Internet Structure & Protocols

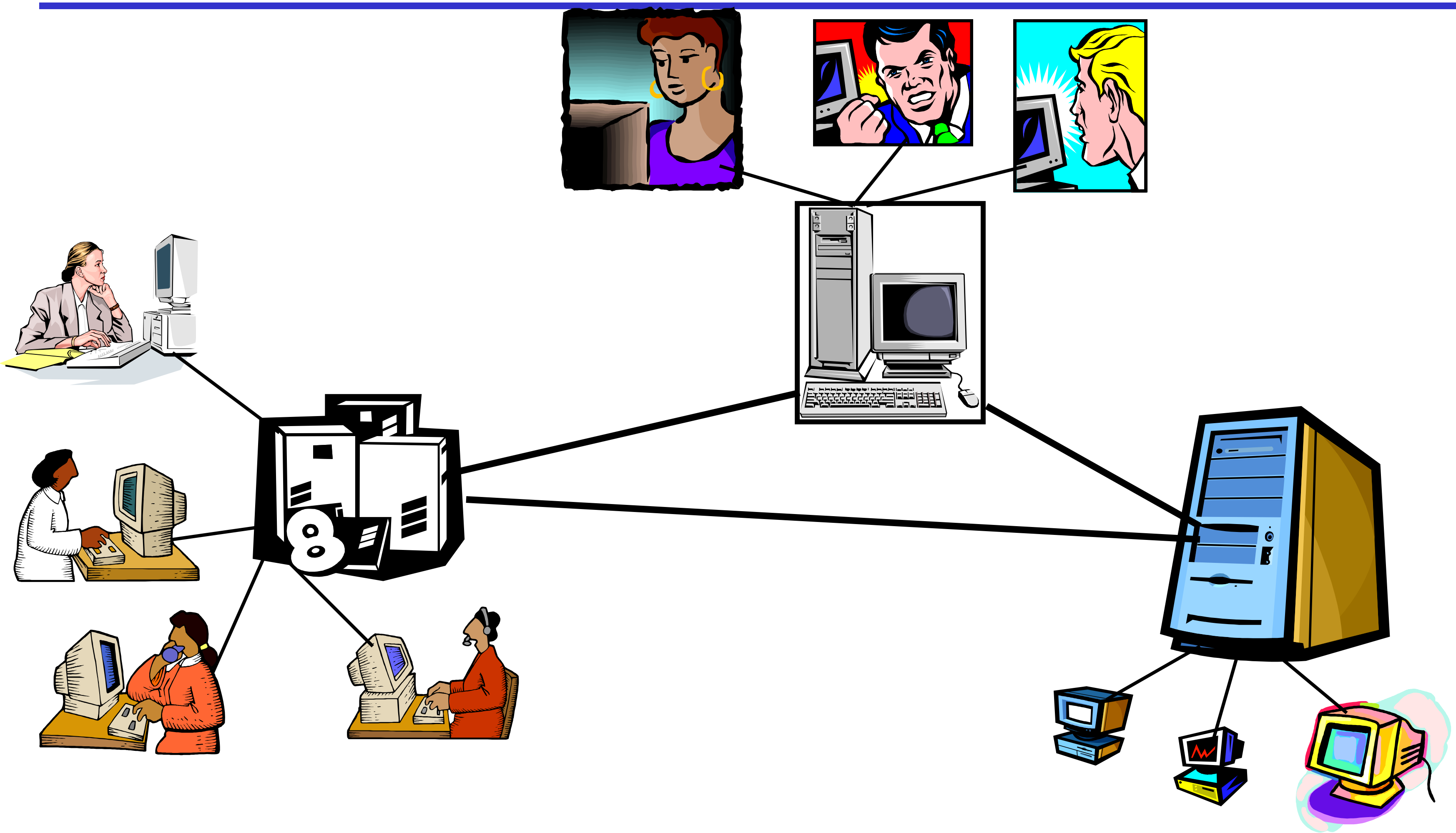
Emin Gun Sirer

Internetworking Origins

- Expensive supercomputers scattered throughout the US
- Researchers scattered differently throughout the US
- Need way to connect researchers to expensive machinery

