CS 88: Security and Privacy

16: PKI and Introduction to Networking

10-27-2022

slides adapted from Dave Levine, Jim Kurose

SWARTHMORE COLLEGE

Reading Quiz

Network Security!

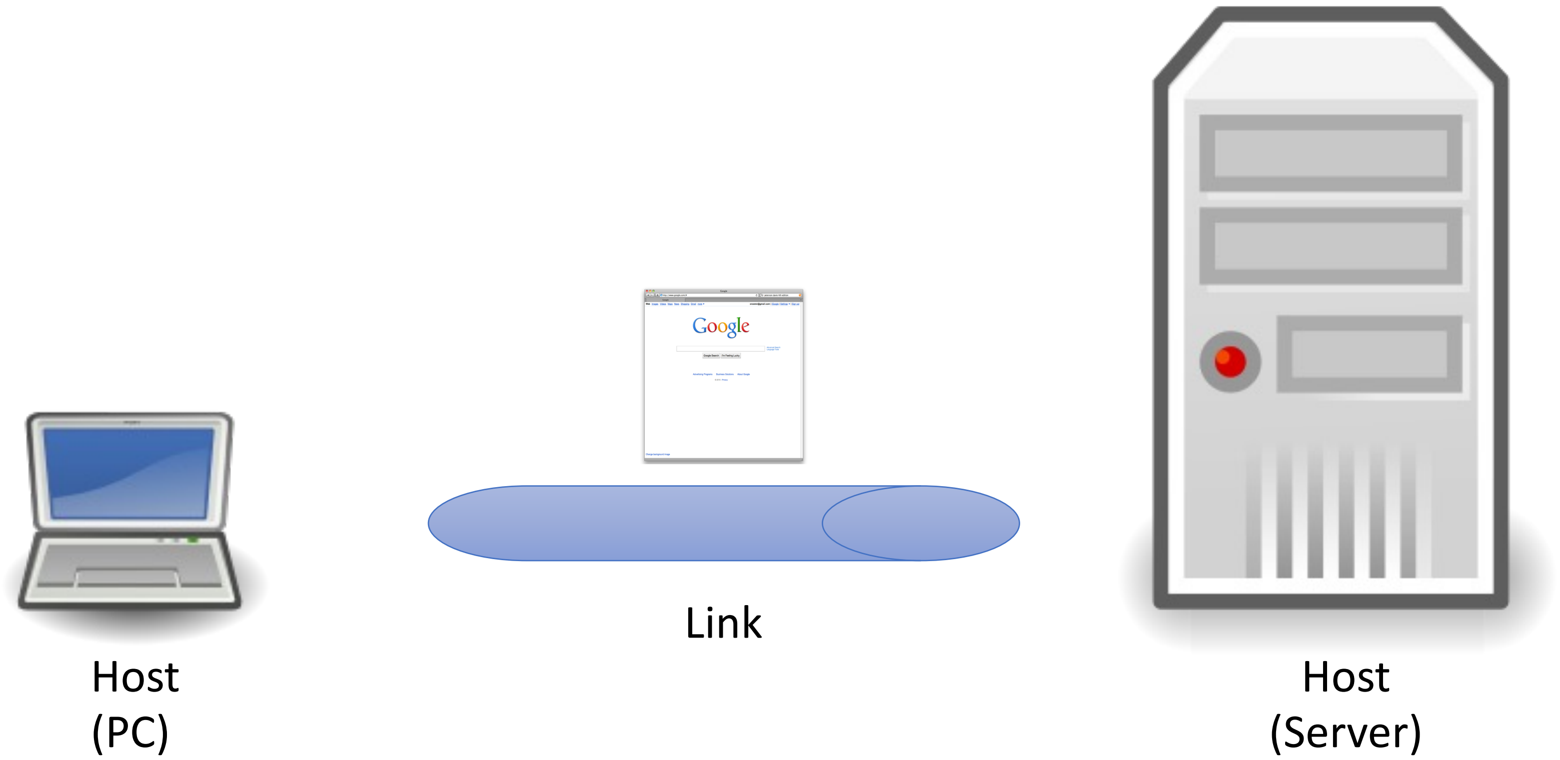
What is the goal of a network?

- Allow devices communicate with one another and coordinate their actions to work together.

- Piece of cake, right?

A “Simple” Task

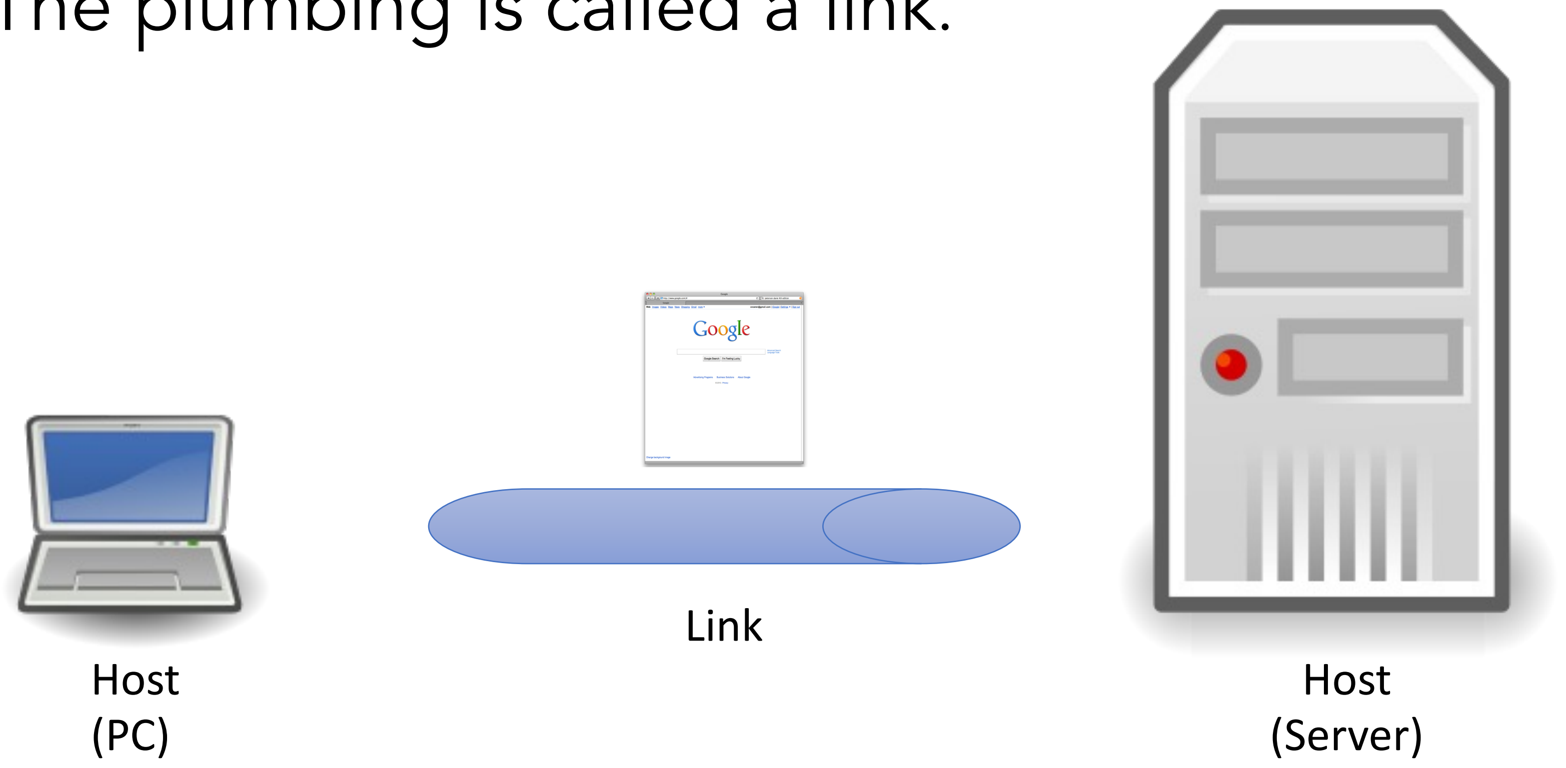
<u>Send information from one computer to another</u>



A "Simple" Task

<u>Send information from one computer to another</u>

- hosts: endpoints of a network
- The plumbing is called a link.

