What's most important in Project 1

- gain some experience with the techniques of protocol implementation
  - Learn to design timeouts and resending
  - Peer to peer interface and protocol interface
  - Protocol design: what data needs to be exchanged and kept track of; frame structure
  - Code for handling finite sequence number space
  - What metrics should be used to judge a design
  - How to manage time in a program

Most important (cont'd)

- experience first-hand the effects of different link characteristics and protocol choices on protocol performance
  - Get a real feel of what happens when you send packets over a network that drops and delays packets
  - Experience the significance of protocol parameters such as packet size, window size, delay
- Create a reliable protocol over an unreliable network

Important for individuals

- Using threads, semaphores
- Distributed debugging
- Understand SWP implementation in detail
- 1st experience implementing a protocol
- Network programming experience

From from Most Important

- Advantages and disadvantages of the 3 experimental links, and packet sizes
- Pros/cons of different implementations
- Working with data flow: when to send, when to ack
- How to do repetitive tasks in the most efficient manner
- C language
- Struct timeval
- Different messages in different directions?
- Pointer in sliding window protocol
- Learn about TCP/IP

Far From …

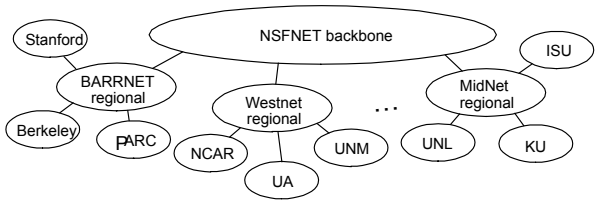
- Difficulties in programming packets
- Use of the emulator interface
- The emulator
- Get SWP to work with as few errors as possible
- How does sliding window work
- Feasible buffering
- How does UDP work
- How to improve sliding window
- How to match send and receive
- Recording unack'ed packets
- How to choose packets for retransmission
- Exception processing
- Appreciation of difficulty of protocol implementation

How to Make Routing Scale

- Flat versus Hierarchical Addresses
- Inefficient use of Hierarchical Address Space
  - class C with 2 hosts (2/255 = 0.78% efficient)
  - class B with 256 hosts (256/65535 = 0.39% efficient)
- Still Too Many Networks
  - routing tables do not scale
  - route propagation protocols do not scale

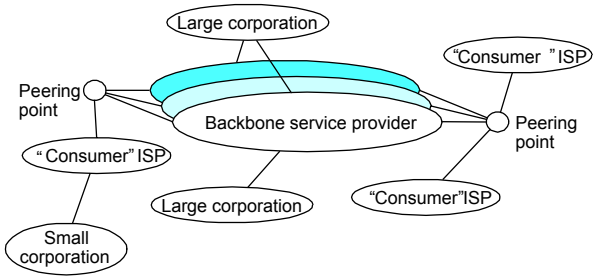
Internet Structure

Early 1990's



Internet Structure

Today



Different Routing: Interior & Exterior

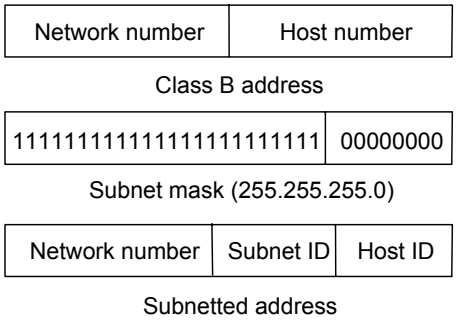
- Autonomous System (AS)
  - corresponds to an administrative domain
  - examples: University, company, backbone network
  - assign each AS a 16-bit number
- Within an AS (Autonomous System) – interior routing – RIP or OSPF or … (up to AS administrators)
- Between AS's – exterior routing – formerly EGP, now BGP – internet standard

