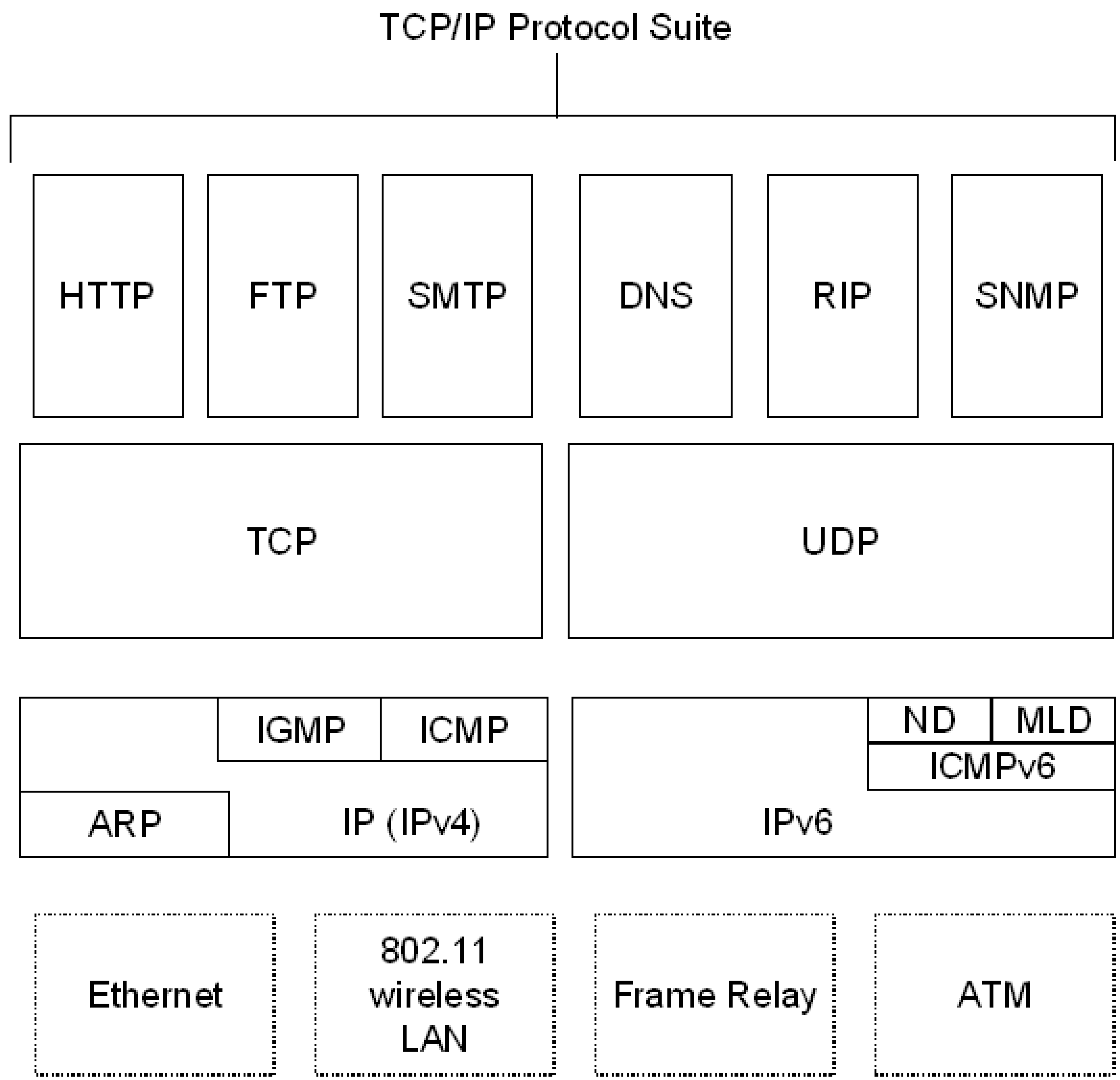
Layered Architectures and Applications

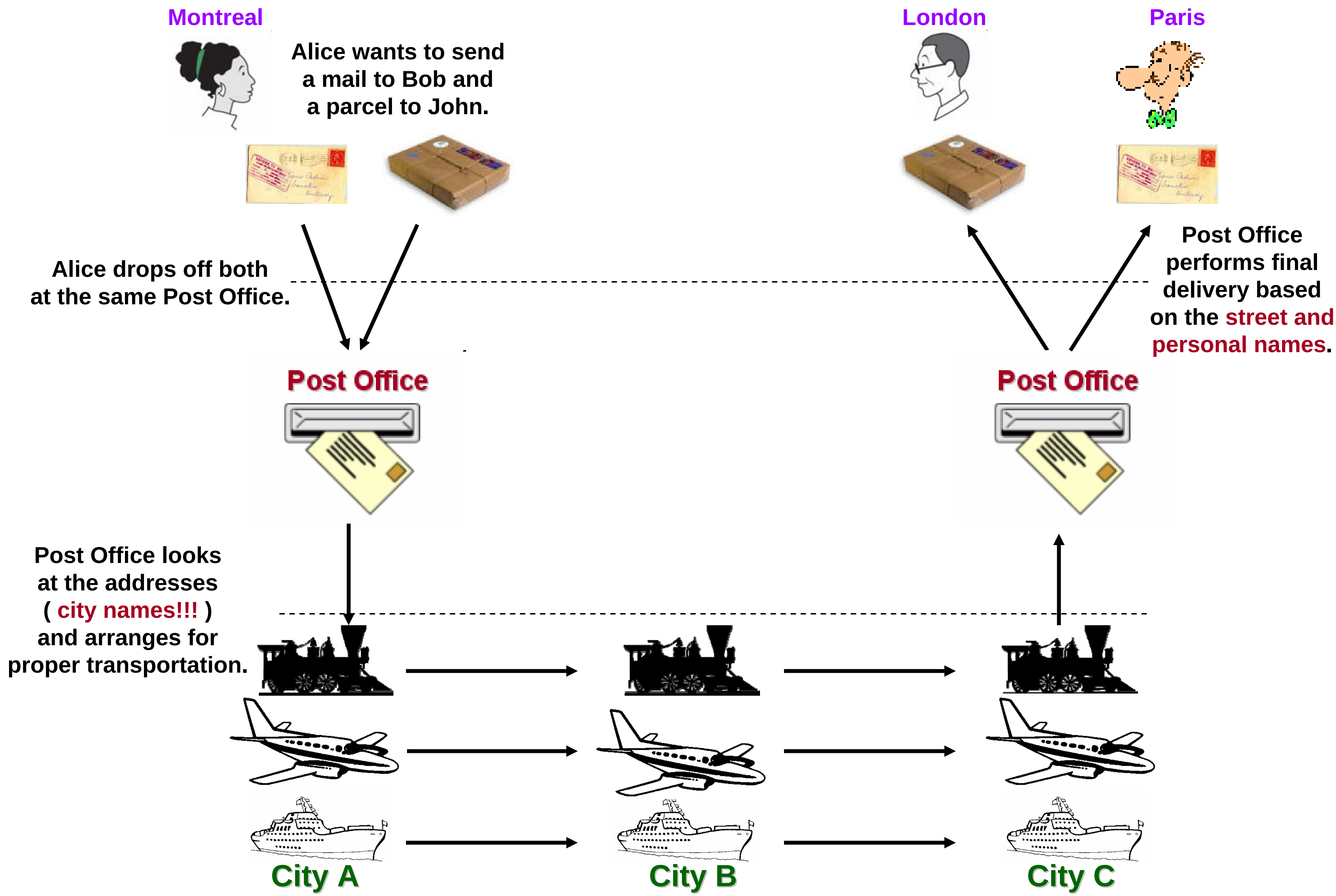
Required reading: Garcia 2.1, 2.2, 2.3

CSE 3213, Fall 2010

Instructor: N. Vlajic


TCP/IP Protocol Suite
HTTP
FTP
SMTP
DNS
RIP
SNMP
TCP
UDP
IGMP
ICMP
ARP
IP (IPv4)
ND
MLD
ICMPv6
IPv6
Ethernet
802.11 wireless LAN
Frame Relay
ATM

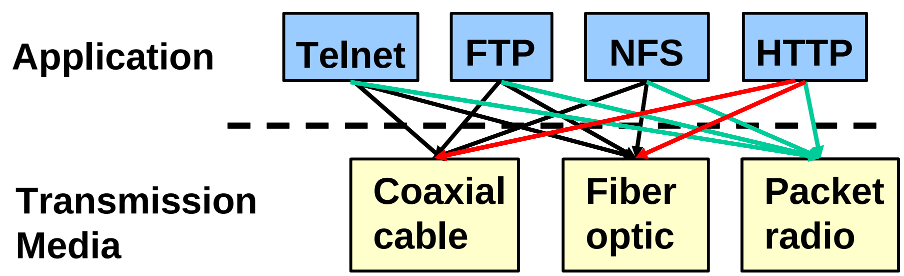
Why Layering?!



Why Layering?! (cont.)

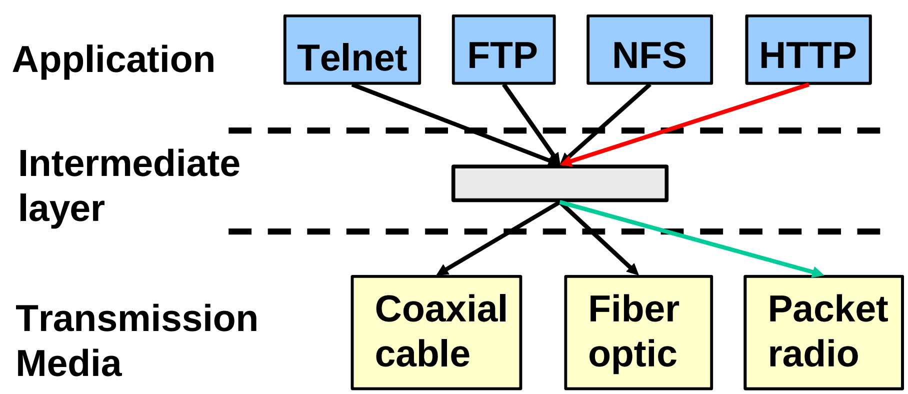
No Layering

- **each new application has to be *re*-implemented for every network technology!**



Layering

- **intermediate layer(s) provide a unique abstraction for various network technologies**



Why Layering?! (cont.)

Why Protocol Layering?