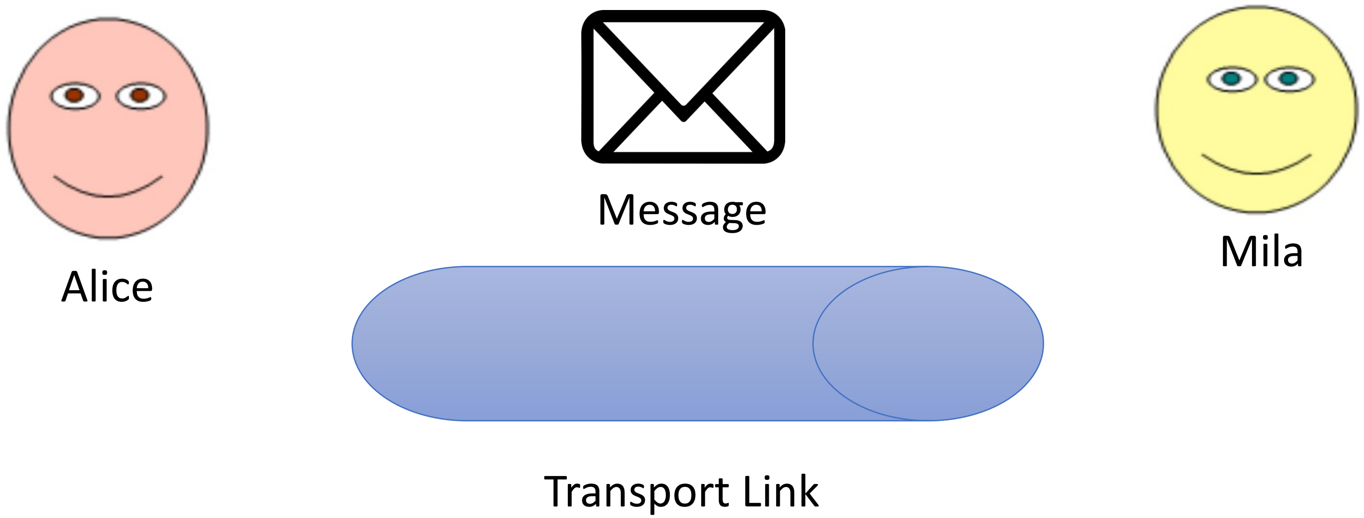


A "Simple" Task: Sending a message from host to destination

But first… let's try the postal system, something we are all (still!) familiar with and address a couple of key challenges..

A “Simple” analogous task: Post-it Note

Alice and Mila are Swatties starting out their semester and are roommates. Alice wants to give Mila a reminder to get milk.



A "Simple" analogous task: Post-it Note

Alice and Mila are roommates, Alice wants to give Mila a reminder to get milk. Figure out some key tasks:

1. <u>Structure of the message:</u>
   - Construct the message that Alice posts to Mila.
2. <u>Organizing a drop-off point.</u>
   - Who chooses the drop-off point?
3. <u>Write a protocol to write a note /post—it to your housemate</u>

A “Simple” analogous task: Post-it Note

Alice and Mila are roommates, Alice wants to give Mila a reminder to get milk.

1. <u>Structure of the message: (Alice to Mila)</u>

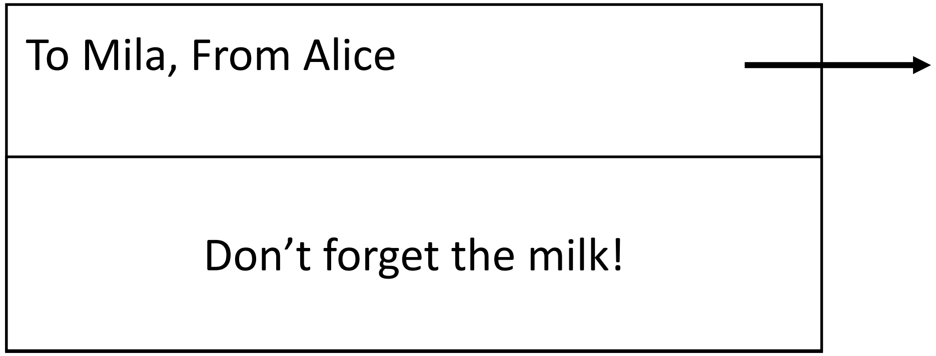
To Mila, From Alice
Don’t forget the milk!

<u>Irrespective of the source and destination, the format of the message stays the same.</u>

A “Simple” analogous task: Post-it Note

Alice and Mila are roommates, Alice wants to give Mila a reminder to get milk.

1. <u>Structure of the message: (Alice to Mila)</u>

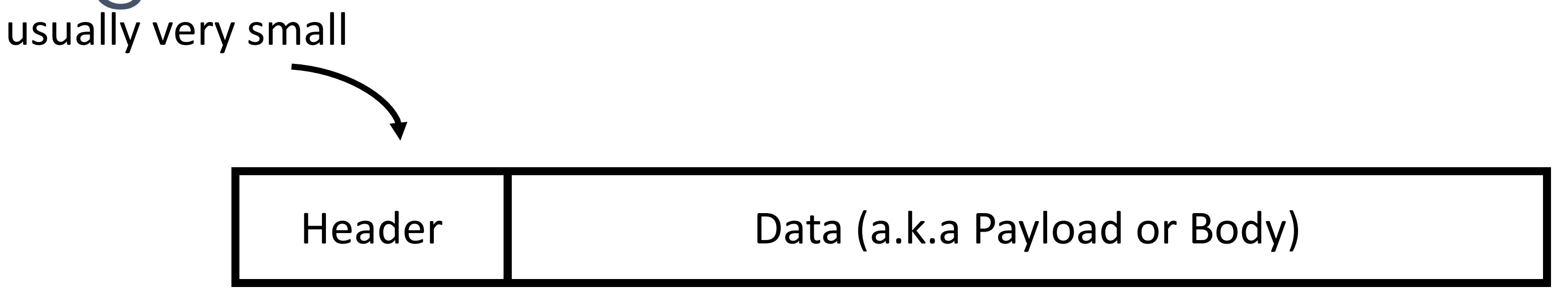


Header: contains sender/receiver information + additional state

- timestamp
- urgent! (priority)
- ordering of messages (1 of 10..)
- error control..

