



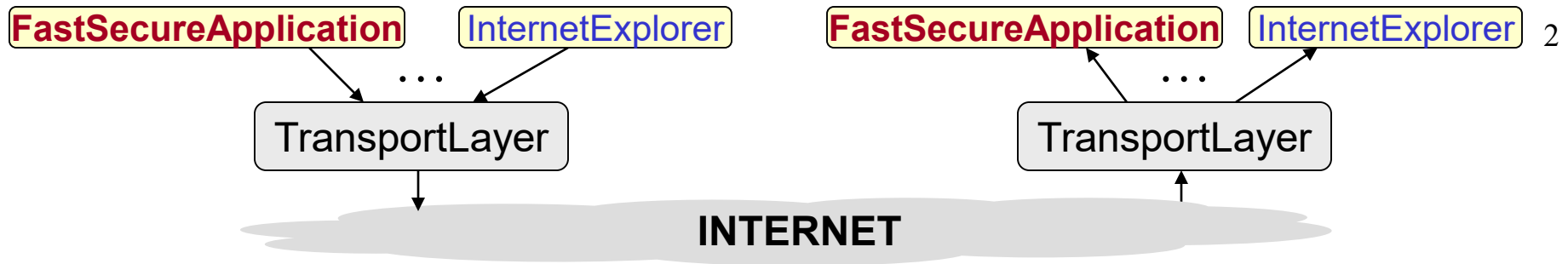
```
public void main(String[] args) {
    ...
    FastSecureApplication.send(file);
    ...
}
```

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

...

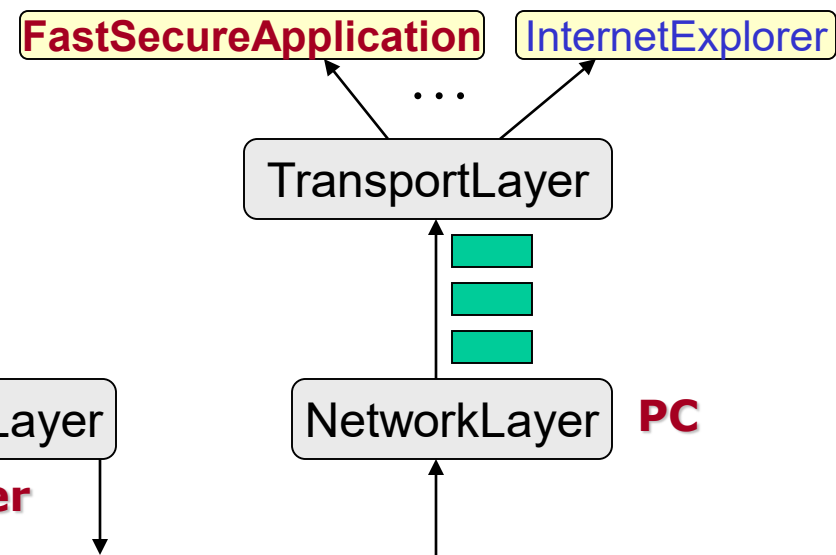
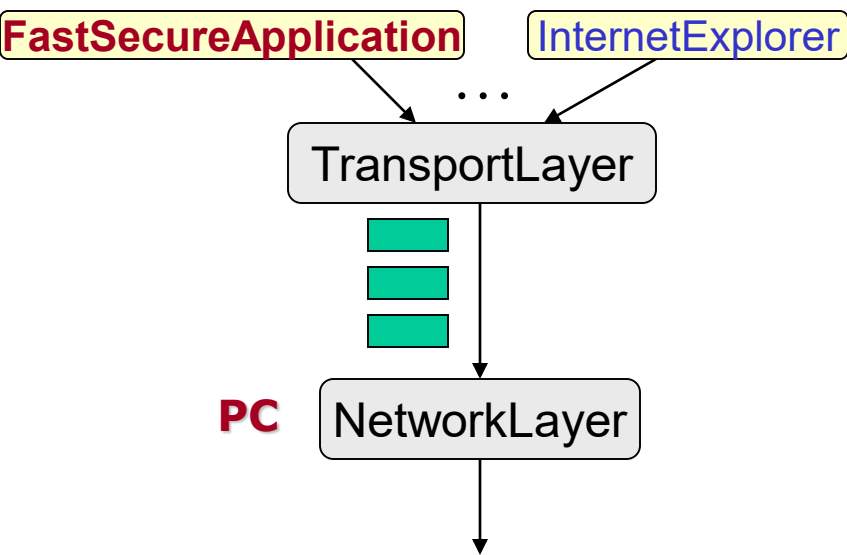
```
static void send(Object file) {
    Object compressedFile = compress(file);
    Object encryptedFile = encrypt(compressedFile);
    TransportLayer.send(encryptedFile);
    ...
}
```

