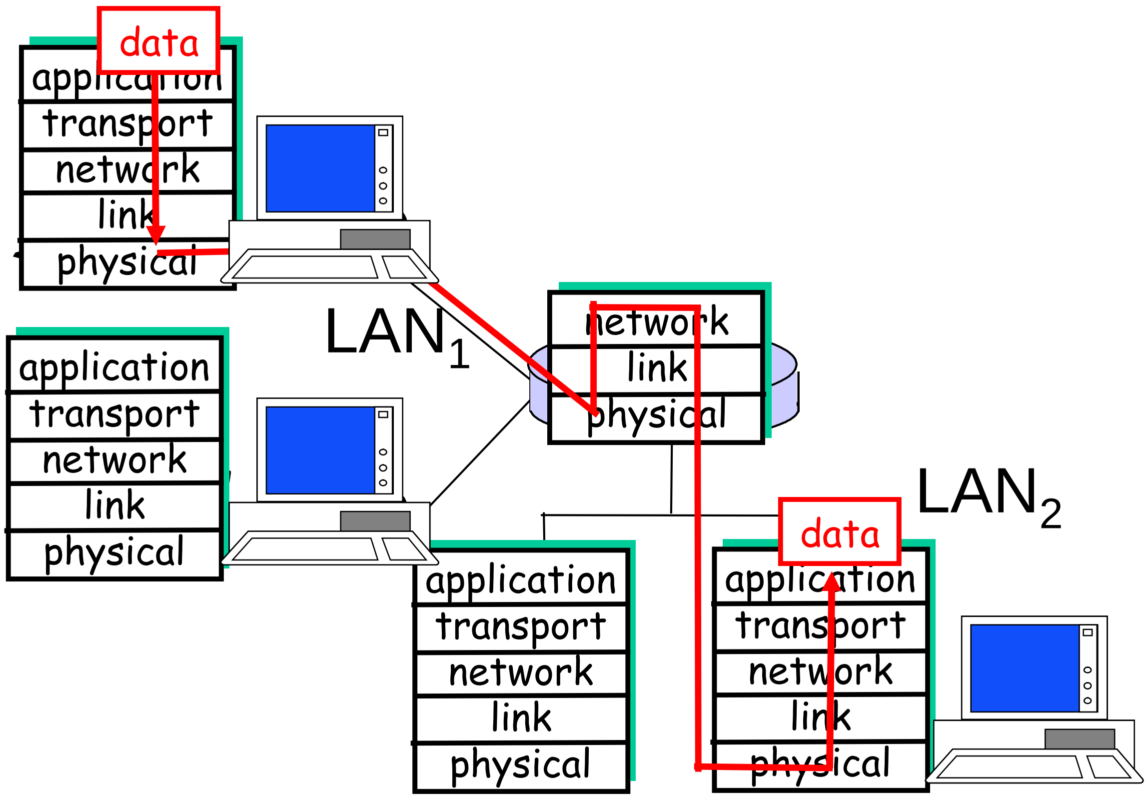


TCP/IP Protocol: How the Layers Work Together (cont.)

Bonus Question [layering – encapsulation]