1) **modularity** – **one problem is decomposed into a number of smaller more manageable subproblems ⇒ more flexibility in designing, modifying and evolving computer networks**

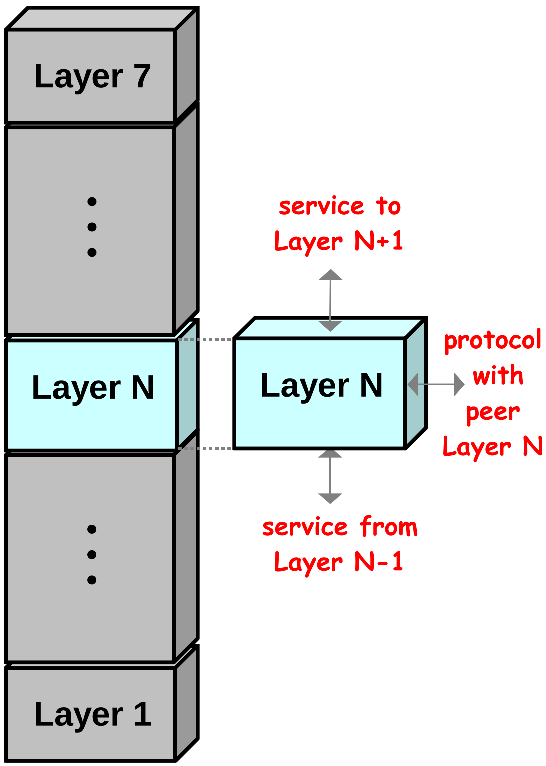
2) **functionality reuse** – **a common functionality of a lower layer can be shared by many upper layers**

A monolithic network design that uses a single large body of hardware and software to meet all network requirements can quickly become obsolete and also is extremely difficult and expensive to modify.

Layered approach accommodates incremental changes much more rapidly.

Layered Architecture

Protocol Layering – grouping of related communication functions into hierarchical set of **layers**



- each layer:
  - (1) performs a subset of functions required for communication with another system
  - (2) **relies on next lower layer** to perform more primitive functions
  - (3) **provides service to next higher layer**
  - (4) implements **protocol** for communication with **peer layer** in other systems
- <u>vertical communication</u> – commun. between adjacent layers – requires mutual understanding of what services and/or information lower layer must provide to layer above
- <u>horizontal communication</u> – commun. between software or hardware elements running at the same layer on different machines

Communication between peer processes is <u>virtual</u>, i.e. indirect.

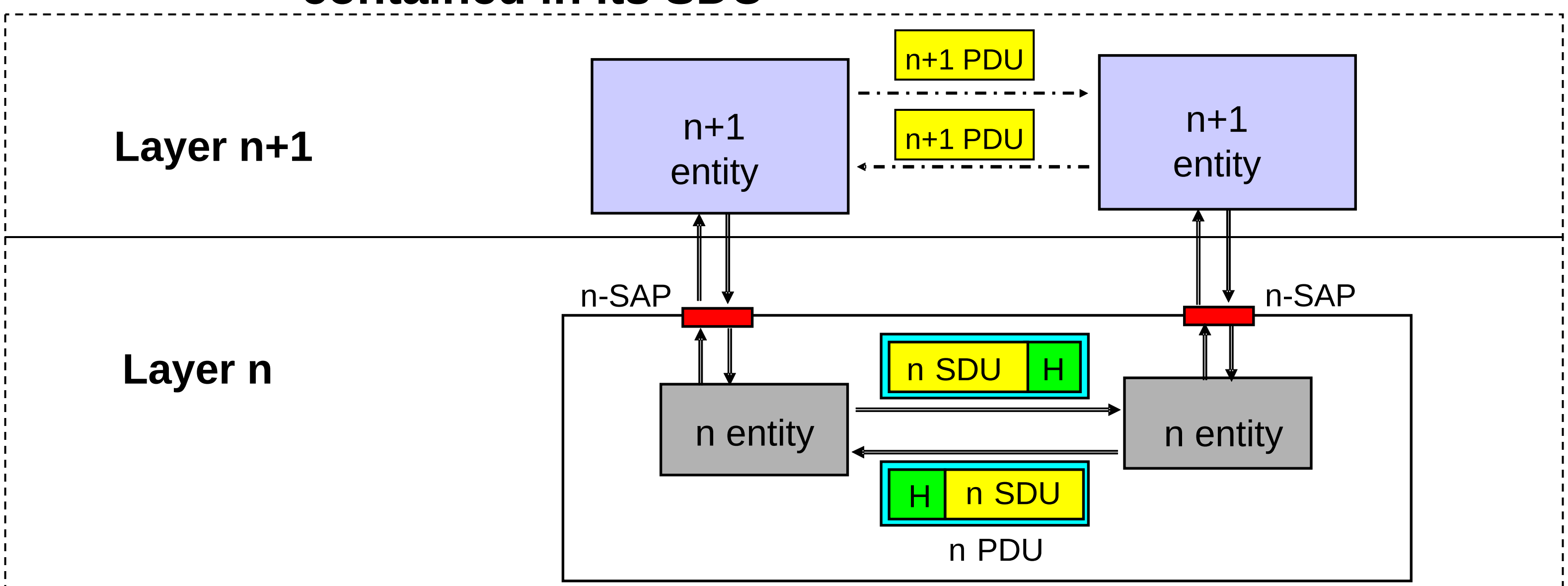
Layered Architecture (cont.)

Protocol – set of rules that govern data comm. between peer entities

- **layer-n peer processes communicate by exchanging Protocol Data Units (PDUs)**

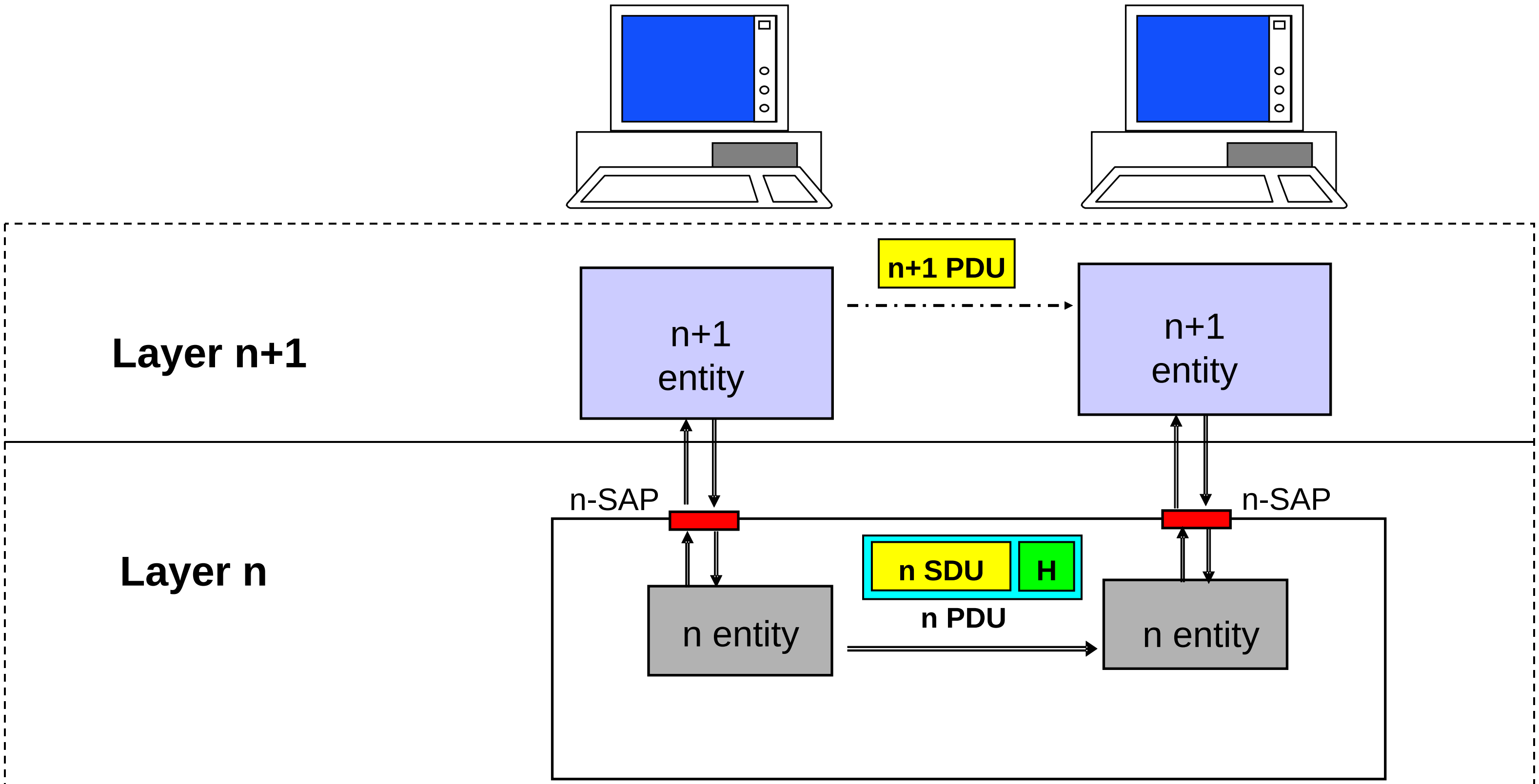
Service – can be accessed through Service Access Points (SAP's)

- **layer n+1 PDU = layer n SDU (SDU = Service Data Unit)**
- **layer n process adds control information (header) to its SDU to produce layer n PDU – encapsulation!**
- **layer n does not interpret or make use of information contained in its SDU**

