



A quick guide to the Internet

David Clark

6.978J/ESD.68J

Lecture 1 V1.0

Spring 2006

Why should you care?

- “People” say: “Why doesn’t the Internet...”
 - Protect me from spam.
 - Control porn
 - Keep terrorists from plotting
 - Etc.
 - We need to translate from a cry of pain to realistic expectations.
 - Must understand (in general terms) the technology to make realistic policy.
-

Defining the Internet

- It is not the applications:
 - Email
 - Web
 - VoIP
 - It is not the technology
 - Ethernet
 - WiFi
 - Fiber optics
 - So what is it?
-

A nice picture

A range of applications operating over a range of technologies, by means of a single interface -- the Internet Protocol IP.

Figure removed for copyright reasons.

See <http://www.nap.edu/openbook/0309050448/html/53.html>

Figure 2.1 in National Research Council. "Realizing the Information Future: The Internet and Beyond." National Academies Press, 1994.

What is a network?

Or...what is the problem we are solving?

- *A shared medium of communications.*
 - Why?
 - To share expensive resources
 - Cannot afford a wire between “everywhere.”
 - To facilitate general communication--
information sharing.
-

How to share?

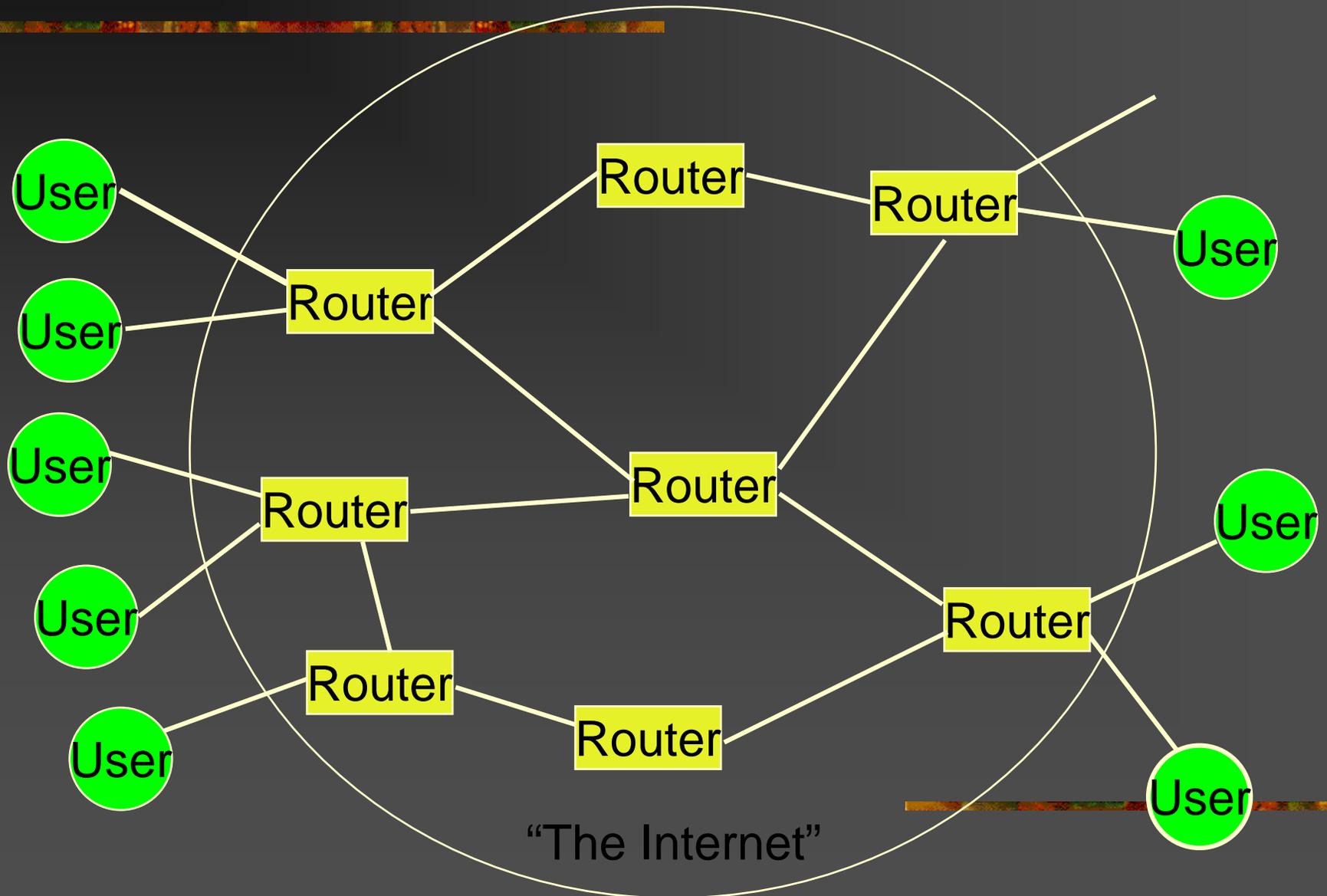
- Computer traffic is bursty.
 - Older sharing method (circuit switching) was inefficient.
 - About 30 years ago, the need for a new mode of sharing was felt.
 - **PACKETS!**
-

