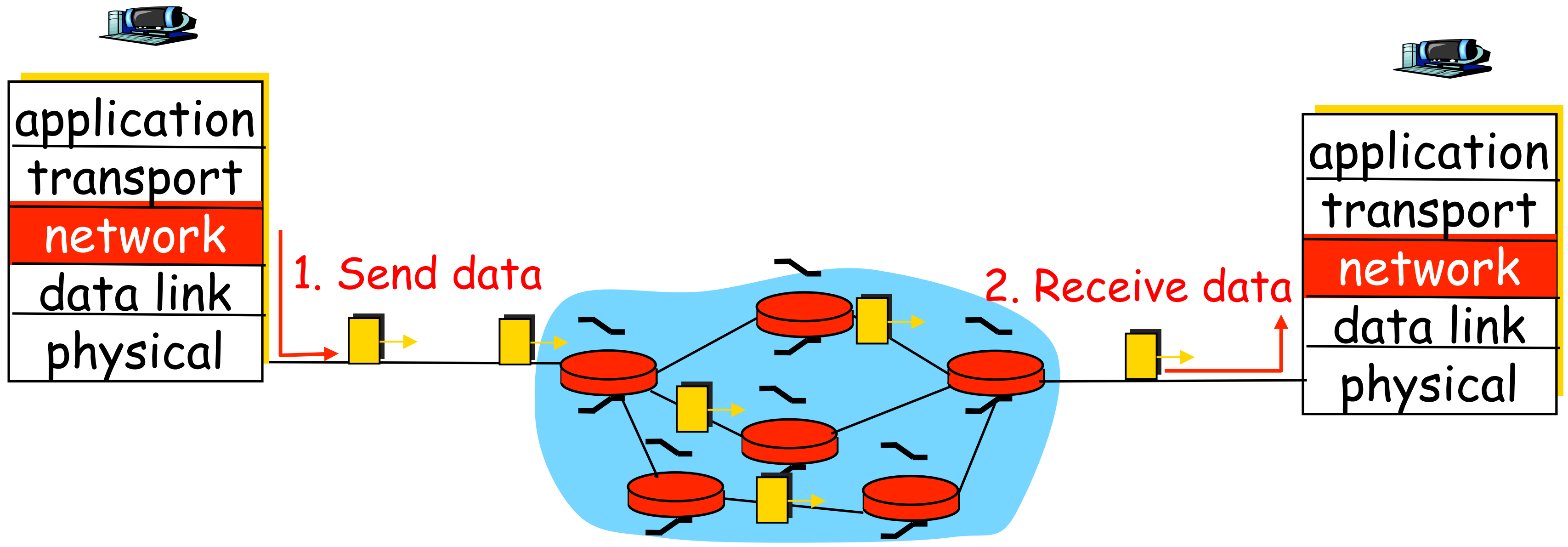
Last Lecture: TCP

1. *Multiplexing and Demultiplexing*
2. *Byte-stream service*
   - Stream of bytes sent and received, not stream of packets
3. *Reliable data transfer*
   - A combination of go-back-N and selective repeat, and performance tuning heuristics
4. *Connection management*
   - Connection establishment and tear down
5. *Flow control*
   - Prevent sender from overflowing receiver
6. *Congestion control* ✓
   - General principles & How TCP does it

This Lecture: Network Layer

1. *Design goals and issues* ✔
    - Debate around service model/design principle question
2. *Routing and Forwarding*
3. *Addressing, Fragmentation and reassembly*
4. *Internet Routing Protocols and Inter-networking*
5. *Router design*
6. *Congestion Control, Quality of Service*
7. *More on the Internet's Network Layer*

Network Layer’s Main Tasks



- Carry packets from a source (host) to a destination efficiently
- Provide “carrier” services to the transport layer

Basic Design Questions

- *What services to provide to the transport layer?*
  - Connection oriented vs connectionless
  - Any other Quality of Service (QoS) guarantee?