Route Propagation

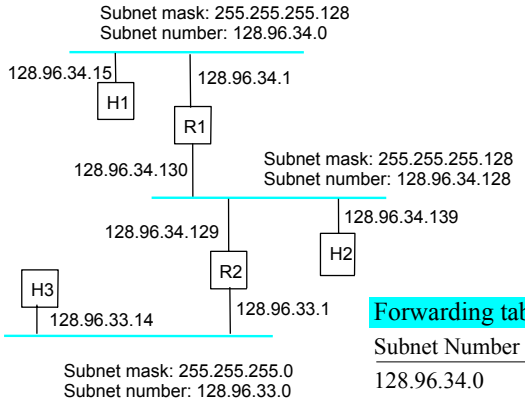
- "Know a smarter router" – use default routes
  - hosts know local router
  - local routers know site routers
  - site routers know core router
  - core routers know everything

Subnetting

- Add another level to address/routing hierarchy: *subnet*
- *Subnet masks* define variable partition of host part
- Subnets visible only within site



Subnet Example



Forwarding table at router R1

Subnet Number	Subnet Mask	Next Hop
128.96.34.0	255.255.255.128	interface 0
128.96.34.128	255.255.255.128	interface 1
128.96.33.0	255.255.255.0	R2

Supernetting

- Assign block of contiguous network numbers to nearby networks
- Called CIDR: Classless Inter-Domain Routing
- Represent blocks with a single pair

  `(first_network_address, count)`
- Restrict block sizes to powers of 2
- Use a bit mask (CIDR mask) to identify block size
- All routers must understand CIDR addressing

Forwarding Algorithm

```
D = destination IP address
for each entry (SubnetNum, NetMask, NextHop)
   D1 = NetMask & D
   if D1 = SubnetNum
      if NextHop is an interface
         deliver datagram directly to D
      else
         deliver datagram to NextHop
```

- Use a default router if nothing matches
- Not necessary for all 1s in net mask to be contiguous (but they almost always are)
- Can put multiple subnets on one physical network
- Subnets not visible from the rest of the Internet

EGP: Exterior Gateway Protocol

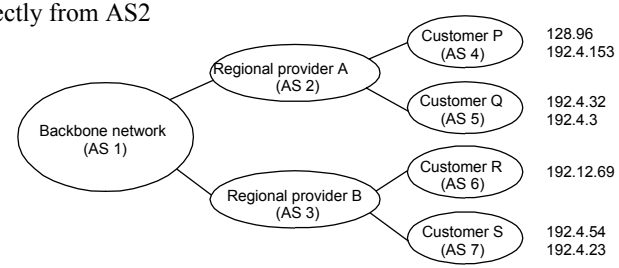
- Overview
  - designed for tree-structured Internet
  - concerned with *reachability*, not optimal routes
- Protocol messages
  - neighbor acquisition: one router requests that another be its peer; peers exchange reachability information
  - neighbor reachability: one router periodically tests if the another is still reachable; exchange HELLO/ACK messages; uses a k-out-of-n rule
  - routing updates: peers periodically exchange their routing tables (distance-vector)

BGP-4: Border Gateway Protocol

- AS Types
  - stub AS: has a single connection to one other AS
    - carries local traffic only
  - multihomed AS: has connections to more than one AS
    - refuses to carry transit traffic
  - transit AS: has connections to more than one AS
    - carries both transit and local traffic
- Each AS has:
  - one or more border routers
  - one BGP *speaker* that advertises:
    - local networks
    - other reachable networks (transit AS only)
    - gives *path* information

BGP Example

- Speaker for AS2 advertises reachability to P and Q
  - network 128.96, 192.4.153, 192.4.32, and 192.4.3, can be reached directly from AS2



- Speaker for backbone advertises
  - networks 128.96, 192.4.153, 192.4.32, and 192.4.3 can be reached along the path (AS1, AS2).
- Speaker can cancel previously advertised paths

IP Version 6

- Features
  - 128-bit addresses (classless)
  - multicast
  - real-time service
  - authentication and security
  - autoconfiguration
  - end-to-end fragmentation
  - protocol extensions
- Header
  - 40-byte “base” header
  - extension headers (fixed order, mostly fixed length)
    - fragmentation
    - source routing
    - authentication and security
    - other options