}



```

class TransportLayer {
    int pickPortNumber();
    Packets[] reassemble(Object file);
    Packets[] addHeaders(Packets[] filePackets, int portNmb);
    static void send(Object applicationLayerFile);
    Object assemble();
    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]);
        }
        ...
    }
}
  
```



```

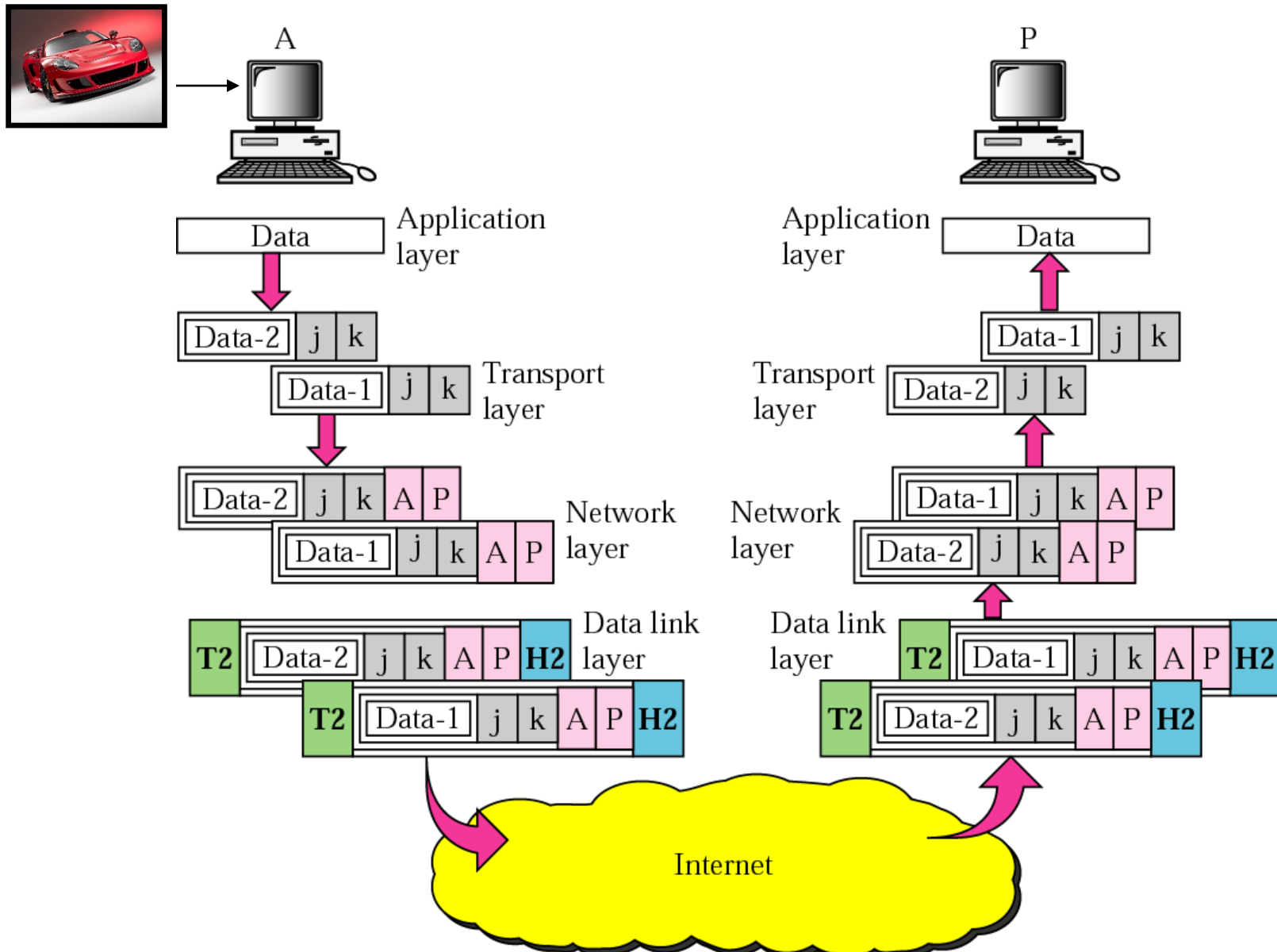
class NetworkLayer {
    ...
    static void send(Object transportLayerPacket) {
        Packet IPPacket = addHeader(transportLayerPacket, IPAddress, etc.);
        DataLinkLayer.send(IPPacket);
        ....
    }

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