- *How to implement those services "efficiently" on top of various different intermediate networks?*
  - Routing
  - Forwarding (datagram, virtual circuit, source routing)
  - Addressing
  - Fragmentation and reassembly
  - Internetworking
  - Congestion control + congestion feedback (to transport)

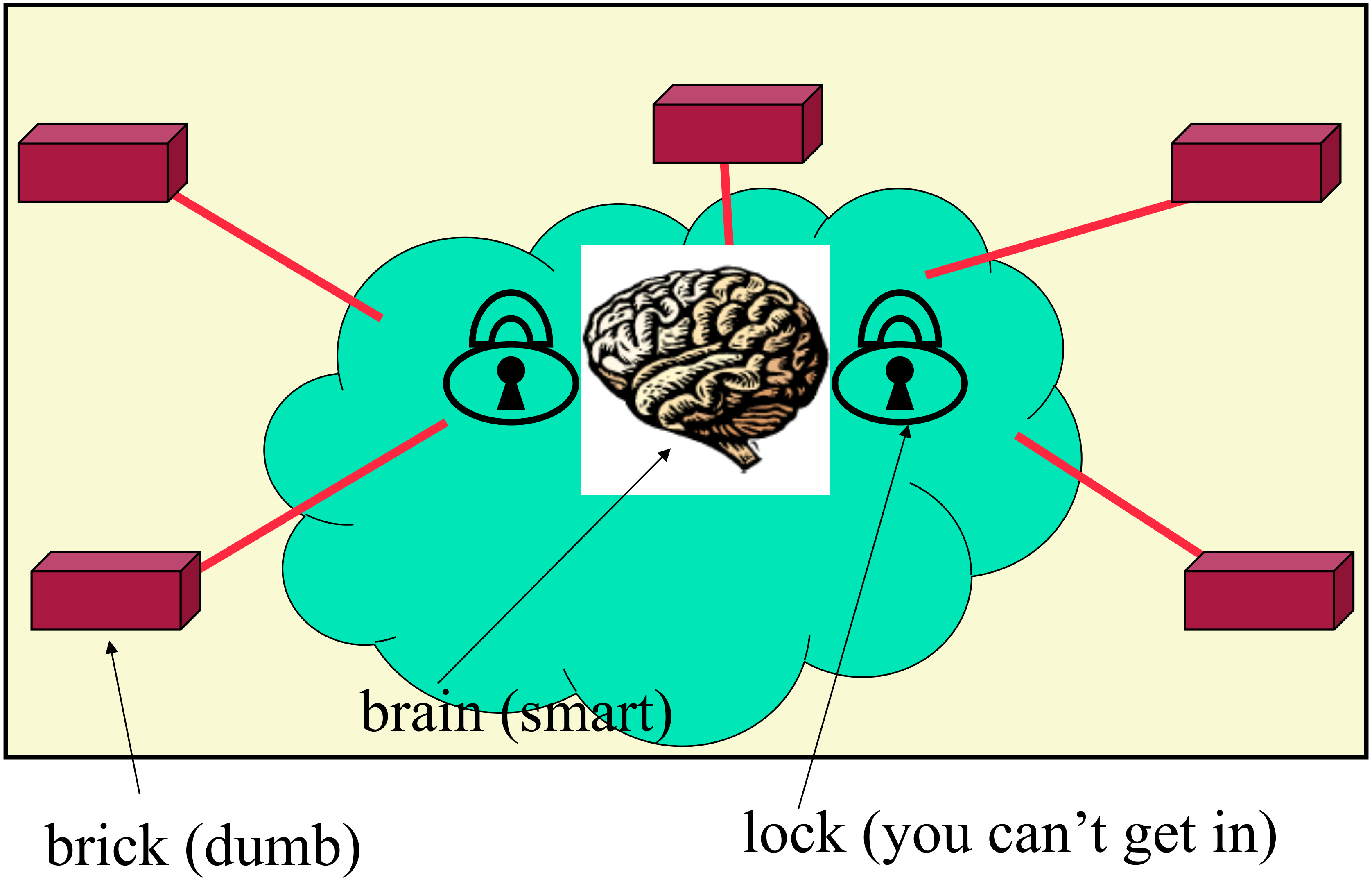
What Services to Provide?

