Teaching Computer Networking
A Hands-on Course

Deniz Gurkan

University of Houston Networking Lab

June 17, 2018
Why Teach Computer Networking
Computer Networking

- TEKS
- Future career opportunities
- Life
Computer Networking
Technology Applications CS and CTE Networking TEKS

- CTE Subchapter K: 130.305, 306. Networking and Lab (One Credit)
- Tech Apps Subchapter C: 126.33, 34, 35. Computer Science I, II, III
- Tech Apps Subchapter C: 126.49, 50. Independent Study I, II
Computer Networking
Future Career Opportunities for Students

- Science, technology, and engineering
- System administration, computer science, cybersecurity
  engineer, network engineer, ...

Career
  - Where:
    - Intern or work part-time in any industry while going to college
    - Office spaces for various professionals all trying to access their data
  - How: Problem solving, communicating with people, and building real life systems
  - What $: Six figure salaries
Computer Networking: Universal life skill

- Commodity in our lives: water, electricity, network, car, bank account, cell phone, food
- Relationships, social networks, road/highway systems, water pipes, electricity grid, etc. are network graphs
Course Components
Course Components

- Teaching computer networking with hands-on protocol observations and dynamic evaluations
  - Lectures
  - Lab Modules
  - Quiz and midterm
- Student and teacher perspectives
Student Experience
Student Experience

- Lecture slides with protocol descriptions
- Hands on lab experiment modules for protocol behavior observations
  - Troubleshooting through a project on a broken network:
    - Procedures
    - Packet formats (data representations)
- Quizzes with immediate feedback on learning of algorithms and procedures
  - Reflection on own learning
  - Algorithm and procedure learnings
Lectures

- Lecture Slides
- Descriptive Text Handouts
Lectures
Protocol Behavior Descriptions to Accommodate Diverse Learning Styles

- Verbal description
- Flow charts
- Flow chart & packet traces on networks
- Animations of protocol actions
- Lab modules and projects
Lectures

Verbal Description of Protocols

MAC Learning Operation

- Listen to every packet at ports
  - Read source MAC address
    - If not there, store in the L2 table:
      - source MAC address
      - incoming port number
  - Read the destination MAC address and check the L2 table:
    - No match for destination MAC address: forward on all ports except the incoming port (= flooding)
    - Match the destination MAC address:
      - if port is the same as the incoming port, drop
      - forward on the specified port (= selective send)
- Each entry in the L2 table expires after some time (aging time)
Lectures
Flow Charts - One for Each Protocol

Well-formed Packet Received

Known src_mac?

no
Add (in_port, src_mac) to match table

yes
Known dst_mac?

no
out_port_list = [all_ports]

yes
out_port_list = [num]

out_port_list = out_port_list - in_port

len(out_port_list) >= 1?

yes
Send Packet out port

no
port = pop(out_port_list)

STOP
Lectures
Flow Charts - One for Each Protocol: Zoom In 1

Well-formed Packet Received

Known src_mac?

Add (in_port, src_mac) to match table

Known dst_mac?

out_port_list = [all_ports]

out_port_list = [num]

out_port_list = out_port_list - in_port
len(out_port_list) >= 1?

**yes**

**port = pop(out_port_list)**

**Send Packet**

**out port**

**no**

STOP
Lectures
Flow Charts - Trace Packets over the Network

```
layer 2 VLAN-tagged packet:

<table>
<thead>
<tr>
<th>MAC</th>
<th>Port</th>
<th>VLAN ID</th>
<th>Age</th>
</tr>
</thead>
<tbody>
<tr>
<td>host2 eth0 MAC</td>
<td>1</td>
<td>0x100</td>
<td>100</td>
</tr>
<tr>
<td>host1 eth0 MAC</td>
<td>1</td>
<td>0x0800</td>
<td>100</td>
</tr>
</tbody>
</table>

payload
```

```
layer 2 packet:

<table>
<thead>
<tr>
<th>MAC</th>
<th>Port</th>
<th>VLAN ID</th>
<th>Age</th>
</tr>
</thead>
<tbody>
<tr>
<td>host2 eth0 MAC</td>
<td>1</td>
<td>0x0800</td>
<td></td>
</tr>
</tbody>
</table>

payload
```
Lectures
Each Protocol with Animated Packet Tracing

Learn the source MAC address

Match on destination MAC address: Flood

Match on destination MAC address: Flood

Learn the source MAC address and match on destination MAC address and selective send
Lab Modules
Everybody gets their own network!