<u>Irrespective of the source and destination, the format of the message stays the same.</u>

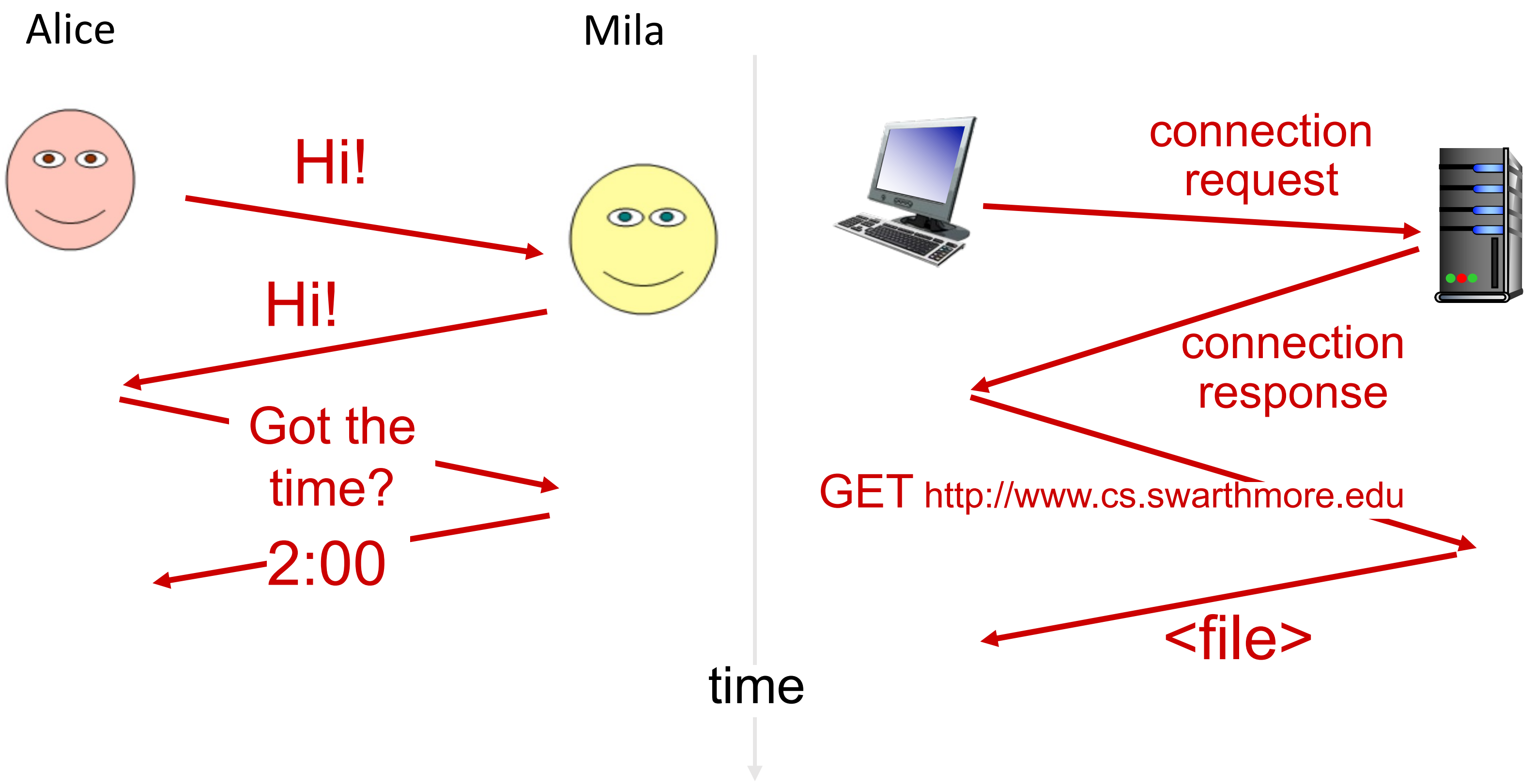
Message



- Message: Header + Data
- Data: what sender wants the receiver to know
- Header: information to support protocol
  - Source and destination addresses
  - State of protocol operation
  - Error control (to check integrity of received data)

What is a protocol?

Protocol: message format + transfer procedure



Human Protocol

Network Protocols (defined in RFCs)

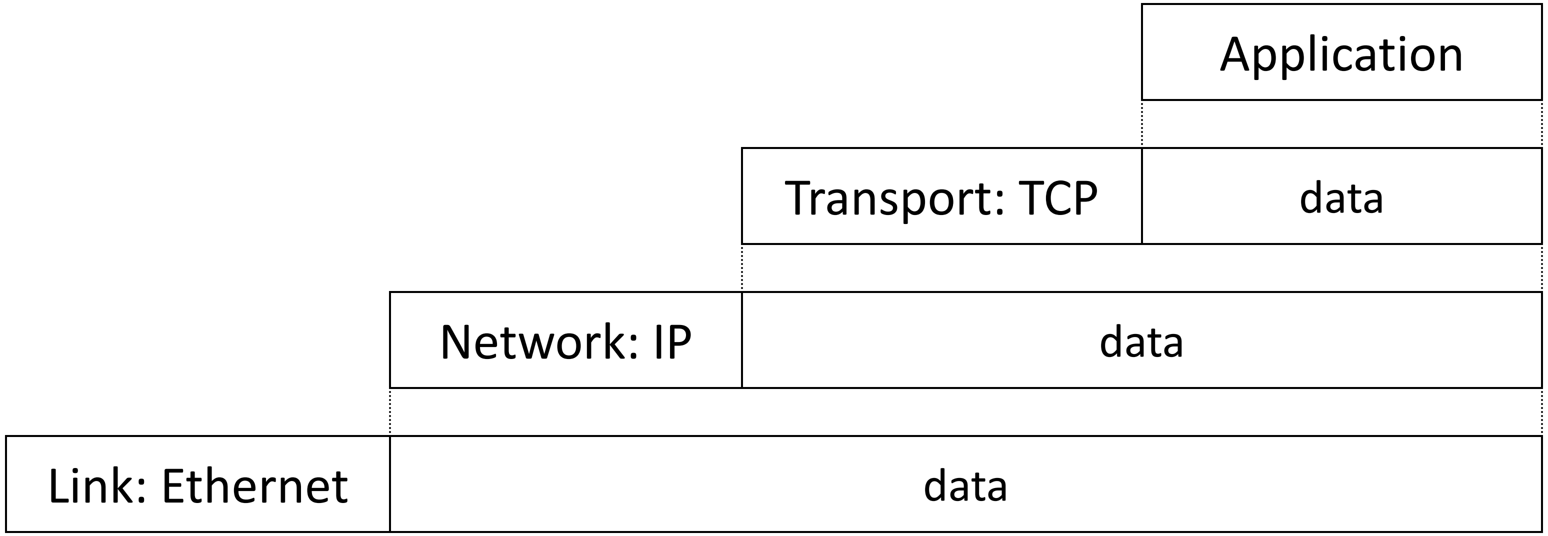
What is a protocol?

Goal: get message from sender to receiver

Protocol: message format + transfer procedure

- Expectations of operation
  - first you do x, then I do y, then you do z, …
- Multiparty! so no central control
  - sender and receiver are separate processes

Message Encapsulation



- Higher layer within lower layer
- Each layer has different concerns, provides abstract services to those above

