Web Technology 2015

Lecture 3. The Internet: TCP/IP (part 2)

Staas de Jong

Notes beforehand...

- Guest: welcome!

- The study material:
    - read it now
    - e.g. before or after each lecture
    - ask questions

    ⇒ less work, more clarity around exam time

>

Notes beforehand...

- Web Technology Reports:
  - This Thursday: *final* deadline for updated proposals.
  - This Friday: teams are assigned subjects!

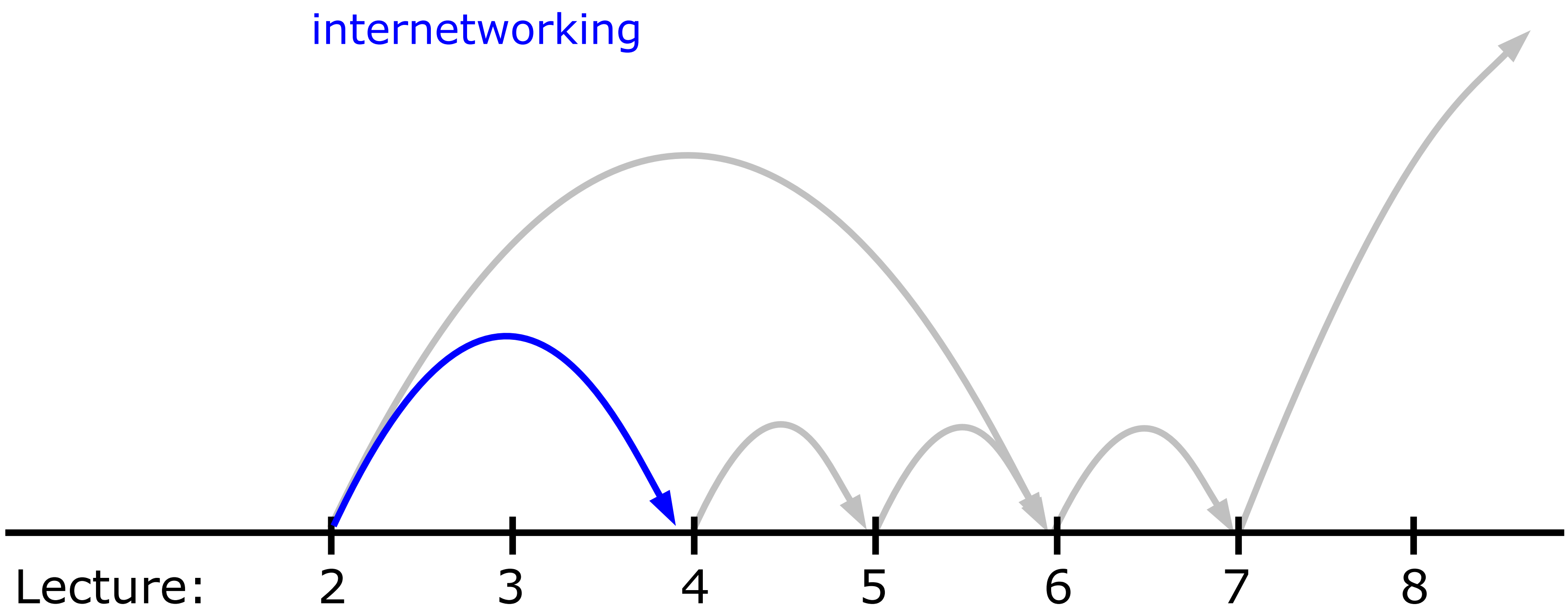
- New suggestions:
  - Since last Monday, “mobile-friendliness” *directly affects Google search ranking*. >

    ⇒ Paper subject: Standards for mobile-friendly web pages.

  - Since last Monday, Android Wear supports *Wi-Fi on smartwatch hardware*.

    ⇒ Paper subject: Android Wear: Smartwatches as a web technology.

Today: closing the *internetworking* arc



⇒ diving into the TCP/IP stack: IP, TCP, UDP, DNS.