What is a packet?

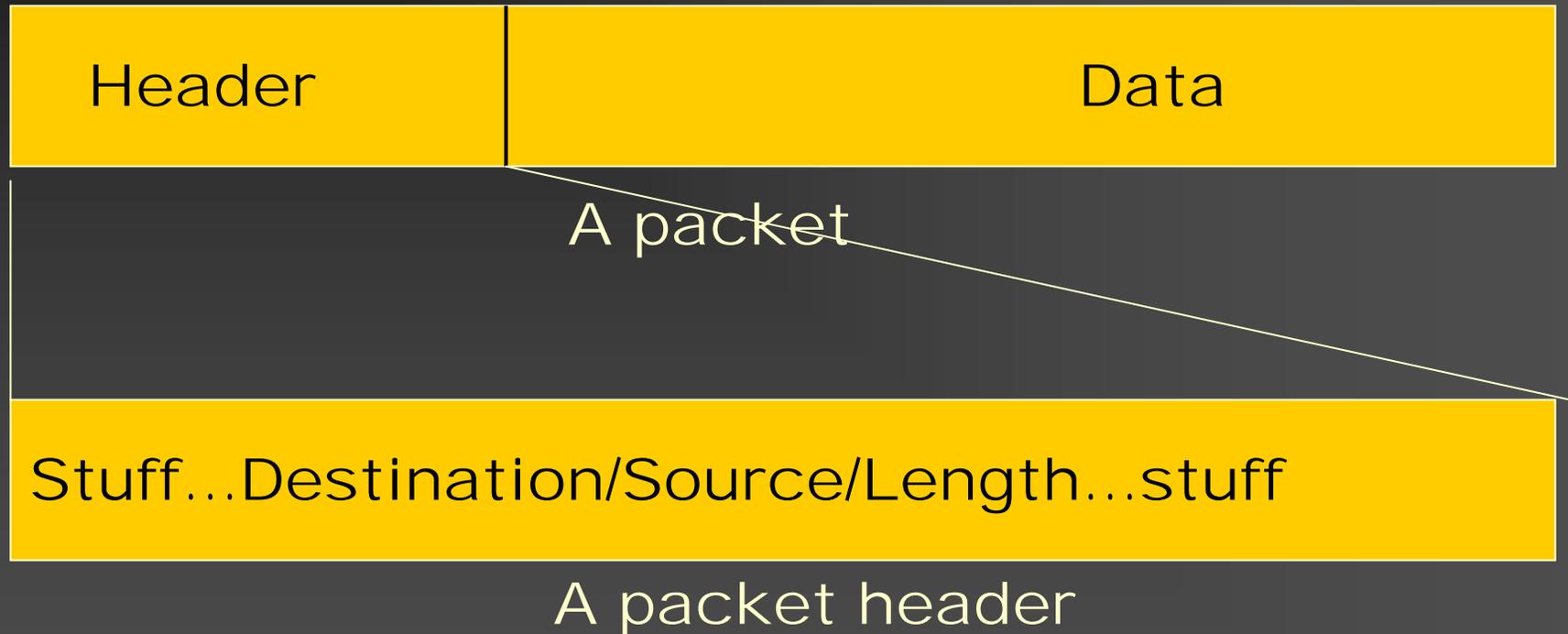
- Packet:
 - Some data with an address on the front.
Specified maximum size
 - Sent serially across a link.
 - Use a computer (a "router" or "packet switch") to manage the link.
 - Statistical sharing.