- Point-to-point connections might have sufficed

Point to point connections



Internetworking Origins

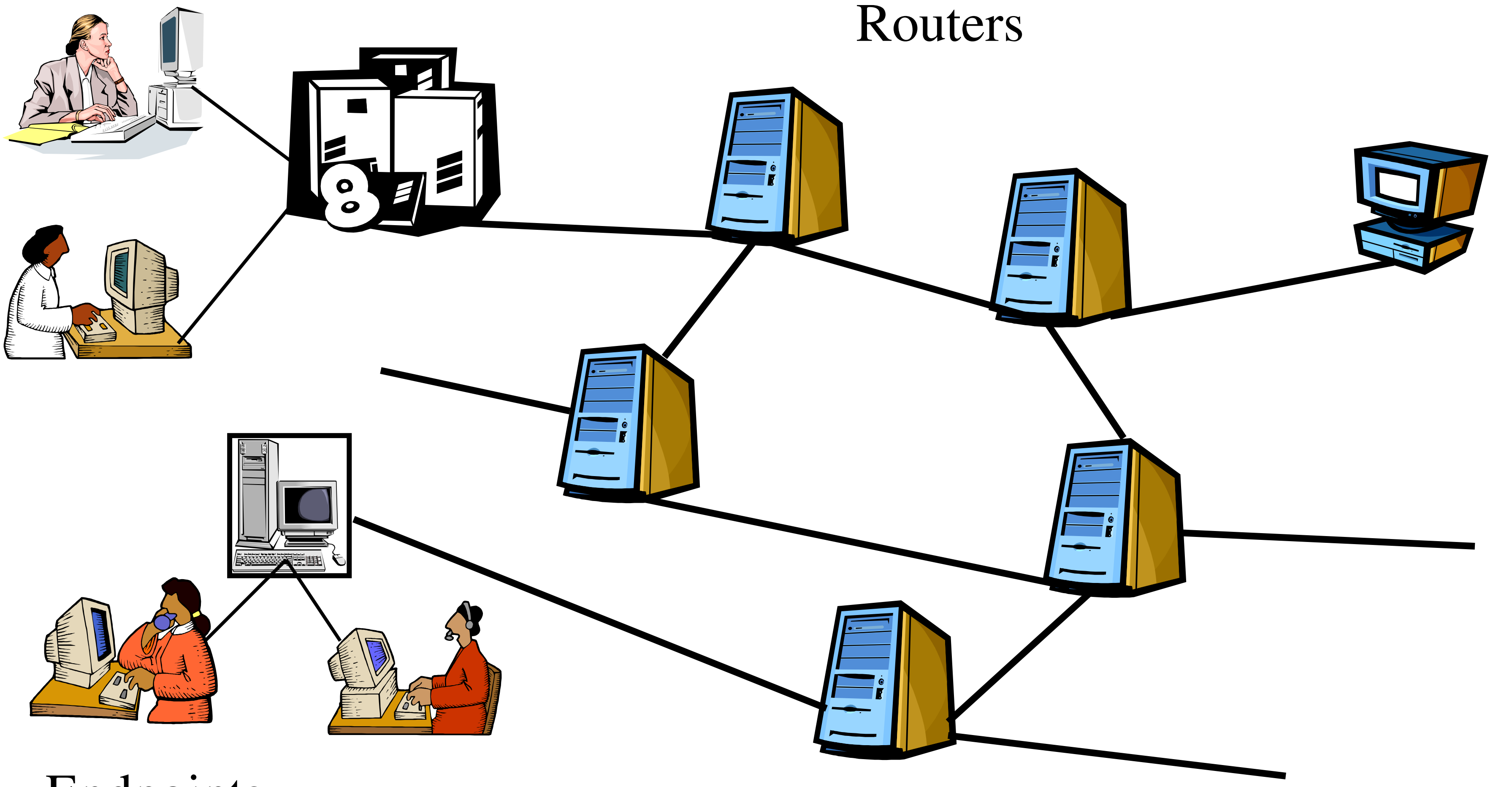
- Department of Defense initiated studies on how to build a resilient global network
    - How do you coordinate a nuclear attack ?
    - Especially, how do you tell people to stop firing missiles during a nuclear war ?
- Interoperability and dynamic routing are a must
    - Along with a lot of other properties
- Result: Internet
- A complex system with simple components