IP: functionality

- IP: the *Internet Protocol*

- Three main components of its functionality:

  - IP provides *addressing*:
    - gives each machine a unique ID;
    - maps these onto hardware addresses.

  - IP provides *routing*:
    - combines smaller networks to form one huge network;
    - lets data traverse networks one by one.

  - IP provides *fragmentation*:
    - where needed, divides packets into smaller parts when they are forwarded.

IP: characteristics

- IP is *connectionless:*
  - There is no notion of a connection starting and ending.
  - Data is sent in packets called *IP datagrams*.
- IP is *unreliable:*
  - Packets can get delayed, lost, duplicated, or corrupted.
  - Corrupted packets are simply discarded without further notice!

IP: addressing

- Classical IP address: 32-bit number, identifying a machine connected to the Internet.

  - E.g.: *10000001001101000000011000000000*.

    ↑ Replacing binary notation by decimal notation: *2167670272*.

- This number is divided into bytes, groups of 8 bits:

  - E.g.: *10000001 00110100 00000110 00000000*.

    or *129 . 52 . 6 . 0*

    ...in dotted decimal notation.

There is structure: *different* bytes have *different* meanings...

IP addresses: structure and meaning

- Higher-order bytes are used to indicate a *network*.

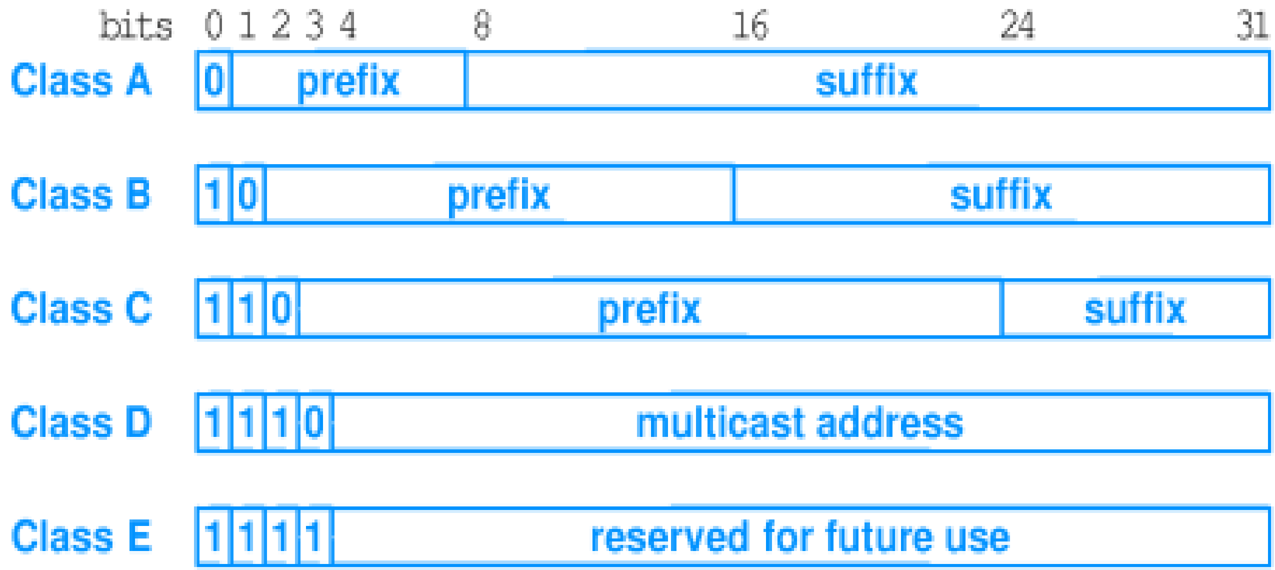
    ↑ The “prefix” part.

- Lower-order bytes are then used to indicate a *host* – *a specific machine* – on that network.

    ↑ The “suffix” part.

- When you encounter an IP address, its very first (highest-order) bits tell you:
    - which bytes form the prefix
    - which bytes form the suffix...