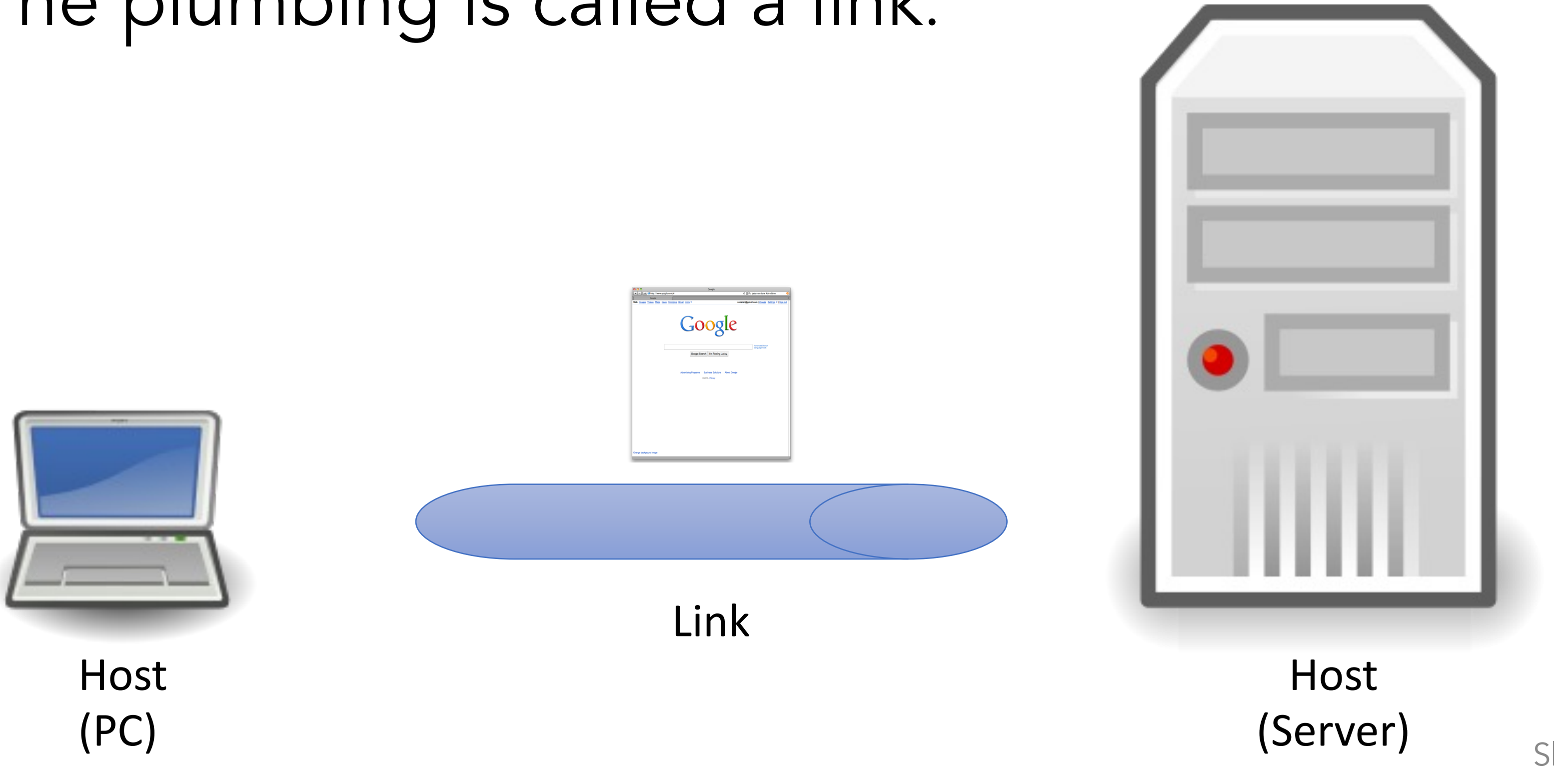
A "Simple" analogous task: Postal Mail

- Many more considerations..
  - Who decides the the sender and receiver addresses? Does someone maintain a mapping peoples' names to addresses?
  - Can Mila always be guaranteed of this delivery date? What factors influence delivery ?
  - What if the mail gets lost – who's responsibility is it? Alice, Mila or someone else?
  - What about security? privacy?