Internet Overview

- Every host is assigned, and identified by, an IP address
- Each packet contains a header that specifies the destination address
- The network routes the packets from the source to the destination

- Question: What kinds of properties should the network provide?

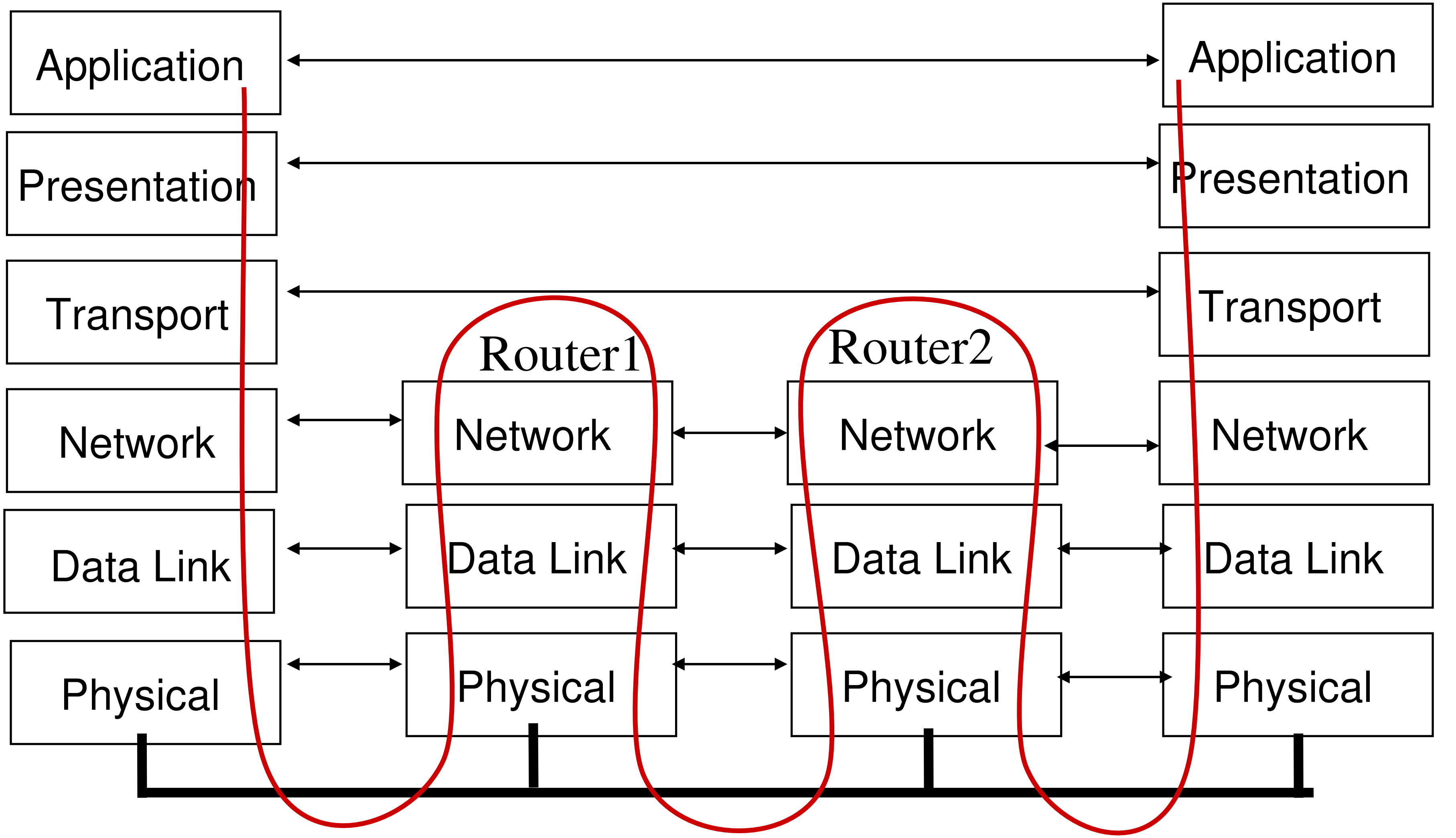
Internet, The Big Picture



Routers

Endpoints

The Big Picture



End-to-End Example

- Should the network guarantee packet delivery ?
    - Think about a file transfer program
    - Read file from disk, send it, the receiver reads packets and writes them to the disk
- If the network guaranteed packet delivery, one might think that the applications would be simpler
    - No need to worry about retransmits
    - But still need to check that the file was written to the remote disk intact
- A check is necessary if nodes can fail
    - Consequently, applications need to be written to perform their own retransmits
    - No need to burden the internals of the network with properties that can, and must, be implemented at the periphery

End-to-End Argument

- An Occam’s Razor for Internet architecture
- Application-specific properties are best provided by the applications, not the network
    - Guaranteed, or ordered, packet delivery, duplicate suppression, security, etc.
- The internet performs the simplest packet routing and delivery service it can
    - Packets are sent on a best-effort basis
    - Higher-level applications do the rest

