Architectural Principles

Brighten Godfrey
cs598pbg August 31 2010

Today

Clark: TCP / IP design philosophy

Architectural principles

Goals of the architecture

1. Interconnect existing networks
2. Survivability
3. Multiple communication services
4. Variety of networks
5. Distributed management
6. Cost effective
7. Easy host attachment
8. Resource usage accountability

I. Interconnect networks

- Assumption: One common architecture
- Technique: packet switching
  - Met target application needs
  - Already used in ARPANET, ARPA packet radio network
- Interconnect with layer of gateways (packet switches)

2. Survivability

- Definition: even with failures, endpoints can continue communicating without resetting high-level end-to-end conversation
  - Except when?
- Did this work?

2. Survivability

Key question for survivability:

Where is connection state stored?

In network

So, must replicate

- Complicated
- Does not protect against all failures

On end hosts

Shared fate

- Simpler
- If state lost, then it doesn't matter

Conclusion: stateless network — datagram packet switching

3. Multiple types of service

- Initially, just TCP

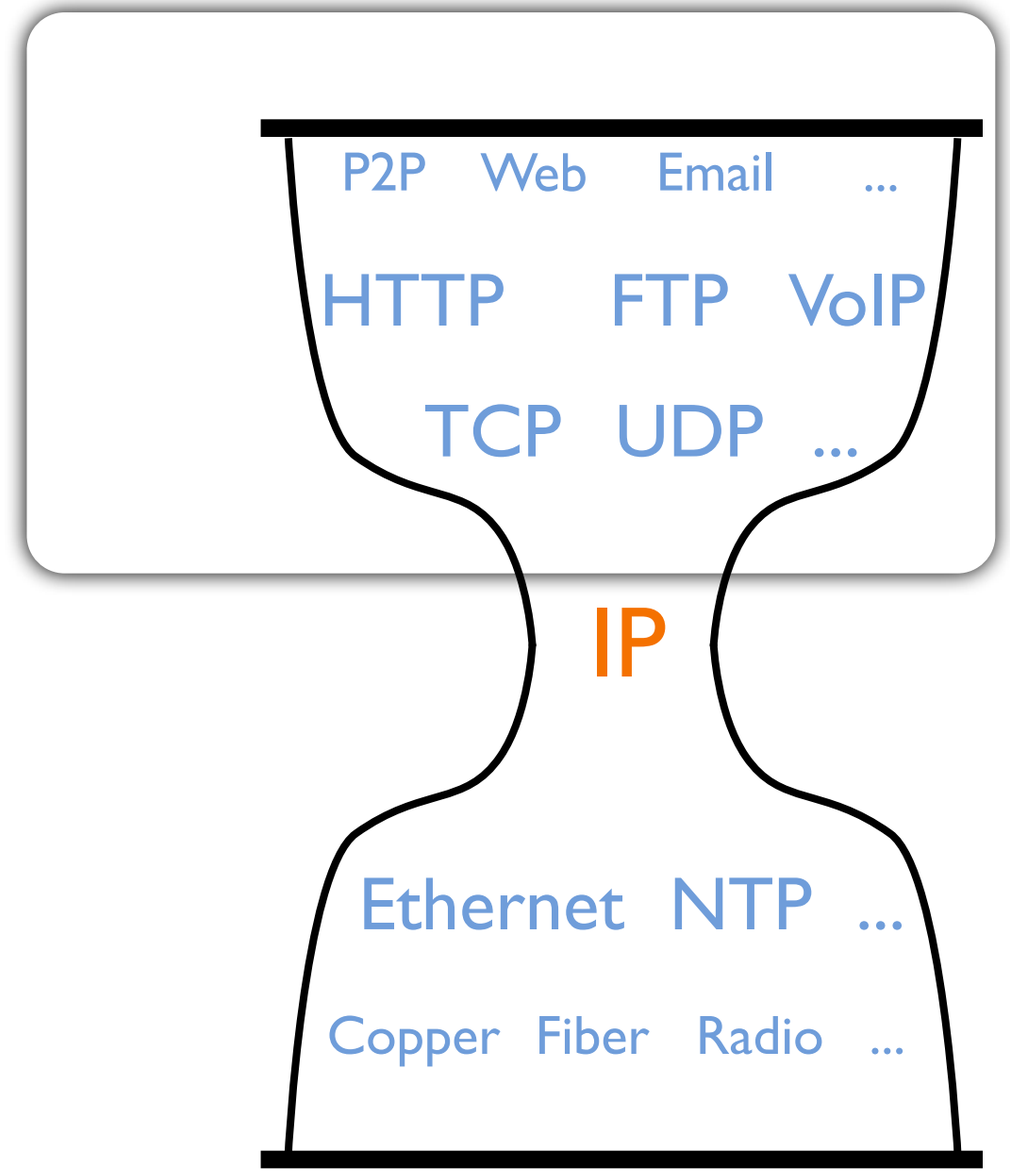
- But some apps

reliability!

(VoIP, XNET)

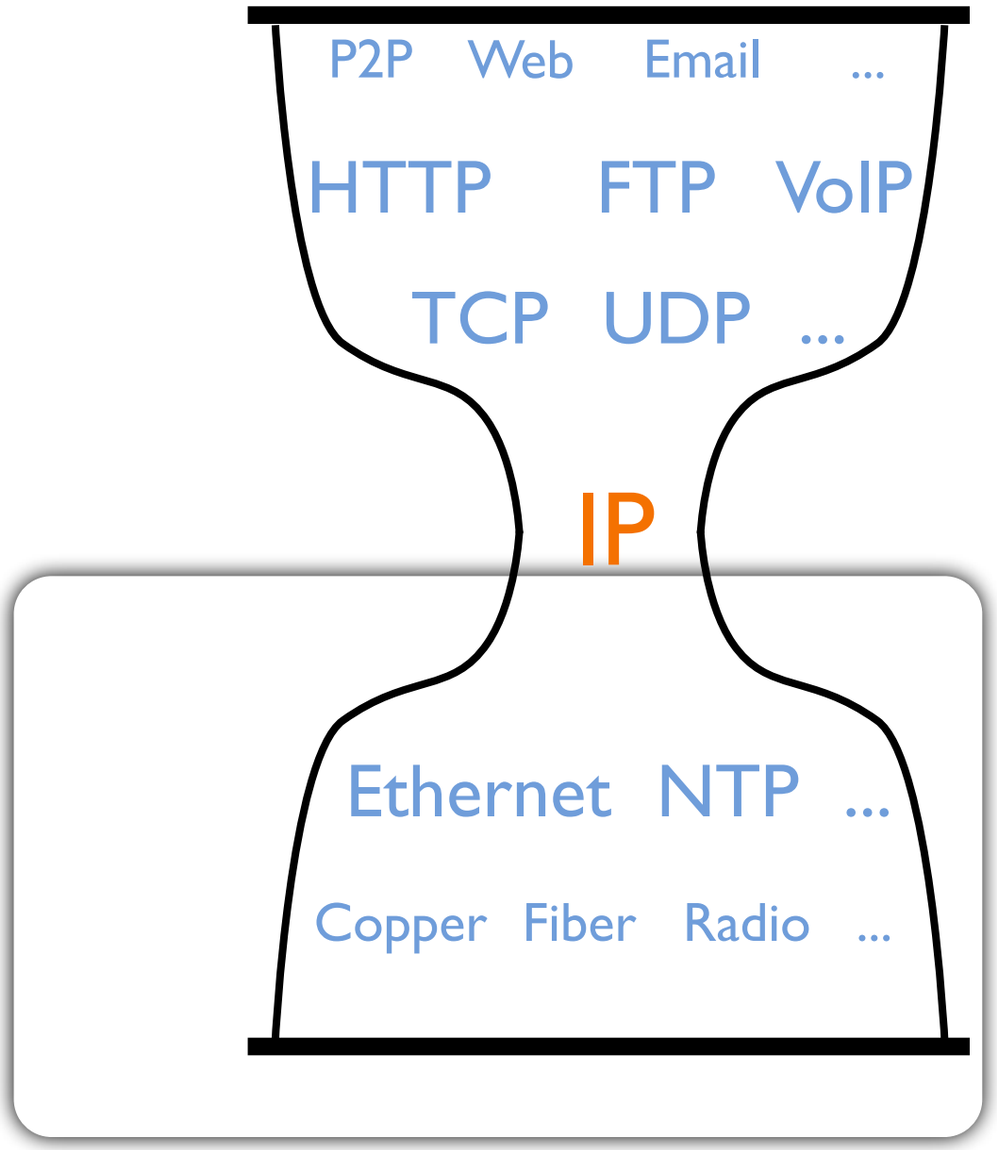
3. Multiple types of service

- So, TCP/IP split: datagram basic building block for many types of service
- Still difficult to support low latency across all networks
  - Hard to remove reliability if underlying network provides it