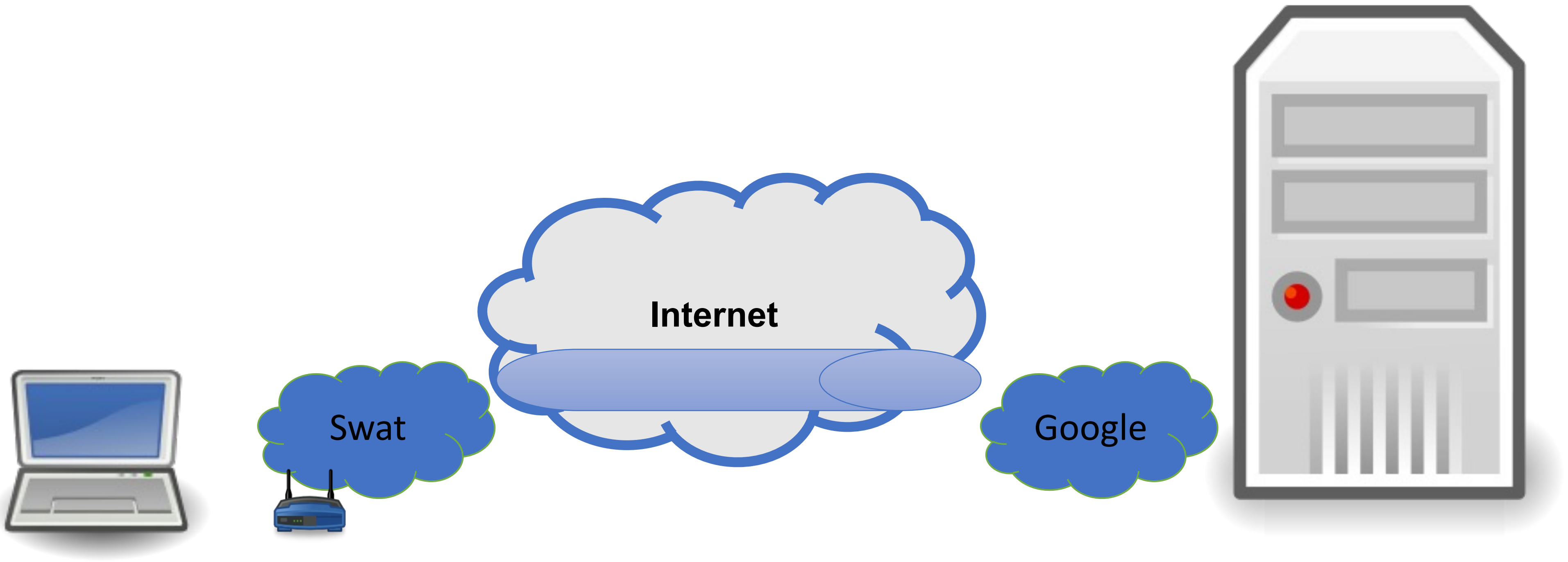
A "Simple" Task

<u>Send information from one computer to another</u>

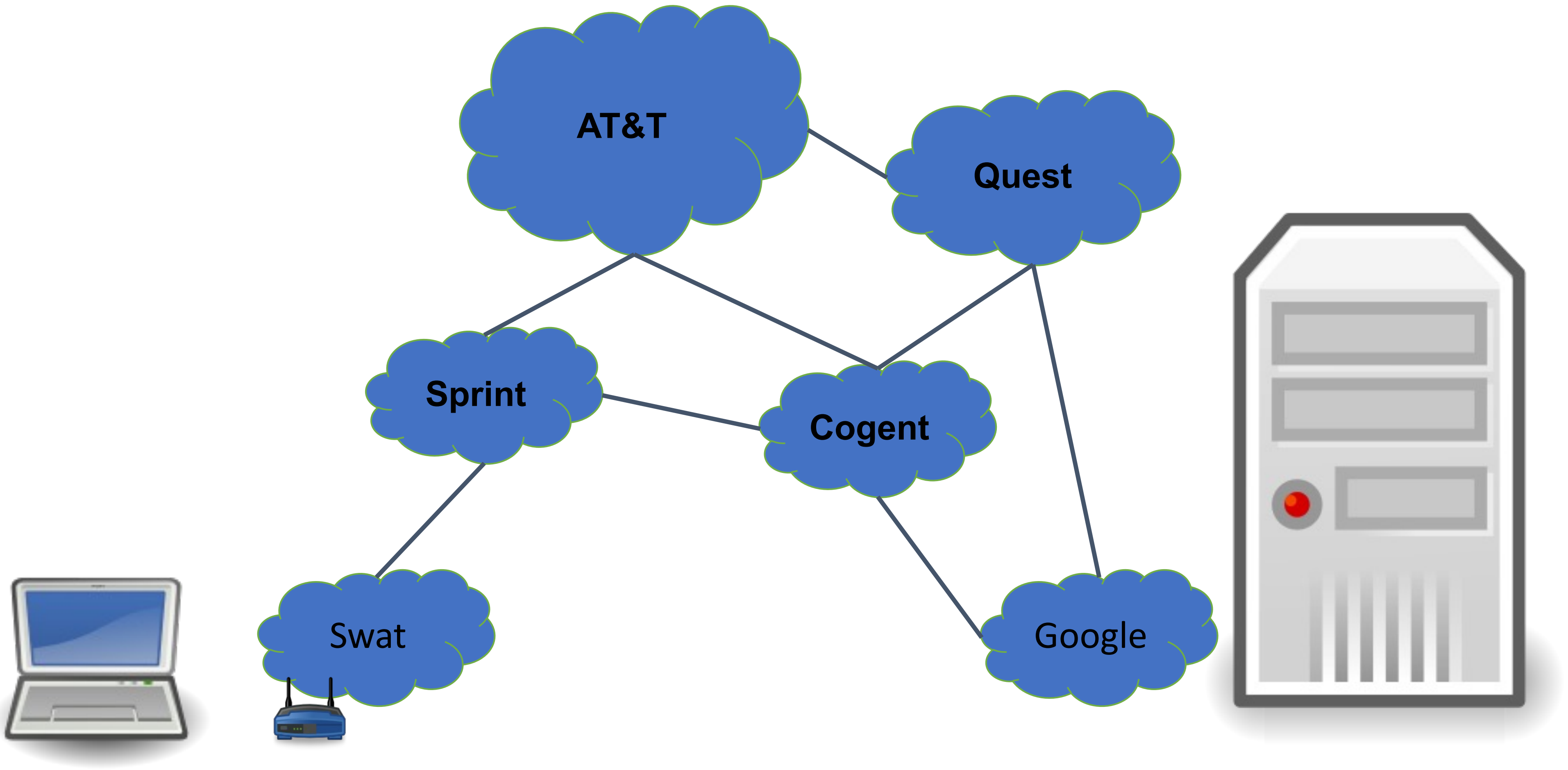
- hosts: endpoints of a network
- The plumbing is called a link.



