CS 60008: Internet Architecture and Protocols
Department of CSE, IIT Kharagpur
Course Instructor: Sandip Chakraborty

Assignment 2

Creating a Static Routing Logic in Mininet

Preface – Traditional Routing

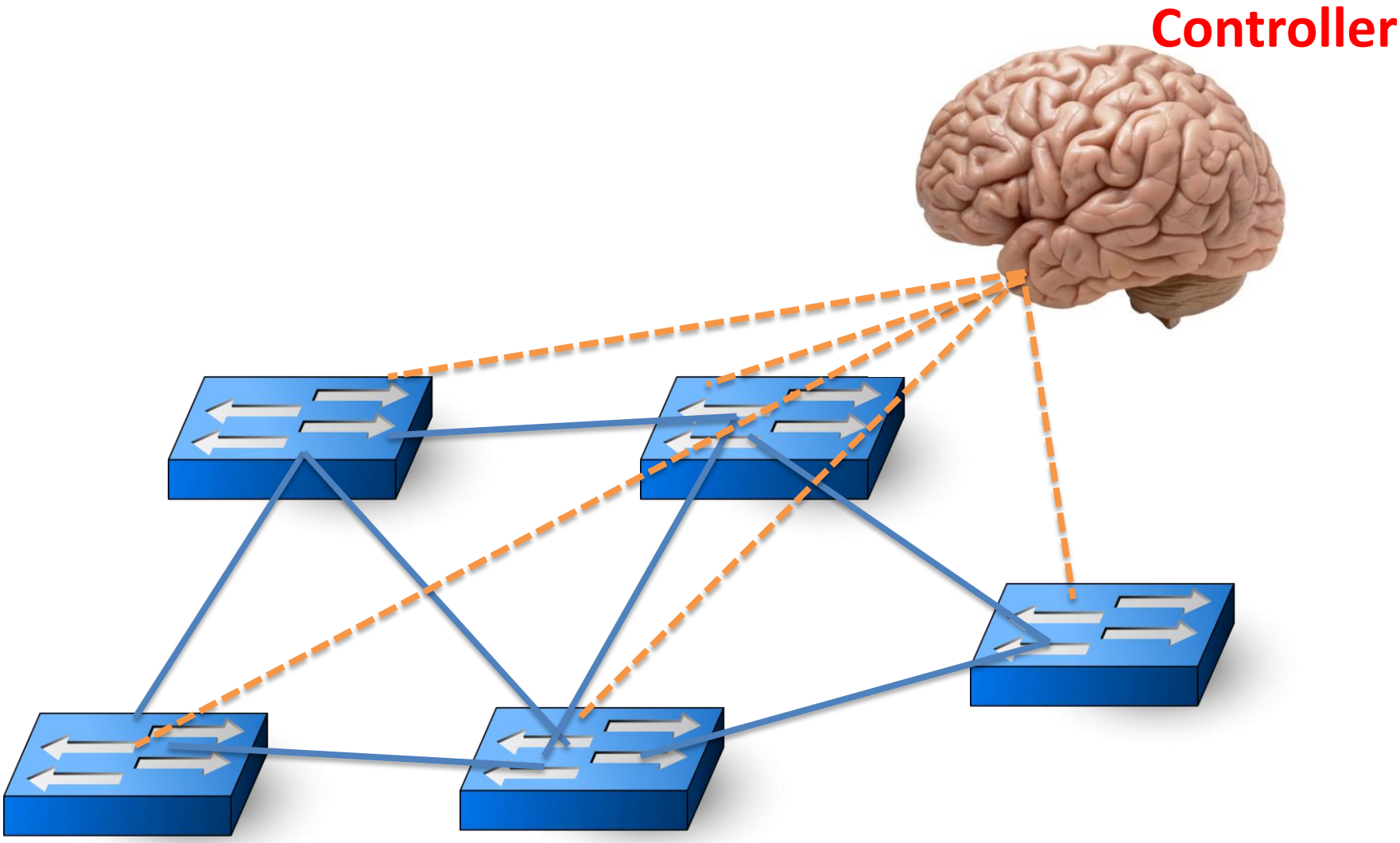
- In a standard network, every router contains its routing logic – A network administrator needs to manually configure the route options at every individual router. Possible route options include
  - Configure the protocols types – what type of routing protocol to use - static routing, RIP or OSPF
  - For a dynamic routing protocol like RIP or OSPF, its own configuration parameters, like frequency of route updates etc.

- The route configurations need to be uniform across all the routers !