3. Variety of networks

- Datagram is simple building block
- Few requirements from underlying network technology



5. Distributed management

> *... some of the most significant problems with the Internet today relate to lack of sufficient tools for distributed management, especially in the area of routing.*

— David Clark, 1988

Still a problem 20 years later!

Later in this course:
easier-to-manage architectures (SDN)

6. Cost effective

- Inefficiencies:
  - 40 byte header
  - retransmission of lost packets
- How much does it matter now?

7. Easy host attachment

- End-hosts must implement net services
  - once caused concern to some people
  - problems if host misbehaves!

8. Accountability

- Difficult to account for who uses what resources
  - How is this done today? Why is it only an approximation?
  - Both an economic and security issue
  - Later in this course: Anderson et al., Accountable Internet Protocol, SIGCOMM 2008.

What it doesn't do

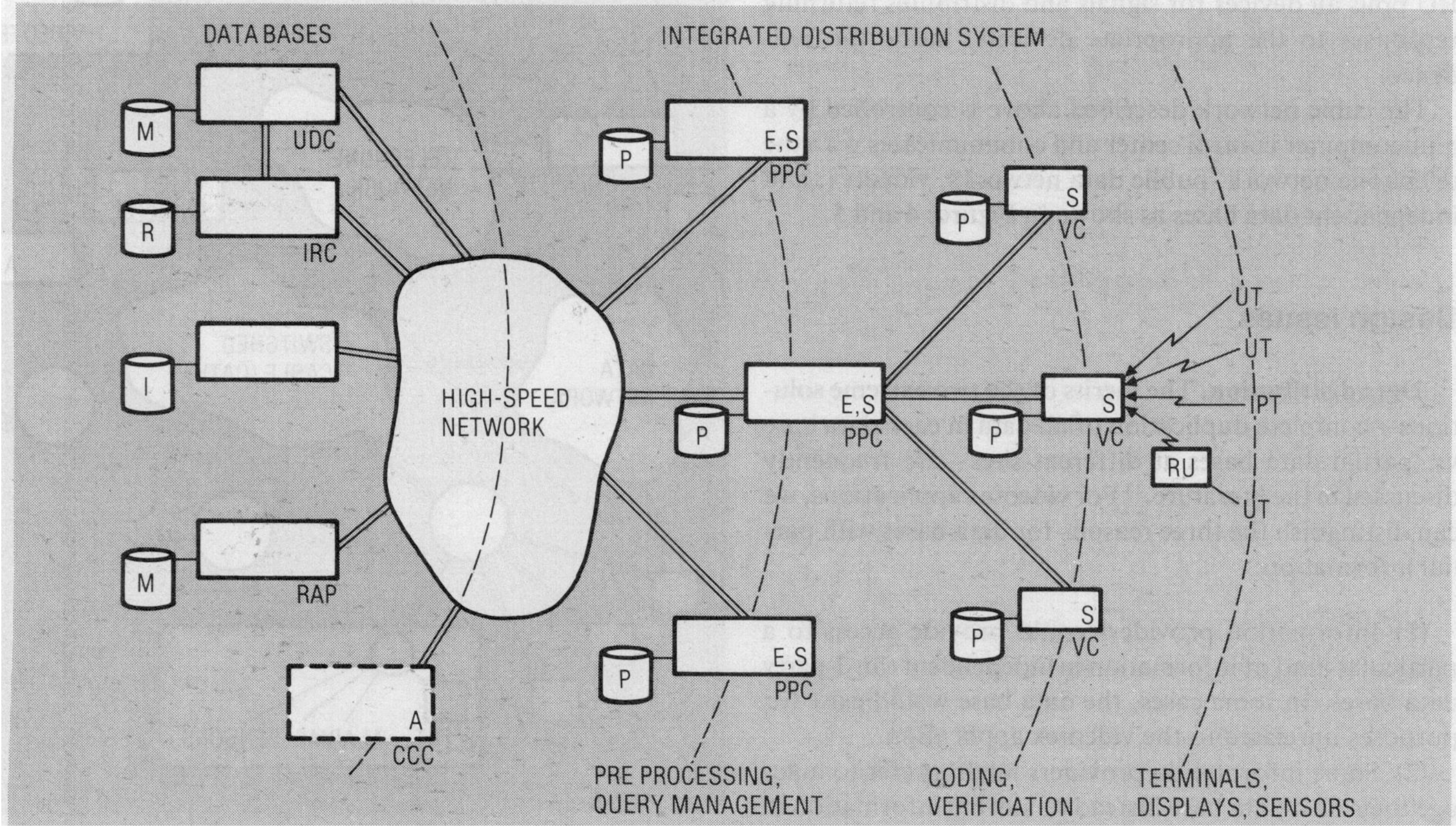
- *"The architecture tried very hard not to constrain the range of service which the Internet could be engineered to provide."*
  Extremely successful! But:
  - Hard for network to report that it failed ("potential for slower and less specific error detection")
  - Resource management (next week!)
  - Multipath
  - Full illusion of reliability during failures
- Security: Clark discusses host misbehavior (briefly) and accountability, but other aspects missing