IP addresses: structure and meaning

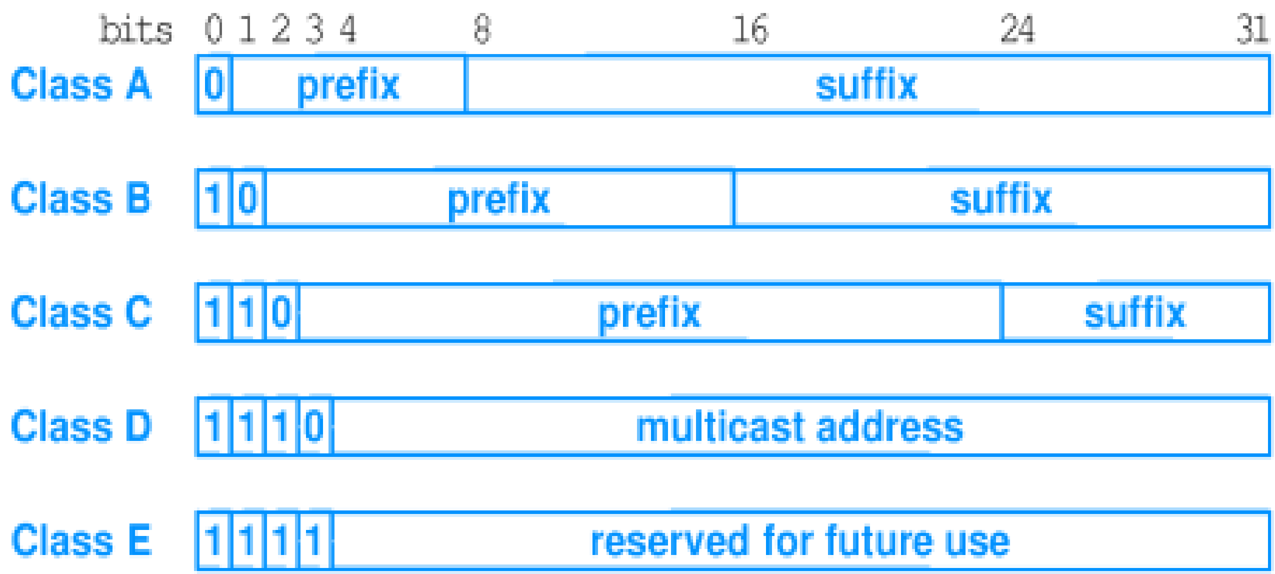


The different prefix/suffix combinations ↑ are called IP address classes.

You can identify the *class* looking at the *first* byte in dotted decimal notation. →

Class	Range of Values
A	0 through 127
B	128 through 191
C	192 through 223
D	224 through 239
E	240 through 255

IP addresses: structure and meaning



↑ Because of IP address classes:

- real-world networks of very different sizes
- could all be combined into a single internet
- in a time when memory was very expensive! ↓

Address Class	Bits In Prefix	Maximum Number of Networks	Bits In Suffix	Maximum Number Of Hosts Per Network
A	7	128	24	16777216
B	14	16384	16	65536
C	21	2097152	8	256

IP addresses: concrete, special examples

- 127.0.0.1 : localhost.

    ↑ Your own machine
    even when not connected to the rest of the Internet.

    - Useful for prototypes & offline testing.

- 192.168.0.x : reserved for private networks.

IP - scalability issue: *classes waste address space*

- **Problem:** The size limits of classes A-C led to large wastes of address space.

Address Class	Bits In Prefix	Maximum Number of Networks	Bits In Suffix	Maximum Number Of Hosts Per Network
A	7	128	24	16777216
B	14	16384	16	65536
C	21	2097152	8	256

Example:

- Imagine a company with 300 host machines.
- They would need a class B address.
- ⇒ And so waste 65536 – 300 = 65236 addresses!

- **Solution:** various techniques to save IP addresses.

⇒ *Takeaway:* today, classes are not used as originally intended.

