Announcements

- Reminder: Homework #2

Internet in a nutshell (protocols in practice)

The gory details first

- The Internet from your computer's view
- Packet-level traces of what happens when you access a web page
- Might not make sense today

What you need to assume

- Every host has a network card with a globally unique, 48-bit hardware address typically expressed as 12 hex digits.
  - ipconfig /all   OR   ifconfig –a
  - This network card = 00-0E-9B-90-1C-50
- Hop-by-hop link layer communication is done via these hardware addresses.
  - Payload may have an IP packet
  - You must know the hardware address of the next hop in order to send a packet there
  - Special hardware broadcast address for discovery

What you need to assume

- Every host has a unique 32-bit IPv4 address (or 128-bit IPv6 address) typically expressed as 4 numbers from 0-255
  - Portland State = 131.252.x.x
  - This machine =
  - Completely decoupled from hardware addresses
  - Structured like postal addresses.
- Every network packet has a source and a destination IP address
  - Routers collaborate to deliver packets based on their destination IP address
- DNS servers collaborate to map names (i.e. www.google.com) to IP addresses (72.14.213.103)

How protocols and packets are structured

- Recall previous lecture
  - Layering of functionality
  - Packets structured according to layers
- Russian doll analogy
  - Innermost doll = Application data (i.e. URL request or web page)
  - Next layer = Transport information (i.e. process address or packet sequence number)
  - Next layer = Network information (i.e. network source and destination addresses)
  - Outermost doll = Data-link layer information (i.e. hardware source and destination addresses)

How protocols and packets are structured

- Mail analogy
  - Application data (i.e. URL request or web page)
    - Contents of a letter
  - Transport information (i.e. process address or packet sequence number)
    - Recipient: Person, Dorm room #, Apt. #
    - Carrier: USPS, UPS, DHL, FedEx
  - Network information (i.e. network source and destination addresses)
    - Street address, City, State, Zip code
  - Data-link layer information (i.e. hardware source and destination addresses)
    - Vehicle or person transporting the mail

How protocols and packets are structured

- Operation
  - End host (web client) creates entire doll (app, transport, network, data-link) and sends it to "next hop"
  - Router pulls off outermost doll (data-link), examines destination address of "network layer", and looks up the "next hop" based on it
  - Creates another outer, data-link layer doll, places the packet within it, and sends it to the next hop's network interface.
  - Eventually reaches other end system (web server) which processes all layers to obtain the request

A day in the life of an Internet host…

- Booting
  - Dynamically configure network settings
    - DHCP request (Dynamic Host Configuration Protocol)
      - UDP (unreliable datagrams)
      - IP and data-link broadcast

Datalink broadcast header	IP broadcast 255.255.255.255	UDP header	DHCP request Host’s datalink (MAC) address 00:50:7e:0d:30:20

-
  -
    - DHCP response from listening server
      - IP address for host to use
      - Netmask (i.e. 255.255.255.0) to determine who is directly connected
      - Default router
      - Local DNS server

Datalink header 00:50:7e:0d:30:20	IP of Host	UDP Header	DHCP reply Host’s network settings

A day in the life of an Internet host…

- Web request http://www.yahoo.com/index.html
  - Step #1: Locate DNS server

    if (DNS server is directly connected) {

    DNS server on local network

    ARP for hardware address of IP_{DNS}

    } else {

    DNS server on remote network

    ARP for hardware address of $IP_{DefaultRouter}$

    }

    - ARP (Address Resolution Protocol)
      - IP address to hardware address mapping
      - Request broadcast for all hosts on network to see
      - Reply broadcast for all hosts to cache

A day in the life of an Internet host…

- Step #2: ARP request and reply

Datalink header broadcast	ARP request: Who has MAC address of IP addr "X"? (X=next-hop router, dns server) MAC address of requestor

Datalink header MAC of requestor or broadcast addr	ARP reply: MAC address of "X" is a:b:c:d:e:f

A day in the life of an Internet host…

- Step #3: DNS request/reply
  - UDP, IP, data-link header
  - DNS request to local DNS server from host

Datalink header (DNS server or next-hop router)	IP of DNS Server	UDP Header	DNS request www.yahoo.com "A" record request

  - DNS reply from local DNS server to host

Datalink header (host)	IP of host	UDP Header	DNS reply www.yahoo.com is 216.115.105.2