```

Example

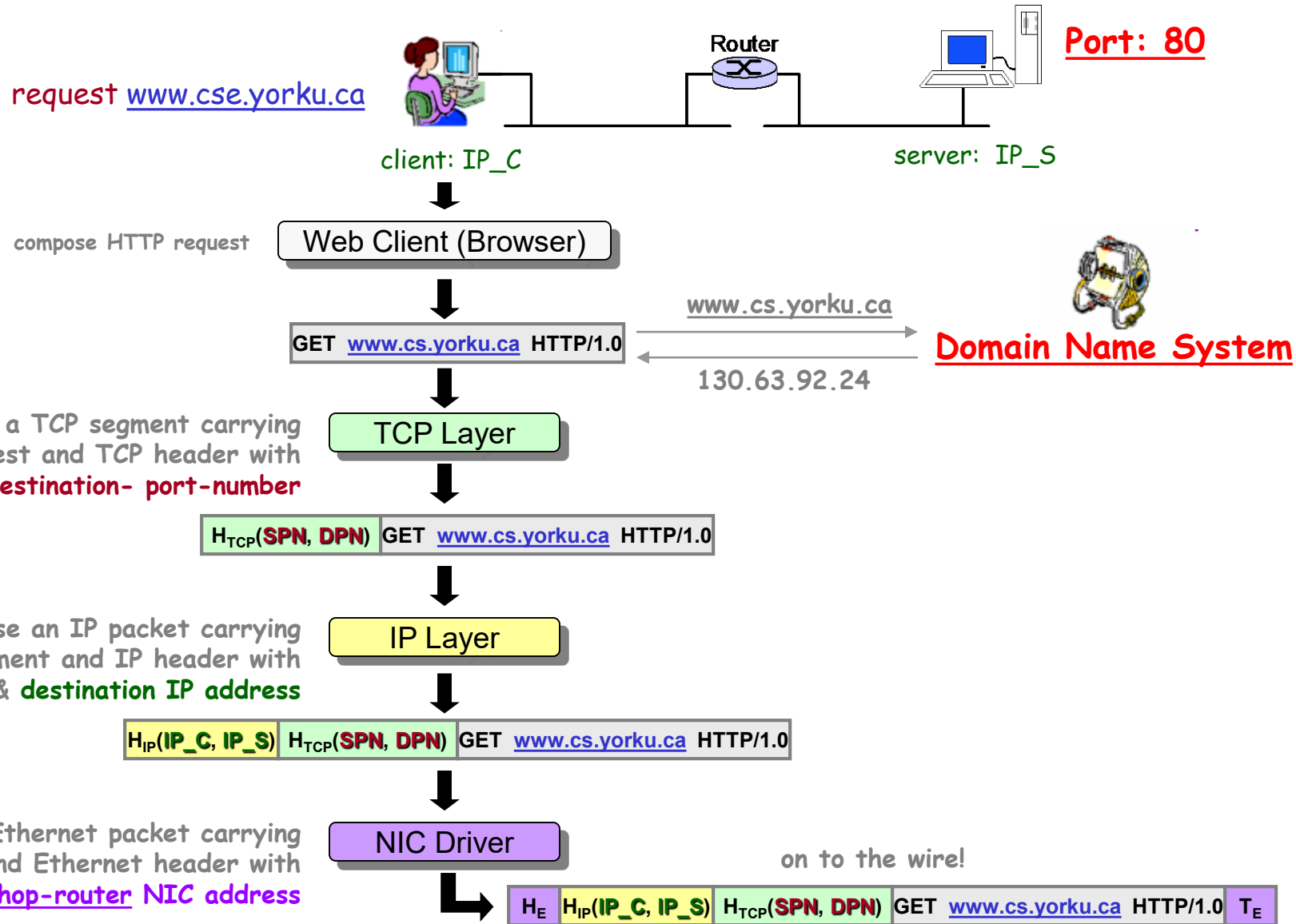
Assume we want to exchange an image between computers A and P. The image, after being compressed, occupies 1000 bytes. The maximum packet size is 500 bytes.