Preface – SDN Routing

- In a SDN based network, the network administrator can load the routing logic at the controller – so this is an one time business!
- Based on the routing logic, the controller creates and dumps the configurations at every individual switch.
- Now, the switches can work as a router, where the controller is the brain of the network !

Preface – SDN Routing



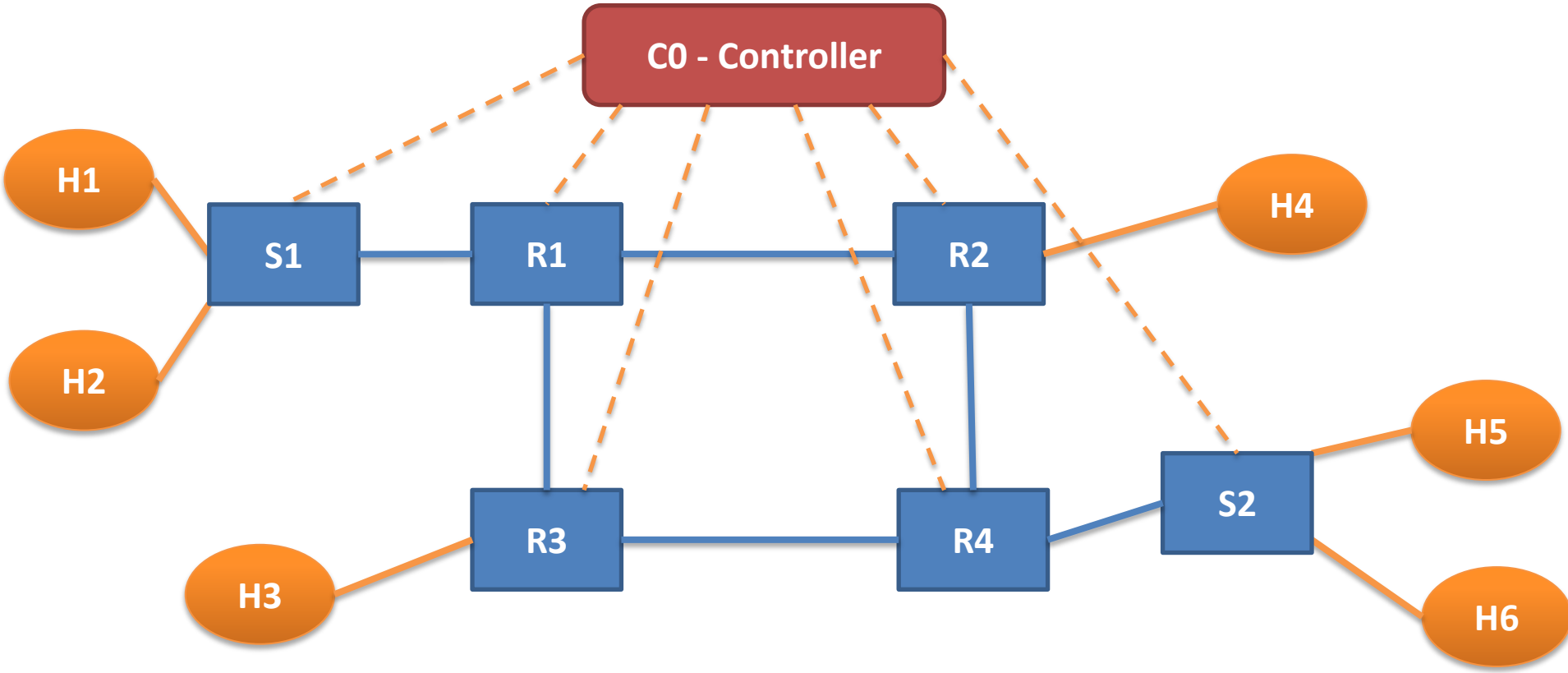
Objectives of this Assignment

- Implement routing logic (IP Forwarding mechanism) in a OpenFlow controller over Mininet.