A day in the life of an Internet host…

- Step #4: TCP connection establishment
  - TCP 3-way handshake (SYN, SYN-ACK, ACK)
  - Session establishment to support reliable byte stream

Datalink header (next-hop router)	IP of 216.115.105.2	TCP Header SYN

Datalink header (host)	IP of host	TCP Header SYN-ACK

Datalink header (next-hop router)	IP of 216.115.105.2	TCP Header ACK

A day in the life of an Internet host…

- Step #5: HTTP request and reply
  - HTTP (application data), TCP, IP, data-link header
  - HTTP request

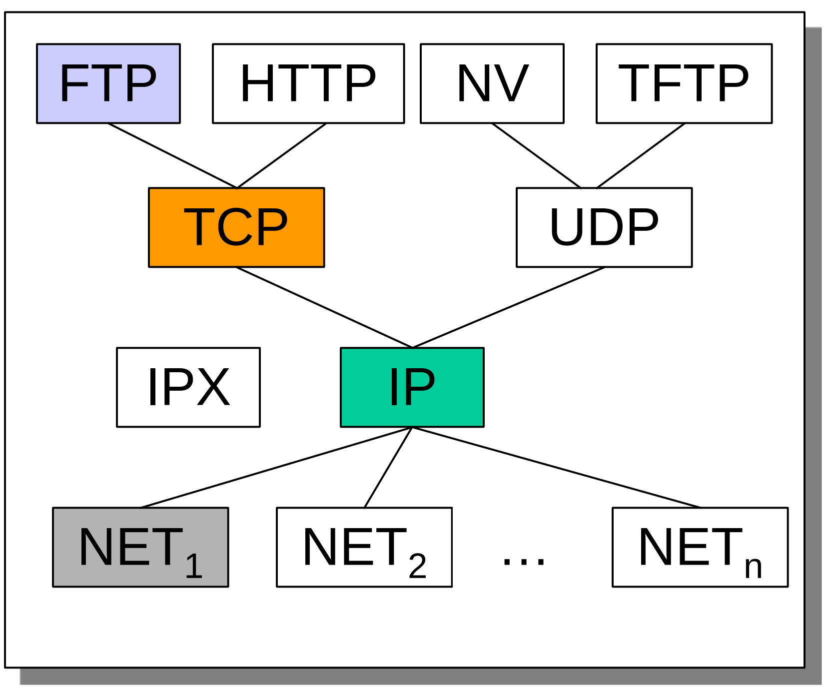
Datalink header (next-hop router)	IP of 216.115.105.2	TCP Header	HTTP request GET /index.html HTTP/1.0

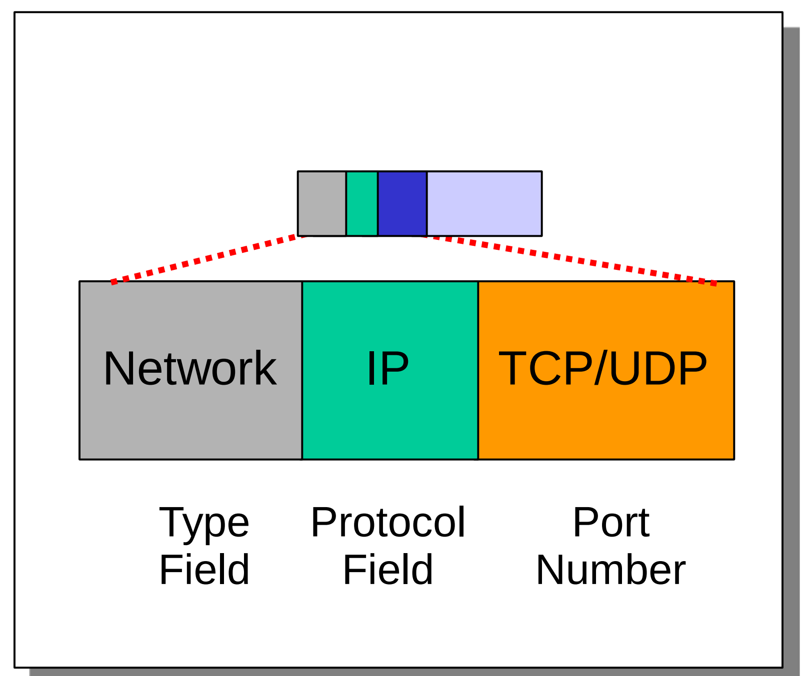
  - HTTP reply

Datalink header (host)	IP of host	TCP Header	HTTP reply HTTP/1.0 200 OK Date: Mon, 24 Sep 2001 Content-Type: text/html <html> …. </html>

A day in the life of an Internet host…

- Role of TCP and UDP?
- Demultiplex at end hosts.
  - Which process gets this request?