Clark's new terms

fate-sharing

flow

soft state



What kind of system is this?

Discussion

- How would the network have been designed if the Internet were commercial?

A commercial 'internet'

- Different priorities: accountability first, survivability last
- Example: Videotex networks
- e.g., France Telecom's Minitel
  - Centralized
  - Reliable
  - Banking, news, stock transactions, ...

photo: wikimedia

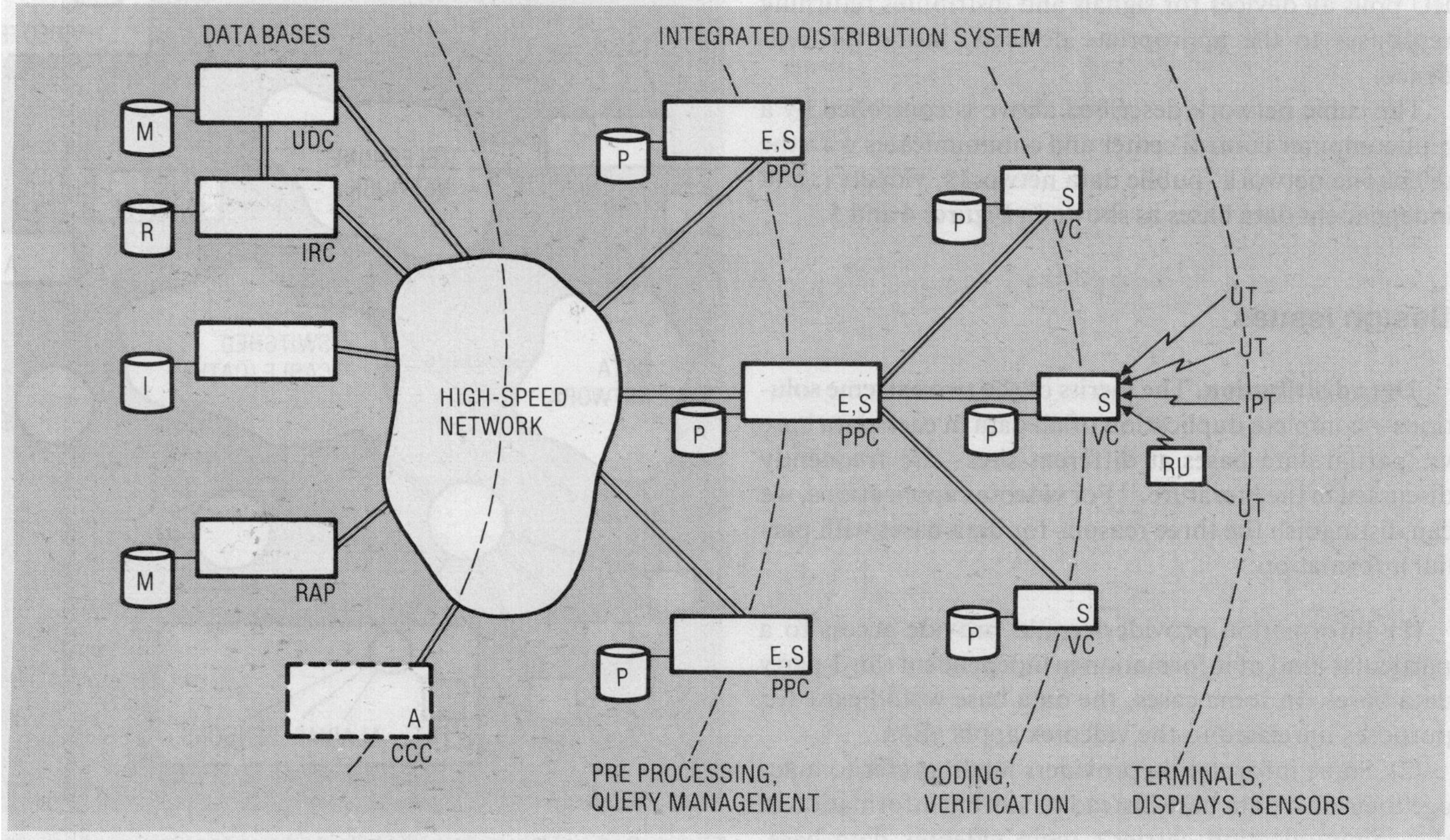


Figure 7. Generic Videotex system.