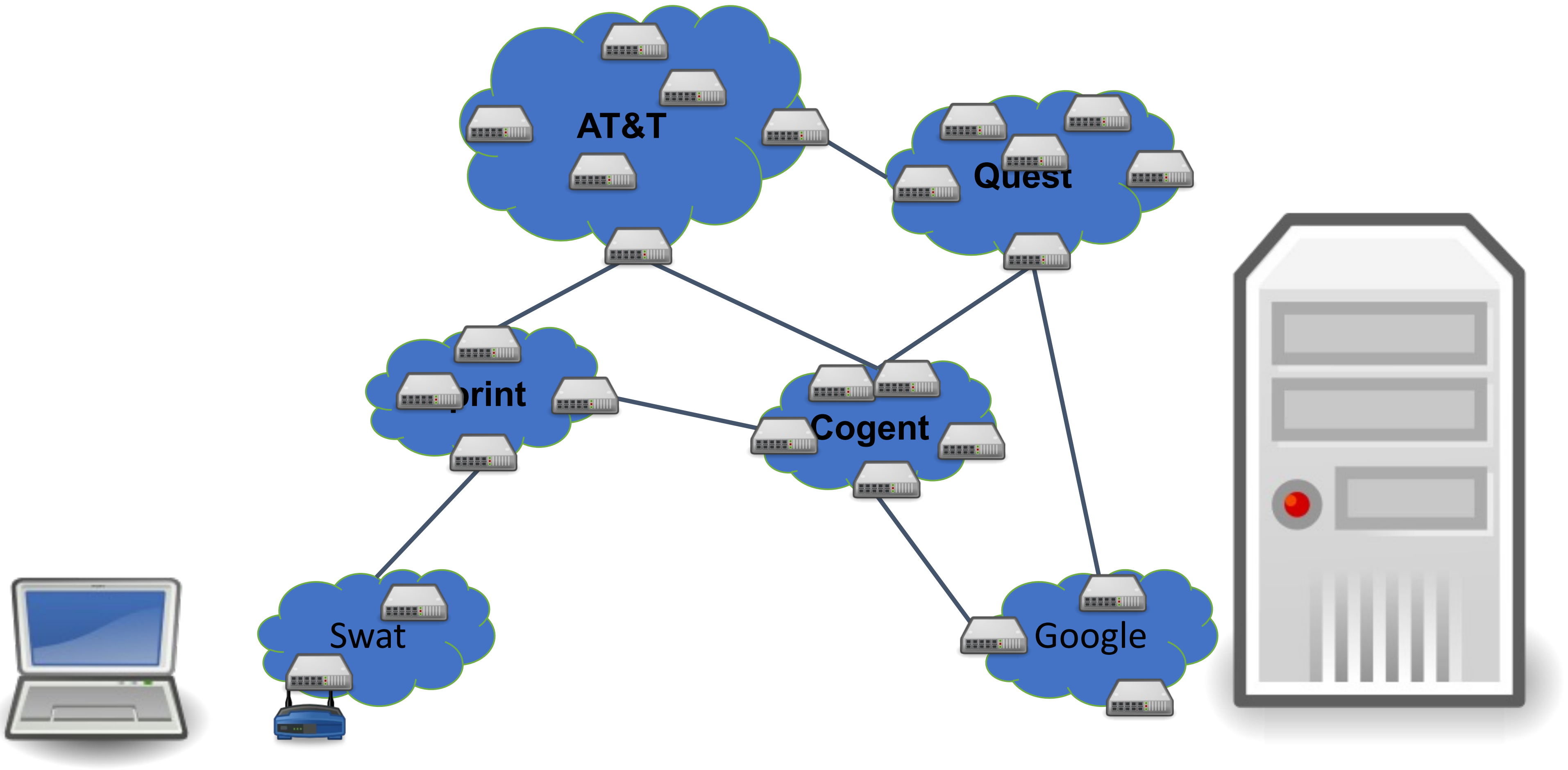
Not Really So Simple…



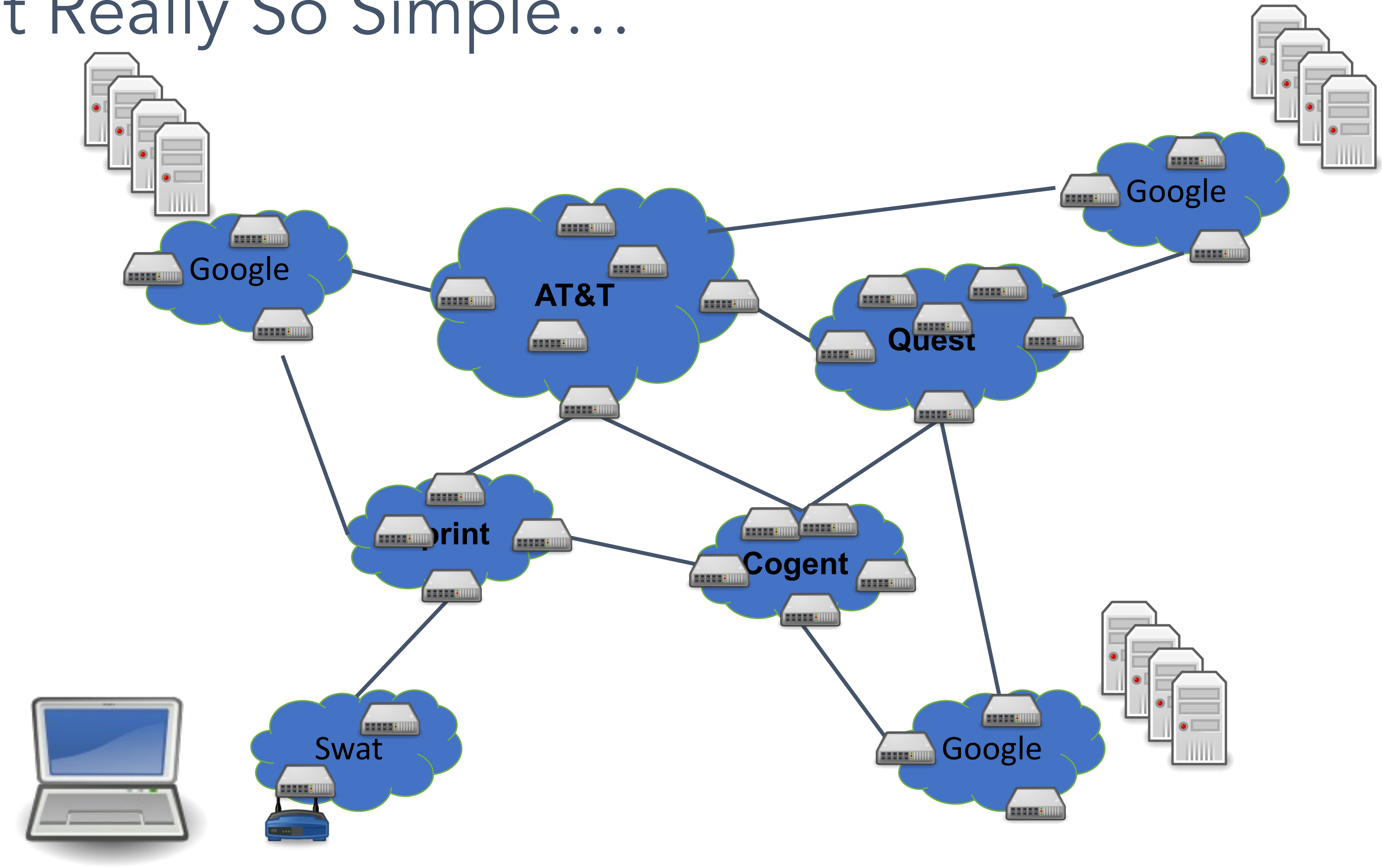
Not Really So Simple…



Not Really So Simple…



Not Really So Simple…



We only need…

- Manage complexity and scale up
- Naming and addressing
- Moving data to the destination
- Reliability and fault tolerance
- Resource allocation, Security, Privacy..

Five-Layer Internet Model

Application: the application (e.g., the Web, Email)

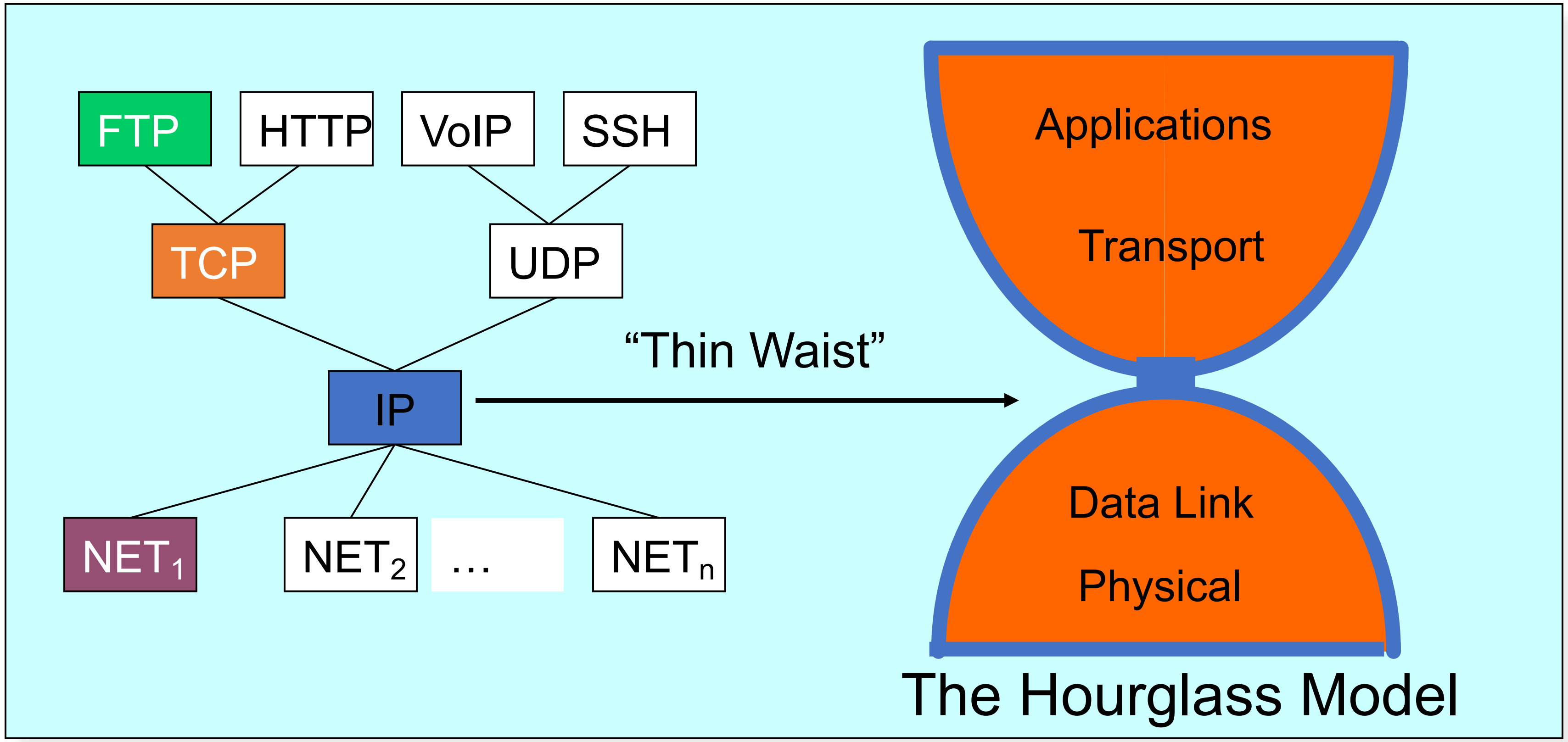
Transport: end-to-end connections, reliability