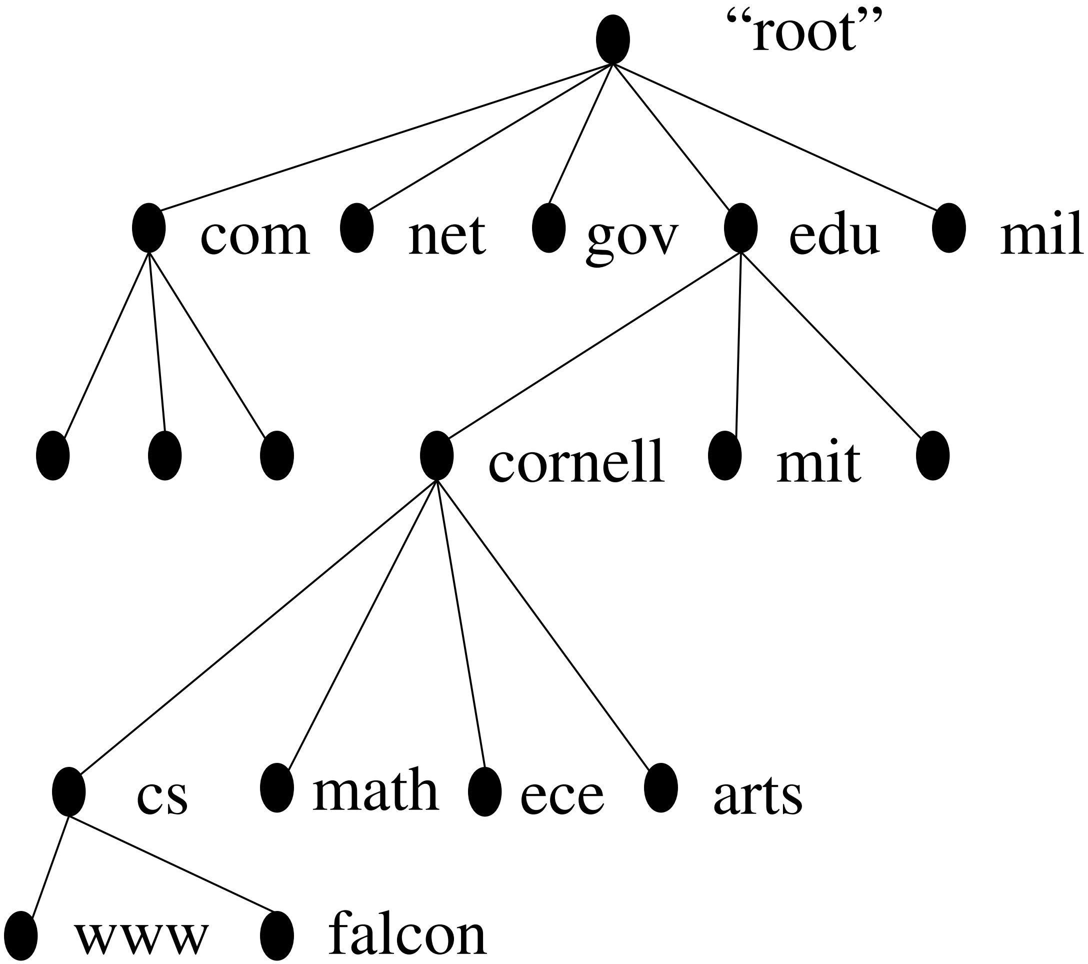
Naming

- Every host on the Internet is identified by an IP address
  - For now, 32-bit descriptor, like a phone number
  - Plans underway to change the underlying protocols to use longer addresses
- IP addresses are assigned to hosts by their internet service providers
  - Not physical addresses: IP address does not identify a single node, can swap machines and reuse the same IP address
  - Not entirely virtual: the IP address determines how packets get to you, and changes when you change your ISP
- Need completely virtual names
  - No one wants to remember a bunch of numbers

DNS

- Protocol for converting textual names to IP addresses
  - www.cnn.com = 207.25.71.25
- Namespace is hierarchical, i.e. a tree.
- Names are separated by dots into components
- Components are looked up from the right to the left

DNS Tree



- All siblings must have unique names
- Root is owned by ICANN
- Lookup occurs from the top down
- DNS stores arbitrary tuples (***resource records***)
- The address field contains the IP address, other fields contain mail routing info, owner info, etc.
- One field stores the cache timeout value

DNS Lookup

1. the client asks its local nameserver
2. the local nameserver asks one of the *root nameservers*
3. the root nameserver replies with the address of the authoritative nameserver
4. the server then queries that nameserver
5. repeat until host is reached, cache result.

DNS Lessons

- Simple, hierarchical namespace works well
    - Can name anything, can share names
- Scales OK
    - Caching