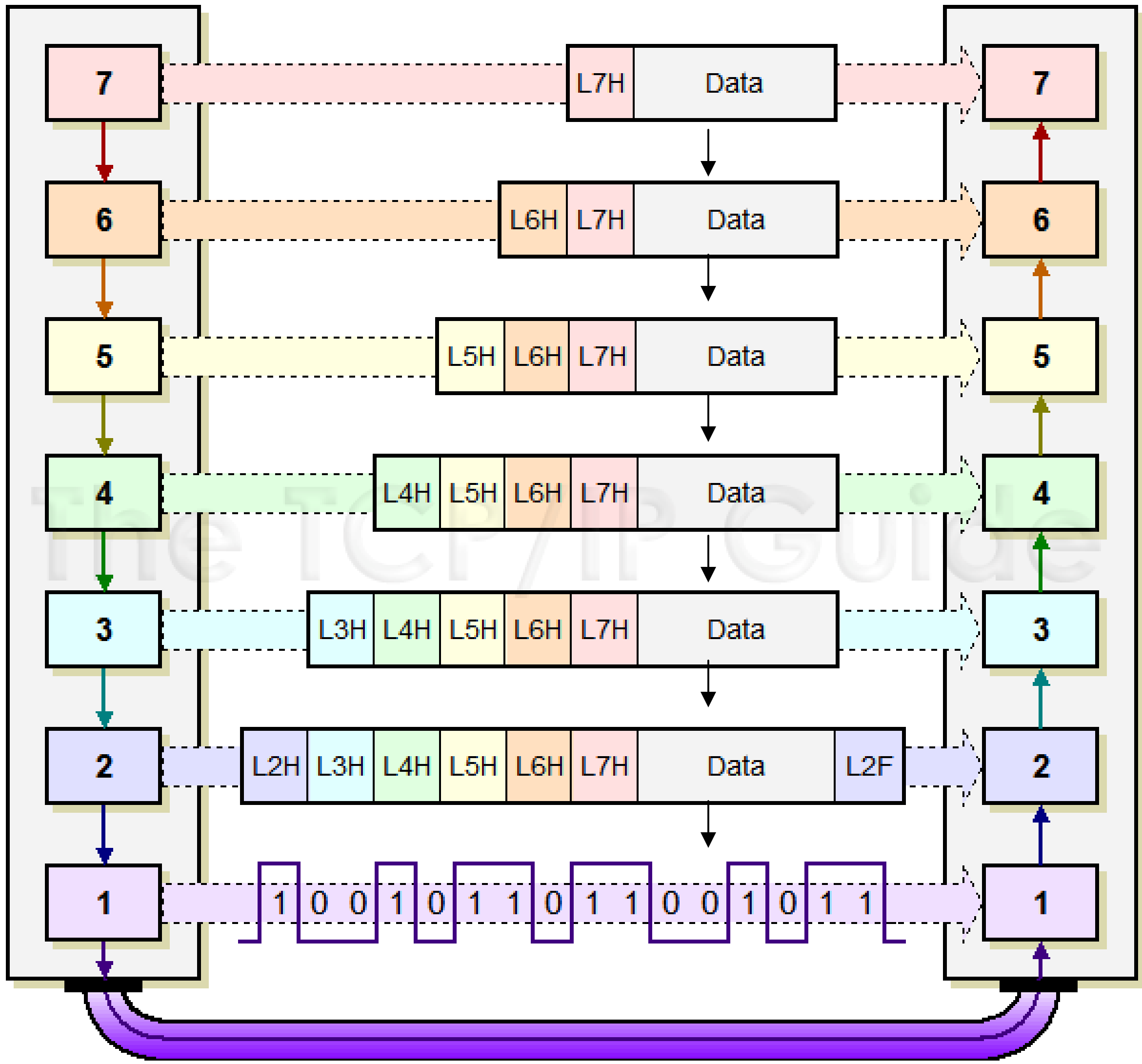


Layered Architecture (cont.)

Example [layering – vertical vs. horizontal flow of information]



Layered Architecture (cont.)



http://www.tcpipguide.com/free/t_DataEncapsulationProtocolDataUnitsPDUsandServiceDa.htm

OSI Model

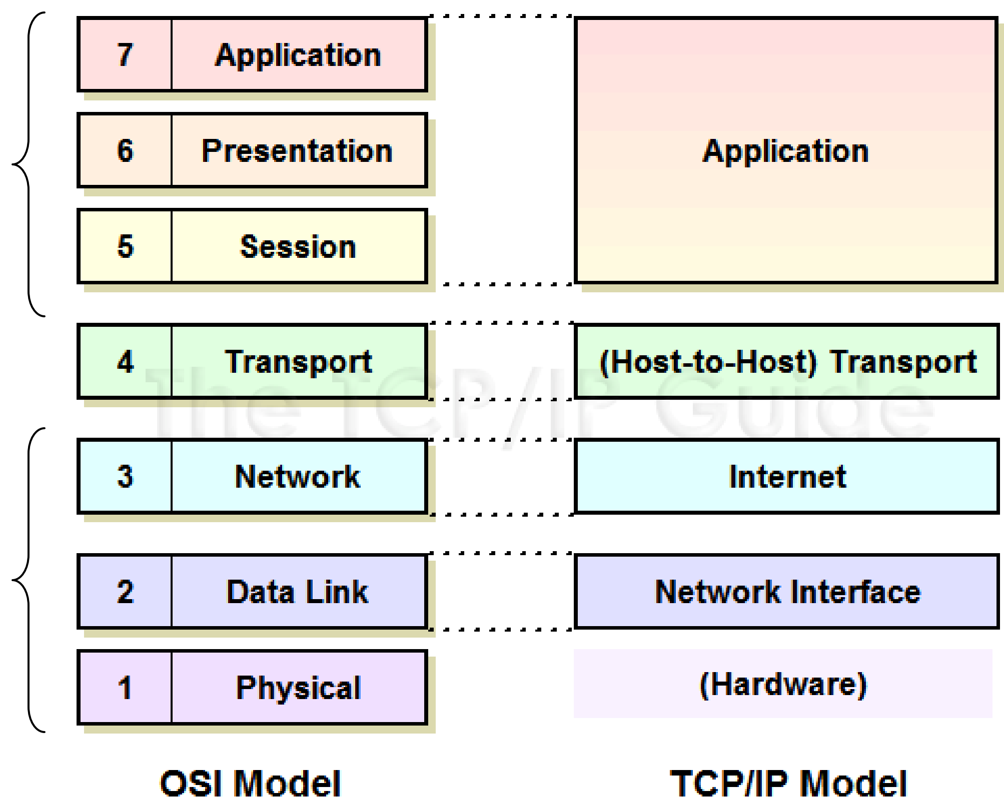
Layered OSI Architecture

- composed of 7 ordered layers
- there is fairly natural correspondence between TCP/IP & OSI layers $\Rightarrow$ TCP/IP architecture can be explained in terms of corresponding OSI layers

application support layers – allows communication with end-user and interoperability among unrelated software systems

transport layer – links upper and lower group – ensures that what lower layers have transmitted is in a form that upper layers can use

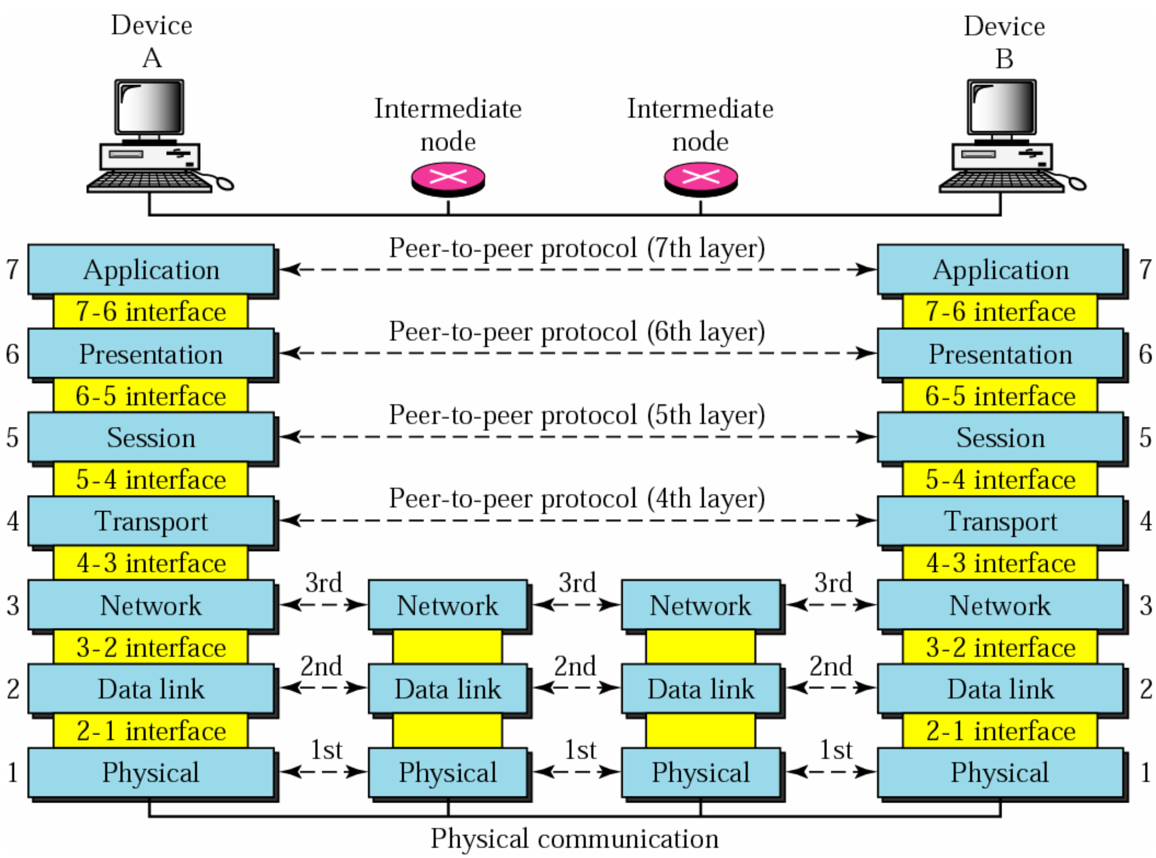
network support layers – deal with physical aspects of moving data from one device to another – across one link and across the whole network



OSI Model (cont.)

