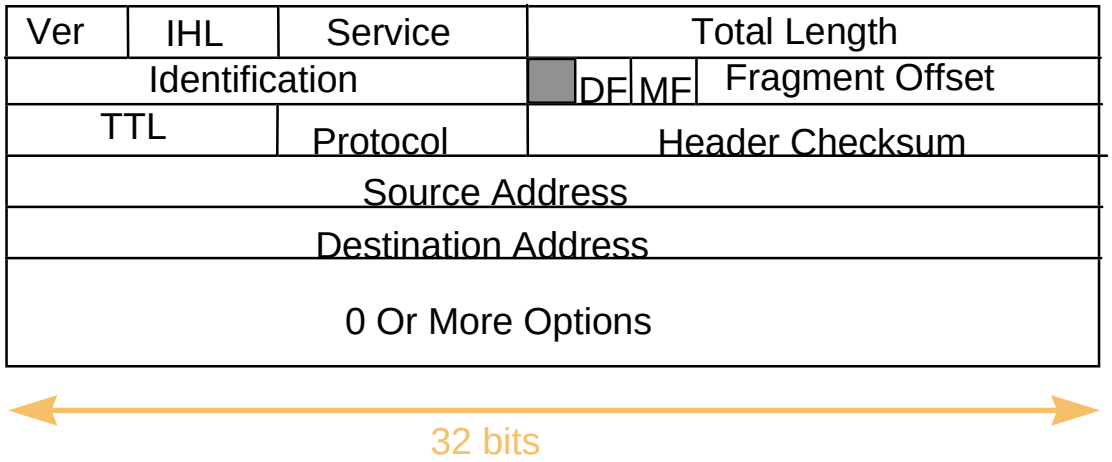
# Announcements

- Project proposal drafts due March 14,1997
- Reading
  - Today: 5.6
  - Tuesday: 6.1-6.2.6

# The IP Protocol

- IP Header
  - source, destination address, total length
  - version, ihl (header length in 32-bit words), ttl, protocol
  - fragmentation support: identification, df, mf, frag. offset
- Options
  - variable length
  - defined options
    - loose source routing
    - timestamp
    - record path



# Semantics of IP Addresses

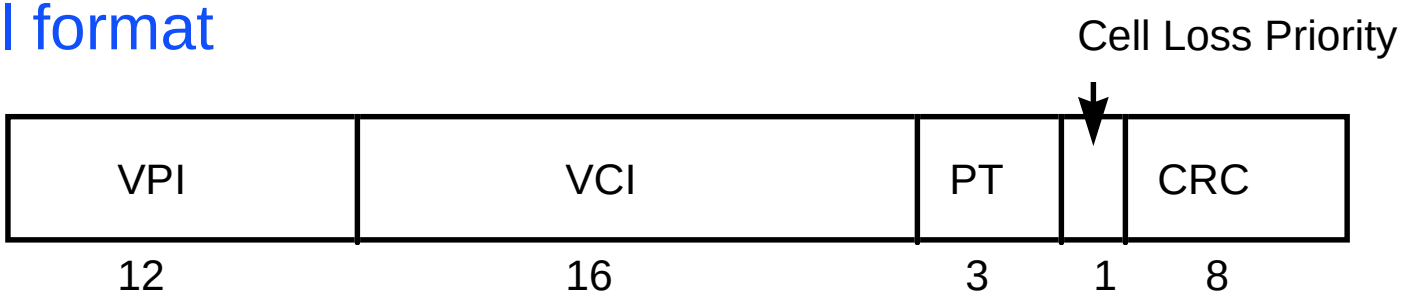
- Each address has a network, subnet, and host part
  - for routing only care about network and subnets not hosts
  - what is the network and subnet part varies depending on
    - what the address is (used to be fixed class A, B, C)
    - where the address is viewed from
      - Maryland subnet  viewed by world is 128.8.X.X
      - CS Dept. subnet  viewed by campus is 128.8.128.X
    - subnets are not visible at higher layers
  - each routing entry is <target,network mask>
    - match dest AND mask with target
    - if multiple matches, one with most 1's in mask wins
- Some special network addresses
  - all 0's --> this host
  - 0's in network address --> host on this network
  - all 1's --> broadcast on this network
  - 127.X.X.X -> loopback (this host)