 - A neat idea that has stood the test of time.
-

A simple view of the Internet



An (over) simple packet picture



Addresses: written in the form 18.26.0.166

What a router does

- When it gets a packet:
 - Check that it is not malformed.
 - Check that it is not going in circles.
 - Look at its destination address.
 - Pick the best link over which to forward it.
 - In the background:
 - Computes the best routes to all destinations.
-

What was at the edge?

- The slide said “user”.
 - It is a “host”, or a “PC”, or a “server”, or a “computer” or an “end node”.
 - The place where application code runs.
 - There might be a person there.
 - Get back to this later...
-

Implications

- Inside the network there are only packets.
 - There is no understanding of higher-level intentions.
 - There is nothing like a “call”, or “placing a call”, in the router’s design.
 - The routers have a limited view of what it means to “operate correctly”.
-

The service model

- The other half of the Internet specification:
 - What is the commitment when I send a packet?
 - Answer: very little.
 - The Internet tries it best, but makes no promises.
 - It can lose, reorder, delay, or duplicate packets.
 - Usually they arrive in good order.
 - If they don't--you have no complaint.
 - Called the “best effort” service.
-

Is this such a good idea?

- Weak expectation means Internet can run over “anything”.
 - Makes the application’s job harder, but not impossible.

 - So, yes, it is a good idea.
 - But now under attack.
-

Congestion

- More than you want to know in one slide...
 - What happens if too many packets get sent?
 - In the short run, queues form in routers.
 - In the longer run, senders (are supposed to) slow down.
 - Why does this work?
 - Application are expected to tolerate it.
 - But if senders do not slow down?
 - Out of aggression, or because they cannot.
 - A raging debate among designers.
-

Quality of Service (QoS)

- Four options.
 - Demand and expect them to slow down.
 - Benign socialist
 - Police them and punish them if they don't slow down.
 - Police state
 - Let them pay to keep going fast.
 - Capitalist
 - Over-provision so net is “never” congested.
 - Pragmatic
-

What was at the edge?

- The slide said “user”.
 - It is a “host”, or a “PC”, or a “server”, or a “computer” or an “end node”.
 - The place where application code runs.
 - There might be a person there.
-

What a “host” does

- Runs the application code
 - Web, email, voip, ssh, doom, etc.
 - Runs software that helps cope with packets and the best effort service model
 - Example: Transmission Control Protocol, or TCP.
-

