CS244A Review Session

Building your own Router
Assignment#2

Friday, January 25, 2008

Clay Collier

(based on slide by Martin Casado)

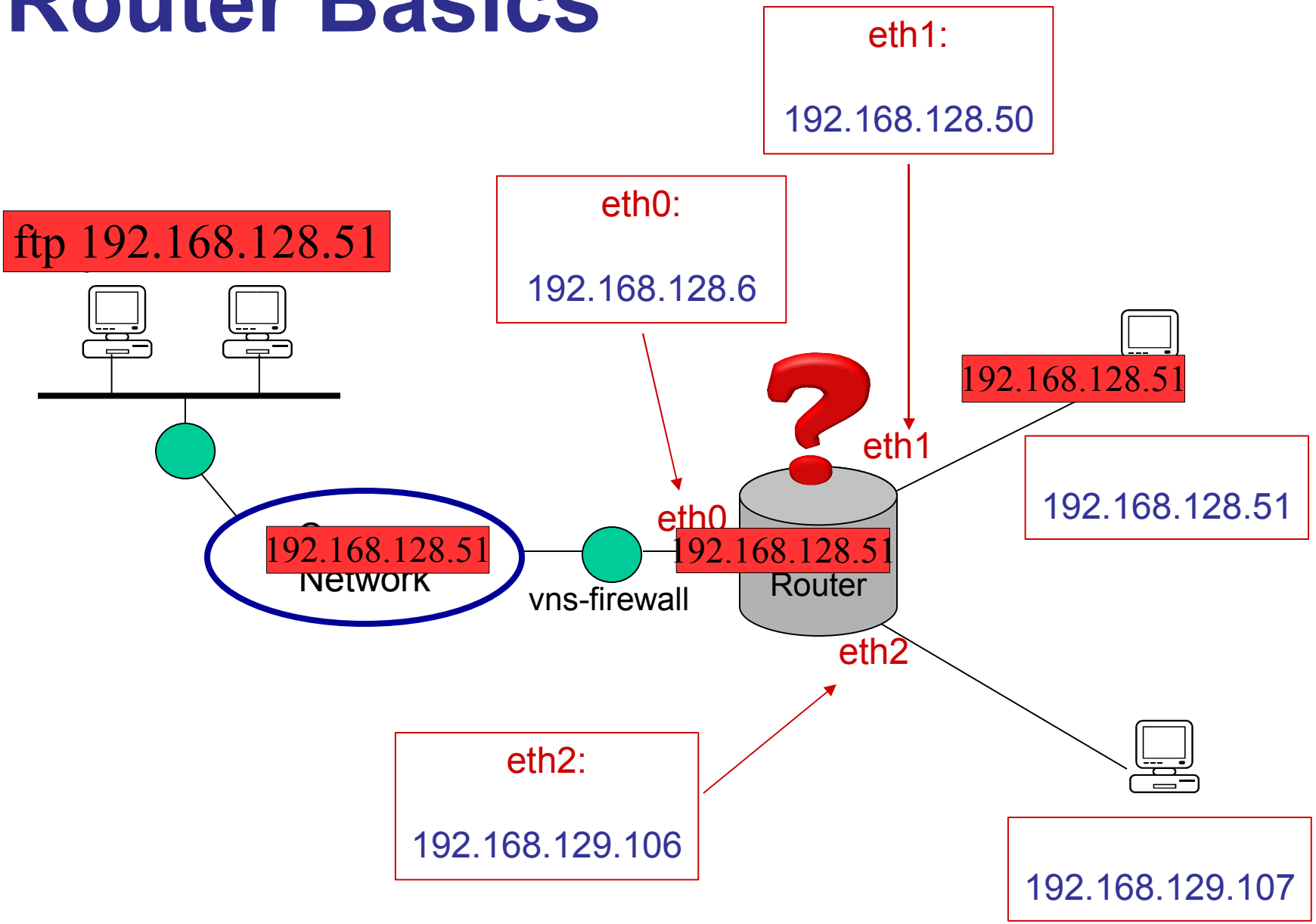
Assignment Overview

- You build 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, February 8th @ 12:00 PM

Getting Started

- Download the stub code from the assignment page
- Create your account and a simple topology
- Connect to your topology as described in 'Test Driving the sr Stub Code' (note that the output will be slightly different)

Router Basics



What you get ....

```
void sr_handlepacket(struct sr_instance* sr,
                     uint8_t * packet,
                     unsigned int len,
                     char* interface)
```

```
int sr_send_packet(struct sr_instance* sr,
                           uint8_t* buf,
                           unsigned int len,
                           const char* iface)
```

Forwarding in an IP Router

1. Remove IP datagram from arriving Ethernet packet.
2. Lookup packet DA in routing table.
    1. If known, determine next-hop IP address.
    2. 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.
    1. If necessary, send an ARP packet to determine MAC address.
5. Encapsulate IP datagram in Ethernet packet.
6. Forward packet to outgoing interface.

Routing table

- Used to decide where to route packets
- There is code provided to read an external table. Use it! No hard coding routes!

IP address	Next-hop IP address	Network Mask	Outgoing interface
192.168.128.51	192.168.128.51	255.255.255.255	eth1
192.168.129.106	192.168.129.106	255.255.255.255	eth2
default	172.24.74.17	0.0.0.0	eth0

Checksums

- IP Checksum
  - Any IP packet, need to check in router. If not correct drop the packet.
  - Read Kurose & Ross, page 327, 427
- ICMP Checksum
  - Need to calculate for outgoing packets
  - Ignore for forwarding packets
- TCP/UDP Checksum
  - End-to-End checksum, routers don’t care

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
- You don’t want to send a request for each packet, instead use an arp cache
- Requests should time out, the arp cache as well (~15 sec)

ARP Cache	
IP address	Ethernet MAC Address
172.24.74.130	00:e0:81:04:08:9b
…	…

What's the required functionality?

For full credit you will need:

- The IP header and MAC addresses are correct in packets sent from your router.
- Your router handles all ARP requests correctly
- You can correctly download a file using http and anonymous-ftp from your application servers
- It responds correctly from any of the elaine machines if you run:

```
traceroute -n <your application server IP>
```

- ICMP 'host unreachable' and 'port unreachable' messages are generated correctly
- Ping to all of your interfaces and the application servers works
- No shortcuts are taken (e.g. replicating every packet on every interface doesn't count)
- **You Can Enforce Guarantees on Timeouts!**

Some tips/hints

- Take a look at Clack (http://yuba.stanford.edu/vns/clack/)

- Use the "dump" option and **`tcpdump`** to examine the packets received and forwarded by your router.
  - 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.

- Use **`jethereal`**, a graphical tool similar to tcpdump/wireshark and it will tell you even more about what's wrong with your packets (http://yuba.stanford.edu/JEthereal/)

- You'll need to find a convenient method to test ICMP host unreachable

More tips/hints

- You don't have to deal with:
  - Multicast
  - Broadcast
  - IP Header Options
  - Routing protocols etc.
- In the supplied include file *sr_protocol.h* there are definitions for the packet headers (e.g. ip, Ethernet)
- If you create structures yourself, use the *packed* keyword
- If you post a question to the newsgroup always include tcpdump output of your packets! (use -n -vvv -e -x command line args) and only paste relevant packets

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/

Feedback from PA1

- Please turn off diagnostic output and status messages by default.
- Make sure your submission is in the correct format- a gzipped tar archive. We will be checking for this at submission time for the next assignment! (Use `make dist`)
- Compile with -Wall.
- Please describe **design decisions** in the README, rather than the assignment mechanics