- Jupyter Notebooks: Interactive Computation Environment in a Browser
  - Note-taking
  - Observations
  - Experiment service with network topology
- Peripheral skills on IT, system administration, and computational thinking
- Individual and team-based learning: both are valuable and may be applied per lab experiment
- Visualization of all aspects of networks:
  - Data: packets, configuration, end device applications
  - Network: topology, network device settings, overall state documentation

1 Jupyter Project: https://jupyter.org
Lab Modules

Jupyter Notebooks in a Browser

Load libraries

In [4]: %load_ext uhed
Jupyter Notebooks in a Browser

Pick a Lab from Dropdown Menu

In [6]: %lab

Lab 1 - Ethernet Bridge MAC learning
Lab 2 - VLANs
Lab 3 - Subnets
Lab 4 - DHCP
Lab 5 - Routing
Lab 6 - Routers
Lab 7 - DNS
Jupyter Notebooks in a Browser

Load libraries

```python
In [4]: %load_ext uheld
```

Load the lab topology: Pick a lab and click on **Build Network**

```python
In [ ]: %lab
```

**Now, your network is ready and a manifest holds the network topology as reserved**

**Step 1:** Display reservation topology diagram by calling `genish.showtopo`. Change your display options to one of `dot`, `circo`, or `neato` for an optimal view of network topology.

```python
In [ ]: genish.showtopo(manifest, 'dot')
```

**Step 2:** An ssh config has been written for you - ssh your reservation hosts using your slice name:

```bash
ssh SLICE_HOSTNAME
```

```python
In [ ]: SITE.getHostTable(context, SLICE, 'bridgel')
In [ ]: SITE.Host.getARPTable(context, SLICE, 'host1')
In [ ]: SITE.getLeaseInfo(context, SLICE, 'dscp0')
In [ ]: SITE.Host.getRouteTable(context, SLICE, 'router')
In [ ]: SITE.Host.getRouteTable(context, SLICE, 'host1')
```

**Conduct your observation lab here using the ppt instructions**

```python
In [ ]: #delete
```
Jupyter Notebooks in a Browser

Typical Note-taking and Lab Experiment

Conduct your observation lab here using the instructions

```
In [24]: SITE.getL2Table(context, SLICE, 'bridge')
Out[24]: Port VLAN MAC Age

In [26]: SITE.getL2Table(context, SLICE, 'bridge')
Out[26]: Port VLAN MAC Age
1 0 22:89:95:95:5b 179

In [28]: SITE.getL2Table(context, SLICE, 'bridge')
Out[28]: Port VLAN MAC Age
1 0 22:89:95:95:5b 177
2 0 46ae60f8:9057 37
```
Jupyter Notebooks in a Browser

Show Network Topology
Jupyter Notebooks in a Browser
Make Observations

Conduct your observation lab here using the instructions

```
In [24]: SITE.getL2Table(context, SLICE, "bridge")
Out[24]:
    Port  VLAN  MAC  Age

In [26]: SITE.getL2Table(context, SLICE, "bridge")
Out[26]:
    Port  VLAN    MAC  Age
       1    0  22:f8:95:83:95:fb   176
```
Jupyter Notebooks in a Browser

Track the state of bridge L2 table and delete reservation

```
In [24]: SITE.getL2Table(context, SLICE, "bridge")
Out[24]:
Port  VLAN  MAC     Age
      0  22:f8:95:83:95:fb  176

In [26]: SITE.getL2Table(context, SLICE, "bridge")
Out[26]:
Port  VLAN  MAC     Age
      0  22:f8:95:83:95:fb  177
      2  46:ae:6d:c8:90:57   37

In [ ]: %delete
```
Jupyter Notebook in a Browser
Many Observations in One Place

tcpdump -i eth1 -e not ether dst 01:80:c2:00:00:00
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth1, link-type EN10MB (Ethernet), capture size 262144 bytes
03:25:38.529984 2a:f9:c4:8c:92:c4 (oui Unknown) > aa:bb:cc:dd:ee:5d (oui Unknown), ethertype Unknown (0x0820), length 19:
0x0000: 68 65 6c 6c 6f hello
Lab Modules: Observations and Investigations