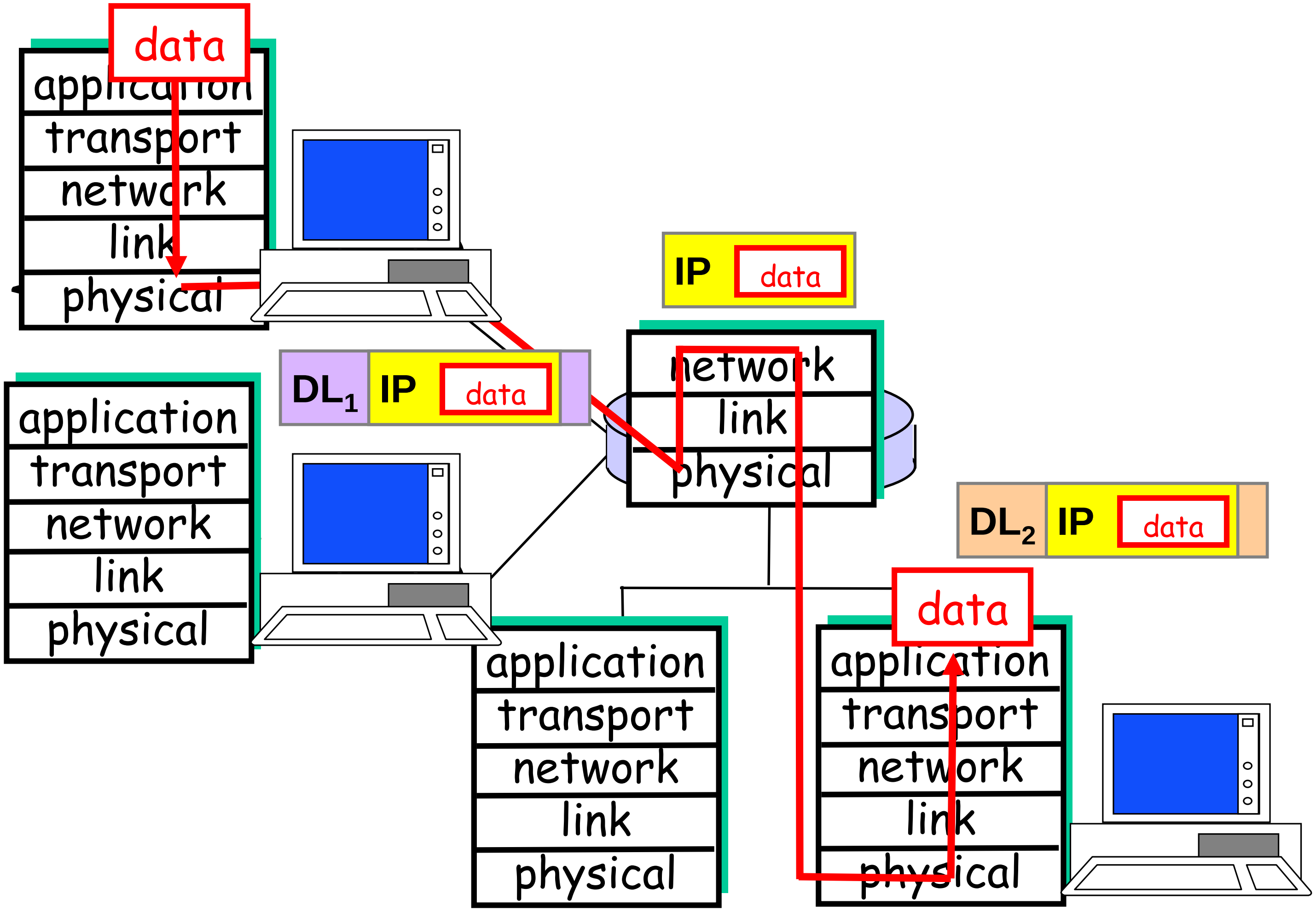
Assume two computers, situated on two distant LANs - with <u>different</u> data-link technologies, communicate with each other over the Internet.

Does each of these computers have to be aware of the data-link technology / protocol run in the LAN of the other computer?



(Source: Kurose & Ross)

TCP/IP Protocol: How the Layers Work Together (cont.)

