



Project Introduction

Jeffrey Shafer



Class Projects

- The projects begin today
- They will continue throughout the semester
- Projects are staggered so that groups can be exposed to both hardware and software elements
- All project details will be posted on the course web site

Groups

- All groups are responsible for both the hardware and software components of the project
 - Students focusing primarily on software should be involved (to some extent) on the hardware
 - Students focusing primarily on hardware should be involved (to some extent) on the software
- All students should participate in design discussions for both hardware and software components
- All students should participate in writing project reports and the final demo



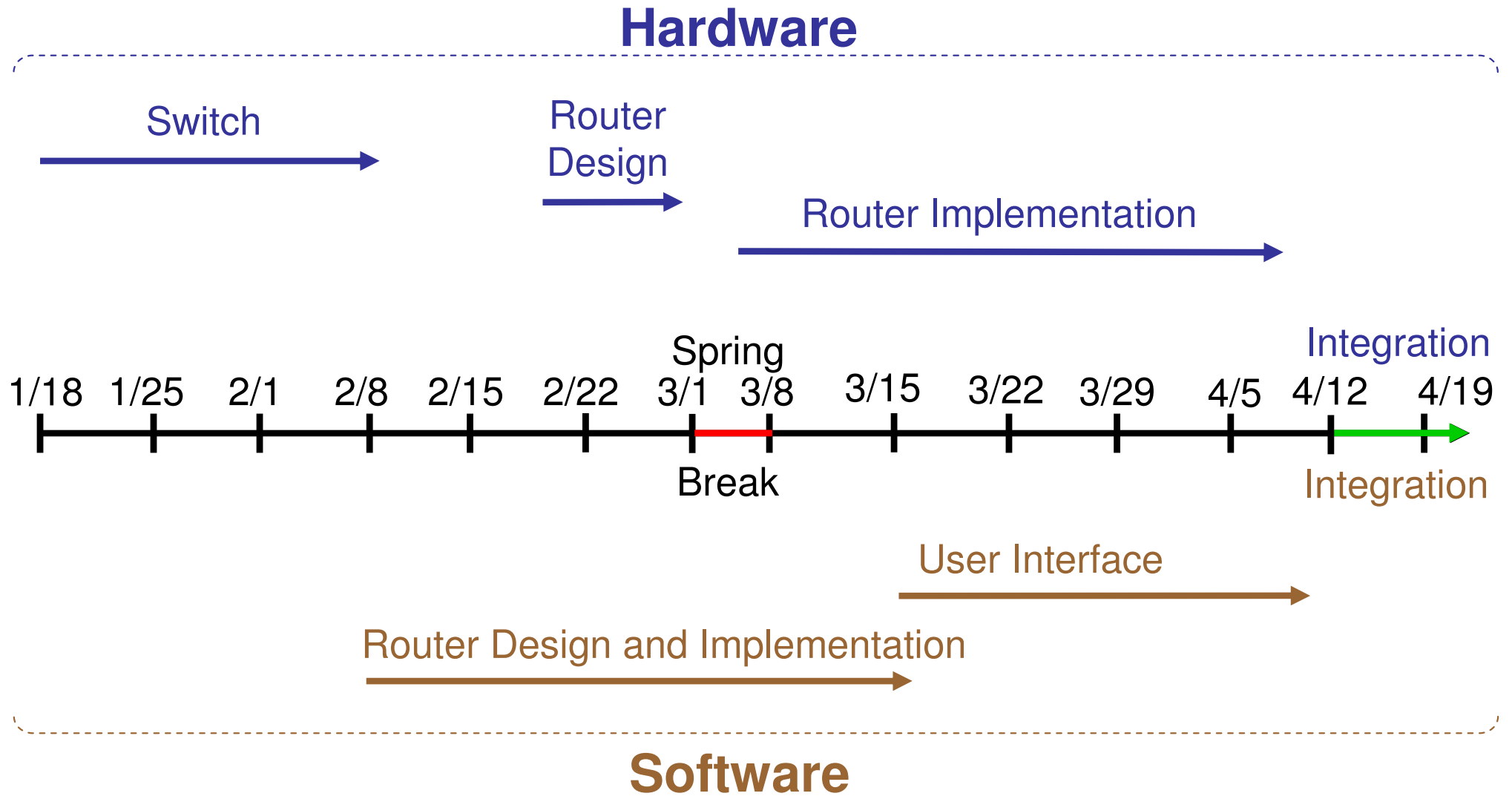
Project Meetings

- We will meet in Abercrombie A123 every Friday for the rest of the semester for either tutorials or group meetings
- With each group, we will
 - Discuss progress
 - Assess designs
 - Answer questions
- Be prepared with evolving design documents
 - Doesn't need to be pretty
 - Should reflect the current design