# Internet Control Message Protocol (ICMP)

- Used to configure and run an IP network
- Just a transport protocol (more or less)
- Message Types
  - destination unreachable
  - time exceeded (ttl count reached 0)
  - parameter problem (invalid header)
  - redirect (inform router of possibly bad path)
  - echo request/response (AKA ping packets)
  - timestamp request/response (timestamped pings)
  - Address Resolution Protocol
    - finding out who owns an IP address on the subnet
    - send link level broadcast with a request
      - response is IP address of destination

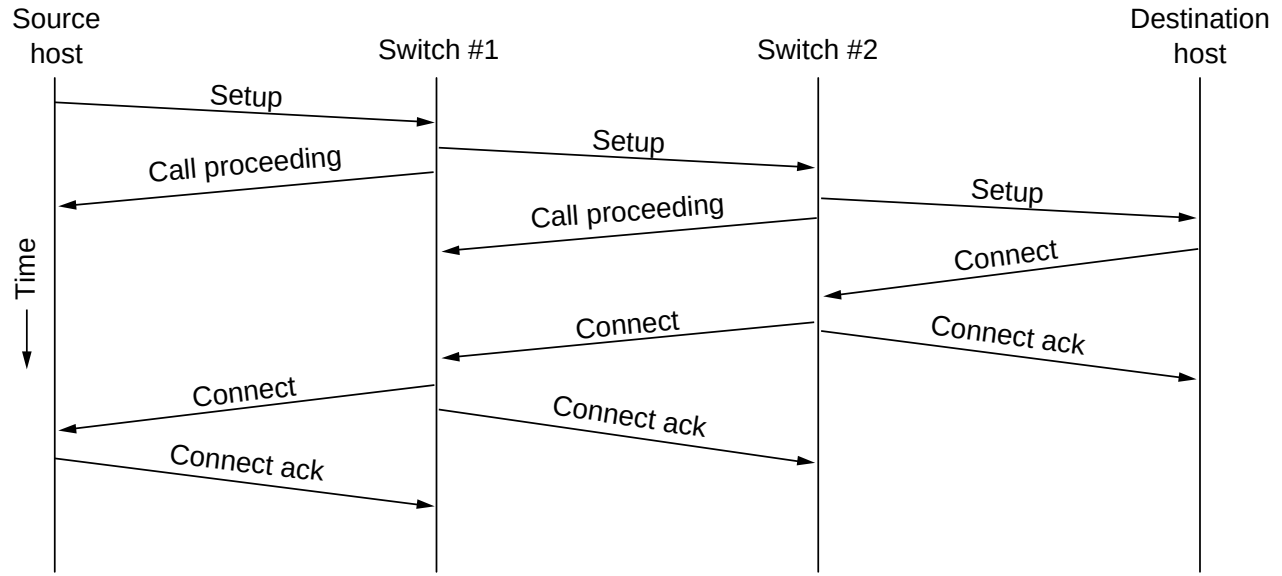
# ATM Network Layer

- Connection oriented
- Cell format

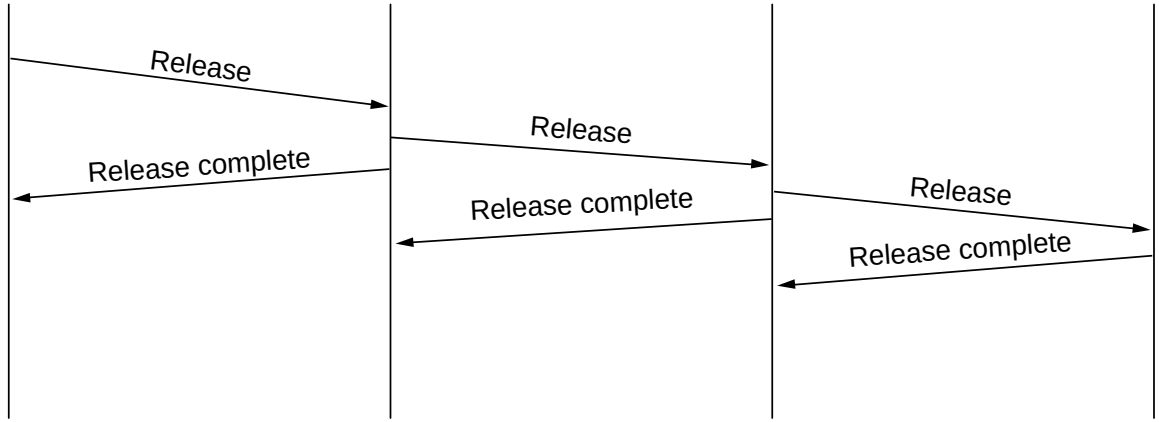



- Messages
  - Setup - establish a virtual circuit
  - Call Proceeding - request seen
  - Connect - request for connection ok
  - Connect ACK - thanks for accepting
  - Release - please terminate (either party)
  - Release Ack - ok, hanging up

# ATM Network Signaling




(a)




(b )

From: *Computer Networks*, 3rd Ed. by Andrew S. Tanenbaum, (c)1996 Prentice Hall.

# What is Software Engineering?

- Stepping back from the process of writing programs to think about how to do it right.
- Mostly just nuggets of common sense
- However, you must take it to heart

# Four Steps

- Abstraction
  - Goal: bury complexity through abstraction
    - Look at the big picture and break it down
    - What are the logical components?
  - Are there recurring patterns that can be generalized?
    - example: timers
- Specification:
  - define precisely the interface of each component
  - Three issues:

    Names - "**big_queue**", "**Big_queue**", or "**bigQueue**"

    Types - function signatures

    Functionality - exactly what each function does

        Pre- and Post- Conditions

# Four Steps (cont.)

- Delegation
  - Who is doing what and by when?
  - Delegate coding but not understanding
  - Separate coding and testing responsibilities
- Verification
  - incrementally test each component
  - Test each other's code
  - Challenges:
