

CSC 458/CSC 2209: Computer Networks

Handout # 5: Building your own Router Assignment#1



Alireza Sahraei
Department of Computer Science
University of Toronto

Friday, September 25

Based on slides by Clay Collier , Martin Casado, Monia Ghobadi and Geoff Salmon

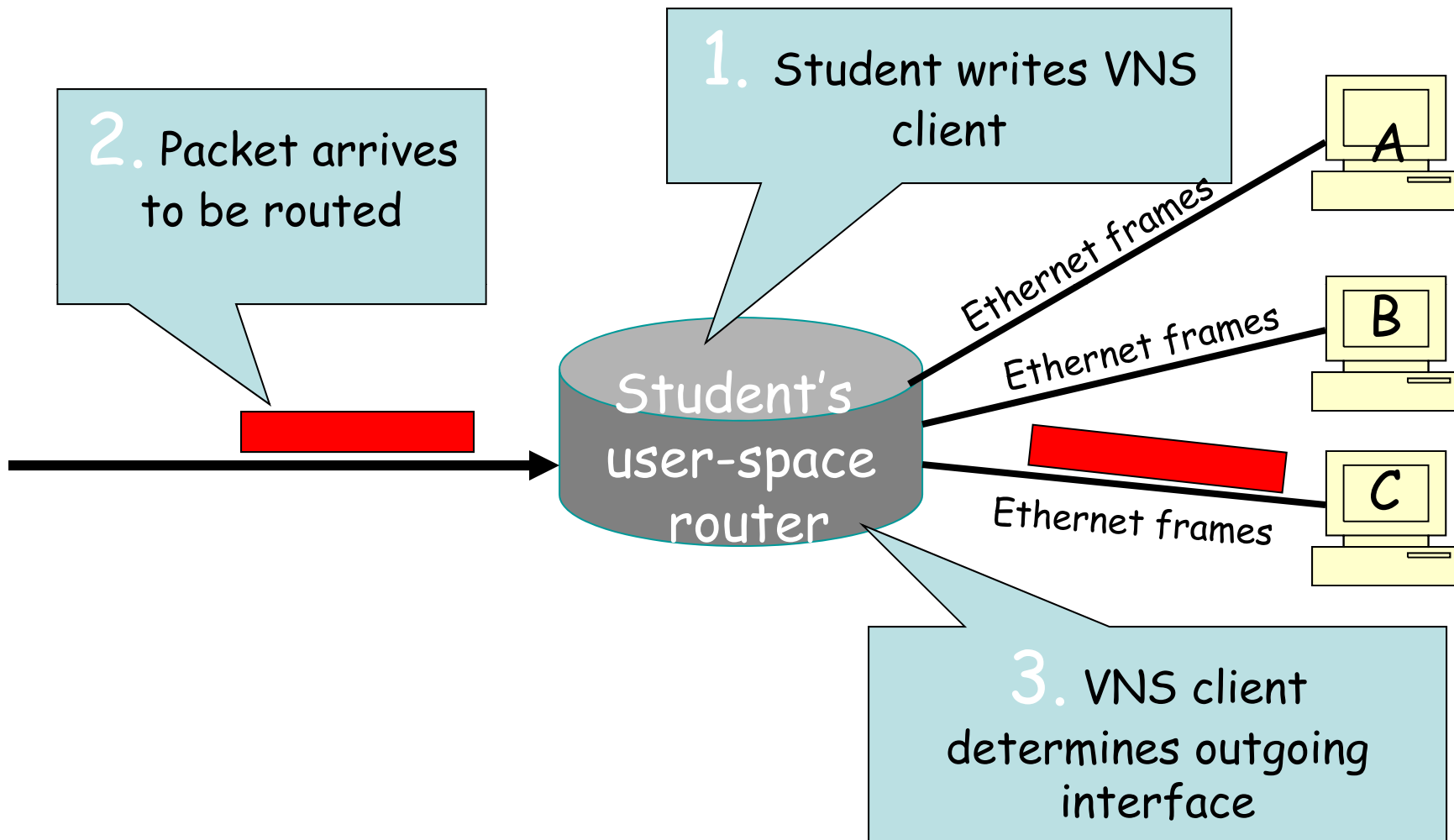
Assignment overview

- You will be given a virtual network topology
- You write a router in C
- Your router will route real packets sent over the Internet from standard clients (i.e, Firefox)
- Each of you has their own router, topologies, and IP addresses.
- Due Friday, October 23rd @ 5pm