- Implement supporting protocols at controller –
  - ARP (Explore why we require ARP support)
  - ICMP (Explore why we require ICMP)

Assignment – Topology Construction

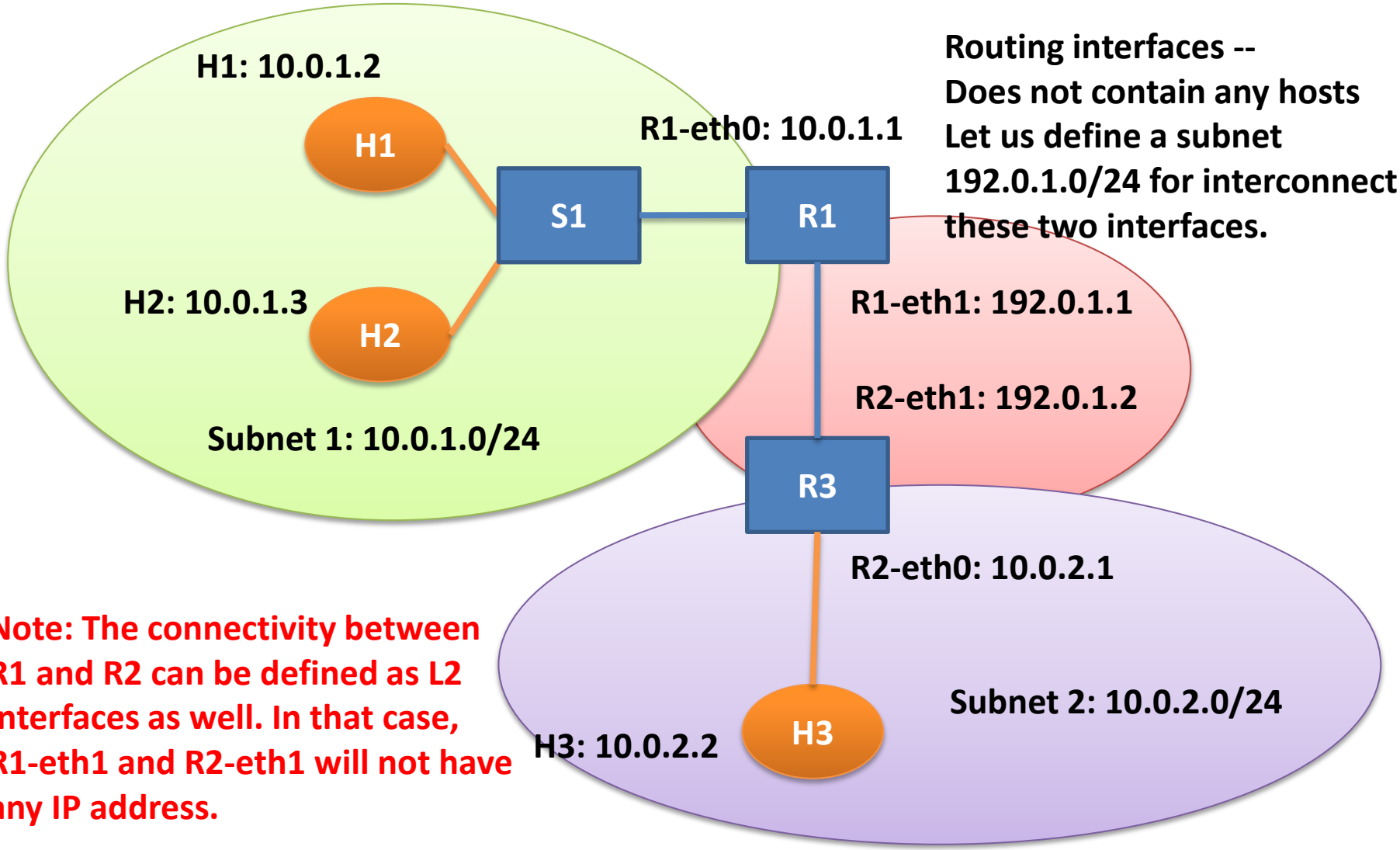
- Create a routing topology in Mininet as follows.



Topology Details

- Four OpenFlow switches that act as routers (or L3 switches)
  - R1 (Subnet: 10.0.1.0/24)
  - R2 (Subnet: 10.0.2.0/24)
  - R3 (Subnet: 10.0.3.0/24)
  - R4 (Subnet: 10.0.4.0/24)
  - Assign interface IPs for these L3 switches
- Two L2 switches S1 and S2 – they do not act as routers, works only at Layer 2.
- Six Hosts – Assign the IP to the hosts based on the subnet IPs of the corresponding switches
- One Controller

How to Assign IPs to L3 Switches



Note: The connectivity between R1 and R2 can be defined as L2 interfaces as well. In that case, R1-eth1 and R2-eth1 will not have any IP address.

Assignment – Task 1 (Create a Static Routing Table)

- In this assignment, we'll use static routing. Read RFC 1812 (https://tools.ietf.org/html/rfc1812) Section 7.4 to know the requirements of static routing.