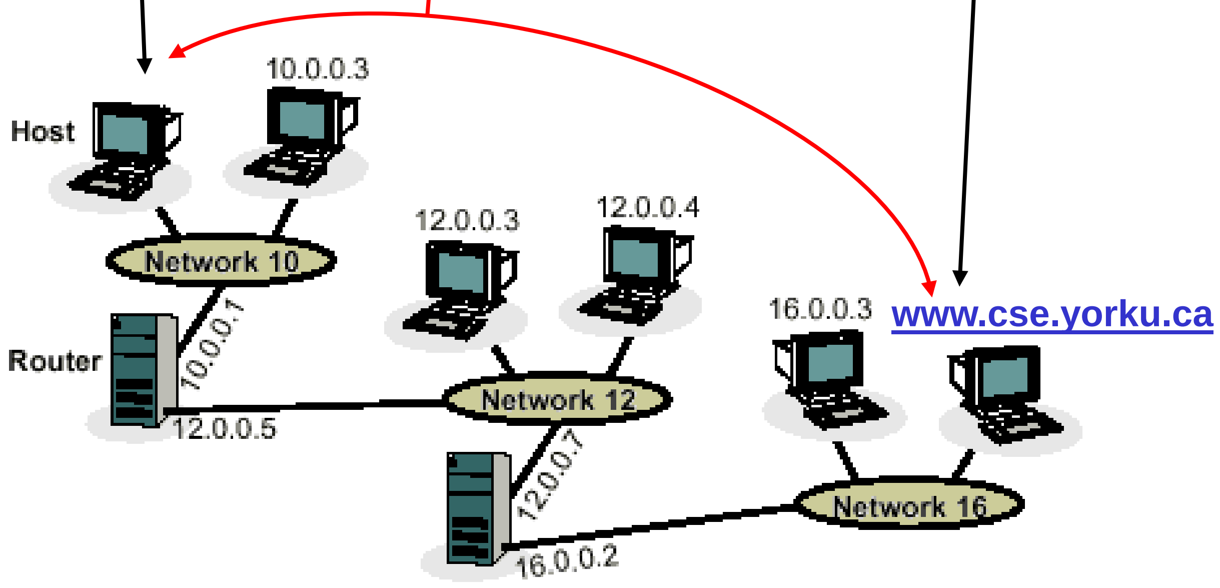


(Source: Kurose & Ross)

How to determine own IP & MAC address(es)?

How to determine the number and identity of intermediate routers?

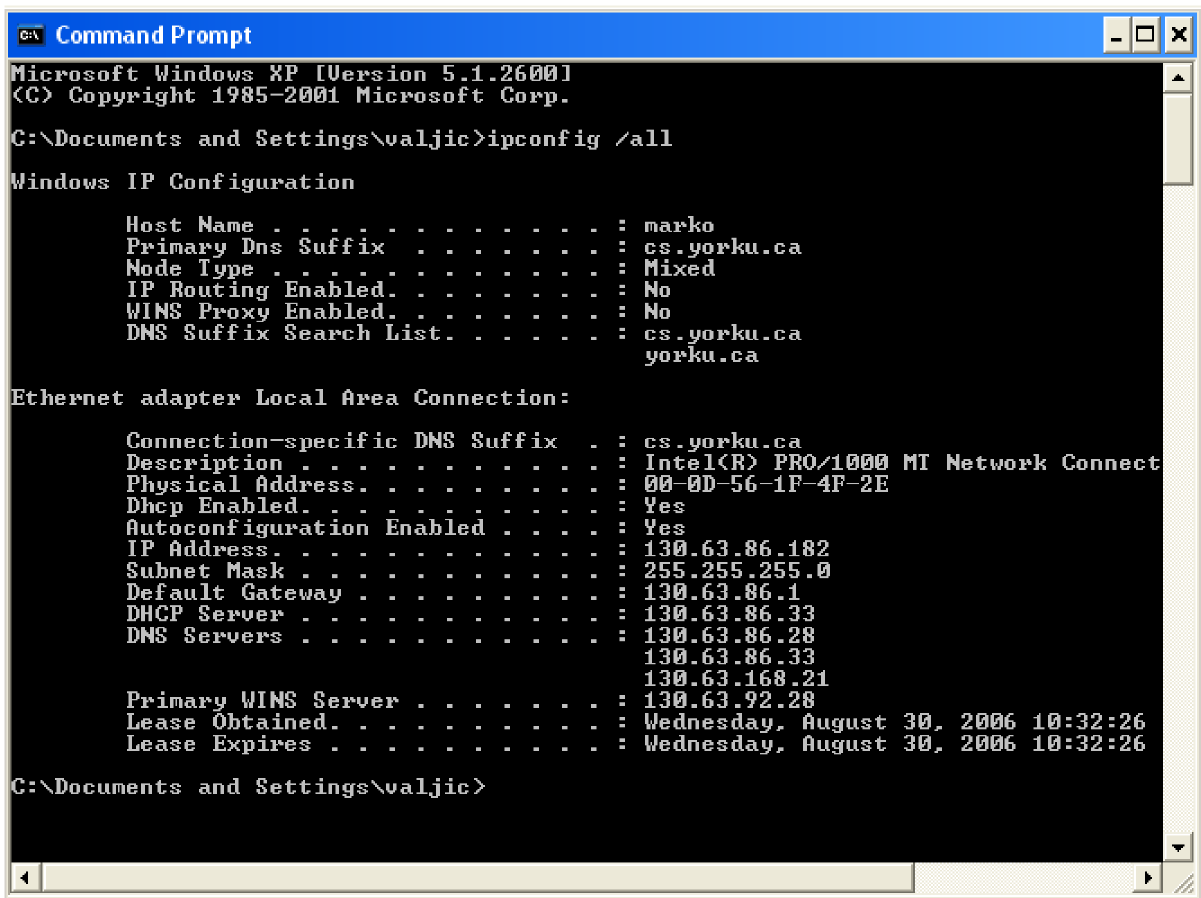
How to determine IP address of another remote machine?



