CSC458/2209 PA1
Simple Router

Based on slides by: Antonin

Yinan Liu
Overview

• Your are going to write a “simplified” router
  ➢ Given a static network topology
  ➢ Given a static routing table
  ➢ You are responsible for writing the logic to handle incoming Ethernet frames:
    • Forward it
    • Generate ICMP messages
    • Drop it
    • And more ...
But how to do it???

• Where will my routing logic run?
• Where will the traffic come from?
• How will I test my code?
• No hardware router 😊
• Network topology emulated with Mininet: your router connects 2 servers to a client
• Your router will handle real traffic
• The topology is emulated on CDF machines!
Emulated Topology

HTTP Server 1
192.168.2.2
eth1
192.168.2.1

Router
10.0.1.11
eth3

HTTP Server 2
172.64.3.10
eth2
172.64.3.1

Client
10.0.1.100

Topology for Simple Router

CSC458/2209 - Computer Networks,
University of Toronto
Emulated Topology

1. Packet sent to SR
2. Routing decision made
3. Action is taken

Interaction Router – SR thanks to POX and Openflow

HTTP Server 1
107.21.41.195
eth1
107.23.34.64
HTTP Server 2
107.21.17.129
eth2
107.21.14.129
Router
eth3
10.0.1.11
Internet
Emulated Topology

Your routing decision:
- Look at the routing table
- Figure out on which interface to forward the packet
- Make necessary changes to the packet

Ping 107.21.17.129
What your routing logic needs to do?

• Route Ethernet frames between the client and the HTTP servers
• Handle ARP request and replies
  ➢ Maintain an ARP cache
• Handle traceroutes
  ➢ Generate TTL Exceeds Message
• Handle TCP/UDP packets sent to one of the routers’ interfaces
  ➢ Generate ICMP Port Unreachable
• Respond to ICMP echo requests

• See course webpage for full requirements
A rough flow chart

1. Receive Raw Ethernet Frame
   - It's an IP packet
     - Cache it, go through my request queue and send outstanding packets
   - It's an ARP packet
     - Construct an ARP reply and send it back
     - Request to me
     - Reply to me
A rough flow chart

1. Receive Raw Ethernet Frame
   - It's an IP packet
     - If it's ICMP echo req, send echo reply
       - Or if it's TCP/UDP, send ICMP port unreachable
     - Not for me
       - Check routing table, perform LPM
         - Match
           - Check ARP cache
             - Hit
               - Send frame to next hop
             - Miss
               - ICMP net unreachable
   - It's an ARP packet
     - Not for me
     - Send ARP request
       - Resent >5 times
       - ICMP host unreachable
A rough flow chart

• Many things missing from this chart
  ➢ Checksums, TTLs
• Read the instructions carefully
• 500+ lines of code, so start early
• Final submission: **Oct. 23\textsuperscript{th} at 5pm**
How to test your code

• Test connectivity with ping from a server or the client
• Traceroute will not work well outside of Mininet:
  ➢ Use Mininet CLI
  ➢ mininet> server1 traceroute –n server2
• HTTP requests with wget, curl
• Don’t forget to test “error” cases!
Some advice

• Be thorough in your testing
• Do not hesitate to change the routing table (what about an incorrect routing table?)
• Be careful when implementing Longest Prefix Match
• Don’t get mixed up with **endinanness**: Linux is little endian, network big endian
  ➢ Try to put the calls to hton, ntoh in a single place
• Write good quality code
  ➢ Do not hardcode constants, avoid code duplication ...
Things that may be useful

• Mininet console, which supports tcpdump, ping, traceroute (apt-get install traceroute on instance)
• Debug functions in sr_utils.c
  ➢ print_hdrs, print_addr_ip_int
• GDB/Valgrind
Start reading!

http://www.cs.toronto.edu/~yganjali/courses/csc458/assignments/simple-router/