How to make an internet: the TCP/IP protocol stack

To create a single virtual network out of multiple physical ones, we will need:

- Universal addressing
  to mask differences in underlying addressing schemes.
- Routing packets across networks
  to forward packets to neighbouring networks.
- Handling different packet sizes
  to mask differences in underlying maximum packet sizes.

All this is handled by the **Internet Protocol (IP)**.

A *protocol* is a set of rules that describe how to communicate so that you and others know what to expect.

The need for protocol *stacks*

Issue: To realize internetworking, higher-level use & lower-level implementation dictate many *other* requirements *as well*.

Q: Can you name a few?

↑ Together, this translates into a plethora of technical challenges.

How to approach this complexity?

⇒ Divide, and conquer.

The need for protocol *stacks*

- A *protocol stack* is a partitioning of overall communication functionality into so-called *layers.*

- Each layer executes a specific *type* of communication,

    - while using the communication functionality of the layer *directly below it*

    - in order to provide communication functionality to the layer *directly above it*.

- The layers of a protocol stack
    - may be implemented in hard- or in software;
    - will often reside within the same physical device.

IP exists within the *TCP/IP protocol stack*

- *Application layer*
  e.g. HTTP | *solves:* application-specific communication.

- *Transport layer*
  e.g. TCP | *solves:* providing reliable data transport.

- *Network layer* or *Internet layer*
  e.g. IP | *solves:* combining networks into one virtual network.

- *Data link layer* or *Networks interface layer*
  e.g. Ethernet, Wi-Fi | *solves:* connecting computers.

- *Physical layer*
  e.g. copper wire, optical fibre, radio...

(See p.49 of Tanenbaum for rationale for using this ↑ model.)

