15: Networking Basics

Last Modified:
7/3/2004 1:47:08 PM

-1

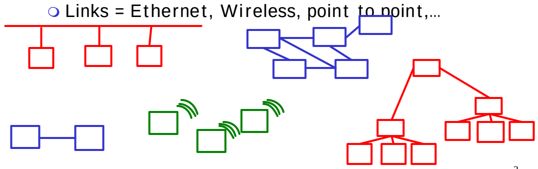
Networking

- So far we have talked primarily about OS support for individual computer systems
- Today we are going to talk about networking computer systems together

-2

A Network

- A network is simply a collection of nodes, connected by links, that communicate and cooperate
  - Nodes = End Hosts (PCs, PDAs, toasters?), Internal Nodes (Routers, switches, hubs,..)
  - Links = Ethernet, Wireless, point to point,...



-3

Questions

- What will be the format of data exchanged? How do we agree on a language among all kinds of nodes?
- Transmission across links is faulty can corrupt/lose data. How can we reliably exchange information?
- How do we find the right path between two nodes? If there are many how do we choose the best one?
- How do nodes refer to one another or address one another?
- What is the operating systems role in all this?

-4

Communication?

- If two entities are going to communicate, they must agree on the expected order and meaning of messages they exchange.
- Asking for the time protocol
  - SUCCESSFUL PROTOCOL EXCHANGE
    - Hi ...Hi...Got the time?....two oclock
  - ABORTED PROTOCOL
    - Hi...Don't bother meXX
  - PROTOCOL MISMATCH
    - Allo...Hello..Quelle heuere a'til .....XX<blank stare>

-5

Protocol

- Defines the format and the order of messages exchanged between communicating entities
- Defines the actions expected to be taken on the receipt or the transmission of a message

-6

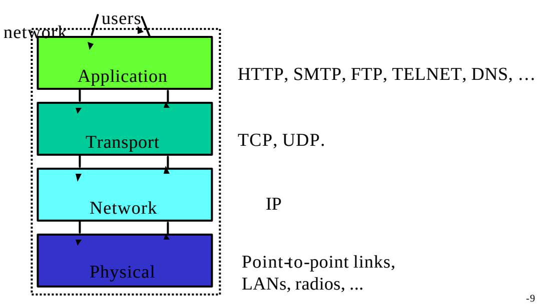
Networking protocols

- Ok lets define the "language" for all interactions over the network??
  - One single language that can support everything from web browsing to email to ftp to distributed file systems?
- Human beings are able to handle lots of complexity in their protocol processing.
  - Ambiguously defined protocols
  - Many protocols all at once
- How do computers manage complex protocol processing?

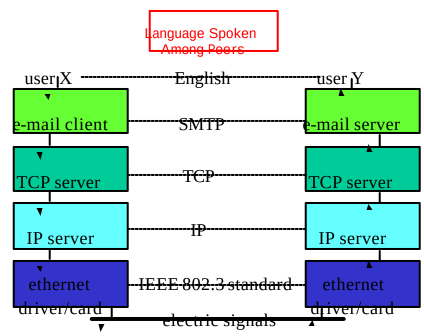
Layered Architectures

- Break-up design problem into smaller, more manageable problems
  - Layers
- Design protocols to support each well defined task
  - Not one language for everything!!