Final Product

- Upon completion of the projects, you should have:
 - A functioning router with integrated control (software) and data (hardware) planes
 - A clear, concise, and complete report that describes your router
 - This should evolve over the course of the semester – do not be afraid to work ahead!
- There will be demos to present your router in the last week of class

Project Timeline





Initial Projects

■ Hardware

- Ethernet Learning Switch – Due 2/13/09
- Router Design Document – Due 2/27/09

■ Software

- Software router implementation – Due 3/13/09

Ethernet Learning Switch

- *Assigned today – You know everything!*
- Learning switch
 - Learn MAC address → port mappings
 - Forward packets based on MAC address
 - Broadcast if destination unknown
 - Single port if destination mapping exists
- First step to building an IP router
 - Much simpler functionality
 - Basic forwarding table
 - Gain familiarity with the NetFPGA design

Software Router

- *Won't be assigned until after classes on IP, ARP, and ICMP...*
- Functioning IP router
 - ARP (requests and replies)
 - ICMP (echo, no route, unreachable, time exceeded)
 - IP
- First step to building an integrated hardware/software router
 - Gain experience with all aspects of basic routing

Deliverables

- For all projects
 - Source (C and/or Verilog)
 - Report – basically a design document describing your implementation
 - Project meetings – ongoing discussion of design and implementation
- The report should evolve over the course of the projects
 - You should *not* write it the night before!
 - We will discuss the state of your design (and the report that describes it) during project meetings
 - Tip: Structure your reports to be easily assembled into one large router document at the end
 - Tip: Update your documentation after each project is graded and returned to save time at the end

Pragmatics

- Resources for this class are shared
 - comp519.cs.rice.edu
 - server1/server2/server3
 - Cisco router
 - Etc.
- It is possible for you to cause problems for other groups!
 - Be a good neighbor – Don't!

comp519.cs.rice.edu

- 8-way Opteron 275 server (64-bit) / 8GB RAM
- Linux distribution: Ubuntu 7.10 (AMD 64)
 - VNS projects can be run anywhere
 - May want to use Ubuntu 7.10 (in a virtual machine?) to ensure compatibility
- Shared resource for all students
 - Don't start too many hardware compiles at the same time...
 - Be polite to other users

Disk space

- 250GB of mirrored space on `comp519.cs.rice.edu`
 - Mounted over NFS to all class machines
- This space is not backed up!
 - *(Unless Jeff runs a manual mirroring script, which just copies it to another server about 20 feet away...)*
 - Do not rely on it to keep your projects
 - Use subversion!

The Servers

- Right now, everyone can log in
 - Useful to run wireshark and debug your hardware
- What happens if you overload/crash/etc. these machines?
 - Everyone loses
 - Limit what you do when logged on to the servers
 - Running wireshark (monitor traffic)
 - Pinging other hosts (low load test)
 - TCP streaming tests (high load test)
- You can send traffic to other group's subnets – Don't!