Need a *service model*

- Connection oriented or connectionless?
  - This is host-to-host, not process-to-process like TCP
- Other QoS guarantee?
  - Bandwidth
  - Inter-packet timing (jitter)
  - Loss-less, in order delivery
  - …
- Security issues
- Performance-related "services"?
  - Congestion feedback to sender?
  - …

This question is a 40+ year old and on-going debate!

Telephone Community View

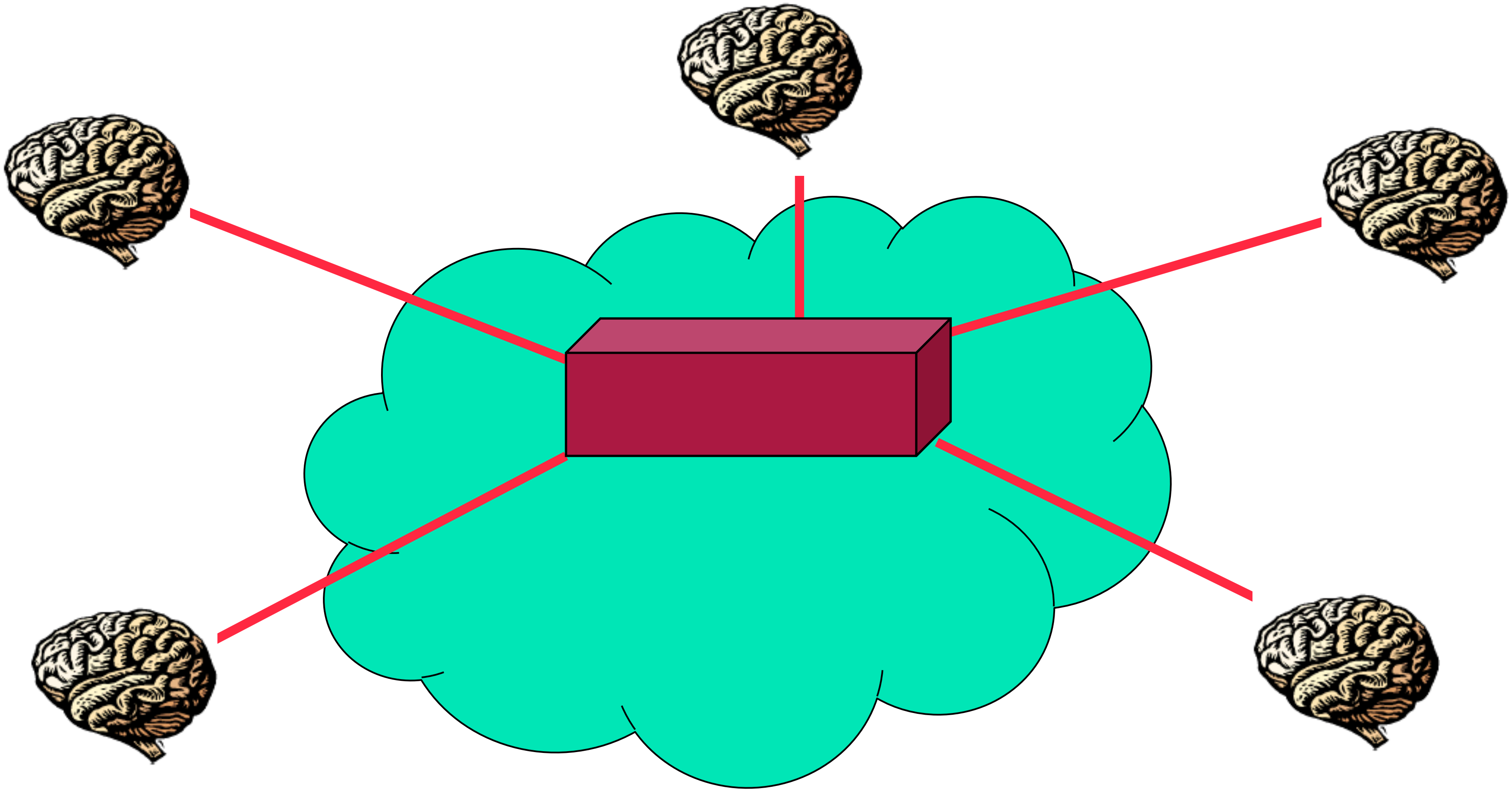


brain (smart)

brick (dumb)

lock (you can’t get in)