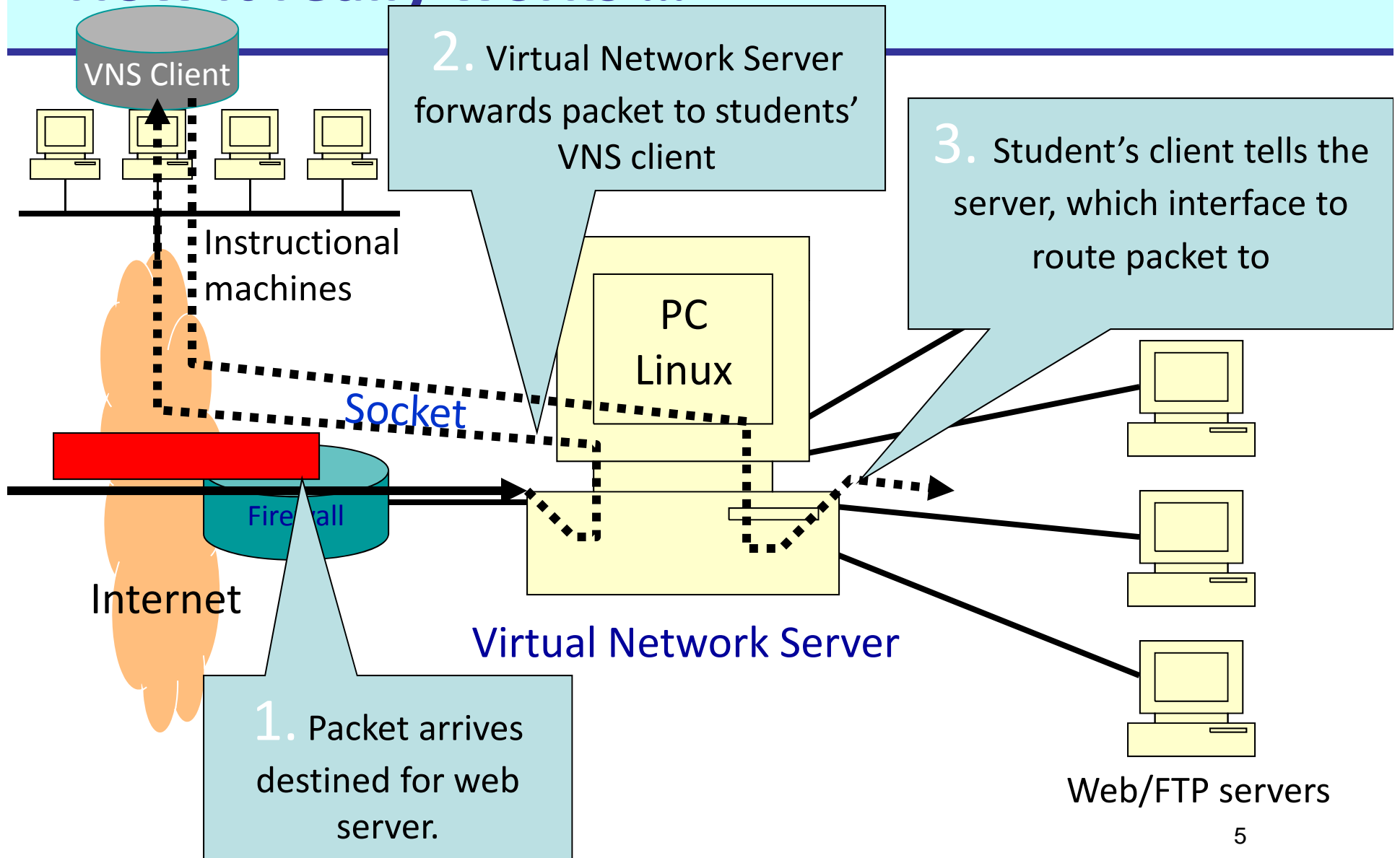
VNS

- We will use Stanford University's Virtual Network System (VNS) for this programming assignments

How VNS works?