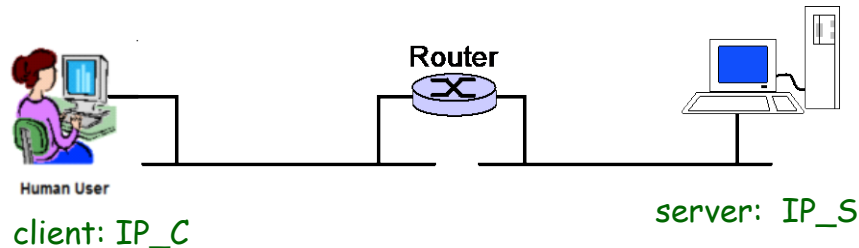
TCP/IP Protocol: How the Layers Work Together

5

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



TCP/IP Protocol: How the Layers Work Together (cont.) 6



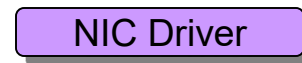
perform sequencing, error recovery, etc.
and pass HTTP segments reliably and in-order
on to specified port number



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



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



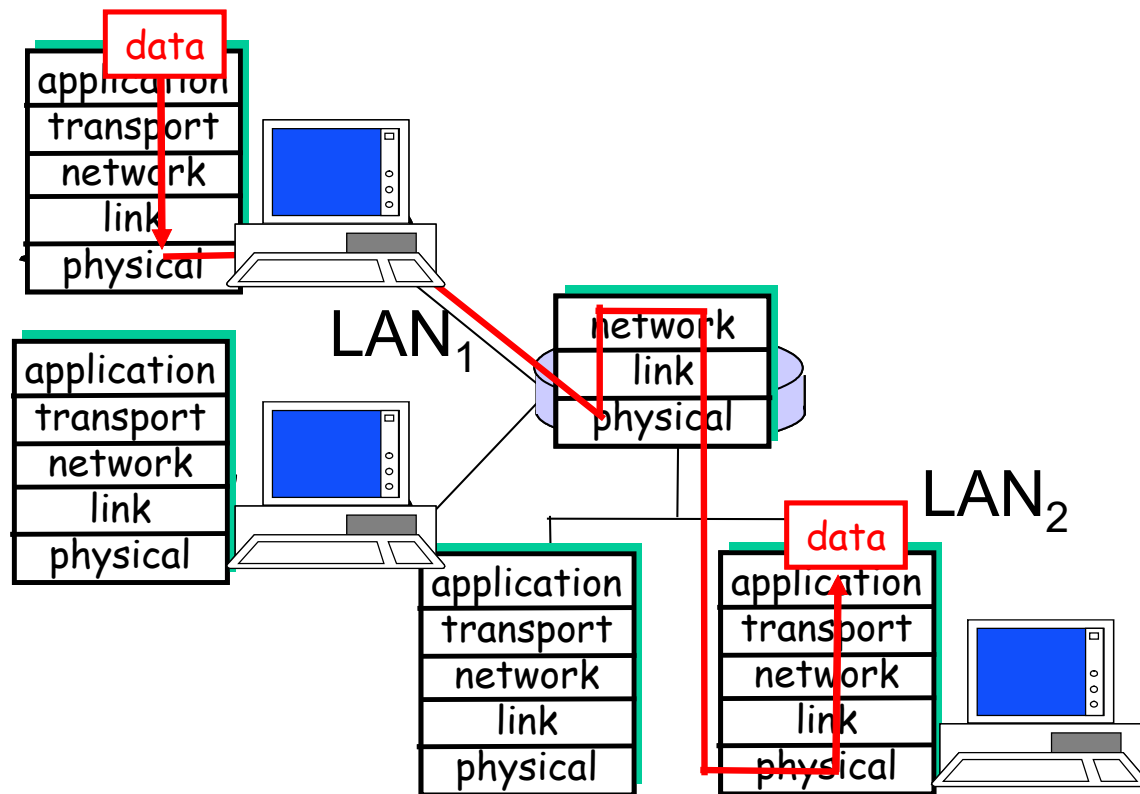
from the wire!