Network: routing

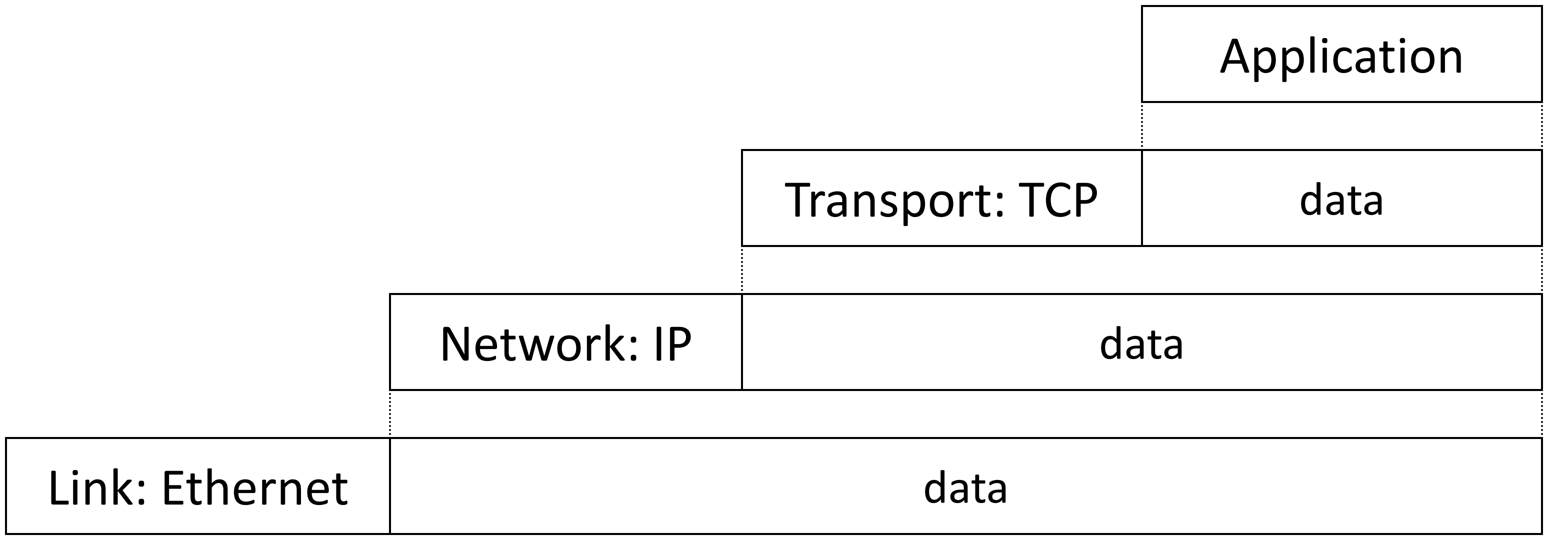
Link (data-link): framing, error detection

Physical: 1’s and 0’s/bits across a medium
(copper, the air, fiber)

Internet Protocol Suite

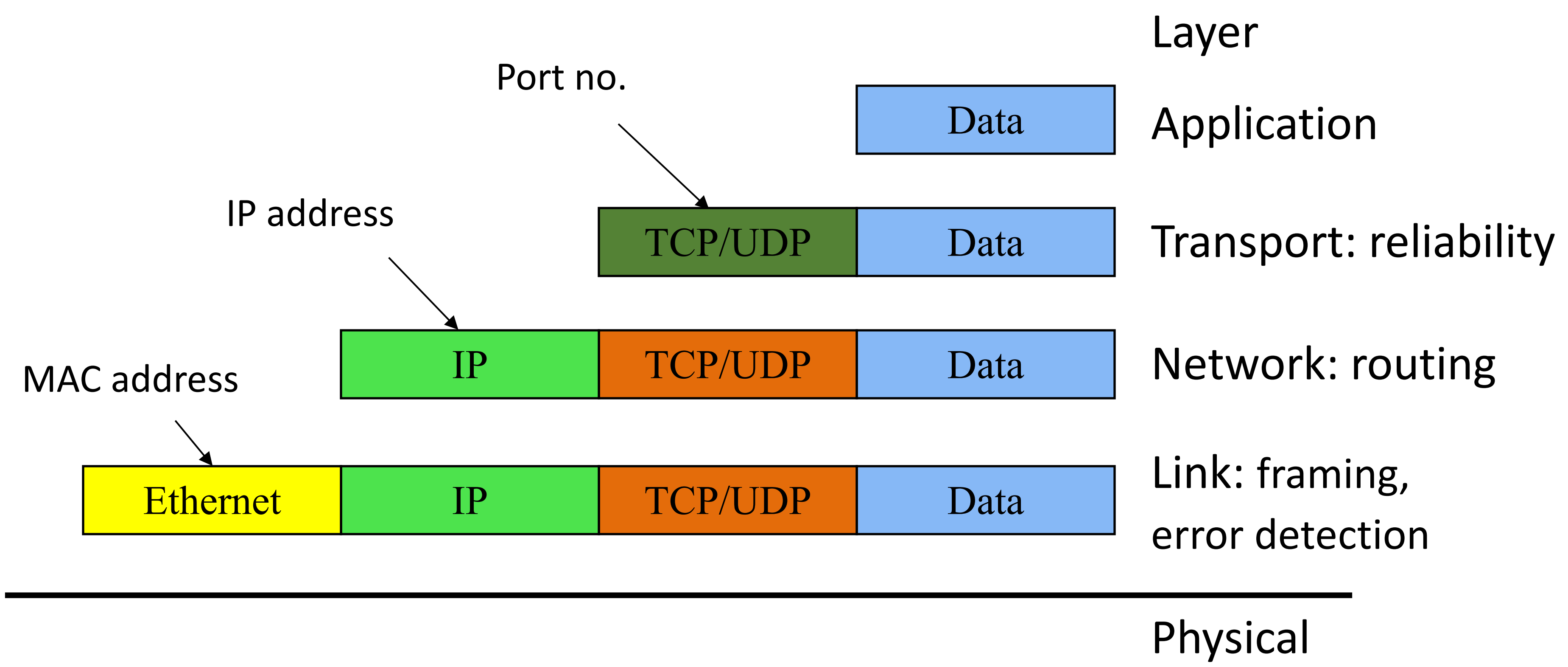


Message Encapsulation



- Higher layer within lower layer
- Each layer has different concerns, provides abstract services to those above