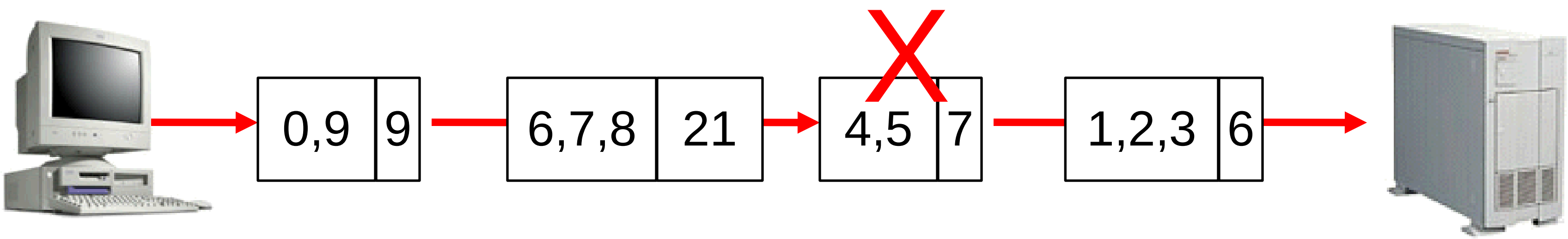
A day in the life of an Internet host….

- What about….
  - Reliability
    - Corruption
    - Lost packets
  - Flow and congestion control
  - Fragmentation
  - Out-of-order delivery
- The beauty of TCP, IP, and layering
  - All taken care of transparently

What if the Data is Corrupted?

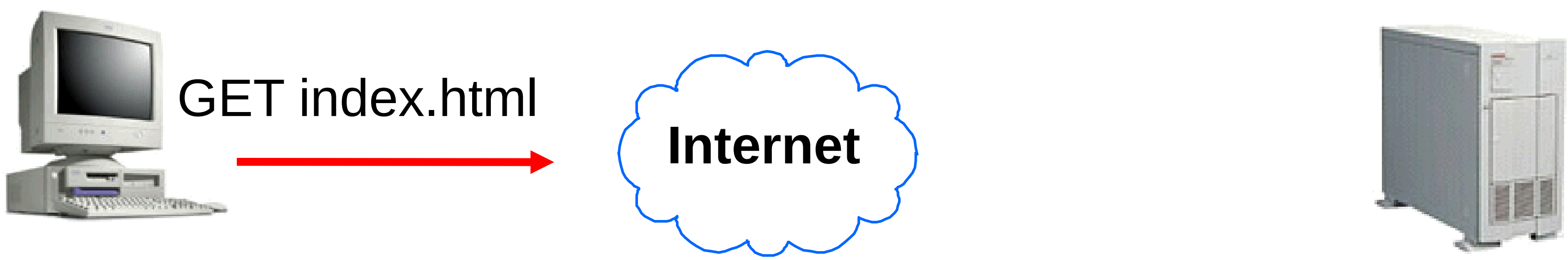
Problem: Data Corruption

Solution: Add a *checksum*

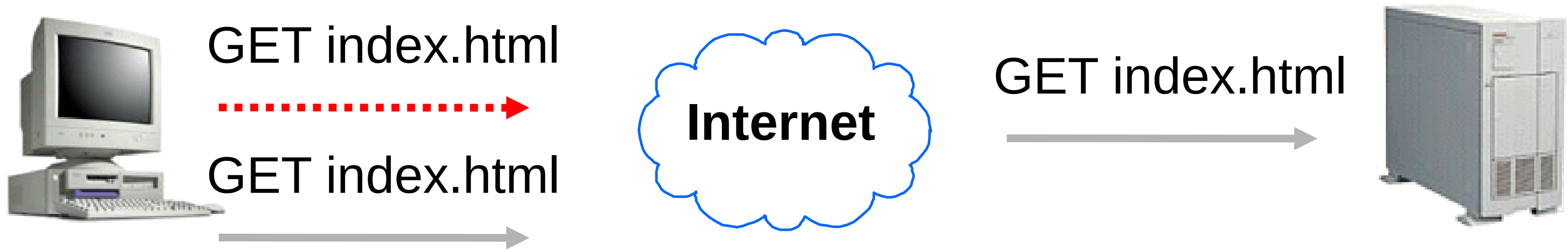


What if the Data is Lost?