IP - scalability issue: *address space itself too small*

- **Problem:** A classical, IP version 4 (IPv4) address is a 32-bit number.

  ⇒ This gives us 4,294,967,296 unique addresses ...only!

  Example: think of the world population.

- **Solution:** IP version 6 (IPv6) uses 128-bit addresses.

  ⇒ This gives us ±50.000.000.000.000.000.000.000.000.000.000 unique addresses ...for *each* of the ±6.5 billion people alive today.

NAT: *an intermediate fix giving real-world surprises*

- **Network Address Translation (NAT):**
  - *On the local network:* Each machine has its own IP address, within a specific range.
  - *To the rest of the Internet:* The local IP range is represented by a **single**, quite different IP address!

  ⇒ (+) This saves address space.

  ⇒ (−) In a project, your IP might not be what you think it is!

- **Example:** home WLAN with router to Internet Service Provider.

My laptop OS says ↓

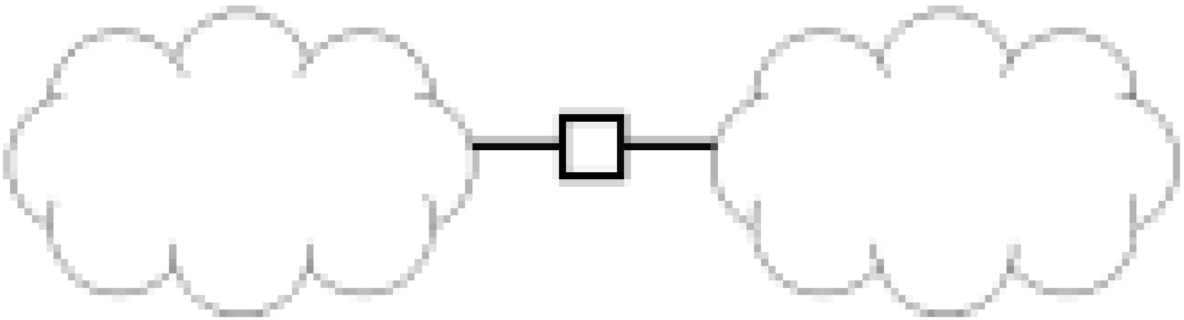
IPv4	
IP Address:	192.168.0.112
Broadcast Address:	192.168.0.255
Subnet Mask:	255.255.255.0