- Packets
- Configuration
- Network

- Objectives
- Network Topology
- Observation Points and Environment
- Experiment Activity
Lab Modules: Description of Methods

Lab 1: MAC Learning in Ethernet Bridges

Objectives
- Load Lab Network
- Topology Orientation
- Observation Plan and Environment

Observations

Loops in Ethernet Topologies
- Spanning Tree Protocol
- Lab 2: Spanning Tree Protocol
- VLAN Scoping
- Internet Protocol
- IP Subnet Address Ranges

The environment includes two terminal screens sending packets, and three others listening on their interfaces to infer the bridge forwarding behavior. The iPython lab notebook will be used to observe how bridge L2 table is populated with source MAC addresses of incoming packets.

Observations
Lab Modules: Packet View

Layer 2 packet:

- **destination MAC**: 10.10.75.42
- **source MAC**: 192.168.1.17
- **type**: N
- **payload**: hello
- **destination IP**: 10.10.75.42
- **source IP**: 192.168.1.17
- **TTL**: 2
- **protocol**: TCP
- **other fields**: 0x0000

TCPdump output:
```
tcpdump -i eth1 -e not ether dst 01:80:c2:00:00:00
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth1, link-type EN10MB (Ethernet), capture size 262144 bytes
03:25:38.529984 2a:f9:c4:8c:92:c4 (oui Unknown) > aa:bb:cc:dd:ee:5d (oui Unknown), ethertype Unknown (0x0820), length 19:
  0x0000: 68 65 6c 6c 6f
```

Teaching Computer Networking University of Houston Networking Lab
Lab Modules: Configuration towards Network State

<table>
<thead>
<tr>
<th>MAC</th>
<th>Port</th>
<th>VLAN ID</th>
<th>Age</th>
</tr>
</thead>
<tbody>
<tr>
<td>host0</td>
<td>0</td>
<td>100</td>
<td>7</td>
</tr>
<tr>
<td>host3</td>
<td>1</td>
<td>100</td>
<td>5</td>
</tr>
<tr>
<td>host1</td>
<td>0</td>
<td>200</td>
<td>1</td>
</tr>
</tbody>
</table>

```
bridge2

In [14]: SITE.getL2Table(context, SLICE, "bridge")
Out[14]:
```

```
<table>
<thead>
<tr>
<th>Port</th>
<th>VLAN</th>
<th>MAC</th>
<th>Age</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>aa:bb:cc:dd:ee:0b</td>
<td>58</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>aa:bb:cc:dd:ee:03</td>
<td>38</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>aa:bb:cc:dd:ee:08</td>
<td>36</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>aa:bb:cc:dd:ee:0e</td>
<td>35</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>aa:bb:cc:dd:ee:0c</td>
<td>19</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>aa:bb:cc:dd:ee:09</td>
<td>18</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>aa:bb:cc:dd:ee:07</td>
<td>17</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Client ID</th>
<th>ifindex</th>
<th>vlan</th>
<th>MTU</th>
<th>Admin State</th>
<th>Link State</th>
<th>RX Bytes (Pkts)</th>
<th>TX Bytes (Pkts)</th>
</tr>
</thead>
<tbody>
<tr>
<td>bridge1:0</td>
<td>2028</td>
<td>None</td>
<td>1500</td>
<td>up</td>
<td>up</td>
<td>0 (0)</td>
<td>2738136 (52652)</td>
</tr>
<tr>
<td>bridge1:2</td>
<td>2018</td>
<td>None</td>
<td>1500</td>
<td>up</td>
<td>down</td>
<td>180100 (3459)</td>
<td>752 (10)</td>
</tr>
<tr>
<td>bridge1:3</td>
<td>2024</td>
<td>None</td>
<td>1500</td>
<td>up</td>
<td>up</td>
<td>2738292 (52655)</td>
<td>763 (12)</td>
</tr>
<tr>
<td>bridge1:1</td>
<td>2016</td>
<td>None</td>
<td>1500</td>
<td>up</td>
<td>up</td>
<td>2738344 (52656)</td>
<td>752 (10)</td>
</tr>
</tbody>
</table>
Lab Modules: Network Topologies