Internet Community View



The Debate

- The debate is more than "*telephone co. vs. freedom fighters*" – though some people feel that way

- The debate is about large system design philosophy

- Internet community loves the *End-to-End principle* of system design
  - Popularized by J. Saltzer, D. Reed, and D. Clark, 1984

E2E Arguments: Summary

- Specific application-level functions usually cannot and preferably should not be built into the lower levels of the system (the network core)

- Sometimes an *incomplete* version of the function provided by the communication system may be useful as a *performance enhancement*

- This philosophy is central in today Internet's design:
  - Functionalities are moved up and out of the core
  - *KISS principle*

E2E Arguments: Some Pros

- *Evolvability*:
  - No central authority imposing what kinds of applications can be developed
  - Easier to maintain backward and forward compatibility
  - “Simple” network layer makes it easier for IP to spread
- *Cost befenit*:
  - Applications that don’t need a particular feature do not have to pay the price (Turn this argument around?)
- *Flexibility, Adaptability, Simplicity*
  - Is it really?
- *Easier to model, describe, implement, and predict*
- *Philosophically pleasing* (liberalism, e.g.)

E2E arguments: Some Cons

- It has been very difficult to follow the philosophy
  - NAT, Firewall, Web caching, …
  - Design decisions are sometimes based on trust, responsibility or performance instead of E2E [*e.g., why is reliable transport not in app. layer? We can also do source routing, or congestion control*]
  - *Performance implications* are not justified by E2E
- New applications have been flourishing, but mostly those sensitive to the E2E design approach
- The "ends" may not be trust-worthy, and may be stupid (less sophisticated users)
  - Spams, DoS, Viruses, Worms, …

Problems Faced by Today’s Internet (1)

1. Untrustworthy world
   - End points can’t be trusted
   - Spam, viruses, worms, DoS, …
2. More demanding applications
   - Best effort can’t support MM apps
   - Might be possible (IntServ, DiffServ) but ISPs won’t cooperate
3. ISP service differentiation
   - ISPs do not want to collaborate to allow E2E implementation, they want ISP-specific services
   - Lead to closed islands of enhanced services

Problems Faced by Today's Internet (2)

3. Rise of third-party involvement
   - Officials of organizations (corporate networks, ISPs, …)
   - Officials of governments (Vietnam, …): law enforcement, political censorship, public safety, …
4. Less sophisticated users
   - Installation, configuration, upgrades, maintenance of complex end-system softwares require experts
   - End users want ease of use
   - Other dumb devices join the net (PDAs, sensors, watches, refrigerators, …)
7. Many more network types
   - Sensors, PDAs, other devices
   - Inter-planetary networks (DTN, e.g.)

Examples of new requirements

- Users communicate but don’t trust each other
  - Two parties want to negotiate a binding contract
  - Authentication
  - Communication with anonymity
- End parties do not trust their own hardwares, softwares
- The ends vs. the middle
  - Third party gets in the way of communications
  - E.g, should “traffic analysis” be allowed? How about firewalls? How about government reading your emails?
- Solving problems of spam, worms, phishing, …
- Multiway communications