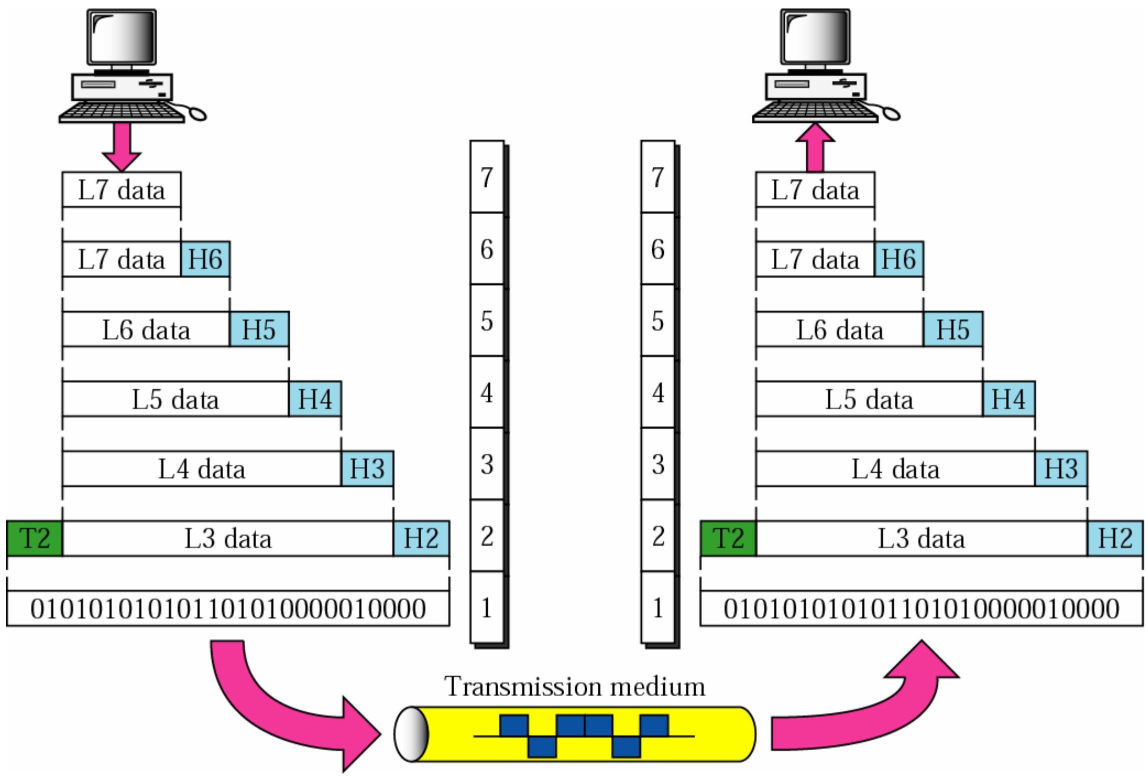
Peer-to-Peer Communication over 7 OSI Layers

- message moves down through layers on sending device, over intermediate nodes, to receiving station, and then back up through layers
- at intermediate nodes (routers), data is pulled <u>only</u> up to network layer, so that next hop could be determined



OSI Model (cont.)

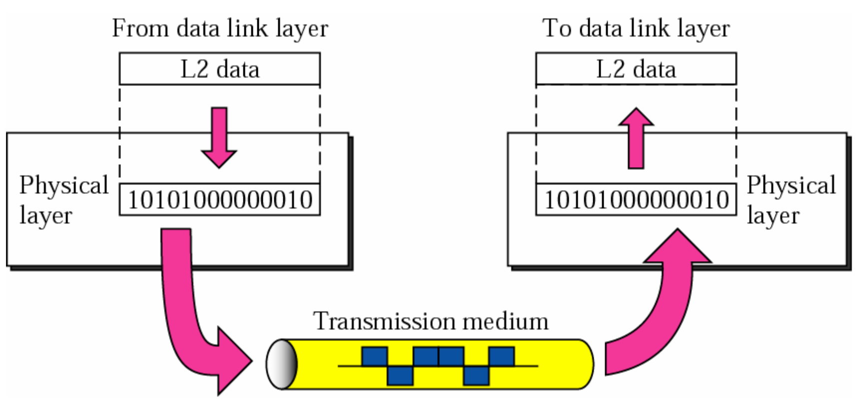
- **each layer in sending device adds its own information to message it receives from layer above it and passes whole package to layer just below it – reverse process occurs at receiving device**
- **when data reaches physical layer, it is changed into electromagnetic signal and sent along a physical link**



1. Physical Layer

- **coordinates transmission of bit-stream over physical medium, including**
  - **representation of bits: to be transmitted, bits must be encoded into signals – electrical or optical; P.L. defines type of encoding – how 0s & 1s are changed to signals (e.g. 1 = +1V, 0 = -1V)**
  - **bit length / data rate: P.L. defines how long a bit lasts and, accordingly, number of bits sent each second**

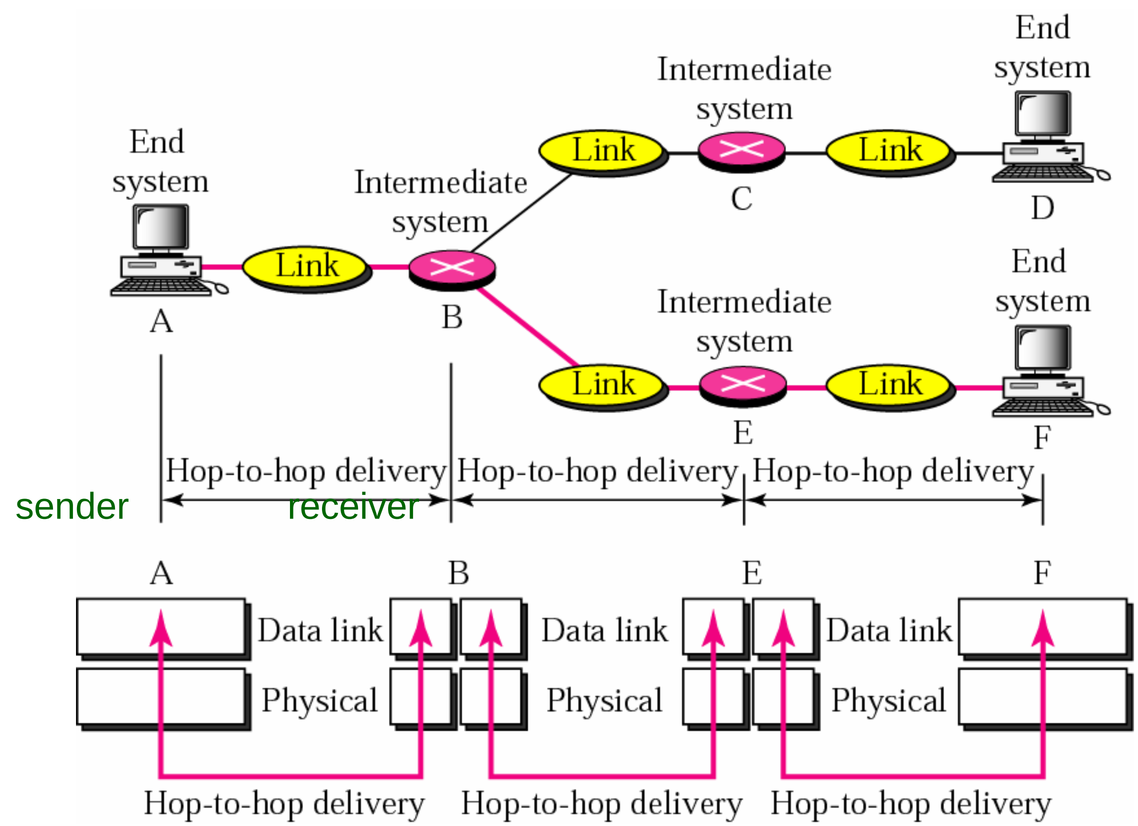
(different values for copper wire, coaxial cable, fiber-optics, …)



2. Data-Link Layer

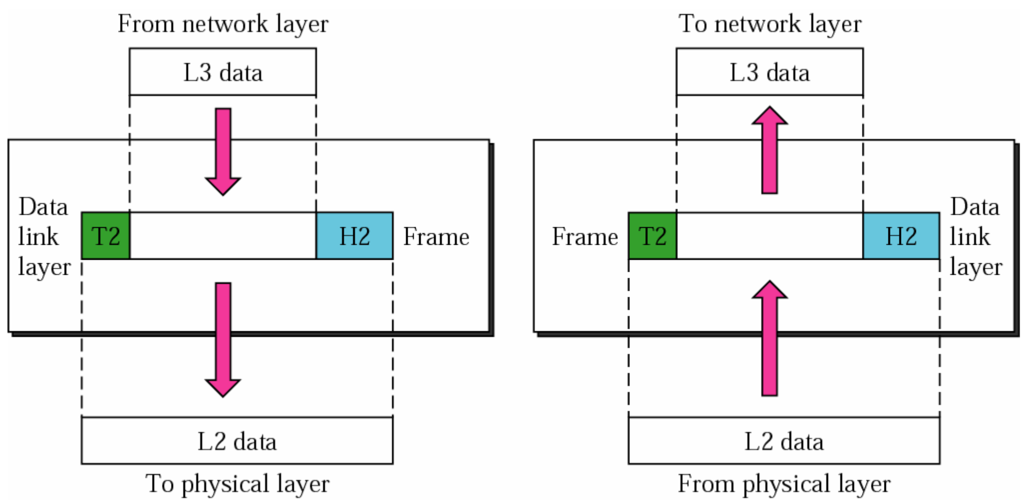
The data link layer transforms the physical layer, a raw stream of bits, to a <u>reliable link</u> between two devices <u>on the same network</u>.

It makes the physical layer appear error-free to the upper layer.



OSI Model: Data-Link Layer

- **framing**: The D.L.L divides the stream of bits received from the network layer into manageable data units called frames.
- **physical addressing**: The D.L.L adds a **header** to the frame to specify the NIC address of appropriate receiver on the other side (of wire).
- **error control**: The D.L.L adds reliability to the physical layer by adding a **trailer** with information necessary to detect/recover damaged or lost frames.
- **access control**: When 2 or more devices are connected to same link, the D.L.L determines which device has control over the link at any given time.
- **flow control**: If rate at which data are absorbed by receiver is less than sender's transmission rate, the D.L.L imposes a flow control over sender.