Layering and encapsulation



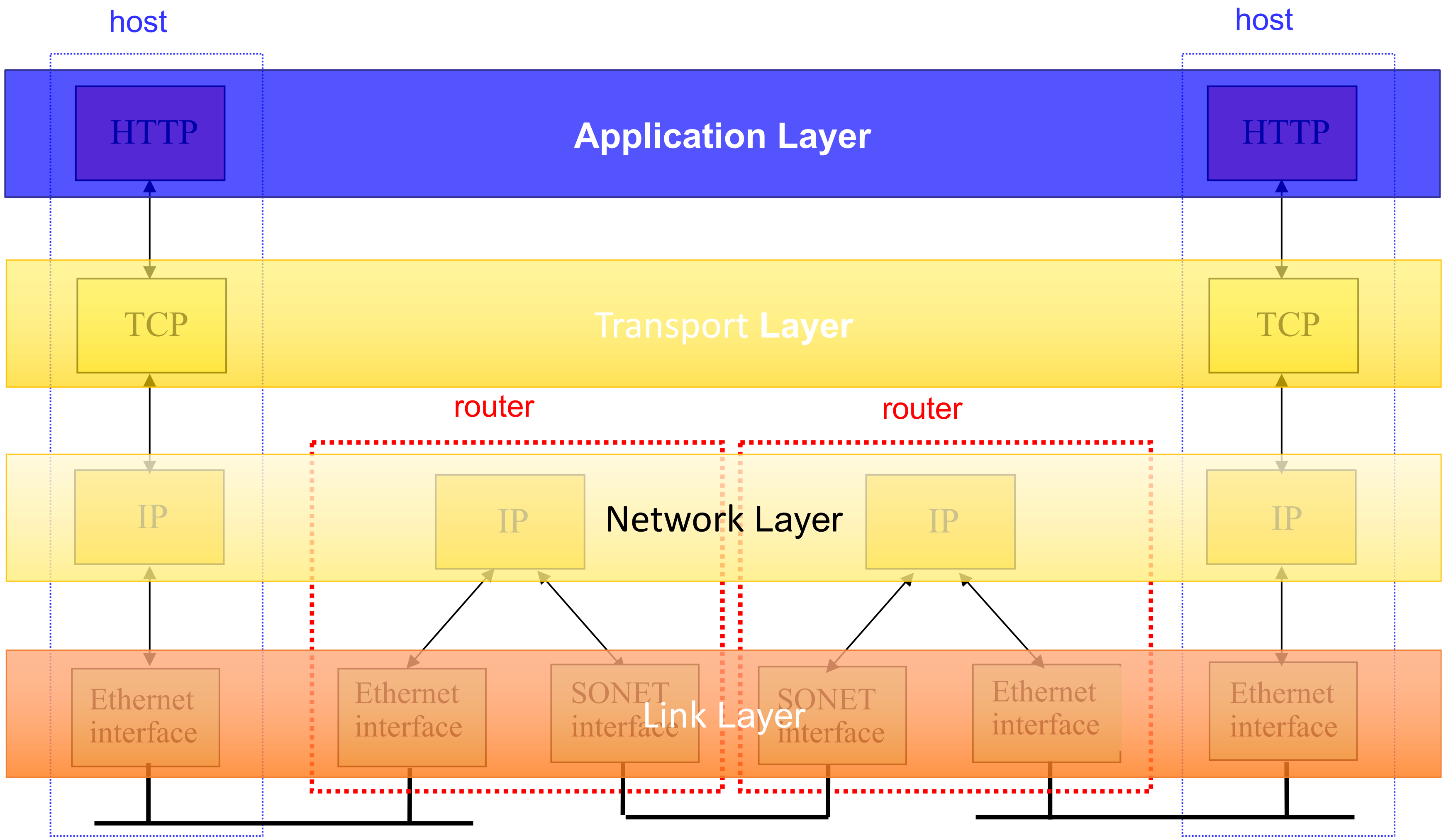
Layering: Separation of Functions

- explicit structure allows identification, relationship of complex system's pieces
  - layered reference model for discussion
  - reusable component design
- modularization eases maintenance
  - change of implementation of layer's service transparent to rest of system,
  - e.g., change in postal route doesn't effect delivery of lette

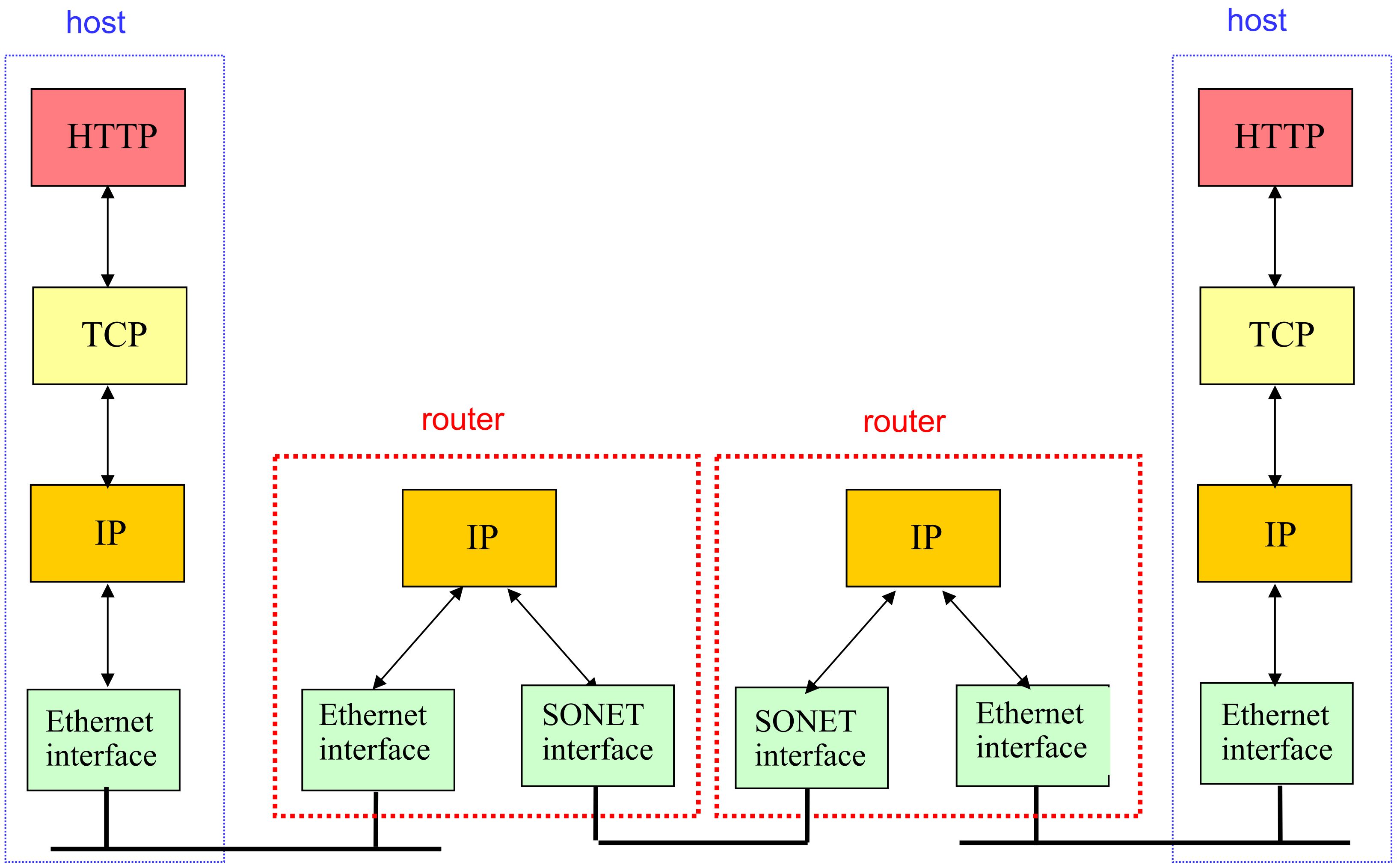
Abstraction!

- Hides the complex details of a process
- Use abstract representation of relevant properties make reasoning simpler
- Ex: Alice and Mila's knowledge of postal system:
  - Letters with addresses go in, come out other side

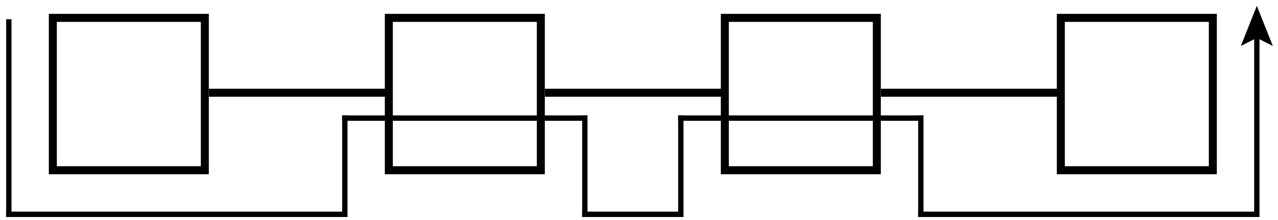
TCP/IP Protocol Stack



TCP/IP Protocol Stack



The “End-to-End” Argument



Don’t provide a function at lower layer if you have to do it at higher layer anyway ...

... unless there is a very good performance reason to do so.

Examples: error control, quality of service

Reference: Saltzer, Reed, Clark, “End-To-End Arguments in System Design,” ACM Transactions on Computer Systems, Vol. 2 (4), pp. 277-288, 1984.

Midterm Discussion