Memory aid for TCP/IP stack: ***Physical DaNeTrAp***

Ap*plication layer*

Tr*ansport layer*

Ne*twork layer*

Da*ta link layer*

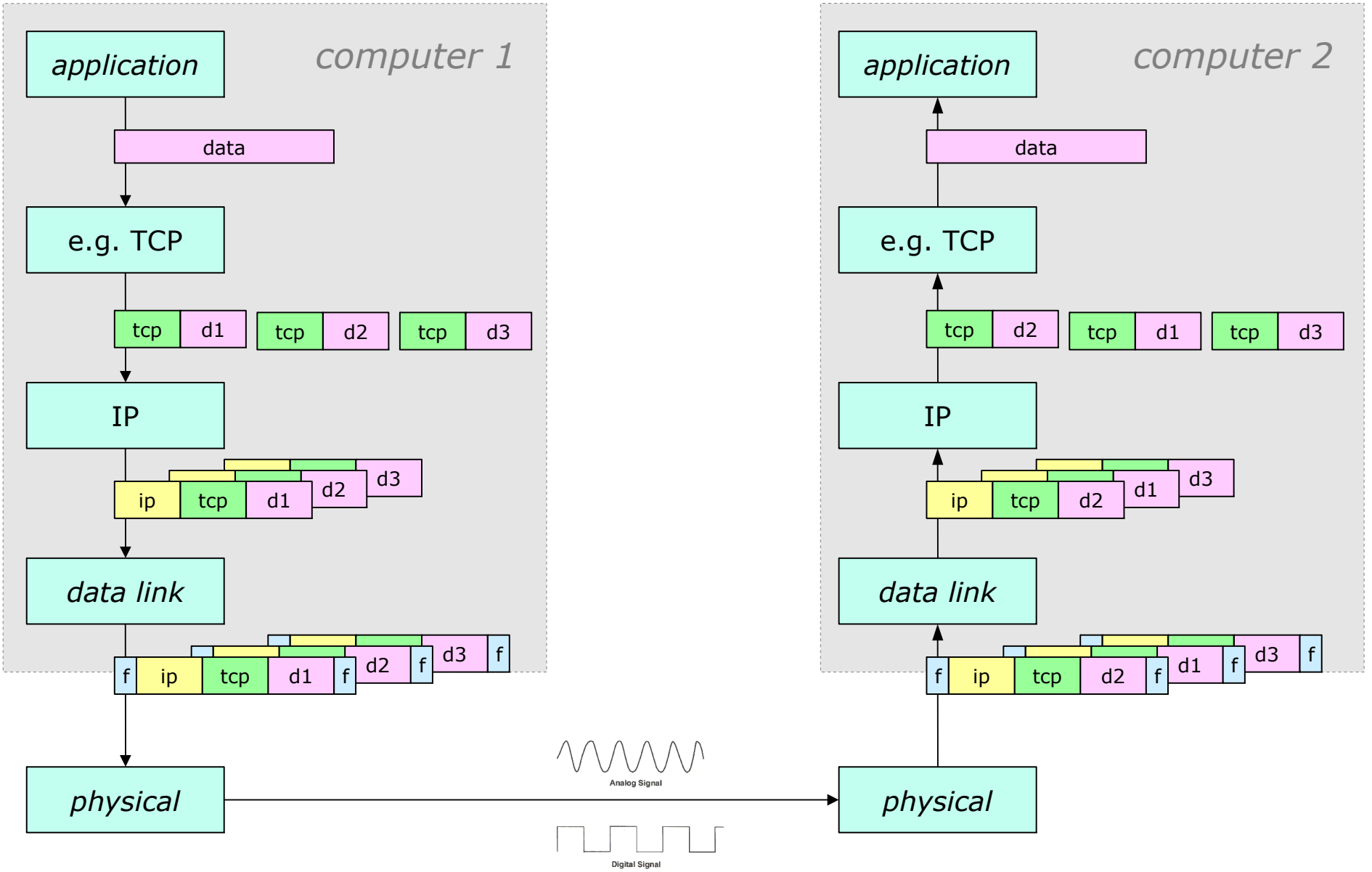
Physical *layer*

What could be a “physical Danetrap”?

How does one physically catch a Scandinavian??