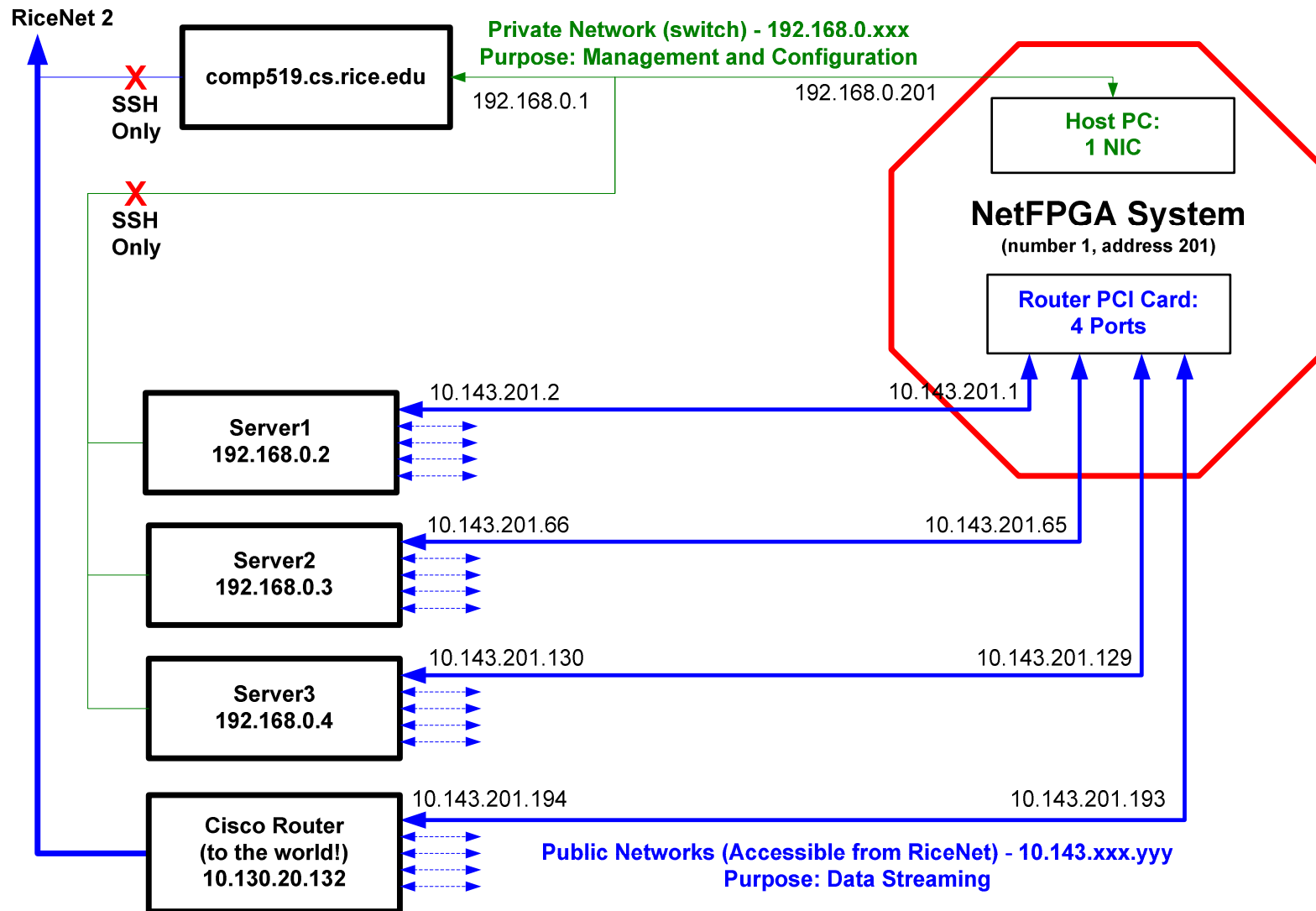
netfpga0X

- Each group is assigned their own machine containing a netfpga board
 - Intel Atom 230 (32-bit)
 - Ubuntu 8.04 LTS
 - Low-power embedded machine with limited resources
 - Recommend that you run most tools on comp519
- Login to these machines will be restricted to members of your group
- You have limited privileges on these machines
 - nfsudo gives root access and wraps X-forwarding
 - wireshark, nf2_download, cpci_reprogram, reboot

Compiling: 32-bit versus 64-bit

- comp519 – 64-bit kernel
 - Do development on this machine
- netfpga0X – 32-bit kernel
 - Make sure software tools that need to run on netfpga0X are re-compiled first

IP Addresses



IP Addresses

- You should *only* access IP addresses that are assigned to your group
 - Using addresses assigned to other groups will cause both you and them problems
- All of the IP addresses (Cisco/server1/server2/server3) are accessible from within Rice
 - Except maybe from Rice visitor networks
- Server1 also has public IP addresses that will be accessible from the Internet (static mapping)

Version Control

- Get used to using version control for *everything*
 - Should never be editing C/Verilog that is not first checked into subversion
 - This can take some getting used to, but is actually simpler to use after you are comfortable with it
- “Sandboxes”
 - Each person should have their own copy of the project checked out
 - Your edits do not affect others in your group
 - When your edits “work”, check them in, so others can see them
 - Never edit someone else’s code – always work in your own sandbox!

Version Control

- Come up with check in policies for your group
 - Only check in after testing?
 - Always check in at the end of your work day?
 - You need to determine what works best for your group
- Subversion *will* save you
 - You have a record of all changes
 - When something breaks, you can travel back in time to a working version!
 - More frequent check ins make this more valuable
 - If comp519 drives fail, you still have everything!



Projects

- All group members should learn about both hardware and software components
- All group members should actively participate in the design
- Everyone should be a good neighbor with shared resources (machines and IP addresses)
- Have fun!

Homework

- A subversion tutorial is on the website
- By Friday:
 - Do the “Initial Setup” task
 - Creating a repository on the campus server
 - Skim the “Information” task
 - Basic tutorial on using SVN
 - Carefully read the “Subversion exercises” task
 - How to create a new hardware project from the reference design, and share it with your group
 - What files should I ***not*** check into SVN?
 - The easiest time to start using SVN is ***before*** you have modified any part of your project

Upcoming Events

■ This Week

- Friday – Tutorial (Hardware Simulation)
- **Reminder: Compile your hardware design before the Friday tutorial**

■ Next Week

- Monday – IP Protocol (#1)
- Wednesday – IP Protocol (#2)
- Friday – Tutorial (Hardware Testing)