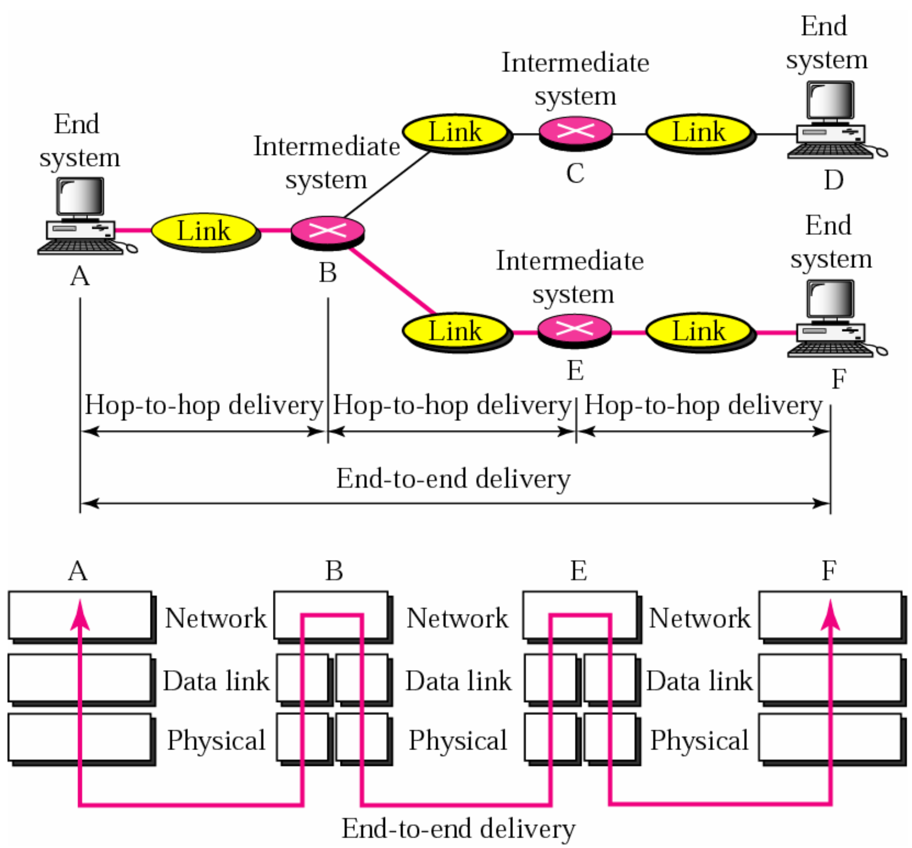


3. Network Layer

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

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

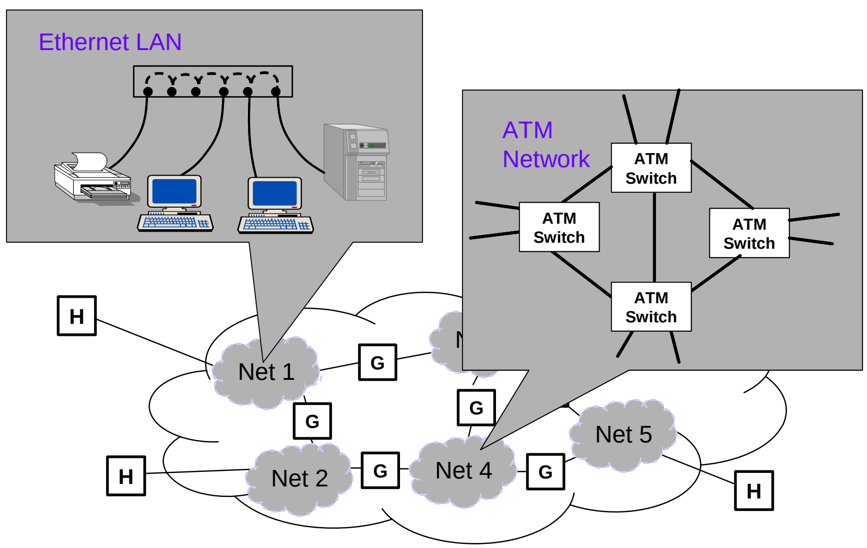
sender



receiver

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 <u>network</u>.**
- **routing**: **The N.L. provides the mechanism for routing/switching packets to their final destination, along the optimal path – across a large internetwork.**
- **fragmentation & 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, send each piece to the D.L.L, and then have pieces reassembled once they arrive at the N.L. on destination machine.**

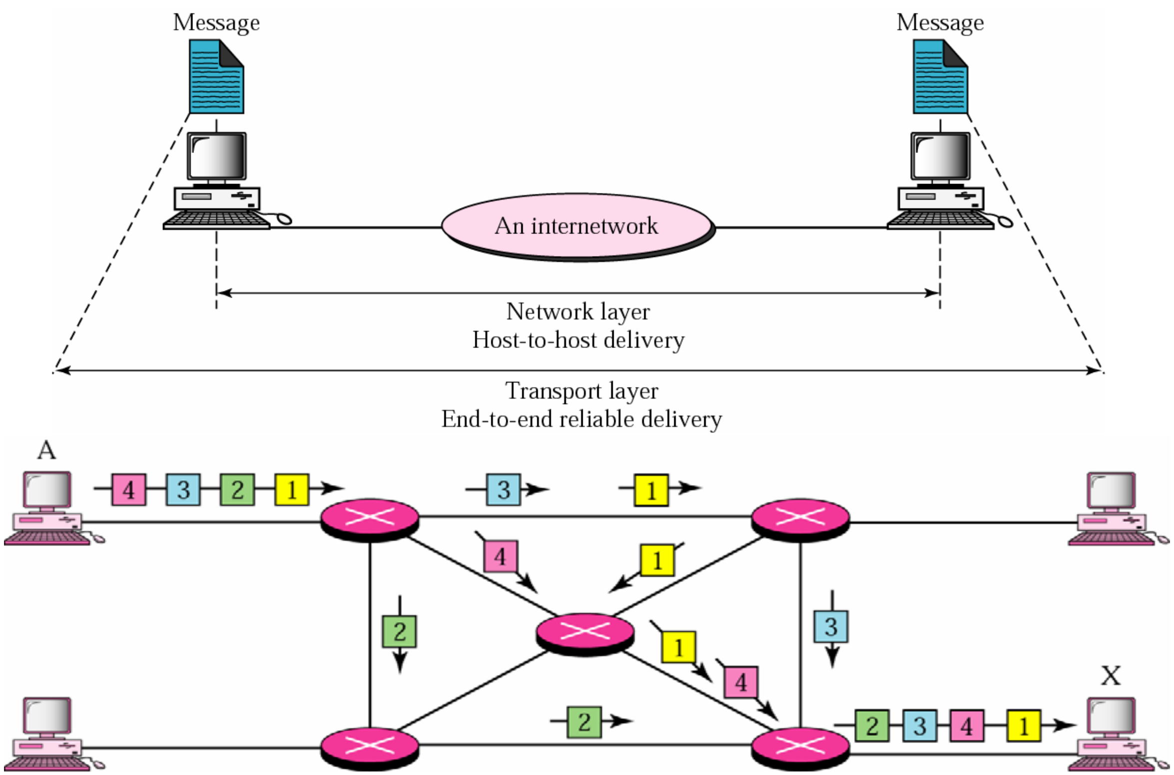


OSI Model: Transport Layer

4. Transport Layer

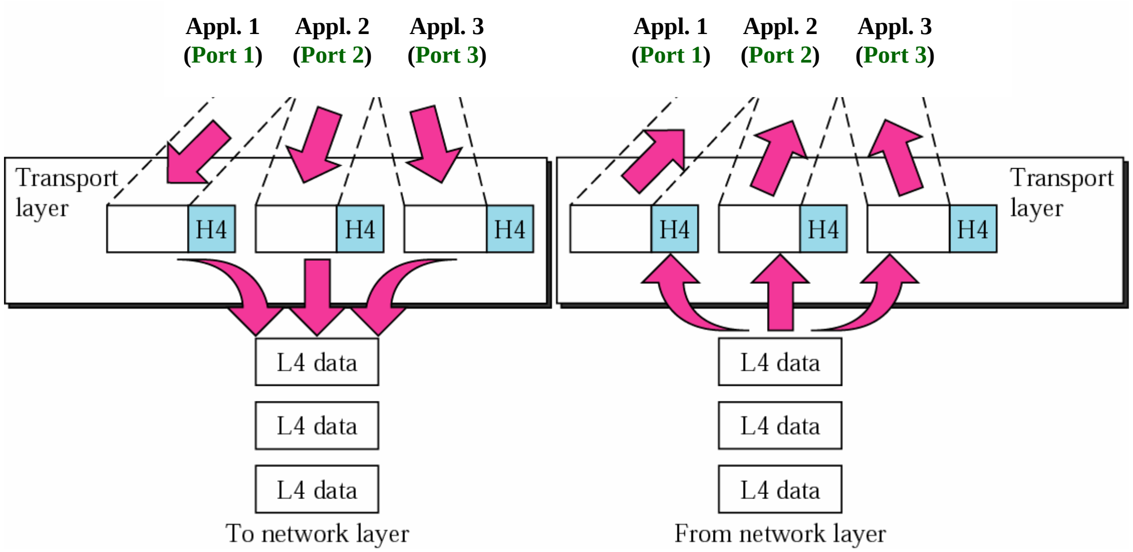
The transport layer is responsible for
process-to-process delivery of an entire message.

While network layer gets each packet to the correct computer, 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. Hence, 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 T.L. 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 T.L. 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.**



OSI Model: Application Layer

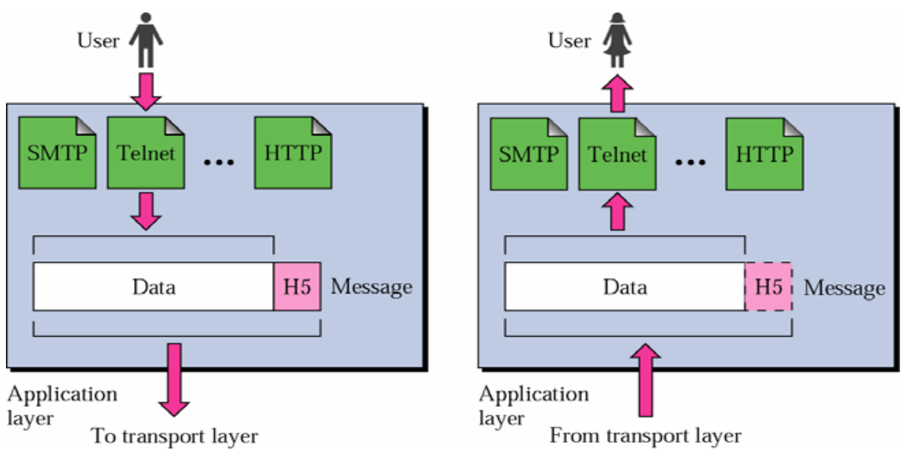
Application Layer (i.e. OSI Session + Presentation + Application Layer)

The application layer provides the actual service to the user.

We want to send a big file to a system that occasionally crashes.