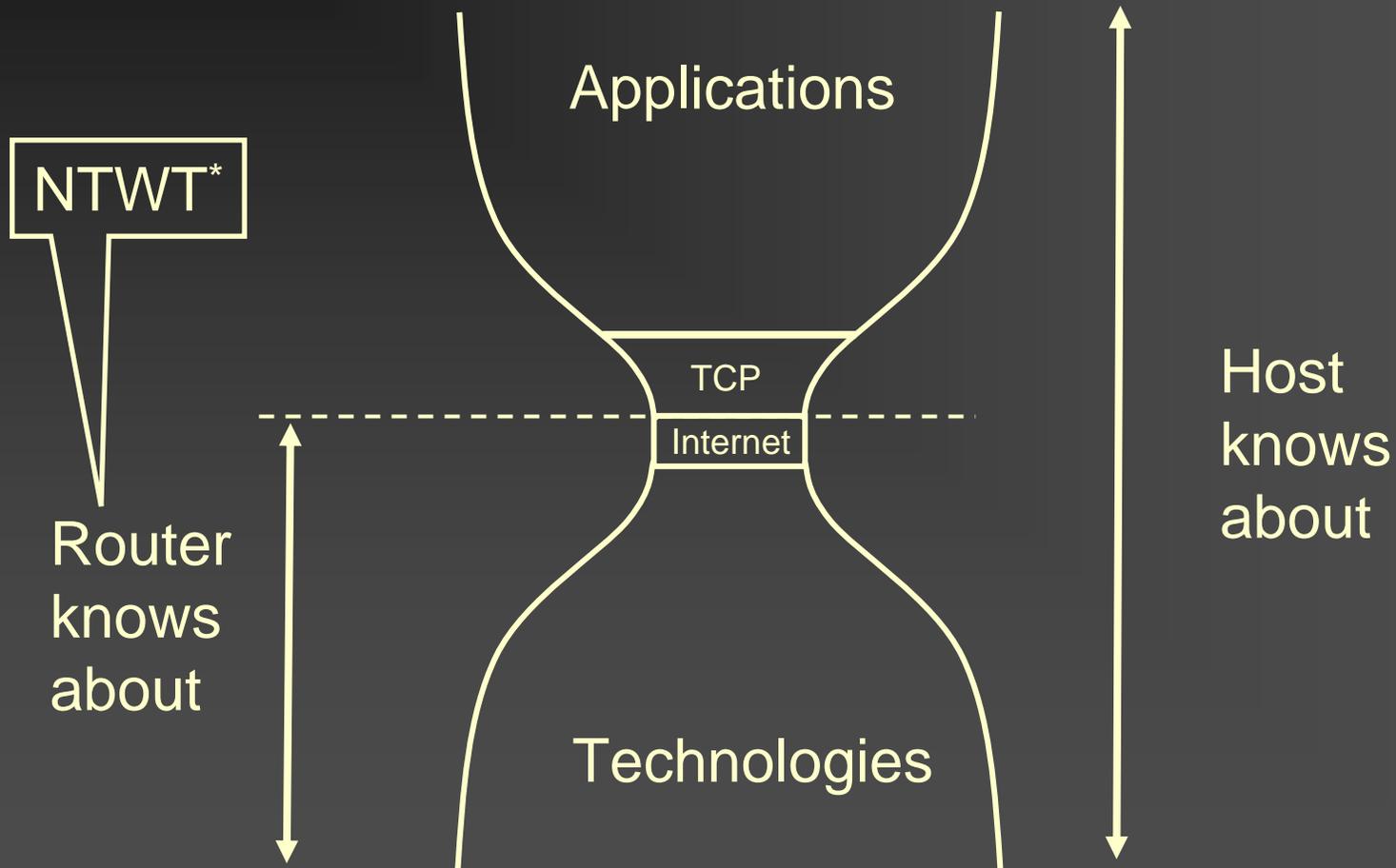
What is a protocol?

- Protocol: A specification of what can be sent when and in what format.
 - A very general term, used to describe many aspects of networking.
 - The voltage to represent a 1 or 0 on a link.
 - The bit sequence to represent characters (e.g. ASCII).
 - The format of the address on the front of the packet.
 - How one reports a lost packet.
 - From the Greek: "Glued on the front."
-

What does TCP do?