```
In [7]: genish.showtopo(manifest)
Out[7]:
```

![Network Topology Diagram](image-url)
Lab Modules: Network troubleshooting experiments as projects

- Students are capable of debugging and fixing networks
- Broken network experiment for students to fix
Lab Modules: Project Topology

 IPv4 Router
 L2 Switch

 R2
 R1
 R0
 R3
 R4
 R6

 S0.0
 S0.1

 S5.0
 S5.1
 S5.2
 S5.3

 S6.0
 S6.1
 S6.2
 S6.3
 S6.4

 DHCP2
 DHCP3
 DHCP4
 DHCP5

 h2:1
 h3:1
 h2:2
 h3:2
 h3:3
 h3:4
 h2:3
 h3:5
 h3:6
 h2:4
 h4:1
 h5:1
 h4:2
 h5:2
 h5:3
 h5:4
 h2:5
 h5:5
 h5:6
 h4:4
 h4:5
 h4:6

 Teaching Computer Networking
 University of Houston Networking Lab
Lab Modules: Broken Network to be Fixed as a Project - Methodology

- Network State Documentation
- Investigative Process
- Testing Tools
Quizzes: did I learn? If yes, what did I learn? If no, what did I not learn?

- Direct evaluation of procedures and algorithms
  - Test with open notes/google/etc
  - Learn by making a mistake - one more way to engage in learning!
- Apply learnings on a network
  - Experience the packet traced on a network using the flow charts
  - Learn by applying
Teacher Perspective
Teacher Perspective

- Lecture slides descriptive diagrams of network topologies, packets, algorithms
- Hands-on lab experiment modules for observations on what lectures have covered
  - Student activity data
  - Isolated learning environment per student
- Quizzes with immediate feedback on learning of algorithms and procedures
  - Autograded: a dynamic evaluation capability
Lectures

- Slides with descriptive diagrams, network topologies, packet formats
- Descriptive text with analogies, implementation insights
Lectures: Slides with packet traces over topologies
Lectures: Slides with packet traces over topologies

Time to live field of packet is set to 1

 originateator

destination IP: 10.0.3.28
source IP: 10.0.1.108
TTL: 1

type: 0x800
destination MAC: router1 eth1
source MAC: origin eth0

router1
TTL: 0

router2

Teaching Computer Networking
University of Houston Networking Lab
Lectures: Slides with packet traces over topologies
Time to live field of packet is set to 1
Lectures: Slides with packet traces over topologies
ICMP TTL Expired

originator

eth0
10.0.1.108

ICMP TTL expired

destination IP: 10.0.1.108
source IP: 10.0.1.254
TTL: 30

type: 0x800
destination MAC: origin eth0
source MAC: router1 eth1

router1
eth1 10.0.1.254
eth2 10.50.0.1

eth1 10.50.0.2

eth0 10.0.3.28

eth0 10.0.1.108
eth1 10.0.1.254
eth2 10.50.0.1
eth1 10.50.0.2
eth1 10.0.3.254
Lectures: Slides with packet traces over topologies
ICMP TTL Expired
Lectures: Descriptive text on all lectures and labs

Lecture & Lab Manuals:
- Analogies for each concept:
- In-depth descriptions of protocols:
- Packet data structure and format:

Example of an analogy:

**Analogy: Envelope and Letter vs Packet Header and Payload**

Ethernet packet format includes addresses for **from** and **MAC** addresses. This is similar to how we write on an **envelope** analogous to the header portion of the packet. Sometimes a particular stamped identifier on the envelope can indicate the **destination** MAC address.

**In-depth description of protocols**:

**Protocol Components**

- **Ethernet Frame**
  - The Ethernet frame includes a **preamble** and FCS (frame check sequence) for network interfaces. The starting bit of the data part of the frame is zero bits. The trailing FCS verifies frame's integrity. The frame format is such that all ethernet-capable interfaces operate under this same protocol.

- **Ethernet Packet**
  - The Ethernet packet is composed of source and destination MAC addresses.