- In a static routing protocol, the routing table (also known as *Forwarding Information Base* or FIB, in short) is developed by the network administrator, and fed to the routers – the routing tables have fixed entries and do not get changed automatically, unless changed by an administrator.

Assignment – Task 1 (Create a Static Routing Table)

- Each network node has a configured subnet.
- If a packet is destined for a host in that subnet, the node acts as a switch and forwards the packet **to a known port** or **broadcast**.
- If a packet is destined for a host in other subnet, it finds out the next hop from the route table (by consulting with the controller if information is not available at the local route table), finds out the MAC of the next hop from ARP, and then it should modify the layer 2 destination, and forward the packet to a correct port

The description has been taken from https://github.com/mininet/openflow-tutorial/wiki/Router-Exercise - although the OpenFlow description is bit old !

Assignment – Task 1 (Create a Static Routing Table)

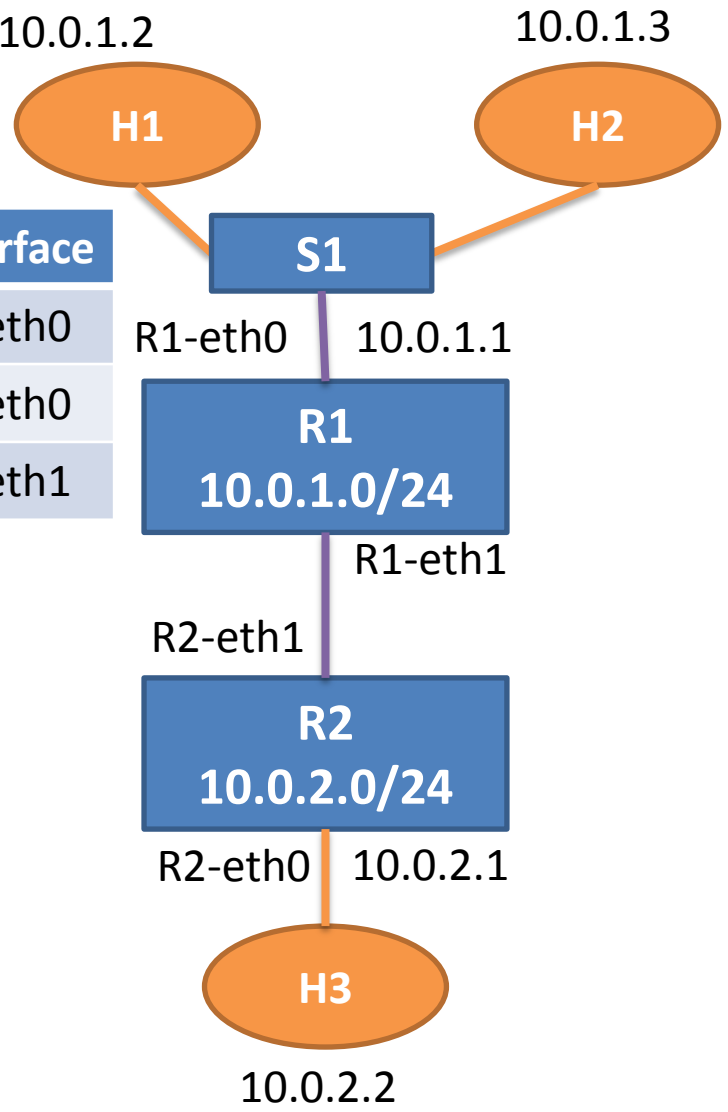
- Your router will make routing decision based on a fixed routing table that you'll define in a text file. The routing table has following entries,
  - Destination IP
  - Next hop
  - Netmask
  - Interface

Assignment – Task 1 (Create a Static Routing Table)

- **A sample routing table for R1:**

IP	Netmask	Next Hop	Interface
10.0.1.2	255.255.255.255	10.0.1.2	R1-eth0
10.0.1.3	255.255.255.255	10.0.1.3	R1-eth0
10.0.2.0	255.255.255.0	10.0.2.1	R1-eth1

- Note: Here I assume that R1-eth1 and R2-eth1 are connected via L2 interfaces.



Assignment – Task 2 (IP Forwarding)

- Develop the IP forwarding logic for the controller:
  - Note that the router receives an Ethernet frame
  - Sanity check the packet – check that the packet meets minimum length and has correct checksum
  - Decrement the TTL by 1, recompute the packet checksum over the modified header.
  - Find out the next hop from the routing table (note the longest prefix match in route lookup)
  - Check the ARP cache for the next hop MAC address corresponding to the next hop IP. If it is there, create the Ethernet frame and send it. Otherwise, send an ARP request for the next hop IP, and add the packet to the queue of packets waiting on this ARP request.