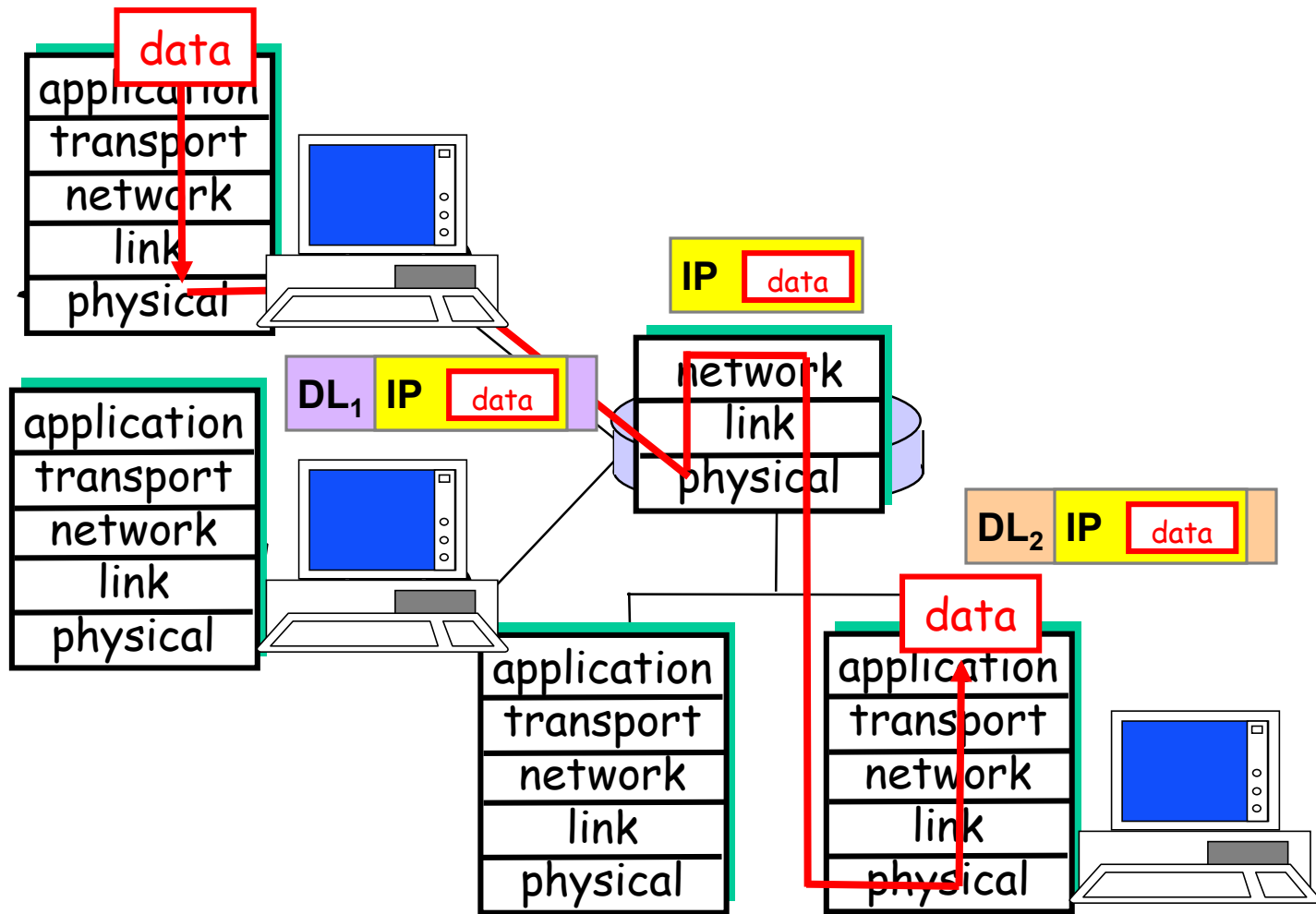
Bonus Question [layering – encapsulation]