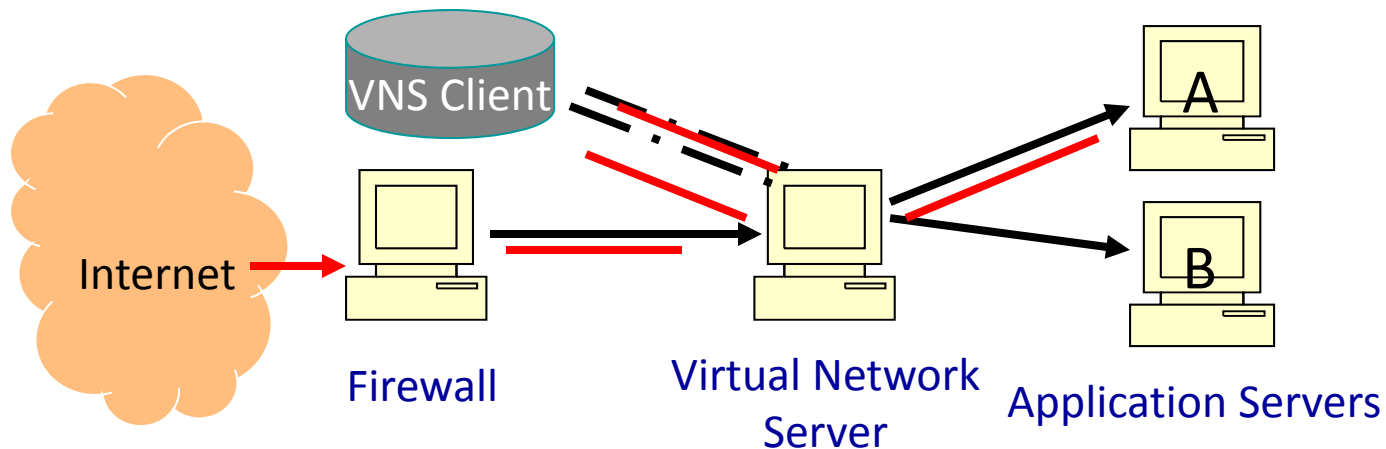


How it really works ...



How it works in excruciating detail ...

- Your VNS client establishes connection with VNS server and is assigned an IP for each virtual interface
- You run standard internet client (ftp) to application server A
- Firewall sends ARP request to VNS server
- VNS server forwards ARP request to client
- VNS client send ARP reply to server destined to the firewall
- Firewall forwards traffic with VNS client's hardware address
- Client determines next hop and sends traffic to VNS server