IP Utilities

IPCONFIG – Microsoft Windows OS tool used to display TCP/IP information about the host - UNIX/Linux equivalents: ifconfig, ip addr

- **in simplest form returns IP address, subnet mask, default gateway**



IP Utilities (cont.)

PING – **standard troubleshooting tool (<u>available on most OS</u>) used to determine**

1) whether a remote computer is currently "alive"

2) round trip delay – max, min, average

- **Windows *ping* sends 4 32-bit packets to destination and reports**
  - **a) how many packets reached another computer**
  - **b) roundtrip delay for each**
- ***ping* makes use of ICMP messages**
- **if host names used instead of IP addresses, ping relies on DNS service to obtain respective IP address**

Command Prompt

```
C:\Documents and Settings\valjic>ping www.cbc.ca

Pinging a1849.gc.akamai.net [209.123.81.16] with 32 bytes of data:

Reply from 209.123.81.16: bytes=32 time=1ms TTL=58
Reply from 209.123.81.16: bytes=32 time=5ms TTL=58
Reply from 209.123.81.16: bytes=32 time=1ms TTL=58
Reply from 209.123.81.16: bytes=32 time=1ms TTL=58

Ping statistics for 209.123.81.16:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 5ms, Average = 2ms

C:\Documents and Settings\valjic>
```

IP Utilities (cont.)

Traceroute – utility (tool) that traces packet from *host_1* to *host_2*, showing **number of hops between hosts** and **how long each hop takes**

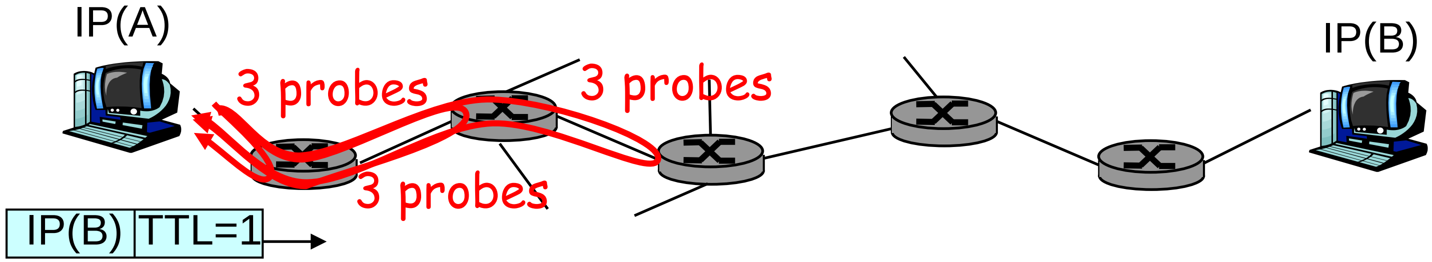
- **works by sending UDP packets with low TTL fields – TTL specifies how many hops packet is allowed to pass before being discarded**

  (1) sender first sends a UDP datagram with TTL=1 as well as an invalid port number to destination host

  (2) 1st router to see datagram sets TTL=0, discards datagram, and sends an ICMP Time Exceeded message to sender – this info enables sender to identify 1st machine in route and associated roundtrip delay

  (3) traceroute continues to identify remaining machines by sending datagrams with successively larger TTLs

- **traceroute repeats above experiment 3 times ⇒ source actually sends 3*N packets to destination (N=number of hops)**



IP Utilities (cont.)

Traceroute Origin – traceroute is a UNIX utility, but nearly all platforms have something similar

- Windows includes a traceroute utility called **tracert** – you can run tracert from MS-Dos Window, by entering tracert followed by domain name, e.g.

  tracert www.cs.yorku.ca

- tracert implementation is different from traceroute !!!

Traceroute Use – traceroute is generally used:

(1) as network debugging tool by pinpointing network connectivity problems

(2) for identifying IP addresses

Example [traceroute]

If you are visiting a Web site and pages are appearing slowly, you can use traceroute to figure out where the longest delay(s) are occurring.

IP Utilities (cont.)