    - It's boring
    - Hard to write realistically complete test suites
  - Early testing can save you last minute nightmares!

# Tools

- Compiler options
  - g - produce symbols for debugging
  - I - search this directory for header files
  - D - define the following flag
- gdb/dbx
  - Look at the source code
  - Set / delete breakpoints
  - Set conditional breakpoints
  - Inspect the state of the threads
  - Inspect the stack (s)
  - Change the values of variables
- Make
  - automatically rebuilds code after changes have been made
  - get a good example **makefile** and hack it
  - Parameterize your actions
  - **makedepend** inserts dependencies into **makefile**

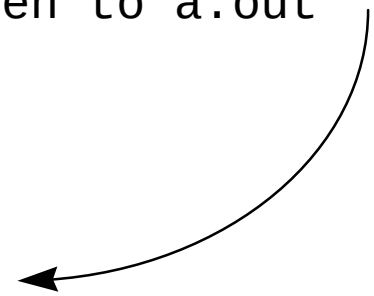
# Revision Control - **cvs**

- Front-end to **rcs**
- Keeps a single copy of the master sources
  - set a single CVSROOT environment variable
- Old versions kept through diffs
- Catches update conflicts
- **checkout**
  - cvs co <directory or file>
- **update**
  - cvs update <directory or file>
  - get latest version of the code
- **commit**
  - cvs commit <directory of file>
  - makes changes visible in the repository

# Miscellaneous - nm, c++filt

- **nm** inspects the symbol table of an object file
- **c++filt** demangles C++ names
- Example:

```
Undefined                first referenced
 symbol                      in file
append__5QueuePc /var/tmp/cca000Ql1.o
ld: fatal: Symbol referencing errors.
No output written to a.out
```



```
~> c++filt append__5QueuePc
Queue::append(char *)
```

# Coding Tips

- Comment code very thoroughly
- Set up **CVS** to imbed log comments in source files
- Use **ASSERT** to double-check pre/post conditions at run-time
- Use **const** instead of **#DEFINE**
- Inheritance: use it, don't abuse it

# Some Words About Threads

- Make your code thread-safe (re-entrant)
- Must also worry about the thread-safety of library calls
- Consider:
  ```
  struct hostent *gethostbyname(const char *name);
  ```
- Read MT-LEVEL sections in the man pages
- See man pages for: **threads** and **thr_create**