[A. J. S. Ball, G.V. Bochmann, and J. Gecsei. Videotex networks. IEEE Computer Magazine, 13(12):8–14, December 1980]

Today

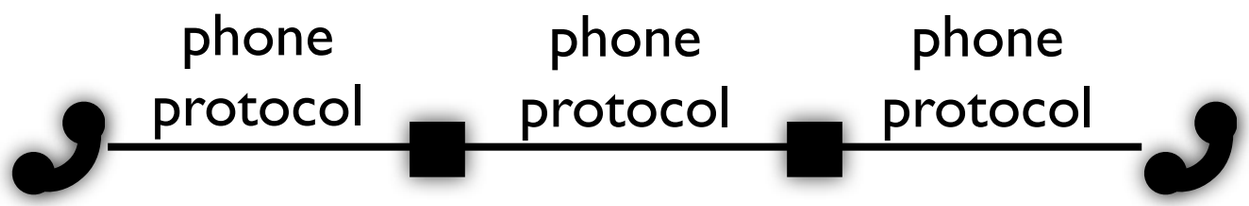
Clark: TCP / IP design philosophy

Architectural principles

Question from last week...

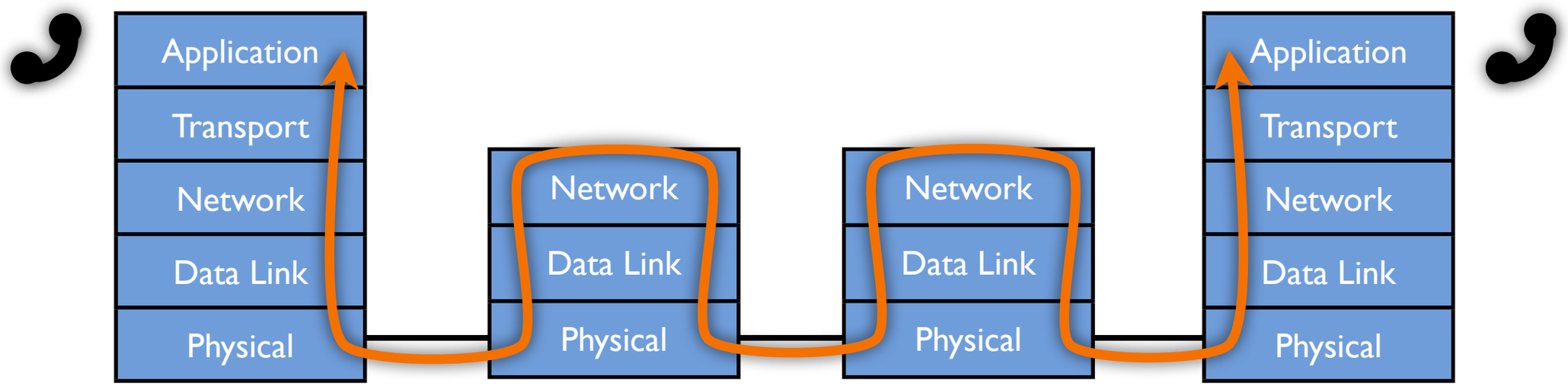
- What was the key to the diversity of innovation that the Internet enabled?
  - Packet switching for efficiency?
  - Packet switching for resilience to nuclear attack?
  - Ability to connect computers?
  - Government funding?
  - ...
- Let's take a step back (in time)