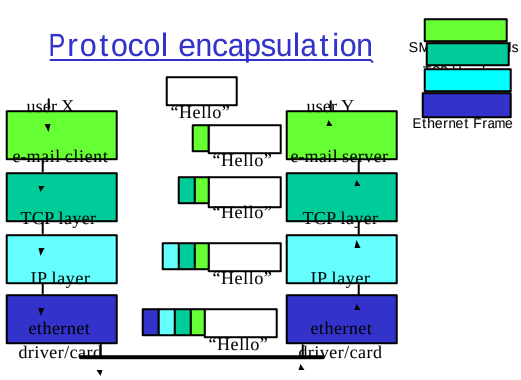
Internet protocol stack



Protocol stack



Protocol encapsulation

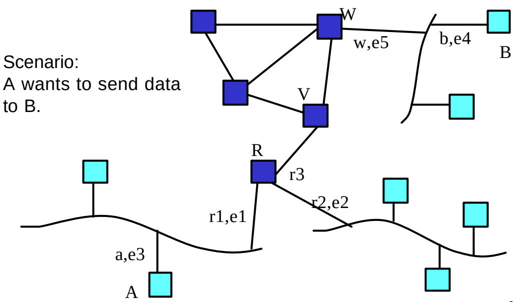


Packet Switching

- Packets indicate their destination
- No predetermined path for a packet to take
- Each intermediate note routes the packet closer to its destination

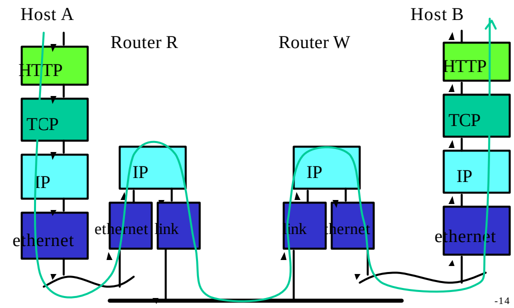
A small Internet

Scenario:
A wants to send data to B.



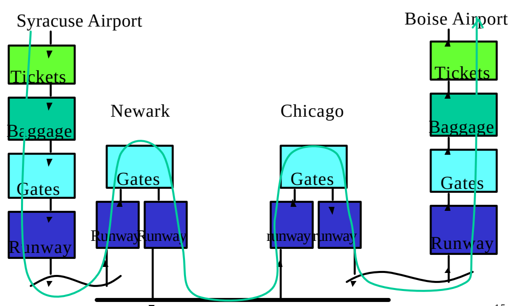
-13

Protocol stack: packet forwarding



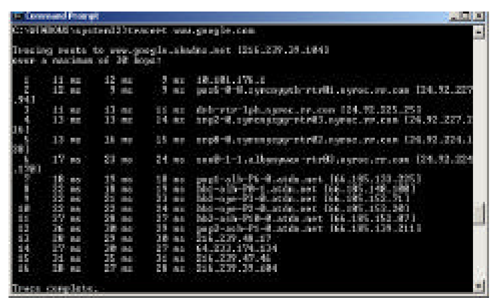
-14

Passenger Forwarding ☺



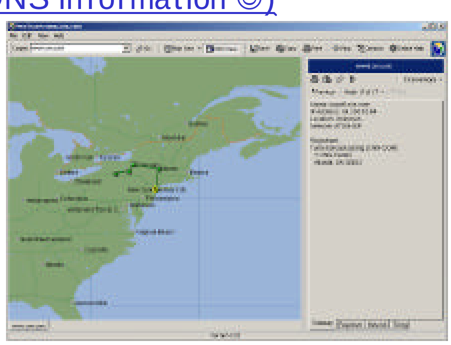
-15

Traceroute/tracert



-16

Graphical Traceroute (plus DNS information ☺)



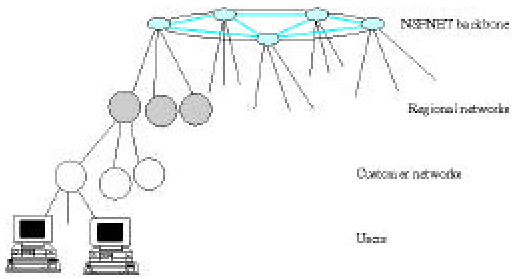
-17

Internet Map

- Traceroute gives one slice through the Internet topology
- What does the Internet really look like?
  - That is a actually a hard question to answer
  - Internet Atlas Project
    - http://www.caida.org/projects/internetatlas/
    - Techniques, software, and protocols for mapping the Internet, focusing on Internet topology, performance, workload, and routing data