Example [traceroute www.bbc.co.uk]

indigo.cs.yorku.ca - PuTTY

```
indigo 302 % traceroute www.cbc.ca
traceroute: Warning: www.cbc.ca has multiple addresses; using 206.167.78.33
traceroute to a1849.gc.akamai.net (206.167.78.33), 30 hops max, 38 byte packets
 1  gateway-92 (130.63.92.1)  0.308 ms  0.283 ms  0.365 ms
 2  core01.gw.yorku.ca (130.63.31.14)  0.737 ms  0.661 ms  0.631 ms
 3  border01.swx.yorku.ca (130.63.27.18)  1.861 ms  1.264 ms  0.883 ms
 4  york-hub-yorku-if.gtanet.ca (205.211.95.129)  0.720 ms  0.732 ms  0.431 ms
 5  ORION-GTANET-RNE.DIST2-TORO.IP.orion.on.ca (66.97.23.125)  0.682 ms  0.816 ms  0.550 ms
 6  DIST1-TORO-GE2-4.IP.orion.on.ca (66.97.16.105)  1.433 ms  1.011 ms  1.013 ms
 7  66.97.16.154 (66.97.16.154)  1.060 ms  1.089 ms  1.092 ms
 8  66.97.17.93 (66.97.17.93)  7.480 ms  7.366 ms  7.812 ms
 9  66.97.23.254 (66.97.23.254)  7.834 ms  7.674 ms  7.722 ms
10  orion-intrarisq.dgtnu-uq.risq.net (132.202.41.53)  7.790 ms  7.584 ms  7.588 ms
11  v2257-colo625.risq.net (132.202.45.14)  10.415 ms  10.443 ms  10.687 ms
12  206.167.78.33 (206.167.78.33)  367.520 ms  365.804 ms  358.620 ms
indigo 303 %
```

IP Utilities (cont.)

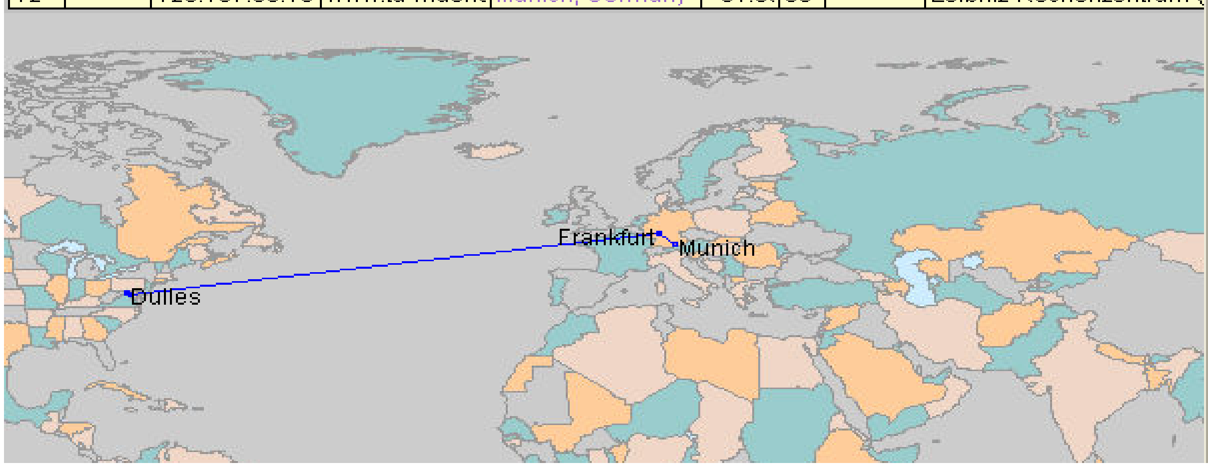
VisualRoute for Internet Performance:

http://visualroute.visualware.com/

Report for www.tu-muenchen.de [129.187.39.10]

Analysis: 'www.tu-muenchen.de' [w3proj5.ze.tum.de] was found in 12 hops (TTL=244). It is a HTTP server (running Apache/1.3.26 (Unix) PHP/4.2.2 AuthMySQL/2.20).