PSTN network architecture



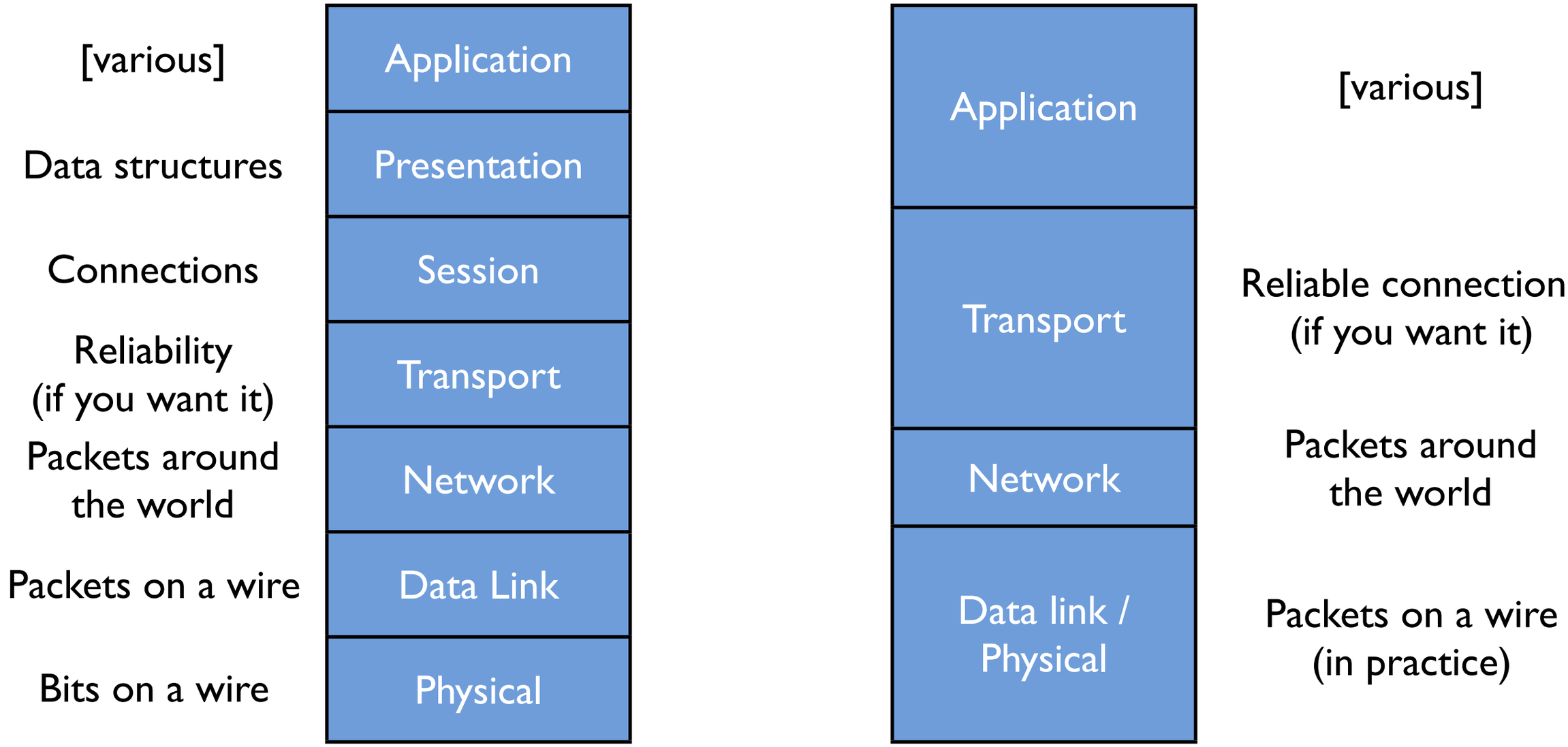
- One protocol spoken by all devices
- One application
- What principle changed this picture?