www.whatsmyip.org says ↓

Your IP Address is

109.163.234.2

IP: Internetwork routing

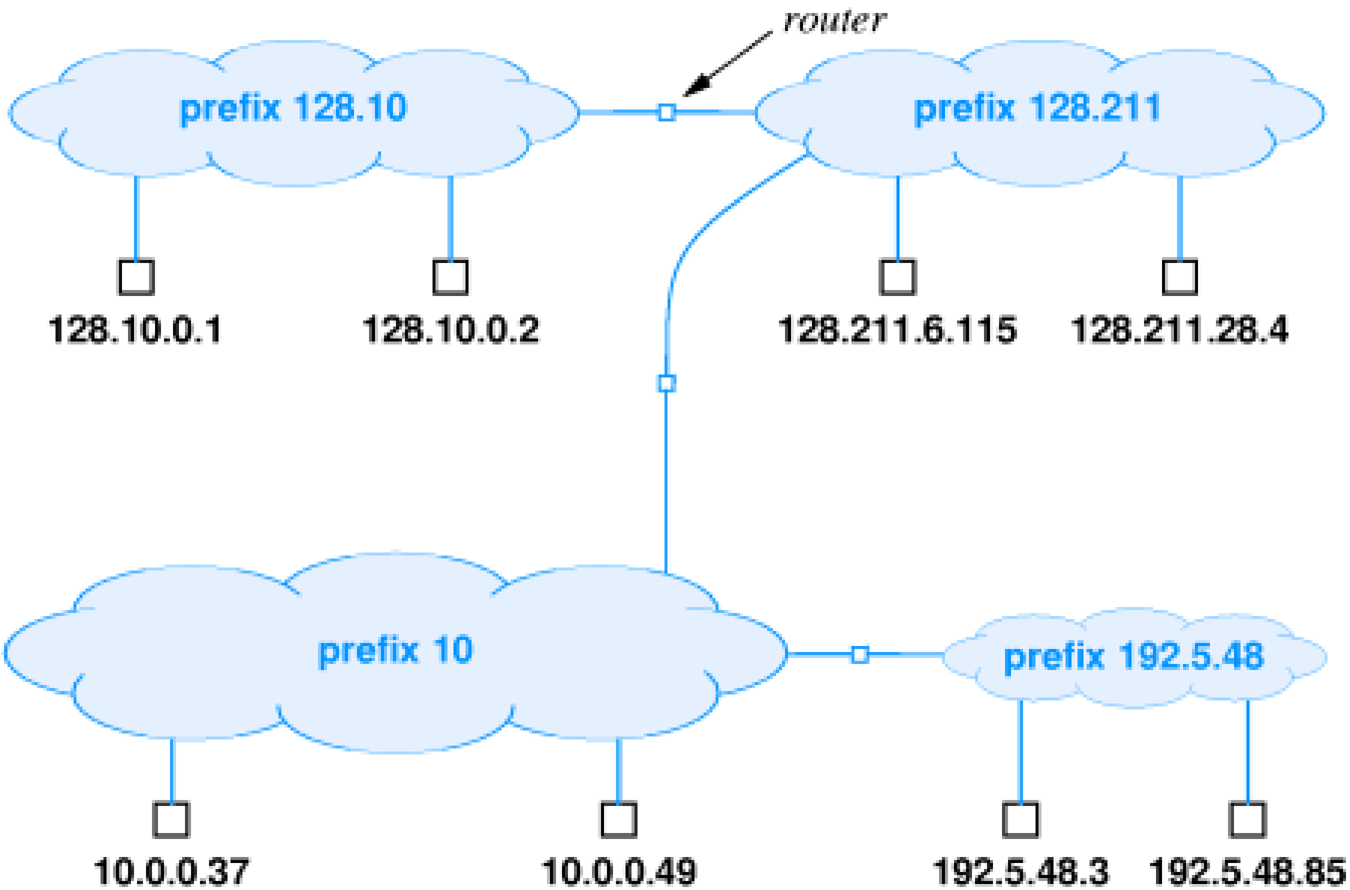


- A router device connects two or more physical networks.
- A router is part of each physical network that it connects.
- Data packets move from one network to the next network through a router.