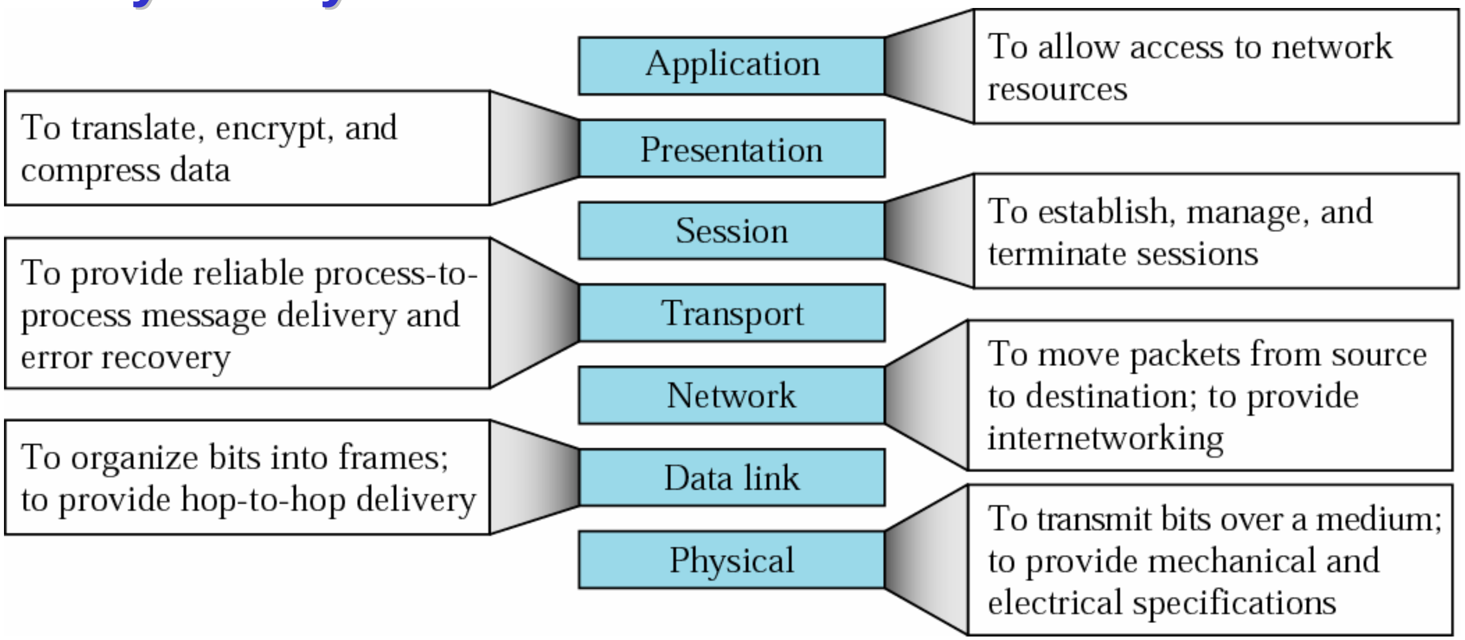
We want to send private data over third-party network.

We want to send multimedia/video data, but network capacity limited.



OSI Model: Summary

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 & presentation layer – provide some useful features; these can be easily provided in application layer**

OSI Model: Summary

Why did OSI Model Fail in Practice?

(1) Bad Timing

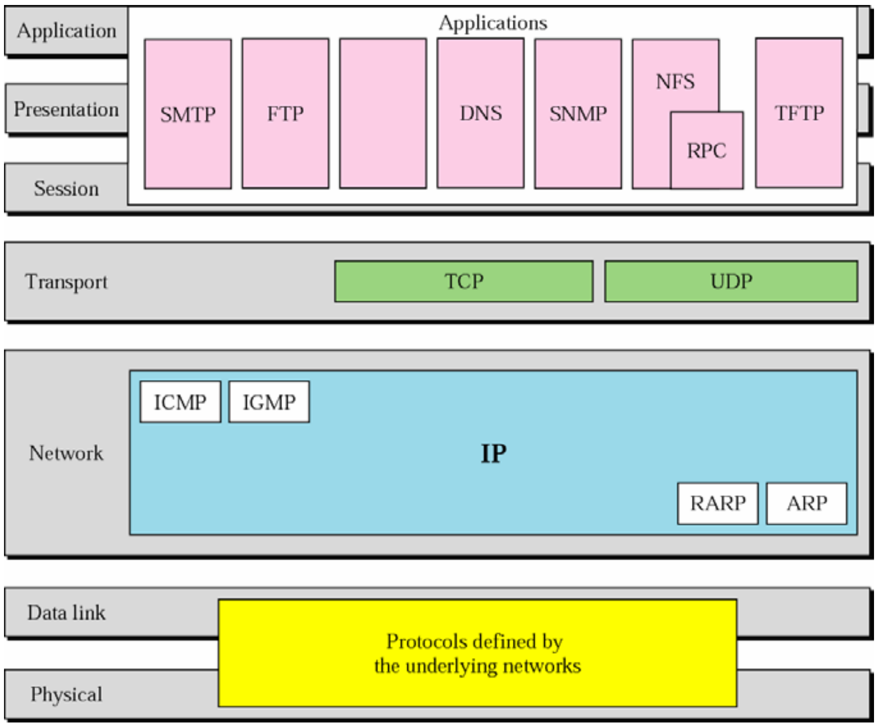
- **although essential elements of OSI model were in place quickly, final standard (model + protocols) was not published until 1984**
- **by the time it took to develop OSI protocol standards, TCP/IP network architecture emerged as an alternative for open system interconnection**
- **free distribution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

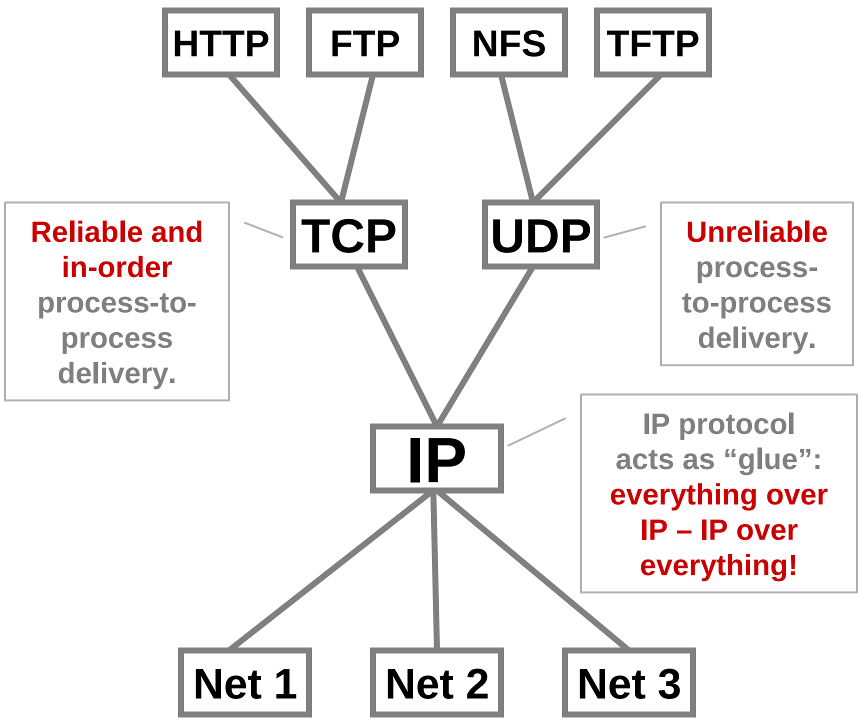
(2) Complexity and Inefficiency

- **7-layer OSI model was specified before there was much experience in designing large-scale OSI networks – some 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

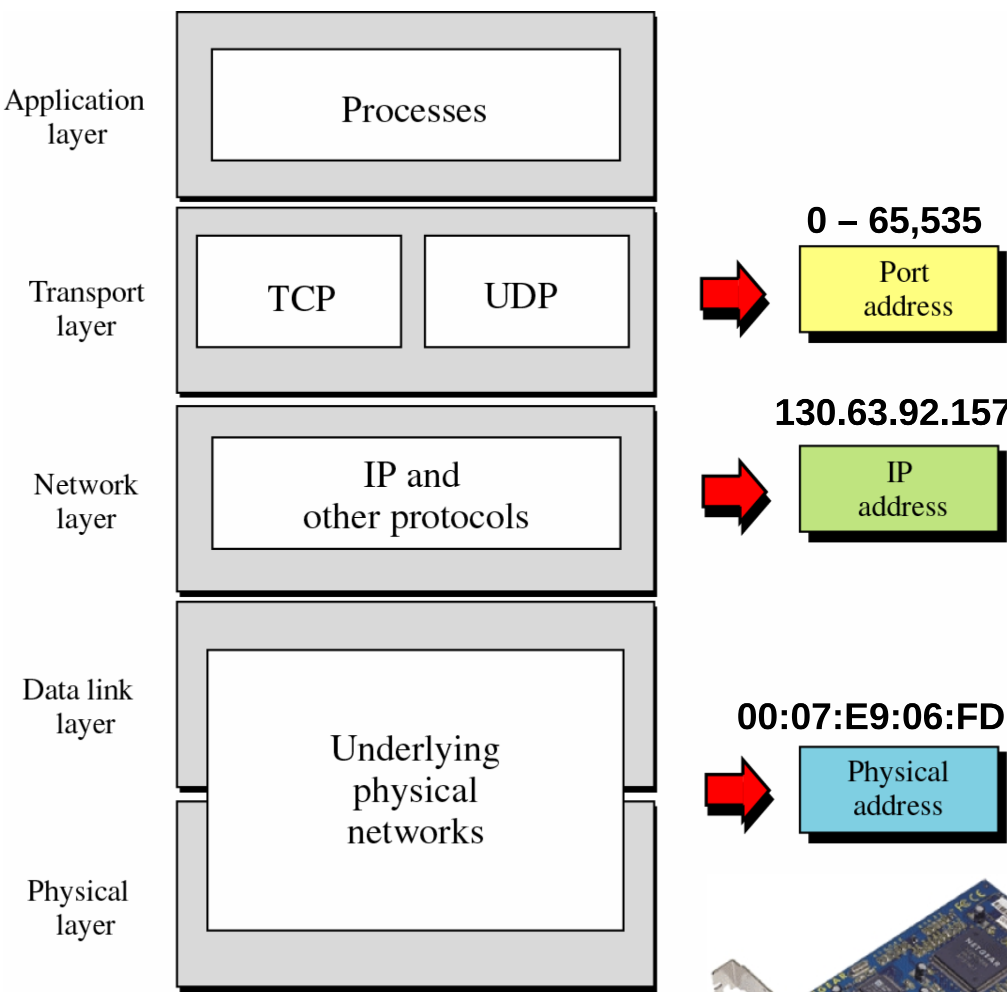




The operation of one single protocol at the network layer (IP protocol) over various networks provides independence from the underlying network technologies. IP over anything, anything over IP!