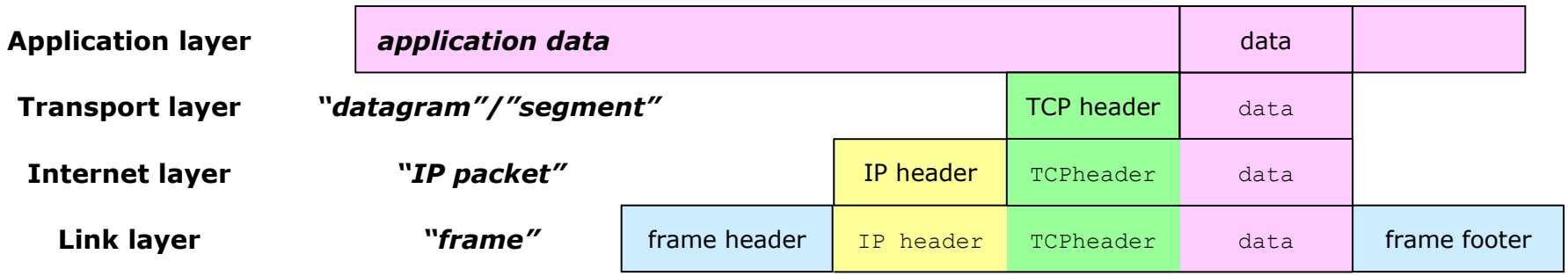
Hypothesis... ↑

Internet protocol stack: data flow



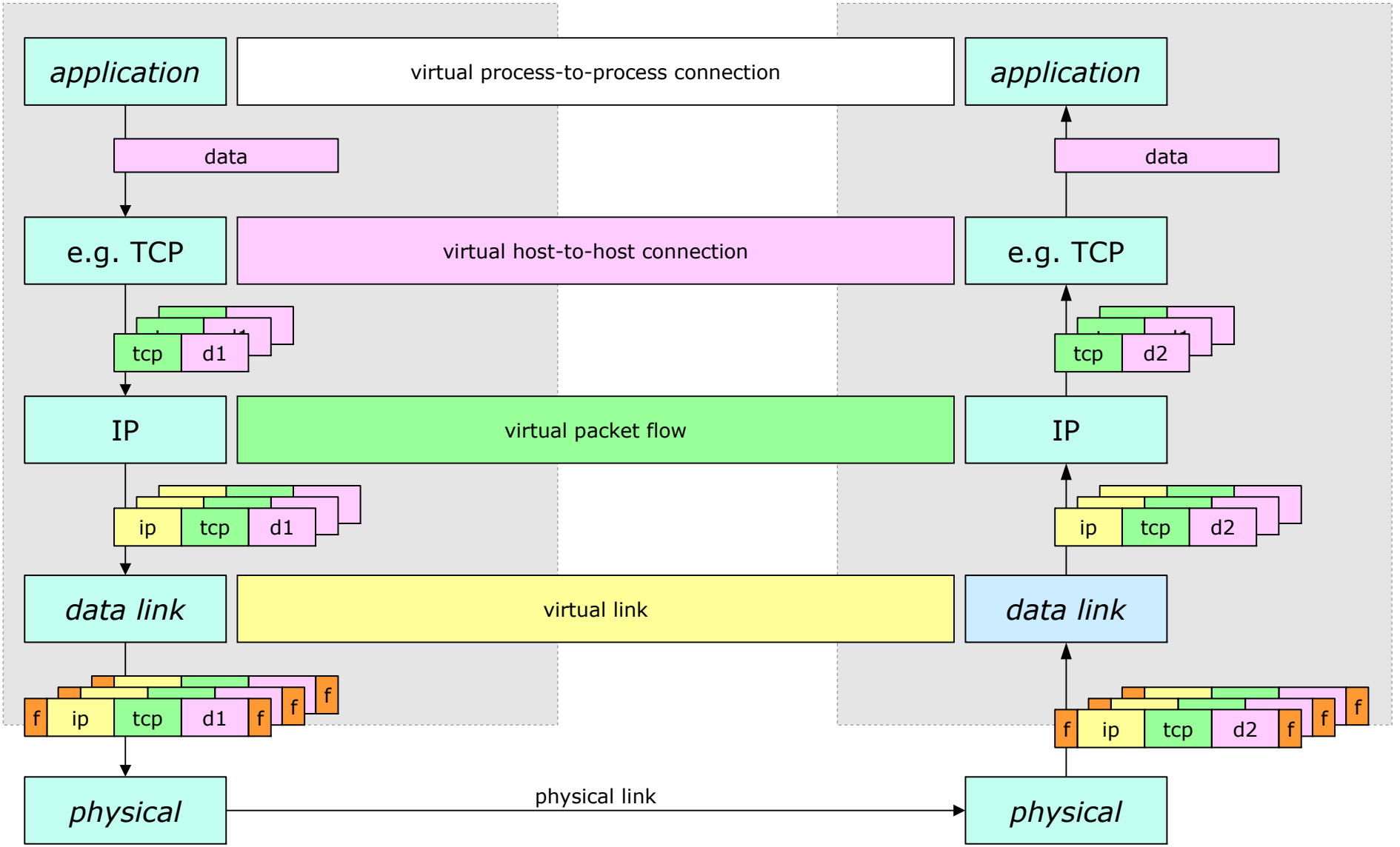
Internet protocol stack: data encapsulation

- Each layer adds data before sending, and removes data after receiving.
- Each layer treats what it gets from above as anonymous data.
- Layers may have different terms for similar types of data formatting.



- Contrast/breach: Deep Packet Inspection ↑

Internet protocol stack: virtual data flows



(Section 1.3.1 of Tanenbaum: lucid analogy for this ↑ process.)

The Internet: worldwide data at the IP level

- In 2012, the Carna botnet was used to do an IPv4 census.
- (“Don't try this at home!”)
- Let's finish by having a look at the obtained data >