Lectures: Descriptive text on all lectures and labs

Analogy: Envelope and Letter vs Packet Header and Payload

Ethernet packet format includes addresses for from and to in the form of MAC addresses. This is similar to how we write on an mailing envelope for a postcard, or a larger package. In this respect, the payload or data is what's in the letter, card, or whatever is in a package.

Protocol Components

Ethernet Frame

The Ethernet frame includes a preamble and FSD (frame sequence delimiter) that network interfaces can detect the starting bit of the data part of the frame, and zero bits. The trailing FCS (frame check sequence) verifies frame's integrity, and the frame format so all ethernet-capable interfaces operate under this spec.

Ethernet Packet

The Ethernet packet is composed of source and destination MAC addresses in the payload, the EtherType, or usually referred to as just the type, and the type that carries the message data. The message data could be the message itself or packet of a layer 3 (L3) protocol, specified in the type field of the packet.
Lectures and Lab Manuals along with Concepts

Lecture & Lab Manuals:

- Computer Networking
- Lab Environment
  - Class Terminology - need alphabetical ordering
- Introduction to Computer Networking
- Ethernet Protocol
  - Protocol Components
    1. MAC Address
    2. Data Type: EtherType
    3. Data
  - Ethernet Device Behavior
- Ethernet MAC Learning
- Lab - 1: MAC Learning in Ethernet Bridges
- Loops in Ethernet Topologies
- VLAN Scoping
- Internet Protocol
Lectures: Analogies, Deep Dives, Implementation Insights

Analogies/Deep dive/Implementation Insights for each concept:

1. **Analogy: Envelope and Letter vs Packet Header and Payload**
   - Ethernet packet format includes addresses for **from** and **to** in the form of MAC address, analogous to the envelope.

2. **Deep Dive: Ethernet Bridge Behavior**
   - **data:**
     - destination MAC
     - source MAC
   - For every incoming packet at bridge ports: Listen to all ports all the time.

   **Tip**
   - Implementation Insight
   - What does it mean for a bridge port to **run a protocol**, e.g., **run STP**?
   - Ethernet bridges are plug-and-play devices. They are placed in networks with a configuration. As mentioned, administrative oversights may happen and loops...
Lectures: In-depth description of protocols

In-depth descriptions of protocols:

Address Resolution Protocol

Any host trying to send a packet to another host with a known IP address needs to find a packet. Because the layer 2 packet is unaddressed. For example, in the scenario topology address, 192.168.1.15.

STP Protocol and Algorithm

The network graph with loops for the physical communication protocol that runs between the exchange messages to map the existing topology numbers, and configuration preference of the existing bridges. The root bridge forms the
**Lectures: Packet data structures**

### Packet data structure and format:

#### Layer 2 packet:
- **destination MAC**
- **host 1 MAC**
- **type**
- **payload**

#### Layer 3 packet:
- **192.168.1.15**
- **192.168.1.67**
- **TTL**
- **protocol**
- **other fields**
- **payload**

#### ARP request:
- **destination MAC**
- **source MAC**
- **0x0806**
- **payload**

#### DHCP discover:
- **FF:FF:FF:FF:FF**
- **host MAC 0x0800**
- **payload**
  - **DHCP message type**
  - **Request information**
  - **other fields**

#### Layer 2 packet:
- **destination MAC**
- **source MAC**
- **type**
- **payload**

#### Layer 3 packet:
- **destination IP**
- **source IP**
- **TTL**
- **protocol**