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?



TCP/IP Protocol: How the Layers Work Together (cont.) 8

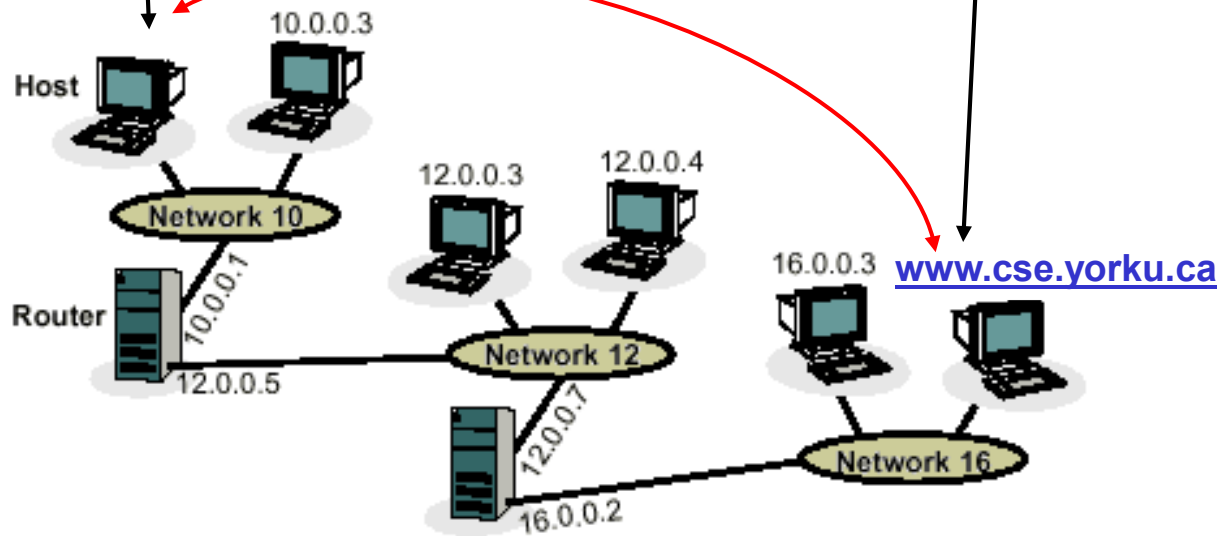


(Source: Kurose & Ross)