Assignment – Task 2 (IP Forwarding – Role of Controller)

- Note that initially the L3 switches (or routers) do not have forwarding information. The global routing table is stored at the controller.

- Whenever, a L3 switch receives a packet for which it does not have any information, it forwards the packet to the controller.

- The controller figures out the next hop information for that packet and sends it to the L3 switch (via OpenFlow API).

- For L2 forwarding and error handling, the controller defines appropriate API for ARP and ICMP, as discussed next.

Assignment – Task 3 (ARP)

- In general, a router responds to ARP requests by maintaining a ARP cache, however, in our SDN based scenario, the router is a dumb device!
- Therefore, the controller handles the ARP requests, once it receives a ARP request, it should send back a ARP reply.
- Configure the controller such that it can construct a ARP reply, and can send it back (check the sample code for load balancer to get an example of how to handle ARP packets).
- The ARP structure in a POX controller contains:
  - hwdst (hardware address of destination - what we are asking for)
  - hwsrc (hardware address of source)
  - protodst (IP address of destination)
  - protosrc (IP address of source)
  - opcode (type of arp package[REQUEST, REPLY, REV_REQUEST, REV_REPLY])

Assignment – Task 4 (ICMP)

- Note that the routers need to manage any error that may occur during the packet forwarding, for example, it may receive a packet destined for a host which does not exists. In such a case it should send a “ICMP: Host Unreachable” packet.
- Study the ICMP protocol to figure out available error management options. You can check RFC 792 (https://tools.ietf.org/html/rfc792)
- For such an error, your controller may receive the ICMP packets, therefore you should write proper routine to handle such messages
  - Example: packets for unreachable subnets should be responded to with ICMP network unreachable message (i.e. you can make a traceroute from any host to any other host)

Test Cases

- You should be able to,
  - Ping the routers and all the hosts from any other host
  - Traceroute a host from any other host
  - If a host or subnet does not exists, the traceroute should return a host unreachable or network unreachable message.
  - Run iperf to test tcp and udp traffic performance.
    - Measure the tcp bandwidth
    - For UDP, measure the performance by increasing UDP data generation rate (use –b option for iperf) – plot a graph of bandwidth (x-axis) vs throughput (y-axis)

Report

- Produce a report (**in pdf format**) that should contain the followings.
  - The complete network topology with appropriate IP assignment to all the L3 interfaces.
  - The routing tables for R1, R2, R3 and R4
  - Screenshot of ping results from H1 to all other hosts
  - Screenshot of ping results from H1 to IP 10.0.5.2
  - Screenshot of traceroute from H2 to all other hosts
  - Screenshot of traceroute from H2 to IP 10.0.9.2 where 10.0.9.0/24 is a invalid subnet (this should return a network unreachable)
  - Screenshot of traceroute from H2 to IP 10.0.4.9 where this is not a valid IP of any host (this should return a host unreachable)
  - A table measuring TCP bandwidth between H1 and all other hosts
  - Bandwidth vs Throughput graph for UDP between H1 and H6.
  - Bandwidth vs Throughput graph for UDP between H1 and H2.
  - Provide a point-wise summary of observations that you made from this experiment.
  - Contribution of every member of the group.

Submission Instructions

- **Submission Deadline: 14/02/2016** (If you want to enjoy the Valentine's Day, you are allowed to submit earlier, but not later ☺)
- Create a zip file containing
  - The source code that you have written. Create a folder "source" and put the source code there.
  - A readme.txt file mentioning the procedure to compile and execute your code. Additional credits will be given for providing a script for automatic compilation and execution of the code.
  - The pdf file containing the report
- If you want to put any additional items, first consult with the instructor or the TAs.
- Rename the zip file as IAP_2_G where G is your group number. Upload the zip file at TutorSpace by the deadline. Additionally, you may mail the zip file to sandipc@cse.iitkgp.ernet.in with a cc to sumitros@gmail.com .

Online Resources

- A must read tutorial on Mininet: https://github.com/mininet/mininet/wiki/Introduction-to-Mininet

- This is a bit old, but a good tutorial on how to set up a OpenFlow controller in Mininet: https://github.com/mininet/openflow-tutorial/wiki/Learn-Development-Tools


KEEP
CALM
AND
HAPPY
CODING