Internet Model (cont.)

Addresses in TCP/IP Model



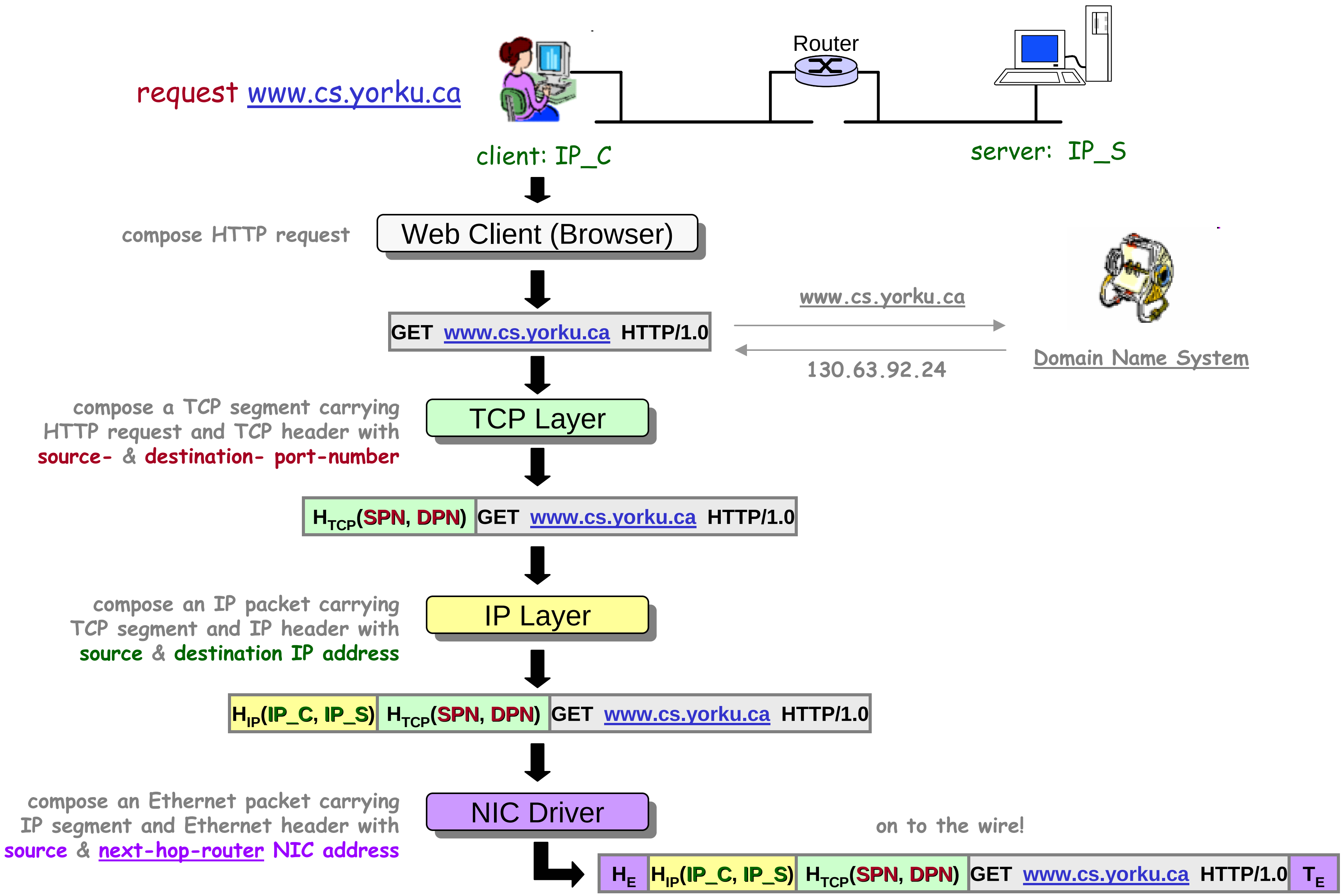
- **locally unique logical address** used to differentiate between applications sharing the same IP address

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

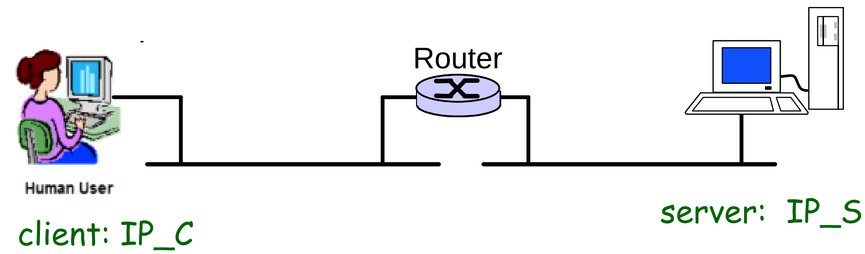
- **globally unique NIC address** used to located corresponding node on a LAN
- **each NIC on a subnetwork may have different manufactures ⇒ we cannot aggregate physical addresses in routing tables ⇒ large networks cannot use these addresses to identify hosts !**

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.)



Web Server

GET www.cs.yorku.ca HTTP/1.0

perform sequencing, error recovery, etc.
and pass HTTP segments reliably and <u>in-order</u>
on to specified port number

TCP Layer

H_{TCP}(SPN, DPN) | GET www.cs.yorku.ca HTTP/1.0

extract destination IP address and
check if it is meant for this computer –
if so, extract TCP segment and pass it on to the TCP layer,
otherwise forward the packet through appropriate outgoing link

IP Layer

H_{IP}(IP_C, IP_S) | H_{TCP}(SPN, DPN) | GET www.cs.yorku.ca HTTP/1.0

data-link layer processing
(header removal, error checking, etc.)

NIC Driver

from the wire!

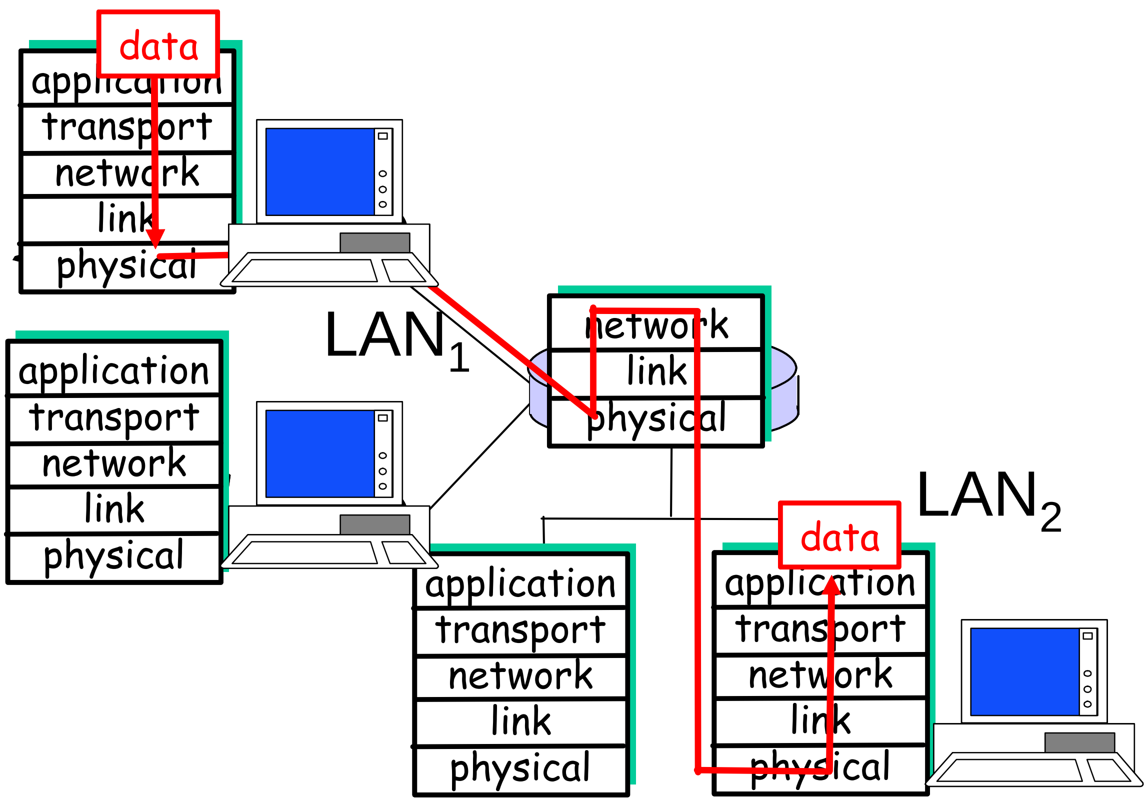
H_E | H_{IP}(IP_C, IP_S) | H_{TCP}(SPN, DPN) | GET www.cs.yorku.ca HTTP/1.0 | T_E

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

Bonus Question [layering – encapsulation]