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

```

C:\Documents and Settings\valjic>ipconfig /all

Windows IP Configuration

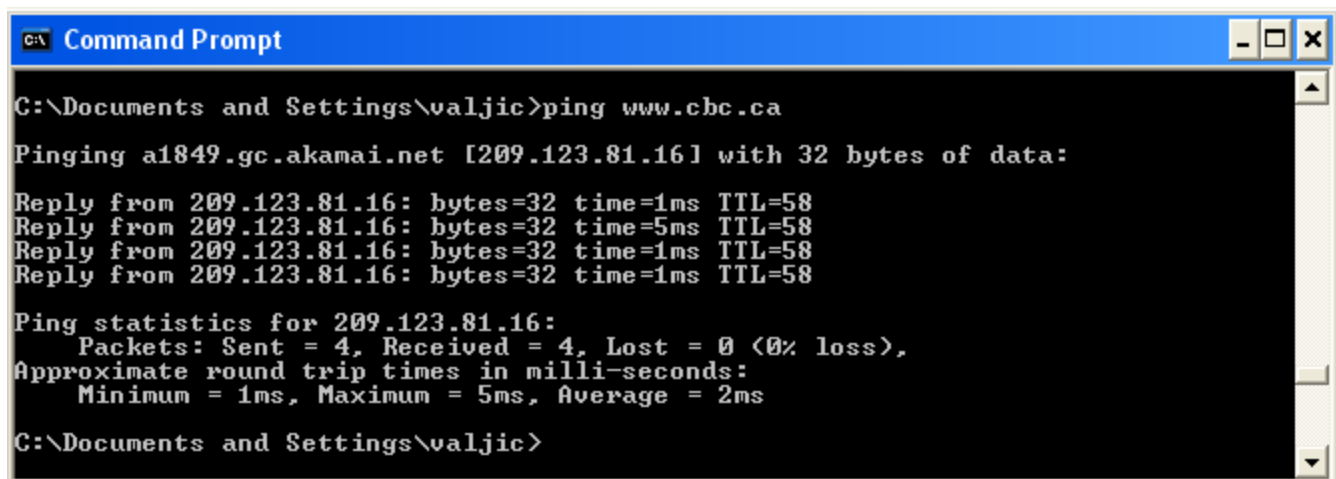
    Host Name . . . . . : marko
    Primary Dns Suffix . . . . . : cs.yorku.ca
    Node Type . . . . . : Mixed
    IP Routing Enabled. . . . . : No
    WINS Proxy Enabled. . . . . : No
    DNS Suffix Search List. . . . . : cs.yorku.ca
                                        yorku.ca

Ethernet adapter Local Area Connection:

    Connection-specific DNS Suffix . : cs.yorku.ca
    Description . . . . . : Intel(R) PRO/1000 MT Network Connect
    Physical Address. . . . . : 00-0D-56-1F-4F-2E
    Dhcp Enabled. . . . . : Yes
    Autoconfiguration Enabled . . . . : Yes
    IP Address. . . . . : 130.63.86.182
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 130.63.86.1
    DHCP Server . . . . . : 130.63.86.33
    DNS Servers . . . . . : 130.63.86.28
                          130.63.86.33
                          130.63.168.21
    Primary WINS Server . . . . . : 130.63.92.28
    Lease Obtained. . . . . : Wednesday, August 30, 2006 10:32:26
    Lease Expires . . . . . : Wednesday, August 30, 2006 10:32:26

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

- PING** – standard troubleshooting tool (available on most OS) 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



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

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

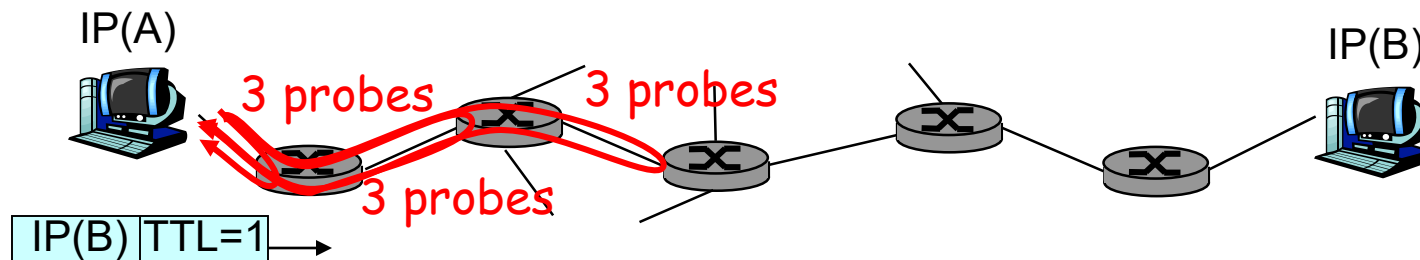
Reply from 209.123.81.16: bytes=32 time=1ms TTL=58
Reply from 209.123.81.16: bytes=32 time=5ms TTL=58
Reply from 209.123.81.16: bytes=32 time=1ms TTL=58
Reply from 209.123.81.16: bytes=32 time=1ms TTL=58

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>
```