Hop	%Loss	IP Address	Node Name	Location	Tzone	ms	Graph	Network
0		161.58.180.11	WIN10115.visu	Dulles, VA, USA	-05:00		0 110	Verio, Inc. VRIO-161-058
1		161.58.176.12	-	Englewood, CO		0		Verio, Inc. VRIO-161-058
2		161.58.156.14	-	Englewood, CO		0		Verio, Inc. VRIO-161-058
3		129.250.28.20	xe-1-2-0-3.r20.	Ashburn, VA, USA	-05:00	0		Verio, Inc. VRIO-129-250
4		129.250.2.35	p64-0-0-0.r21.	Ashburn, VA, USA	-05:00	0		Verio, Inc. VRIO-129-250
5		208.51.6.33	-	Phoenix, AZ, USA	-07:00	0		Global Crossing GBLX-6I
6		67.17.65.54	so5-0-0-2488M	Frankfurt, German	+01:0(	78		Global Crossing GBLX-1
7		208.48.23.142	Dante-Frankfu	Frankfurt, German	+01:0(	88		Global Crossing GBLX-6
8		188.1.18.25	cr-muenchen1	Munich, Germany	+01:0(	94		IP networking on DFN's G
9		188.1.74.2	ar-muenchen1	Munich, Germany	+01:0(	99		IP networking on DFN's G
10		188.1.37.14	csrwan.lrz-mue	(Germany)	+01:0(	93		IP networking on DFN's G
11		129.187.1.244	csrkb1.lrz-mue	Munich, Germany	+01:0(	98		Leibniz-Rechenzentrum (
12		129.187.39.10	www.tu-muenc	Munich, Germany	+01:0(	93		Leibniz-Rechenzentrum (



http://www.visualware.com/resources/tutorials/tracert.html

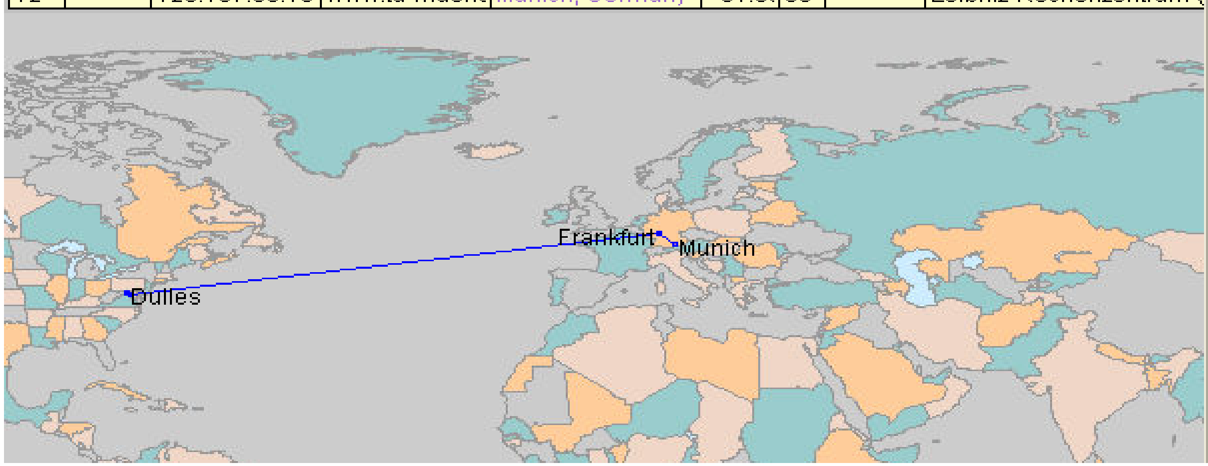
IP Utilities (cont.)

VisualRoute for Internet Performance:
http://visualroute.visualware.com/

Report for www.tu-muenchen.de [129.187.39.10]

Analysis: 'www.tu-muenchen.de' [w3proj5.ze.tum.de] was found in 12 hops (TTL=244). It is a HTTP server (running Apache/1.3.26 (Unix) PHP/4.2.2 AuthMySQL/2.20).