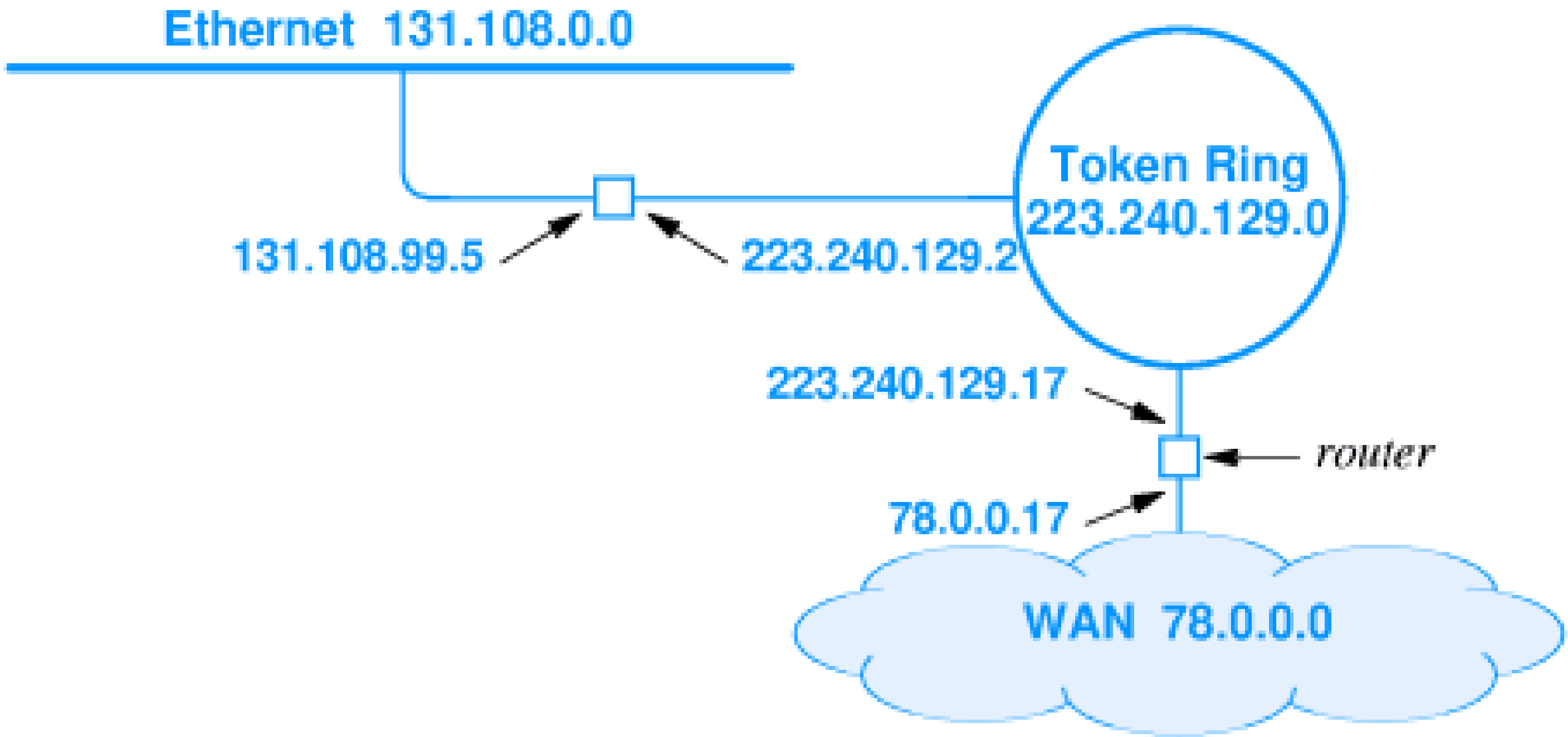
⇒ Routers are responsible for forwarding IP traffic to the right destination.

...are the devices that enable the Internet.