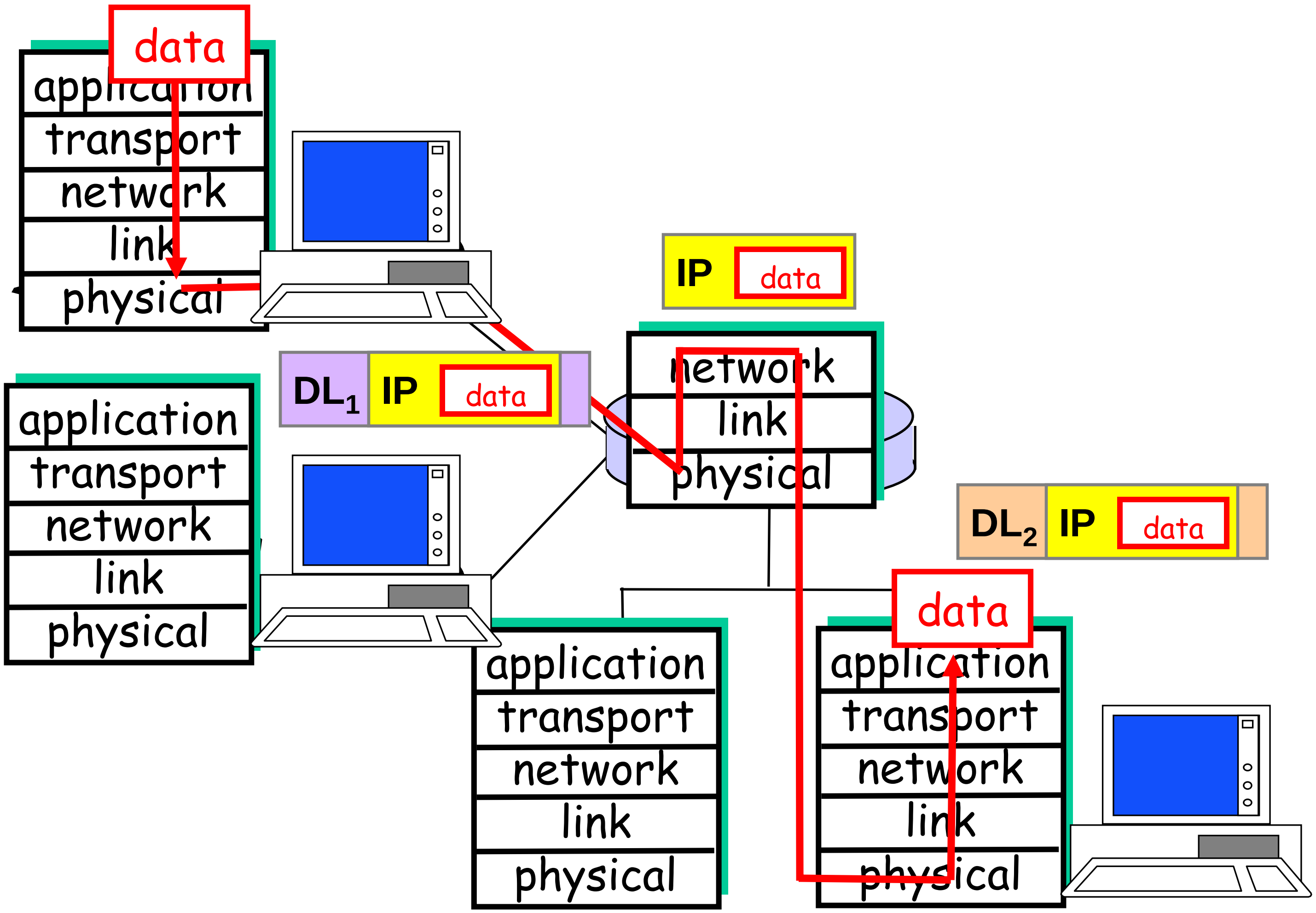
Assume two computers, situated on two distant LANs - with different data-link technologies, communicate with each other over the Internet.

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.)

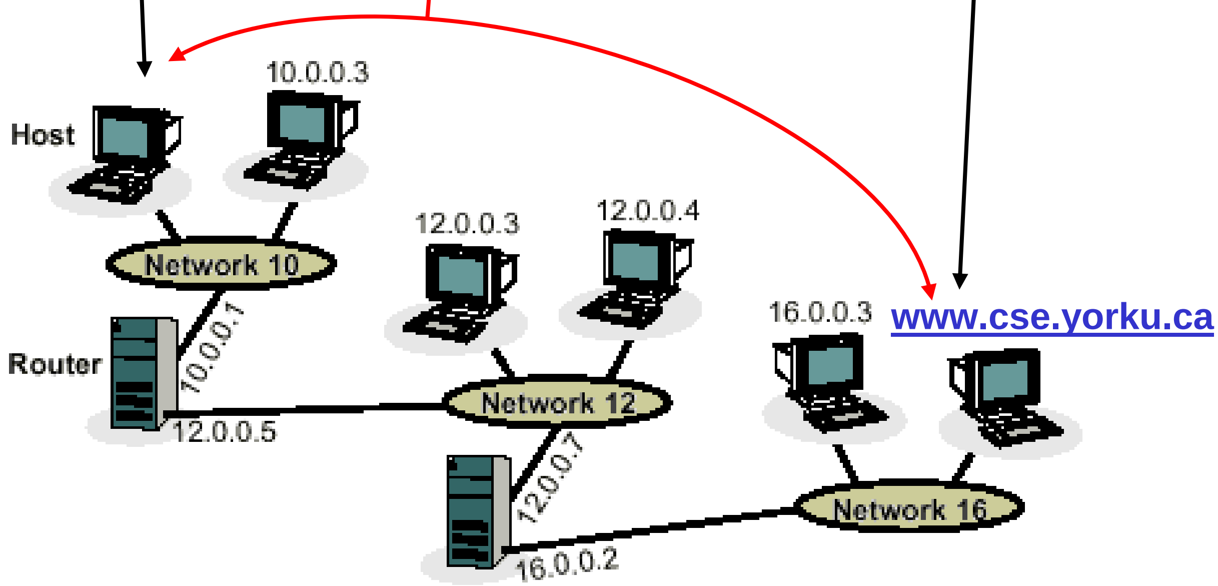


(Source: Kurose & Ross)

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

How to determine the number and identity of intermediate routers?

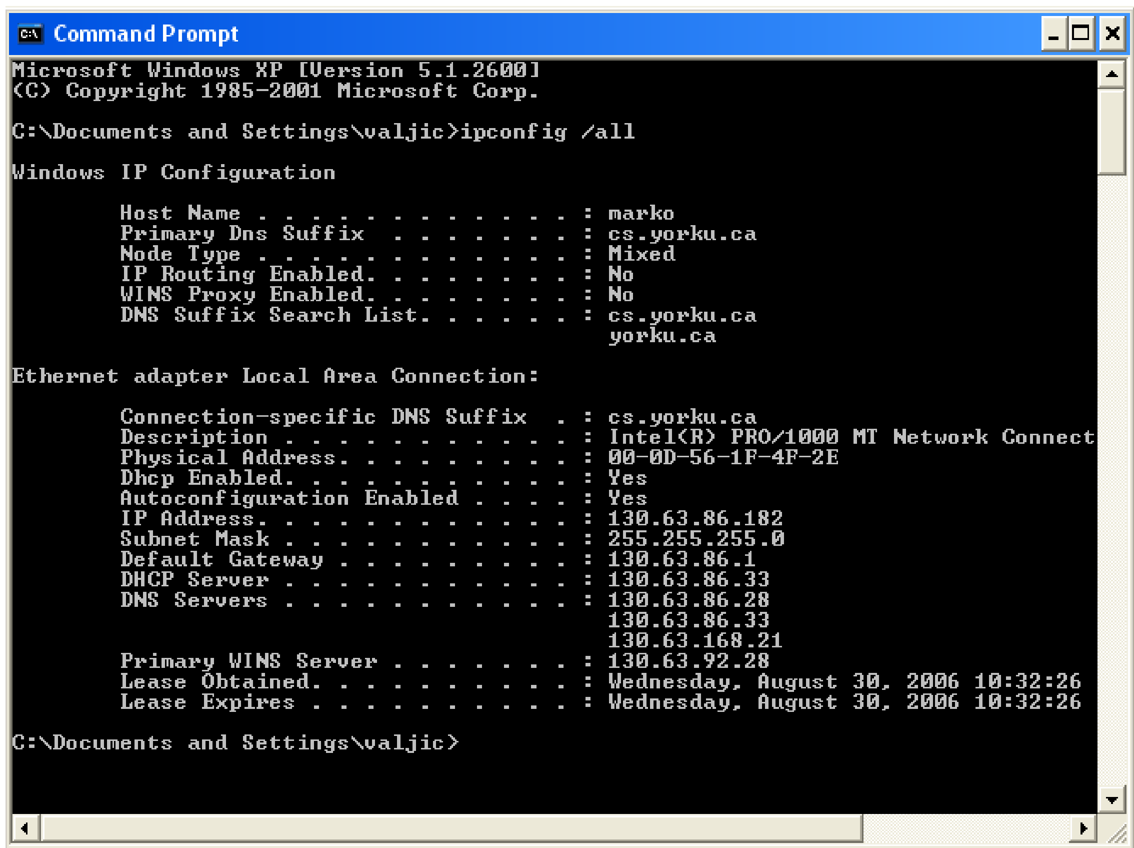
How to determine IP address of another remote machine?



IP Utilities

IPCONFIG – **Microsoft Windows OS tool; UNIX/Linux equivalents: ifconfig, ip addr**

- **in simplest form returns IP address, subnet mask, default gateway**
- ***ipconfig /all* – returns above and DNS hostname, physical address, DNS and DHCP Server addresses, etc.**



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 ⇒ additional delay!**

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 Origin – **UNIX utility, but nearly all platforms have something similar**

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

  tracert www.cs.yourku.ca

- **tracert & traceroute have different implementation !**

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 %
```

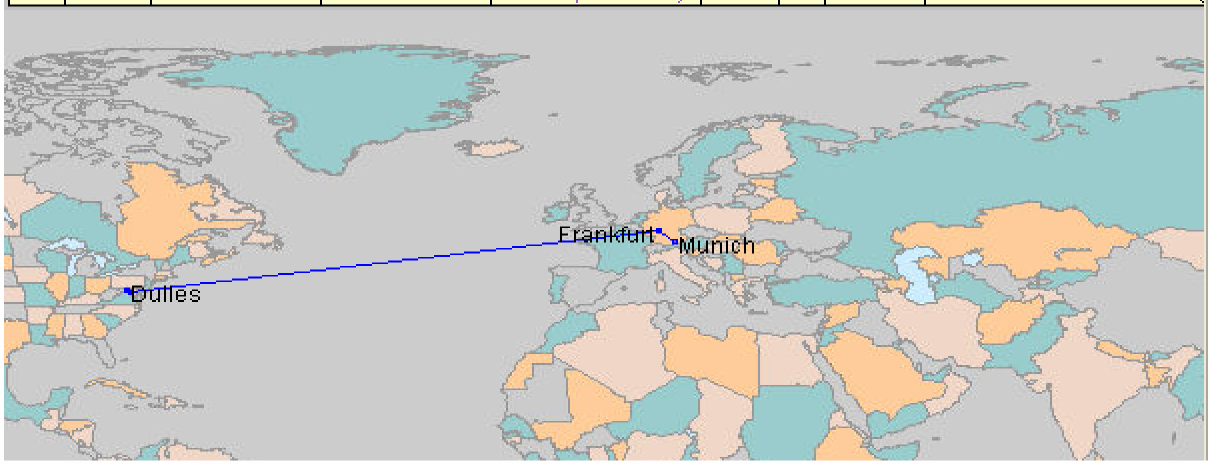
IP Utilities (cont.)

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-6
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

File Edit View Favorites Tools Help

Back Search Favorites Media

Address http://books.google.ca/books?id=YKWfuGGSmXMC&pg=PA302&lpg=PA302&dq=traceroute+shorter+RTT+time&source Go Links

Contents Page 302 Link Feedback

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

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

Done Internet

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.