Layering



- A kind of modularity
- Functionality separated into layers
  - Layer *n* implements higher-level functionality by interfacing only with layer *n*-1
  - Hides complexity of surrounding layers: enables greater diversity and evolution of modules

OSI vs. TCP/IP



OSI Reference Model:
A committee thought
it looked good in theory

TCP/IP layers:
People found it met
their needs in practice

Layering before IP

- Layering in ARPANET
  - "Along with the basic host-host protocol, we also envisioned a hierarchy of protocols, with Telnet, FTP and some splinter protocols as the first examples. If we had only consulted the ancient mystics, we would have seen immediately that seven layers were required." – Stephen Crocker on the 1969 development of ARPANET [RFC1000, 1987]
- Layering in computer systems
  - examples?

Discussion

- Layering is a guiding principle, not a law
- When is layering violated? (layer *n* interacts with layers other than *n*-1 and *n*+1)
  - Web-based authentication for wireless networks
  - NATs
  - Web caches
  - ...
- Is there any connection between packet switching and layering? Does one require the other?

Organizing the layers

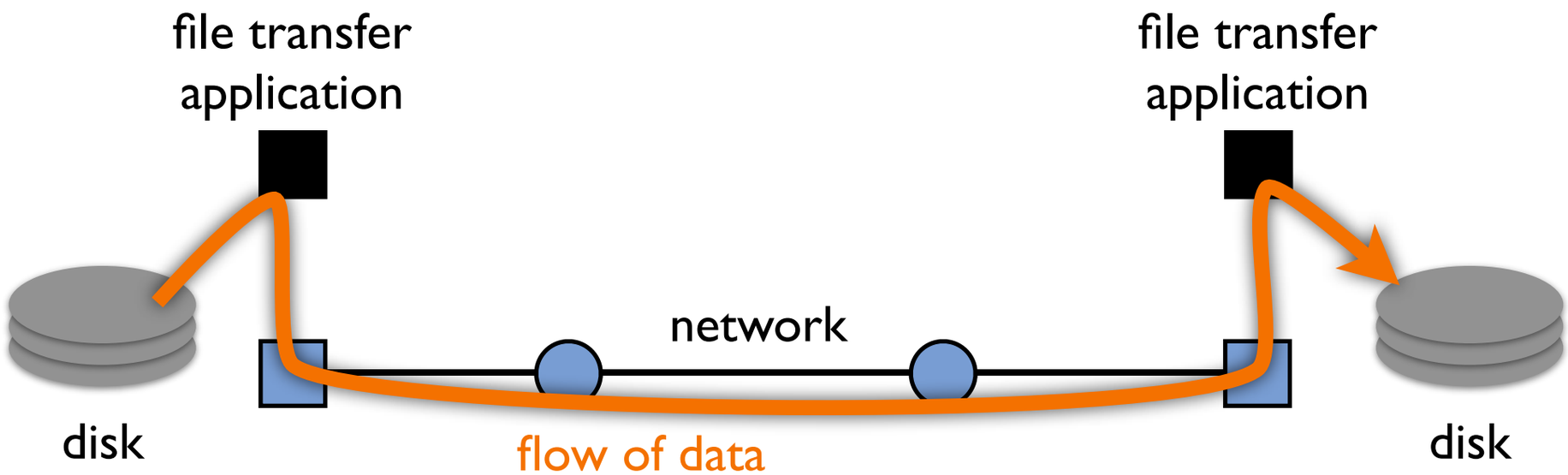
- Layering doesn't tell you what services each layer should provide
- What is an effective division of responsibility between layers?