Some Technical/Non-technical Solutions

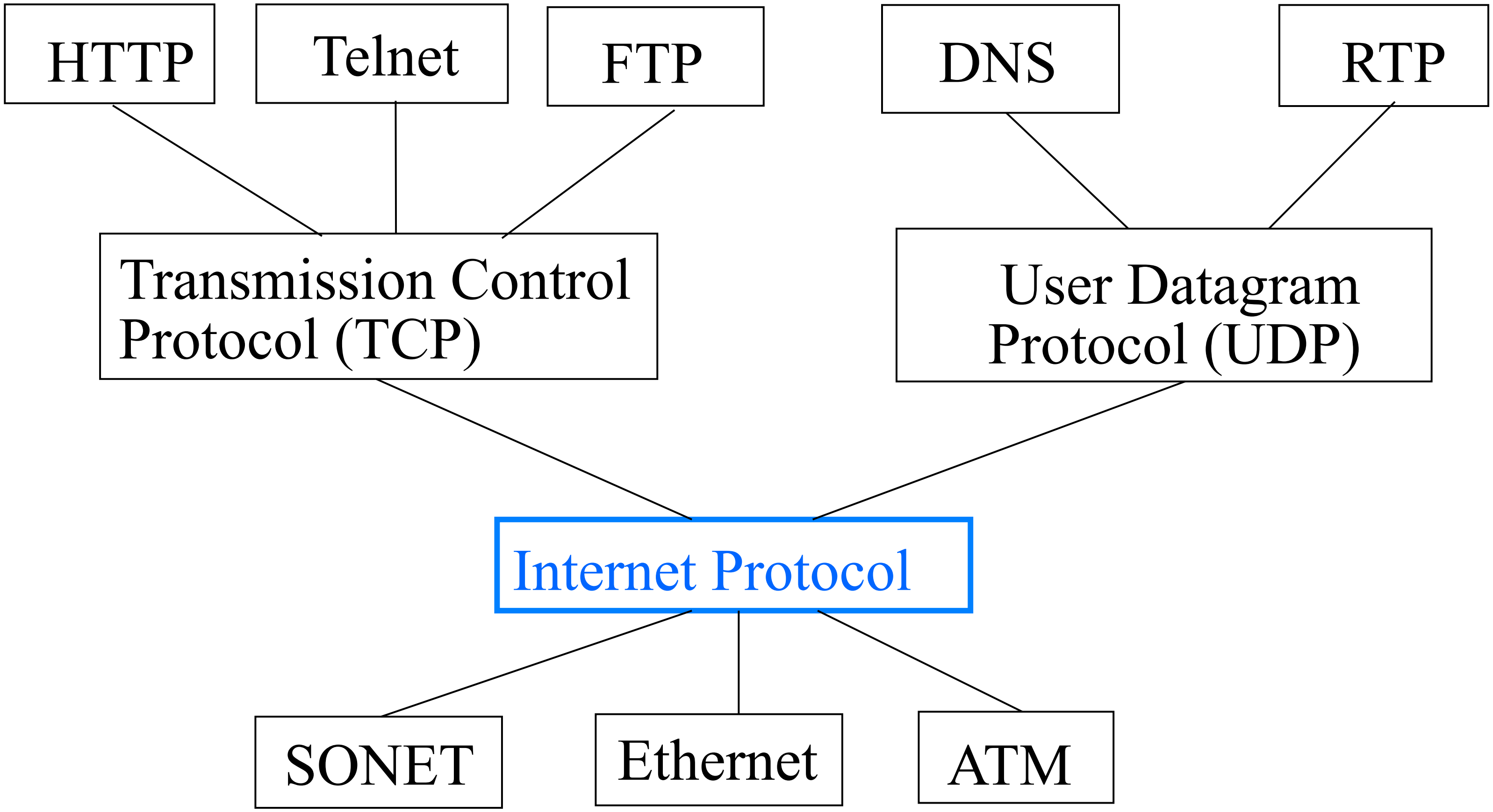
- More functionalities in the end nodes
  - Personal firewalls, filtering softwares
  - E2E smart MM applications (Real, WMP)
  - Use trusted third parties, more cryptographic communications (PGP and others)
- Adding functions to the core (deeply violate E2E)
  - Firewalls & other traffic filters
  - NAT elements
- Laws in cyberspace

Proposals for Re-Engineering the Internet

- Add a knowledge plane [Clark et al, 2003]
- Plutarch: network pluralism [Crowcroft et al, 2003]
- Role-based architecture [Braden et al, 2002]
- Triad Project [Stanford]
- …

- Your proposal?

Our Focus


HTTP
Telnet
FTP
DNS
RTP
Transmission Control Protocol (TCP)
User Datagram Protocol (UDP)
Internet Protocol
SONET
Ethernet
ATM