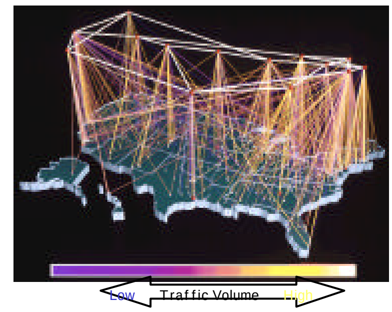
-18

The Internet around 1990



CAIDA: NSFNET growth until 1995

Backbone nodes elevated



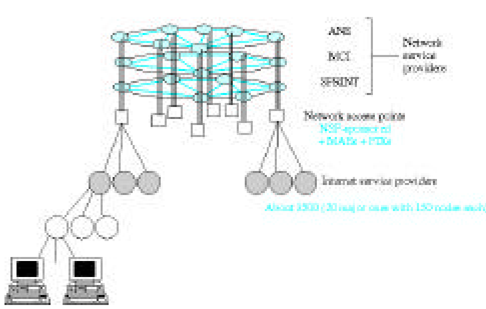
NSF Networking Architecture of Late 1990s

- NSFNET Backbone Project successfully transitioned to a new networking architecture in 1995.
  - vBNS (very high speed Backbone Network Services) - NSF funded, provided by MCI
  - 4 original Network Access Points (NSF awarded)
  - NSF funded Routing Arbiter project
  - Network Service Providers (not NSF funded)

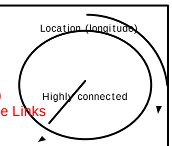
Network Access Point

- Allows Internet Service Providers (ISPs), government, research, and educational organizations to interconnect and exchange information
- ISPs connect their networks to the NAP for the purpose of exchanging traffic with other ISPs
- Such exchange of Internet traffic is often referred to as "peering"

The Internet in 1997



CAIDA's skitter plot

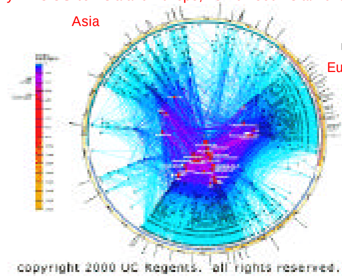


Top 15 ASes are in North America (14 in US, 1 in Canada)
Many links US to Asia and Europe; few direct Asia/Europe Links

Asia

Few connections

Europe



North America

Skitter data

16 monitors probing approximately 400,000 destinations

626,773 IP addresses

1,007,723 IP links

48,302 (52%) of globally routable network prefixes

DNS: Domain Name System

People: many identifiers:
- SSN, name, Passport #

Internet hosts, routers:
- IP address (32 bit) - used for addressing datagrams
- “name”, e.g., gaia.cs.umass.edu - used by humans

Q: map between IP addresses and name ?

Domain Name System:
- *distributed database* implemented in hierarchy of many *name servers*
- *application-layer protocol* host, routers, name servers to communicate to *resolve* names (address/name translation)
  - note: core Internet function implemented as application-layer protocol
  - complexity at network’s “edge”

-25

Names and addresses: why both?

- Name: www.google.com
- IP address (one of them): 216.239.39.147
  - (Also Ethernet or other link-layer addresses.)
- IP addresses are fixed-size numbers.
  - 32 bits. 216.239.39.147 = 101011000.11101111.00100111.10010011
- Names are memorizable, flexible:
  - Variable-length
  - Many names for a single IP address.
  - Change address doesn’t imply change name.
  - iPv6 addresses are 128 bit – even harder to memorize!

-26

Mapping Not 1 to 1

- One name may map to more than one IP address
  - IP addresses are per network interface
  - Multi-homed machines have more than one network interface - each with its own IP address
  - Example: routers must be like this
- One IP address may map to more than one name
  - One server machine may be the web server (www. foo,com), mail server (mail. foo.com)etc.

-27

How to get names and numbers?

- Acquisition of Names and numbers are both regulated
  - Why?

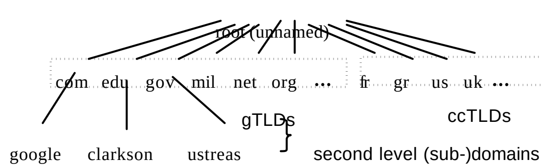
-28

How to get a name?