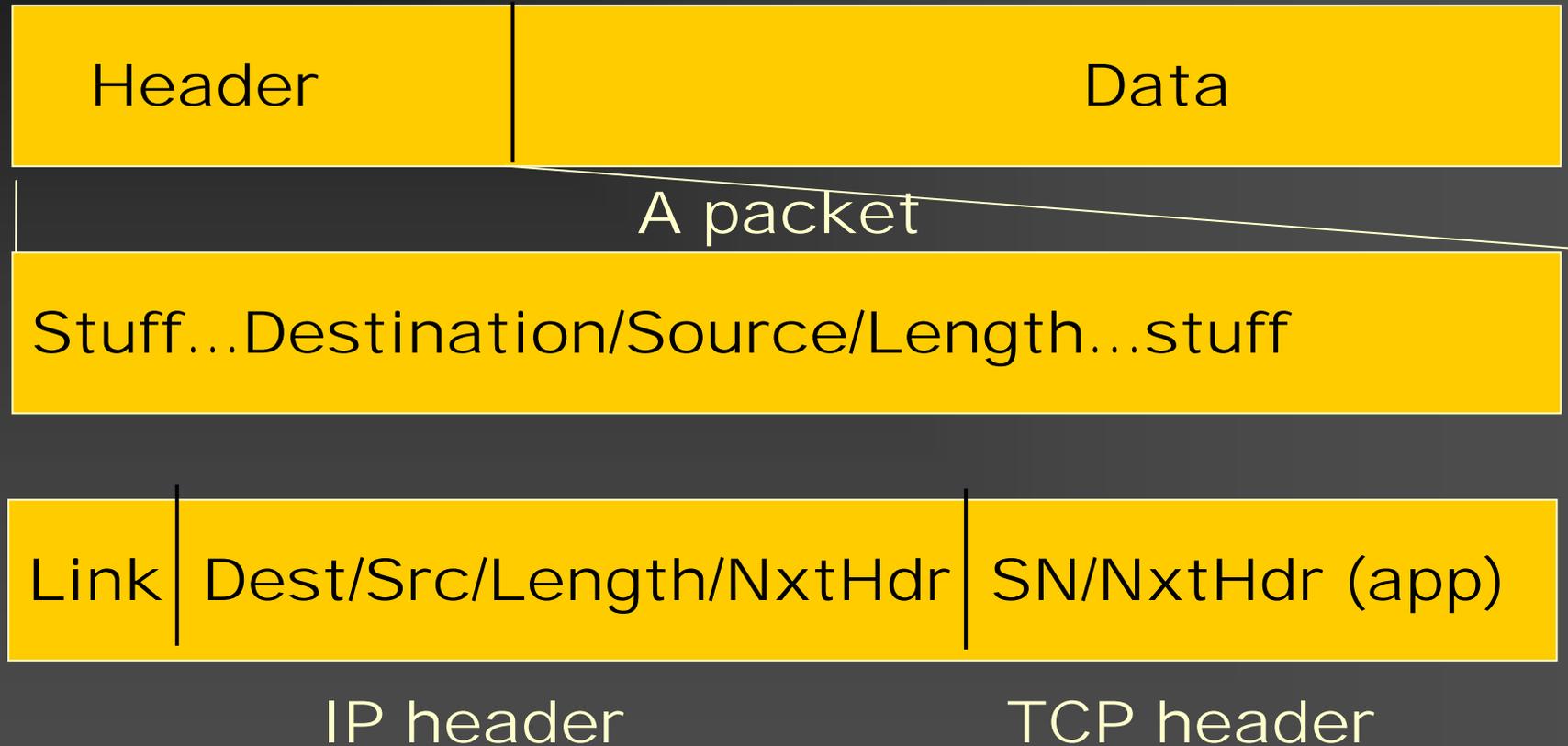
- Breaks a chunk of data (what the application wants to send) into packets at the sender.
 - Numbers the packets.
 - Keeps sending them until it gets an acknowledgement.
 - Puts them in order at the receiver.
 - Passes the data to the right application.
 - Provides a very simple failure model.
-

Host vs. router



*Not the whole truth

A (less) simple packet picture



The *end-to-end arguments*:

The lower layers of the network are not the right place to implement application-specific functions. The lower layers of the network should implement basic and general functions, and the applications should be built “above” these functions, at the edges.

- E.g. move functions “up and out”.
 - This causes function migration to the end-node.
 - The network should be “as transparent as technology permits”.
-

Benefits of end-to-end

- User empowerment.
 - Run what you please.
- Flexibility in the face of unknown applications.
 - A network to hook computers together.
- Lower cost in core of network.
 - Eliminate special “features”.
 - Rely on edge-node equipment.
- More robust applications.
 - No unexpected failures of third-party nodes.
- An example of “getting it wrong”: make the network reliable.