End-to-end principle

(a slight rephrasing of the paper)

If a function can completely and correctly be implemented only with the knowledge and help of the application standing at the endpoints of the communication system, then providing that function as a feature of the communication system itself is not possible.

Example: file transfer



- Suppose the link layer is reliable. Does that ensure reliable file transfer?
- Suppose the network layer is reliable. Does that ensure reliable file transfer?

Interpretation

- Assume the condition (if ...) holds. Then...
- End-to-end implementation
  - Correct
  - Simplifies, generalizes lower layers
- In-network implementation
  - Insufficient
  - May help – or hurt – performance. Examples?
- Be wary to sacrifice generality for performance!

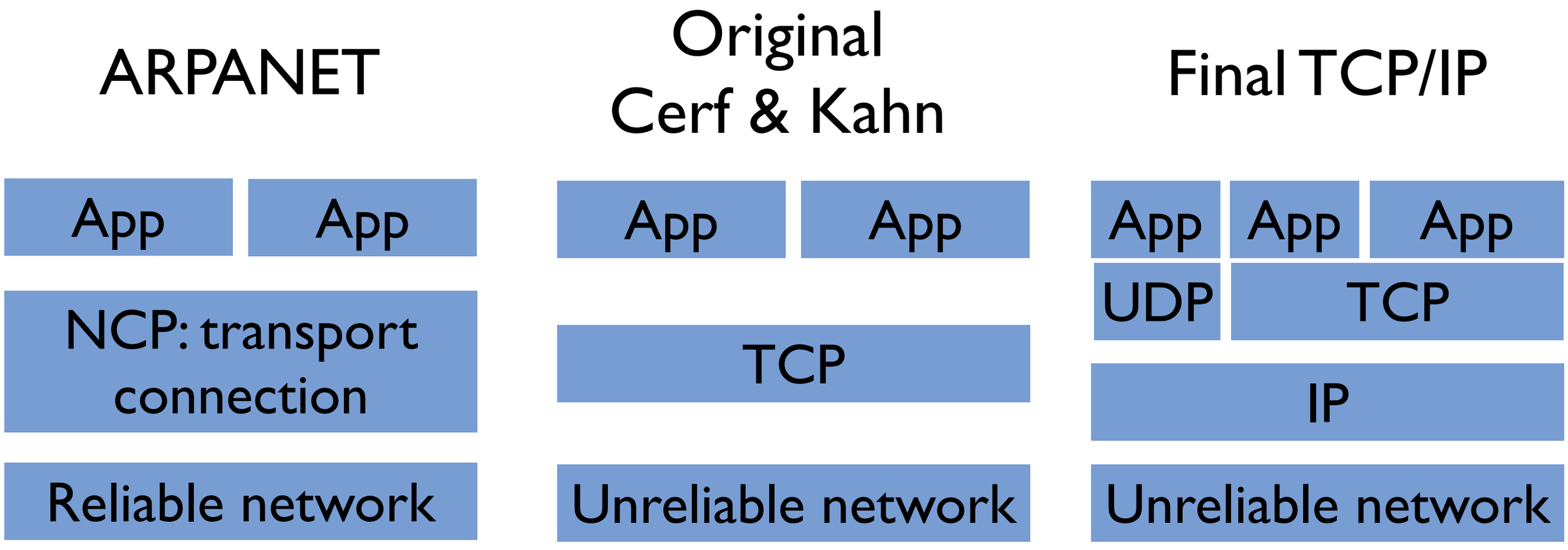
Where should these be?

- Failure avoidance
- Congestion control
- Routing
- Caching web requests

Discussion

- Layers change naturally over time
  - TCP won over OSI; HTTP as a new layer on TCP; Ajax as a new layer on HTTP
- Then, why do we spend so much time arguing about the details of layering?
  - IP is the universal layer that everyone agrees upon
  - Important to get this right to enable innovation below and above IP
  - There has been very little change in IP itself!

Implications for IP



We're done! ... right?

- Two main principles
  - Layering: a modular design
  - End-to-end: guides what the modules should do
- Is that a complete Internet architecture?
  - Resource management?
  - What are the right layers above, e.g. Naming?
  - Routing? Security? Interaction among entities? ...
- Internet experienced organic growth with fewer clear principles in other parts of the architecture