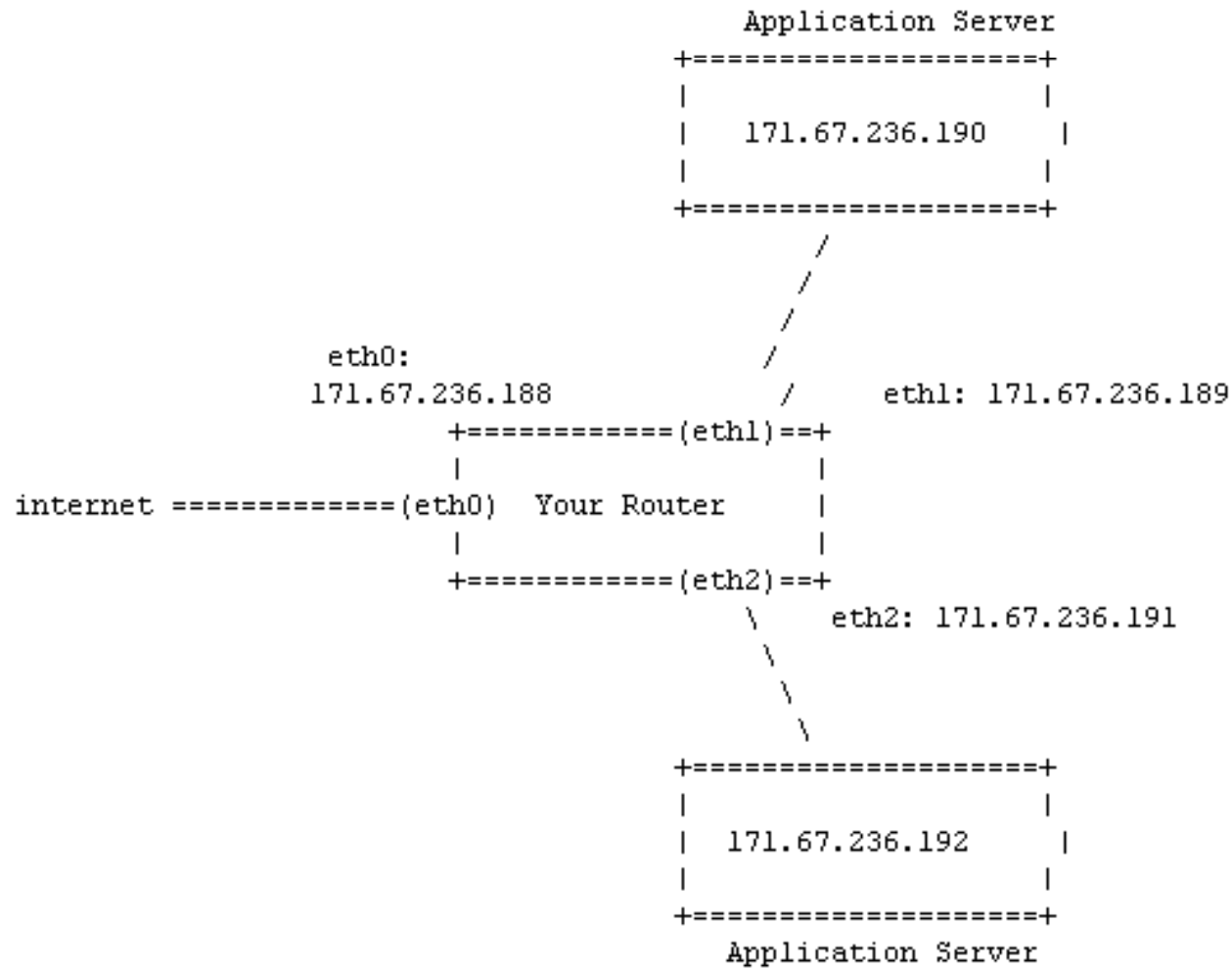
Getting started

- You already have your topology and rtable (Ignore the part between “Loading router table” and “< - - ready to process packets - - >” in README file)
- Download the stub code from the assignment page. It connects and communicates with the VNS server.
- Compile the code and connect to the VNS server: `./sr -s vns-1.stanford.edu -t <topo-id>`

... Getting started

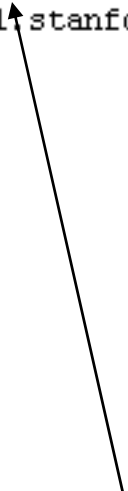
- After you connect successfully, the server will send you a description of the host including all the interfaces and their IP addresses.
- The routing table is constructed from the file rtable and by default consists of only the default route which is the firewall.

Example



... Example

```
greywolf:~/csc458/stub$ ./sr -t 10 -s vns-1.stanford.edu
Loading routing table
-----
Destination      Gateway           Mask      Iface
171.67.236.190   171.67.236.190   255.255.255.255 eth1
171.67.236.192   171.67.236.192   255.255.255.255 eth2
0.0.0.0          172.24.74.17     0.0.0.0   eth0
-----
Client c7ghobad connecting to Server vns-1.stanford.edu:12345
Requesting topology 10
Router interfaces:
eth0   HWaddr70:00:00:0a:00:01
       inet addr 171.67.236.188
eth1   HWaddr70:00:00:0a:00:02
       inet addr 171.67.236.189
eth2   HWaddr70:00:00:0a:00:06
       inet addr 171.67.236.191
<-- Ready to process packets -->
^█
```



Routing table is used to decide where to route packets

Forwarding in an IP router

1. Remove IP datagram from arriving Ethernet packet.
2. Lookup packet DA in routing table.
 - If known, determine next-hop IP address.
 - If unknown, drop packet and send ICMP message.
3. Decrement TTL, update header Checksum.
 - If TTL== 0, send ICMP message.
4. From next-hop IP address, determine outgoing interface and next-hop Ethernet MAC address.
 - If necessary, send an ARP packet to determine MAC address.
5. Encapsulate IP datagram in Ethernet packet.
6. Forward packet to outgoing interface.

ARP

Why do you need ARP?

- Your routing table contains ip addresses for next hop, however you send ethernet frames to ethernet addresses
- The web/ftp server and the router that connects you to the internet need to know your hardware address

What you have to do:

- Generate ARP requests and parse ARP replies
- Listen to ARP requests and send ARP replies
- Cache ARP replies to avoid sending requests for every packet
- Both ARP requests and ARP cache entries should timeout
- Send ICMP host unreachable messages if ARP requests fail.
(Note: Must do this in a fixed period of time. This is tricky!
Read last point in “Required Functionality” carefully)

Some hints/tips

- Use the logging option ‘-l <filename>’ of **sr** to write all packets received and forwarded by your router to a log file .
- Use **tcpdump** (\usr\sbin\tcpdump on CDF machines) to examine the packets in the log file.
 - Use ‘-r <filename>’ to specify the log file
 - Try ‘-v’ or even ‘-vv’ for more analysis
 - Use -e to print MAC addresses
 - Use -x to print out packet in hex, -xx for link layer headers
 - Will detect incorrect checksums, malformed packets etc.
- You don’t have to deal with:
 - Multicast
 - Broadcast
 - IP Header Options

For Further Reading...

- Read the assignment & FAQ
- Peek at the RFC on routers (RFC 1812) but don't worry too much about it
- For ICMP details read the RFC (RFC 792)
- If RFCs are too cryptic to read, try the RFC sourcebook at Network Sorcery:

<http://www.networksorcery.com/>