Problem: Lost Data

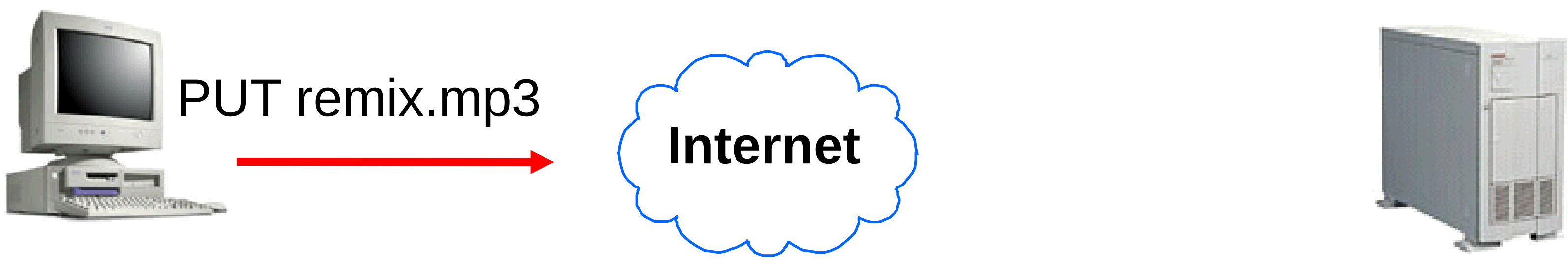


Solution: Timeout and Retransmit



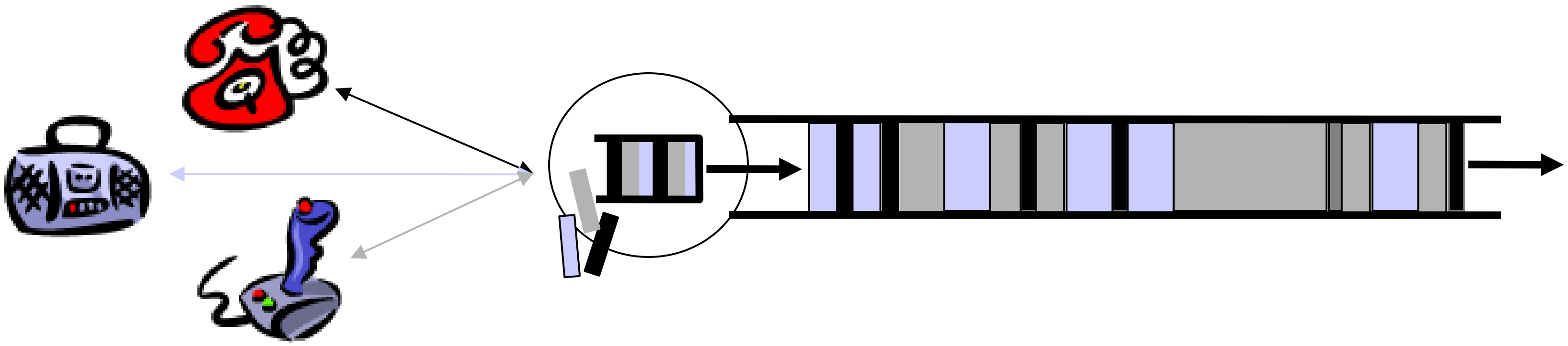
What if receiver has no resources (flow control)?

Problem: Overflowing receiver buffers



Solution: Receiver advertised window

What if Network is Overloaded?