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** \Rightarrow source actually sends $3*N$ packets to destination (N=number of hops)



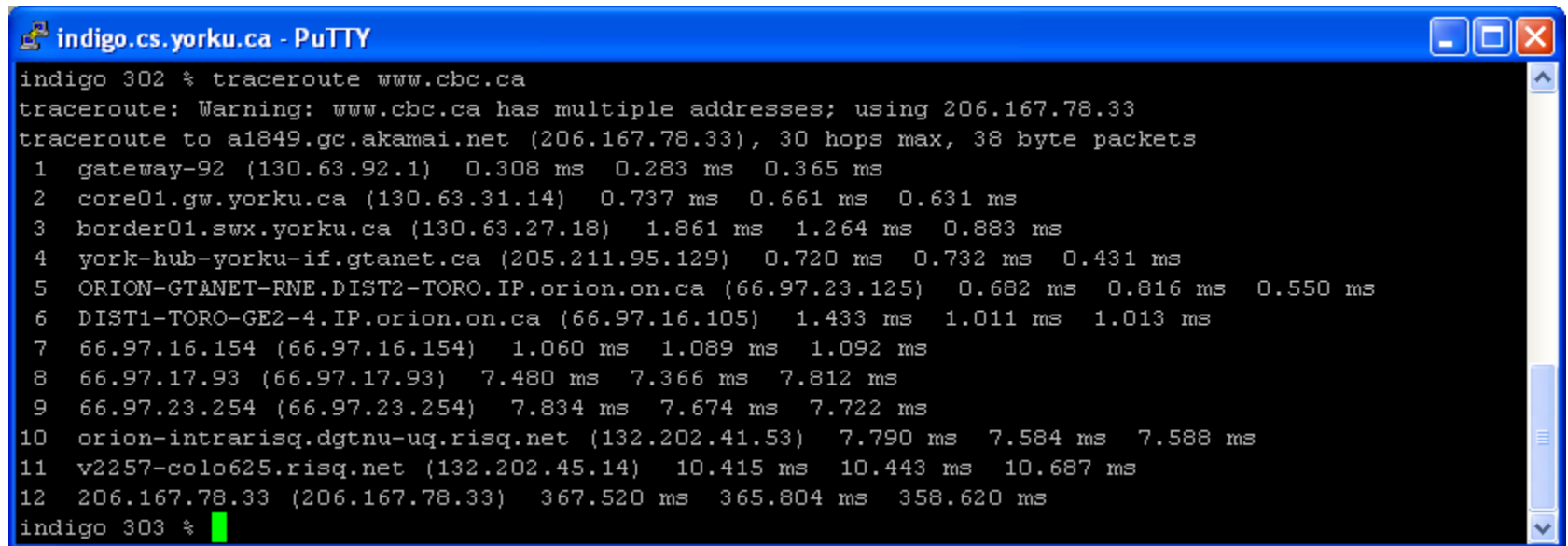
- Traceroute Origin** – traceroute is a UNIX utility, but nearly all platforms have something similar
- Windows includes a traceroute utility called **tracert** – you can run tracert from MS-Dos Window, by entering tracert followed by domain name, e.g.
`tracert www.cs.yourku.ca`
 - **tracert implementation 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.

Example [traceroute www.bbc.co.uk]



The screenshot shows a PuTTY terminal window titled "indigo.cs.yorku.ca - PuTTY". The terminal displays the execution of the command "traceroute www.cbc.ca". The output shows a warning that www.cbc.ca has multiple addresses and that 206.167.78.33 is being used. The traceroute proceeds through 12 hops, showing the IP address of each hop and the round-trip time in milliseconds. The final hop is 206.167.78.33.

```
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 %
```

Open Visual Trace Route for Internet Performance: <https://visualtraceroute.net/>