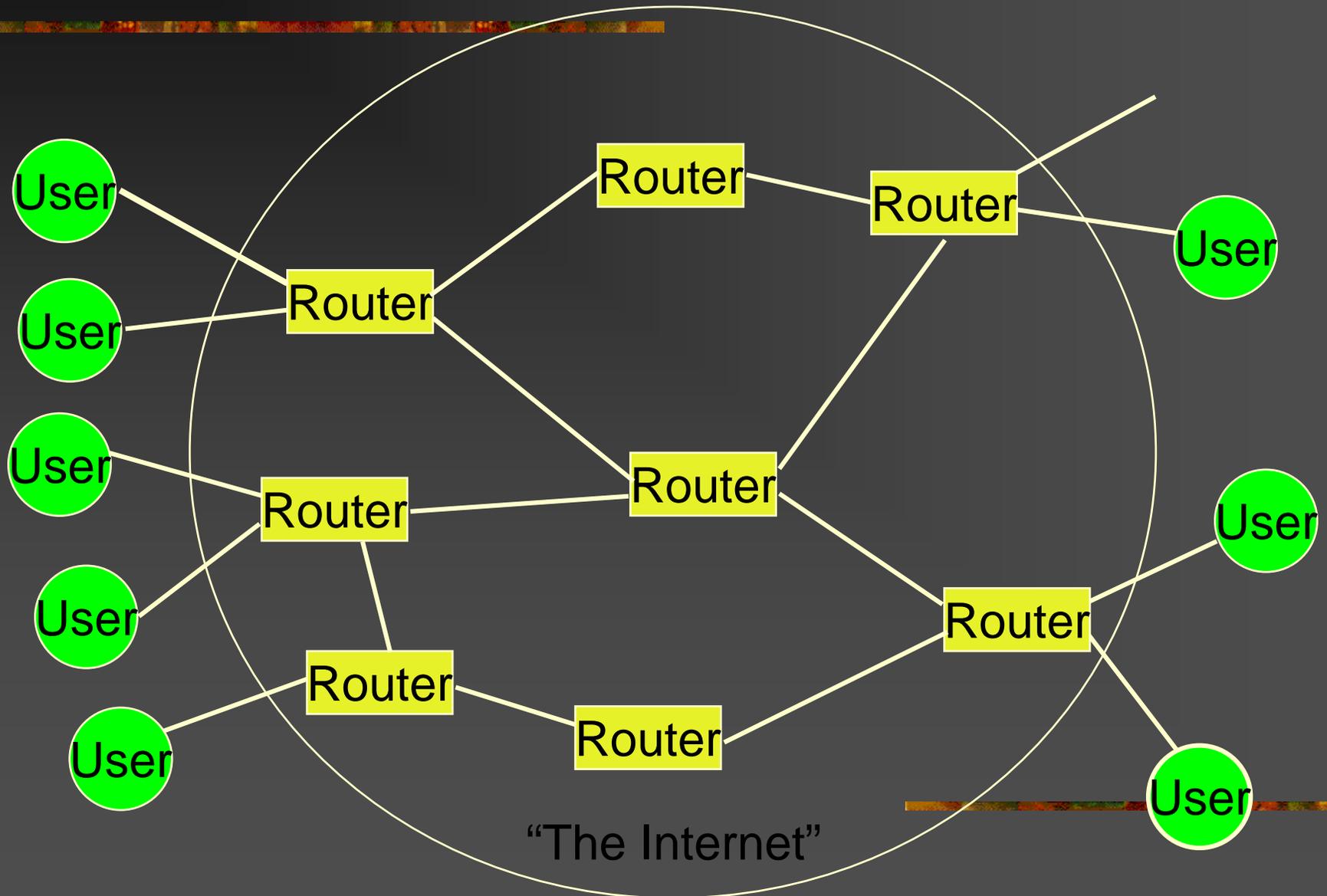
Summary

- What “the Internet” does is very simple:
 - It forwards packets.
 - It is oblivious to the purpose of the packets.
 - Packets allow effective/efficient sharing.
 - Lots of applications run on the Internet
 - And there will be more tomorrow.
 - Each has its own design.
 - There is a tension/tradeoff as to where functions are placed.
 - The Internet can exploit lots of technologies.
-