- First, get a domain name then you are free to assign sub names in that domain
  - How to get a domain name coming up
- Before you ask for a domain name though
  - Should understand domain name structure…
  - Know that you are responsible for providing authoritative DNS server (actually a primary and one or more secondary DNS servers) for that domain and registration information through “whois”

-29

Domain name structure



gTLDs= Generic Top Level Domains
ccTLDs = Country Code Top Level Domains

-30

Top-level Domains (TLDs)

- Generic Top Level Domains (gTLDs)
  - .com - commercial organizations
  - .org - not-for-profit organizations
  - .edu - educational organizations
  - .mil - military organizations
  - .gov - governmental organizations
  - .net - network service providers
  - New: .biz, .info, .name, …
- Country code Top Level Domains (ccTLDs)
  - One for each country

-31

How to get a domain name?

- In 1998, non-profit corporation, Internet Corporation for Assigned Names and Numbers (ICANN), was formed to assume responsibility from the US Government
- ICANN authorizes other companies to register domains in com, org and net and new gTLDs
  - Network Solutions is largest and in transitional period between US Govt and ICANN had sole authority to register domains in com, org and net

-32

How to get an IP Address?

- Answer 1: Normally, answer is get an IP address from your upstream provider
  - This is essential to maintain efficient routing!
- Answer 2: If you need lots of IP addresses then you can acquire your own block of them.
  - IP address space is a scarce resource - must prove you have fully utilized a small block before can ask for a larger one and pay $$ (Jan 2002 - $2250/year for /20 and $18000/year for a /14)

-33

How to get lots of IP Addresses? Internet Registries

RIPE NCC (Riseaux IP Europiens Network Coordination Centre) for Europe, Middle-East, Africa

APNIC (Asia Pacific Network Information Centre) for Asia and Pacific

ARIN (American Registry for Internet Numbers) for the Americas, the Caribbean, sub-saharan Africa

Note: Once again regional distribution is important for efficient routing!

Can also get Autonomous System Numnbers (ASNs from these registries

-34

End-to-End Example

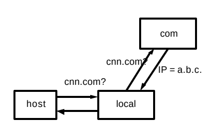
- Click -> get page
- page from local or remote computer
- link: http://www.cnn.com

  specifies
  - protocol (http)
  - location (www.cnn.com)

-35

Locating Resource

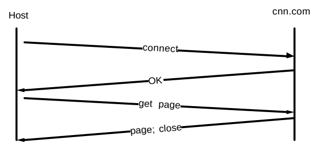
- www.cnn.com is the **name** of a computer (and, implicitly, of a file in that computer)
- Use DNS to translate name to address



-36

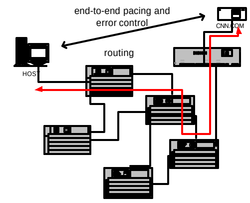
Connection

- The protocol (http) sets up a **connection** (another protocol, tcp) between the host and cnn.com to transfer the page
- The connection transfers the page as a byte stream, without errors: **flow control + error control**



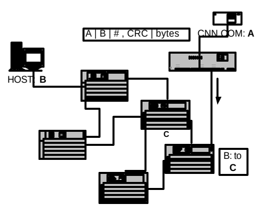
Data flow

- The byte stream flows from end to end across many links and switches: **routing (+ addressing)**
- That stream is regulated and controlled by both ends: **retransmission** of erroneous or missing bytes; **flow control**



Packets

- The network transports bytes grouped into packets
- The packets are "self-contained" and routers handle them one by one
- The end hosts worry about errors and pacing
  - Destination sends ACKs
  - Source checks losses



Port Numbers

- When a packet arrives at its destination, the operating system uses the destination port number to identify which application should receive it.
- This is called demultiplexing.

Bits

- Equipment in each node sends the packets as a string of bits
- That equipment is not aware of the meaning of the bits