IP routing: *an example topology*

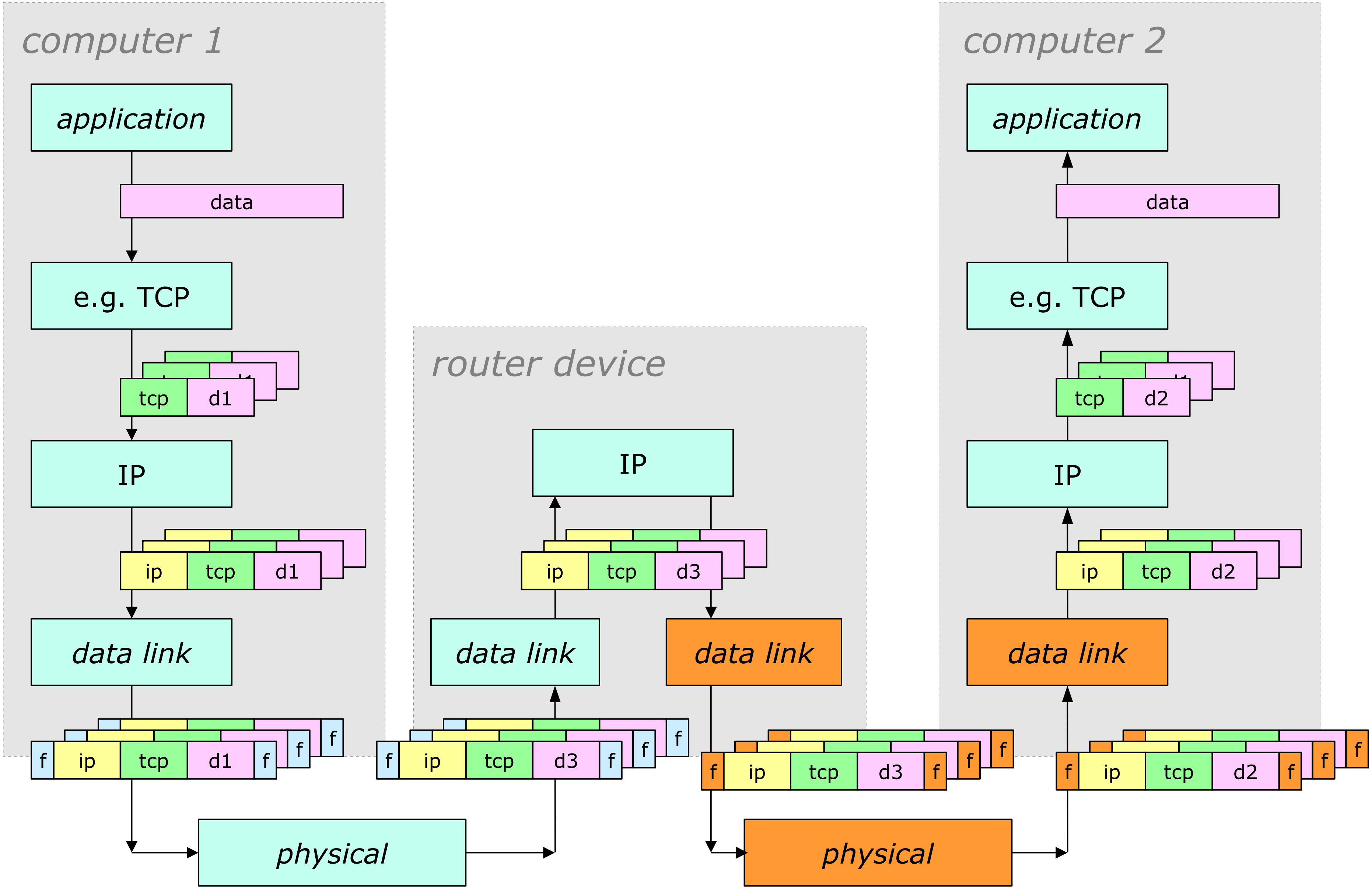


IP routing: *two example routers, in more detail*

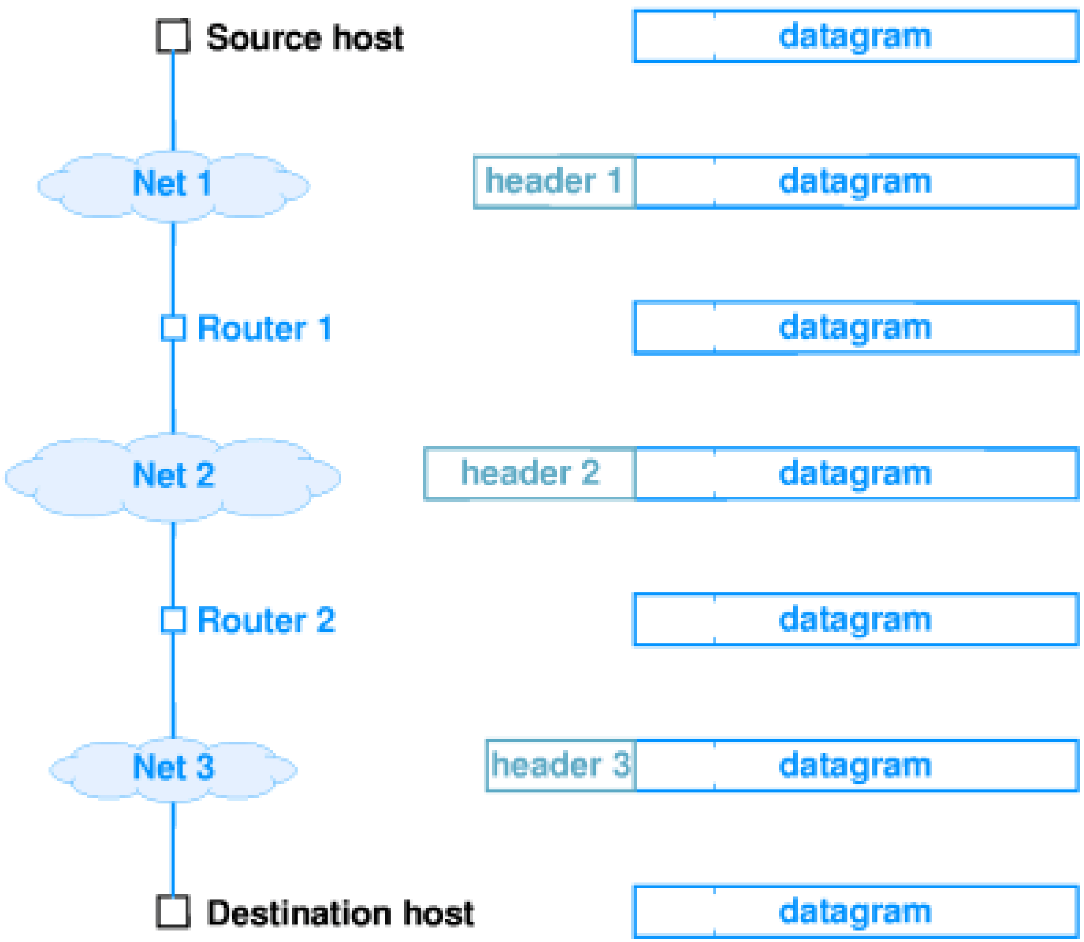


↑ The routers *themselves* **have an IP address,**
in each network they connect to.

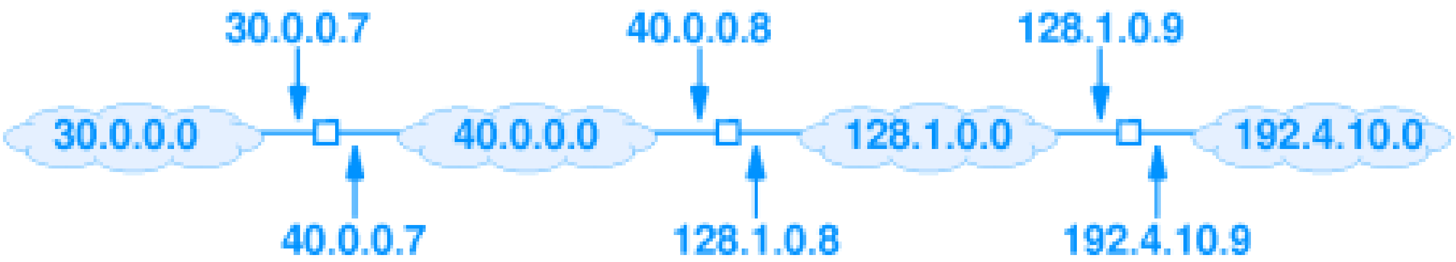
Routers and the Internet protocol stack



Hop-by-hop routing



Hop-by-hop routing: via *routing tables*



(a)

Destination	Mask	Next Hop
30.0.0.0	255.0.0.0	40.0.0.7
40.0.0.0	255.0.0.0	deliver direct
128.1.0.0	255.255.0.0	deliver direct
192.4.10.0	255.255.255.0	128.1.0.9

(b)

↑ Table *(b)* shows the routing table for the middle router in diagram *(a)*.

- *Destination host is on router's networks?* ⇒ Deliver the packet directly.
- *Not?* ⇒ Deliver the packet to a router closer to the destination host..


BREAK!