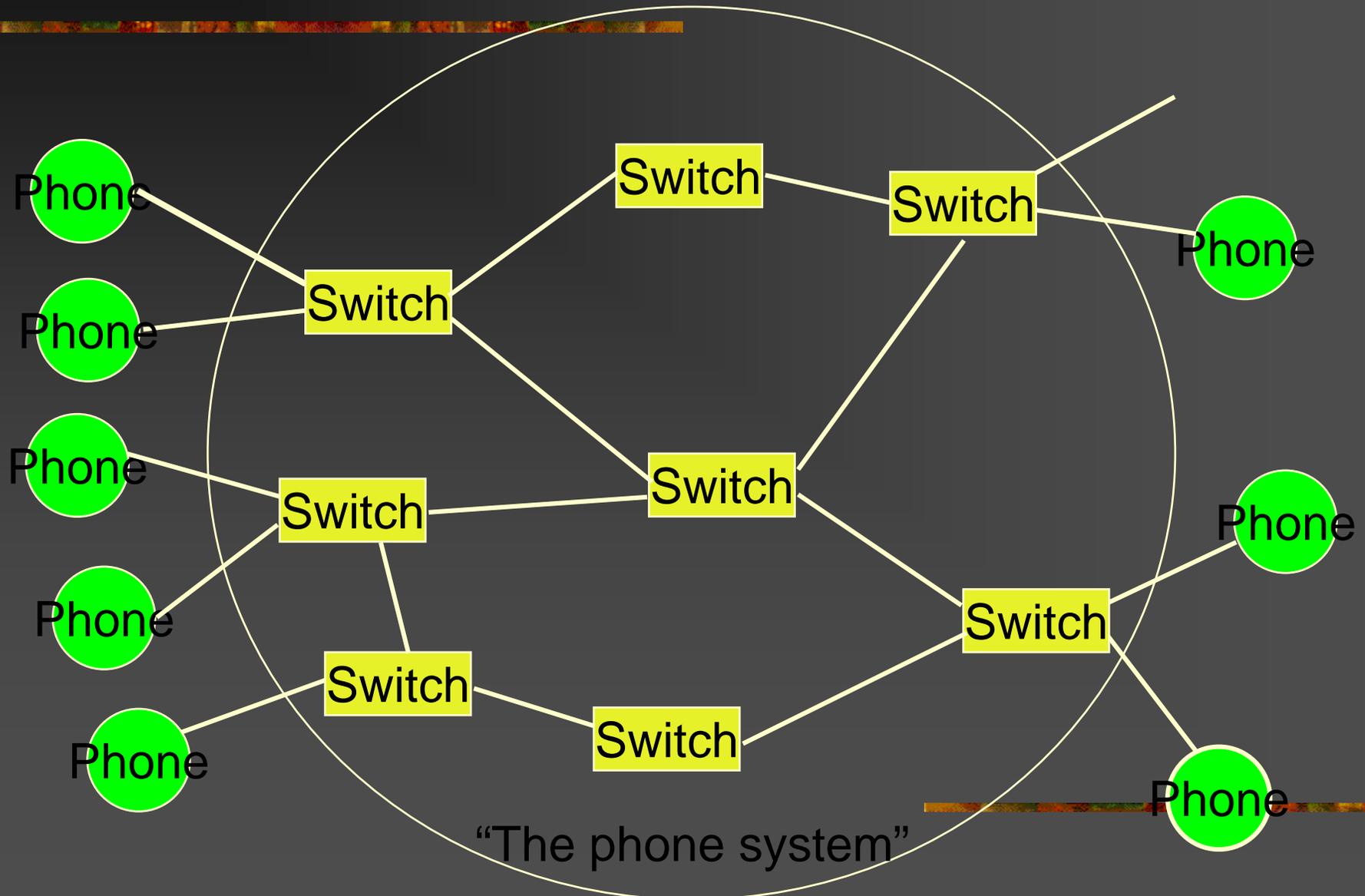
How about the phone system?

- How does it differ from the Internet?
 - And *why*?
- What are the implications for policy?

A simple view of the Internet



A simple view of the phone system



The differences?

- Switches are powerful, because phones are simple.
 - The knowledge of what the phone system is for is embedded in the switches.
 - It “knows” that its purpose is to carry voice.
 - Routers are simple, because end-points are powerful.
 - The knowledge of what the Internet is for is embedded in the end-points.
-

And...?

- The phone system has no open interface for complementers.
 - No easy way to attach new applications.
 - Compare the generality of a voice circuit and packet carriage.
 - Very different view of layering. There is no “voice” layer.
-

Next lecture

- How we design applications.
 - Technical design issues.
 - Implications for policy and for industry structure.
 - Options for shaping tussle among stakeholders.
-

Multiple views of system

- Topology view:
 - Routers as expression of physical distribution.
 - Layered view:
 - What is the role of each “box”.
 - What does this imply about limits to action?
 - Administrative view.
 - Who owns/operates each part?
 - Who controls what talks to what?
-

A more realistic picture