- **other fields**
- **payload**

#### Layer 4 packet:
- **header fields of layer 4**
- **payload**
Lab Modules

- Lab experiment detailed description
- Network topologies ready for each student to perform on an isolated real-life environment
- Dynamic student activity data available as an instructional tool
Lab Modules: Dynamic Student Activity Data from Lab Experiment Activities

- What activities did the students finish with the least number of required steps?
- What investigations did the students perform in a repeated fashion?
- What observations have been completed by the class for a given lab experiment?
- How do these activities translate to higher/lower performance in learning as observed in quizzes?
Lab Modules: Student Activity Data

Activity 1: ◇, 2: △, 3: □, 4: ○

Student 1
- get bridge L2 table
- ssh into host
- send a packet
- create network topology

Student 2

Student N

time
Lab Modules: A real-life testbed to realize instructional innovations

- Lab experiment modules readily available
- There is capability to design one’s own experiments too!
  - Different network design options
  - Novel troubleshooting use cases
  - Support student curiosity over protocol observation tasks

- Virtualized environment:
  - Open source software package
  - Per student account
  - Isolated networks per experiment per student
Lab Modules: A real-life testbed to realize instructional innovations
Quizzes and Exams: Dynamic Evaluation of Instructional Methods

- Auto-grading capability of all quizzes and midterm
- What concepts and what aspect of each concept was missed?
  - Quizzed are designed to isolate individual educational elements (learning goals)
  - For example: a quiz on MAC learning, individual answers for direct forwarding aspect of Ethernet bridge behavior - what aspect of the MAC learning was missed/understood well?
- All quiz/midterm/lab experiment topologies are live, instantiated at run time for realistic evaluations
Given the topology, indicate all actions at all devices when host2 pings host4.

<table>
<thead>
<tr>
<th>Node (Interface)</th>
<th>Action</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>host2.eth1</td>
<td>dhcp-request</td>
<td>port 0, MAC H2E1, no VLAN</td>
</tr>
<tr>
<td>br2</td>
<td>flood</td>
<td></td>
</tr>
<tr>
<td>dhcp.eth1</td>
<td>dhcp-reply</td>
<td></td>
</tr>
<tr>
<td>br2</td>
<td>output</td>
<td>port 0, MAC H2E1, no VLAN</td>
</tr>
<tr>
<td>host2.eth1</td>
<td>arp-request</td>
<td>10.50.50.79</td>
</tr>
<tr>
<td>br2</td>
<td>flood</td>
<td></td>
</tr>
<tr>
<td>host4.eth1</td>
<td>arp-reply</td>
<td>H4E1 (or host4.eth1.mac)</td>
</tr>
<tr>
<td>br2</td>
<td>output</td>
<td>port 0, MAC H2E1, no VLAN</td>
</tr>
<tr>
<td>host2.eth1</td>
<td>echo-request</td>
<td>10.50.50.79</td>
</tr>
<tr>
<td>br2</td>
<td>output</td>
<td>port 2, MAC H4E1, no VLAN</td>
</tr>
<tr>
<td>host4.eth1</td>
<td>arp-request</td>
<td>host2.eth1.ip</td>
</tr>
<tr>
<td>br2</td>
<td>flood</td>
<td></td>
</tr>
<tr>
<td>host2.eth1</td>
<td>arp-reply</td>
<td>H2E1 (or host2.eth1.mac)</td>
</tr>
<tr>
<td>br2</td>
<td>output</td>
<td>port 2, MAC H4E1, no VLAN</td>
</tr>
<tr>
<td>host4.eth1</td>
<td>echo-reply</td>
<td>host4.eth1.ip</td>
</tr>
<tr>
<td>br2</td>
<td>output</td>
<td>port 0, MAC h2E1, no VLAN</td>
</tr>
</tbody>
</table>

Quiz question: Dynamic topology per student
Midterm question: Troubleshooting Network State

Given the topology below:

Route Table (host1/2)

<table>
<thead>
<tr>
<th>Destination</th>
<th>Interface</th>
<th>Gateway</th>
</tr>
</thead>
<tbody>
<tr>
<td>172.18.12.0/24</td>
<td>eth1</td>
<td>-</td>
