Implementation of protocol stack

- Router process packets at Network Layer
  - Operations specified in RFC 1812
- How can we build an IPv4 router?
Software router

- **Hardware**
  - PC with multiple NICs

- **Software**
  - What functions do we need to implement?

Can we write a user-space IPv4 application?
  - Possible, but inefficient

IPv4 as operating system component
  - OS has better access to hardware resources
  - OS can access packet data without copying

Typical
  - IPv4 processing part of kernel
  - Three steps:
    - Input processing
    - Forwarding
    - Output processing

What are the technical challenges?
OS implementation

Figure from http://www.cs.binghamton.edu/~ghose/CS529/linuxTCP/linux-net.html

OS implementation

Linux Kernel 2.4 Packet handling

Network Drivers (drivers/inet)

Figure from http://linux-ip.net/html/linux-ip.html
Interrupts

- Timing of IPv4 processing is tricky
  - Packet arrivals are asynchronous
- Interrupt triggers processing
  - What is an interrupt?
    - Event signal
    - Hardware interrupts: raised by device
    - Software interrupts: raised by software
    - "Interrupt handler" is called to process interrupt
  - Interrupt Priorities
    - Interrupts have different priorities
- How are interrupt levels allocated?
  - Livelock must be avoided

Interrupt Priorities

Scalability

- What is scalability?
  - A system (design) is scalable if it can easily be extended in “size” and performance
    - More ports
    - Faster links
- Scalability important in system design
  - Design process is expensive
  - Ability to easily extend to new requirements is important
  - Performance requirement increase really fast
    - Moore’s Law
  - Systems will eventually be used in a different context
- Is a software router scalable?
Assignments

- Exam I
  - Prepare for next Tuesday
  - Sample problems listed under “assignments” on course web site
- SPARK
  - Assessment quiz