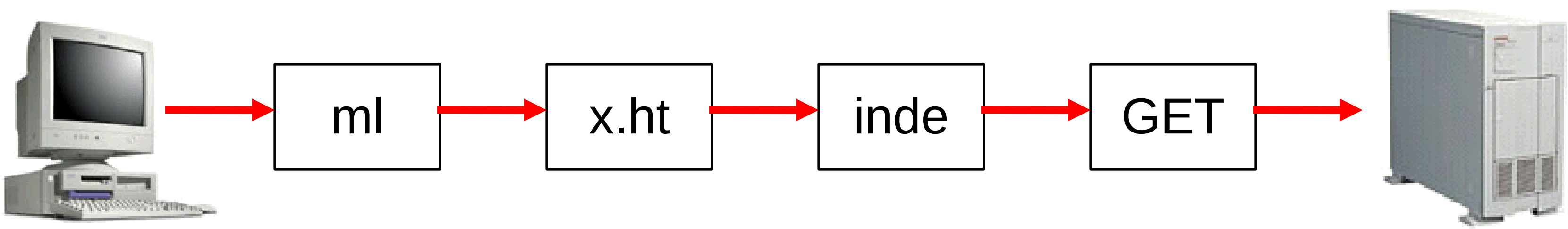
- Short bursts: buffer
- What if buffer overflows?
  - Packets dropped and retransmitted
  - Sender adjusts rate until load = resources
- Called "Congestion control"

What if the Data Doesn't Fit?

Problem: Packet size

- On Ethernet, max IP packet is 1.5kbytes
- Typical web page is 10kbytes

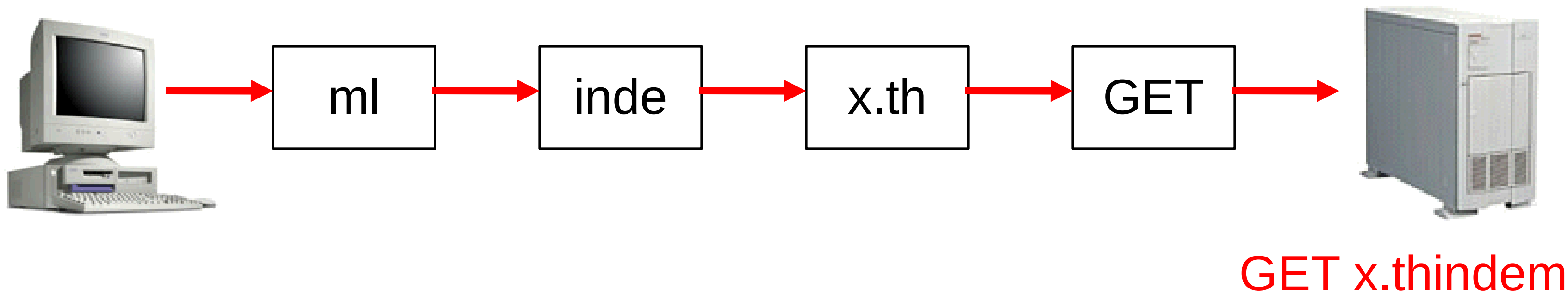
Solution: Fragment data across packets



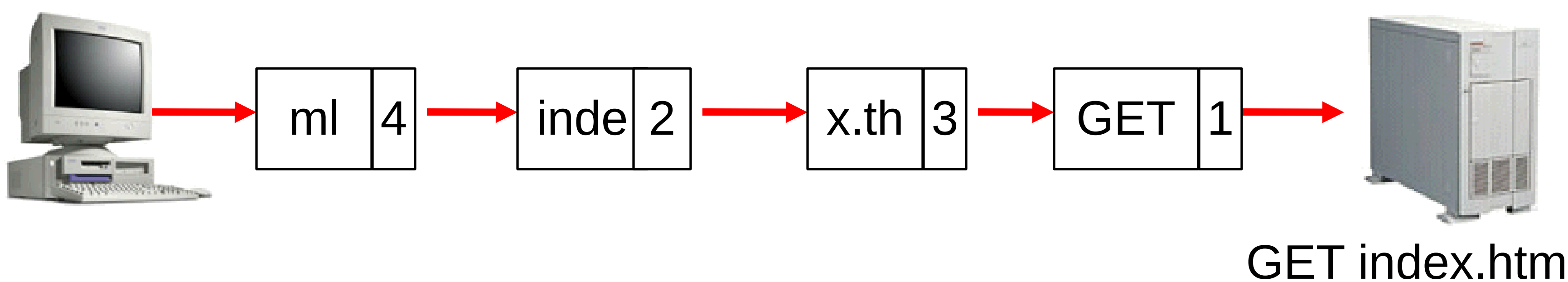
GET index.html

What if the Data is Out of Order?

Problem: Out of Order



Solution: Add Sequence Numbers



tcpdump example

- http://thefengs.com/wuchang/work/courses/cs347u/trace.txt