</tr>
<tr>
<td>0.0.0.0/0</td>
<td>eth1</td>
<td>172.18.12.1</td>
</tr>
</tbody>
</table>

Route Table (R1)

<table>
<thead>
<tr>
<th>Destination</th>
<th>Interface</th>
<th>Gateway</th>
</tr>
</thead>
<tbody>
<tr>
<td>172.18.12.0/24</td>
<td>eth1</td>
<td>-</td>
</tr>
<tr>
<td>10.37.12.0/24</td>
<td>eth2</td>
<td>-</td>
</tr>
</tbody>
</table>

MAC Table (R1)

<table>
<thead>
<tr>
<th>IP</th>
<th>MAC</th>
</tr>
</thead>
<tbody>
<tr>
<td>172.18.12.75</td>
<td>H1E1</td>
</tr>
<tr>
<td>172.18.12.8</td>
<td>H2E1</td>
</tr>
<tr>
<td>10.37.12.81</td>
<td>H3E1</td>
</tr>
<tr>
<td>10.37.12.27</td>
<td>H5E1</td>
</tr>
<tr>
<td>10.37.12.132</td>
<td>H4E1</td>
</tr>
</tbody>
</table>

Route Table (host3/4/5)

<table>
<thead>
<tr>
<th>Destination</th>
<th>Interface</th>
<th>Gateway</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.37.12.0/24</td>
<td>eth1</td>
<td>-</td>
</tr>
<tr>
<td>0.0.0.0/0</td>
<td>eth1</td>
<td>10.37.12.1</td>
</tr>
</tbody>
</table>

NOTE: Unsupplied tables are empty and table entries do not age.

Show the sequence of actions resulting from host1 pinging 10.37.12.132
### Midterm question: Troubleshooting Network State

<table>
<thead>
<tr>
<th>Node (Interface)</th>
<th>Action</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>host1.eth1</td>
<td>arp-request</td>
<td>172.18.12.1</td>
</tr>
<tr>
<td>br1</td>
<td>flood</td>
<td></td>
</tr>
<tr>
<td>R1.eth1</td>
<td>arp-reply</td>
<td>R1E1 (or R1.eth1.mac)</td>
</tr>
<tr>
<td>br1</td>
<td>output</td>
<td>port 2, MAC H1E1, no VLAN</td>
</tr>
<tr>
<td>host1.eth1</td>
<td>echo-request</td>
<td>10.37.12.132</td>
</tr>
<tr>
<td>br1</td>
<td>output</td>
<td>port 1, MAC R1E1, no VLAN</td>
</tr>
<tr>
<td>R1</td>
<td>output</td>
<td>interface eth2, MAC H4E1, no VLAN</td>
</tr>
<tr>
<td>br2</td>
<td>output</td>
<td>port 2, MAC H4E1, no VLAN</td>
</tr>
</tbody>
</table>
Quizzes: Dynamic Networks, Dynamic Evaluation - Autogradable
Quiz question: Dropdown menu to pick answers

<table>
<thead>
<tr>
<th>Vertex</th>
<th>Action</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>host0.eth1</td>
<td>send-packet</td>
<td>Dest. MAC: ff:ff:ff:ff:ff</td>
</tr>
<tr>
<td>dhcp0.eth1</td>
<td>send-packet</td>
<td>Dest. MAC: 66:7f:ef:55:65:71</td>
</tr>
</tbody>
</table>

Submit
Quizzes: Infer learnings on individual concepts

<table>
<thead>
<tr>
<th>Question</th>
<th>Concept</th>
<th>Performance Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAC learning</td>
<td>Flooding</td>
<td>70%</td>
</tr>
<tr>
<td>Direct Forward</td>
<td>60%</td>
<td></td>
</tr>
<tr>
<td>L2 table update</td>
<td>80%</td>
<td></td>
</tr>
<tr>
<td>VLANs</td>
<td>60%</td>
<td></td>
</tr>
<tr>
<td>Host MAC</td>
<td>90%</td>
<td></td>
</tr>
</tbody>
</table>

- Map concepts with statistics on answers
- Class performance on learning elements
Future Directions

- Student activity data may map onto learning objectives
- New lab experiment scenarios and project topologies may be developed using the open source platform
- Support for independent study on topologies by students: design networks, test and observe protocol behaviors
Teacher Training
Teacher Training

- Text
  - Description
  - Analogies
  - Implementation Insights
  - Deep Dive
  - Examples
- Lecture Slides
- Lab Modules
- Projects for Troubleshooting
- Instructional Insight Data
- Testbed Management
Teacher Training

- Text
  - Description
  - Analogies
  - Implementation Insights
  - Deep Dive
  - Examples
- Lecture Slides
- Lab Modules
- Projects for Troubleshooting
- Instructional Insight Data
- Testbed Management

Thank you - Questions?

Email: dgurkan@uh.edu