Hop	%Loss	IP Address	Node Name	Location	Tzone	ms	Graph	Network
0		161.58.180.11	WIN10115.visu	Dulles, VA, USA	-05:00		0 110	Verio, Inc. VRIO-161-058
1		161.58.176.12	-	Englewood, CO		0		Verio, Inc. VRIO-161-058
2		161.58.156.14	-	Englewood, CO		0		Verio, Inc. VRIO-161-058
3		129.250.28.20	xe-1-2-0-3.r20.	Ashburn, VA, USA	-05:00	0		Verio, Inc. VRIO-129-250
4		129.250.2.35	p64-0-0-0.r21.:	Ashburn, VA, USA	-05:00	0		Verio, Inc. VRIO-129-250
5		208.51.6.33	-	Phoenix, AZ, USA	-07:00	0		Global Crossing GBLX-6I
6		67.17.65.54	so5-0-0-2488M	Frankfurt, German	+01:0(	78		Global Crossing GBLX-1:
7		208.48.23.142	Dante-Frankfu	Frankfurt, German	+01:0(	88		Global Crossing GBLX-6/
8		188.1.18.25	cr-muenchen1	Munich, Germany	+01:0(	94		IP networking on DFN's G
9		188.1.74.2	ar-muenchen1	Munich, Germany	+01:0(	99		IP networking on DFN's G
10		188.1.37.14	csrwan.lrz-mue	(Germany)	+01:0(	93		IP networking on DFN's G
11		129.187.1.244	csrkb1.lrz-mue	Munich, Germany	+01:0(	98		Leibniz-Rechenzentrum (
12		129.187.39.10	www.tu-muenc	Munich, Germany	+01:0(	93		Leibniz-Rechenzentrum (



http://www.visualware.com/resources/tutorials/tracert.html

Top-down network design - Google Books - Microsoft Internet Explorer

Result 1 of 1 in this book for **traceroute shorter RTT time**

NOTE Traceroute is used to determine the IP routing path to a remote device. With UNIX and Cisco IOS operating systems, a traceroute packet is a User Datagram Protocol (UDP) "probe" packet sent to a high port number, in the 33,000 to 43,000 range. Microsoft operating systems send a ping rather than a UDP packet. Traceroute works by taking advantage of the ICMP error message a router generates when a packet exceeds its time-to-live (TTL) value. TTL is a field in the IP header of an IP packet.

Traceroute starts by sending a UDP probe or ping packet with a TTL of one. This causes the first router in the path to discard the packet and send back a time-exceeded (TTL exceeded) ICMP message. The traceroute command then sends several packets, increasing the TTL by one after a few packets have been sent at each TTL value. For example, it sends a few packets with TTL equal to 1, then a few packets with TTL equal to 2, then a few packets with TTL equal to 3, and so on, until the destination host is reached.

Each router in the path decrements the TTL. The router that decrements the TTL to zero sends back the time-exceeded (TTL exceeded) message. The final destination host sends back a ping reply (if the sender was using a Microsoft operating system) or a destination unreachable (port-unreachable) ICMP message (if the sender was using UNIX or Cisco IOS), because the high UDP port number is not a well-known port. This process allows a user to see a message from every router in the path to the destination, and a message from the destination.

Unfortunately, traceroute is not dependable. Some routers do not send back time-exceeded messages, either because they are simply not programmed to do so, or because they are configured to rate-limit ICMP or block ICMP for security reasons. Some routers incorrectly use the TTL of the incoming packet to send the time-exceeded message, which does not work. Also, some systems do not send the port-unreachable message, which means that traceroute waits for a long time before timing out. Finally, some service providers purposely change the results of traceroute to hide internal hops so that users think the providers' paths must be shorter than competitors' paths.

Fault Management

Fault management refers to detecting, isolating, diagnosing, and correcting problems. It also includes processes for reporting problems to end users and managers, and tracking trends related to problems. In some cases, fault management means developing workarounds until a problem can be fixed.

Network users expect quick and reliable fault resolution. They also expect to be kept informed about ongoing problems and to be given a timeframe for resolution. After a problem is resolved, they expect the problem to be tested and then documented in some sort

CCNA Questions

Q.1 **Which layer provides logical addressing that routers will use for path determination?**

Q.2 **Which layer is responsible for converting data packets into electrical signal?**

Q.3 **Which layer combines bits into bytes and bytes into frames, uses MAC addressing, and provides error detection?**

Q.4 **Which layer is used for reliable communication between end nodes over a WAN and controlling the flow of information?**

CCNA Questions (cont.)

Q.5 **Which fields are contained within an IEEE Ethernet frame header?**

(a) Source and destination MAC address.

(b) Source and destination network (IP) address.

(c) Source and destination MAC address and source and destination network (IP) address.

Q.6 **When data is encapsulated, which is the correct order?**

(a) Data, frame, packet, segment, bit.

(b) Segment, data, packet, frame, bit.

(c) Data, segment, packet, frame, bit.

(d) Data, segment, frame, packet, bit.