    - Even though it was meant to be hierarchical, people like short names, and use it like a flat namespace
- Arbitrary tuple database
    - Can delegate selected services to other hosts
- No security!
- Namespace = money
    - Innovations in this space are met with resistance from people who control name resolution

IP

- Internetworking protocol
    - Network layer
- Common packet format for the Internet
    - Specifies what packets look like
    - *Fragments* long packets into shorter packets
    - *Reassembles* fragments into original shape
- Some parts are fundamental, and some are arbitrary
    - IPv4 is what most people use
    - IPv6 clears up some of the messy parts, but is not yet in wide use

IPv4 packet layout

<table>
<tr><td>Version</td><td>IHL</td><td>TOS</td><td colspan="2">Total Length</td></tr>
<tr><td colspan="3">Identification</td><td>Flags</td><td>Fragment Offset</td></tr>
<tr><td colspan="2">TTL</td><td>Protocol</td><td colspan="2">Header Checksum</td></tr>
<tr><td colspan="5">Source Address</td></tr>
<tr><td colspan="5">Destination Address</td></tr>
<tr><td colspan="5">Options</td></tr>
<tr><td colspan="5">Data</td></tr>
</table>

IPv4 packet layout

<table>
<tr><td>Version</td><td>IHL</td><td>TOS</td><td colspan="2">Total Length</td></tr>
<tr><td colspan="3">Identification</td><td>Flags</td><td>Fragment Offset</td></tr>
<tr><td colspan="2">TTL</td><td>Protocol</td><td colspan="2">Header Checksum</td></tr>
<tr><td colspan="5">Source Address</td></tr>
<tr><td colspan="5">Destination Address</td></tr>
<tr><td colspan="5">Options</td></tr>
<tr><td colspan="5">Data</td></tr>
</table>