• HW2 due 9am Thursday
  – 0 late days
  – Review in class; solutions posted immediately after class

• Midterm posted after class on Thursday
  – Up to what is covered TODAY
  – Due Friday by 11:59pm
  – Take-home format, designed for ~2hrs
  – Open book, open notes, open Internet
  – No collaboration: no sharing with peers; no posting on StackOverflow, etc.
LINK LAYER:

Switch maps each host-port.

<table>
<thead>
<tr>
<th>Host</th>
<th>Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>H1</td>
<td>1</td>
</tr>
<tr>
<td>H2</td>
<td>2</td>
</tr>
<tr>
<td>N3</td>
<td>1</td>
</tr>
<tr>
<td>NY</td>
<td>1</td>
</tr>
<tr>
<td>N5</td>
<td>2</td>
</tr>
</tbody>
</table>
1. Host 1 to Router 1
   - **Host 1**: 192.168.0.2
   - **Router 1**: 192.168.110
   - **Host 2**: 192.168.112
   - **Routing Table**:
     - **SRC**: 192.168.0.2
     - **DST**: 192.168.110

2. Router 1 to **Host 2**
   - **Routing Table**:
     - **SRC**: 192.168.110
     - **DST**: 192.168.112
$H_1$ Mac Table

$R_{to}$

$H_2$

Hosts on same network
Administivia

- HTA/UTA apps due Wednesday (Mar 16) by 5pm
  - Interested, but already HTAing? Email me
- Summer work apps due Friday (Mar 18) by 11:59pm
  - Part time or full-time, in-person or remote
  - Don’t need to be TA’ing in the fall to do summer work (or vice versa)

Application links on Ed—if you’re interested, please apply and we can talk further!
Warmup

L2
SRC N1 → DST H2

L3 10.0.0.1 → 10.0.0.2
   10.0.0.1

H1 → S → H2

L2: LINK LAYER (MAC ADDRS)
L3: NETWORK LAYER (IP ADDRS)
Today: IP Wrap-up

• More on BGP
• IP Service models
  – Unicast, Broadcast, Anycast, Multicast
• Tunneling
BGP Recap

• Key protocol that holds Internet routing together
• Path Vector Protocol among Autonomous Systems (ASes)
• Route selection based on policy, rather than “optimal” routes
• Important security/stability problems
  – Misconfiguration
  – Prefix hijacking

\[ \text{IMPORT} \rightarrow \text{POLICY} \rightarrow \text{EXPORT} \]

\[ \Rightarrow \text{BGP SPEAKERS CAN LIE ABOUT ROUTES.} \]
What can be done?

Originally: Internet Routing Registries (IRRs): public database listing IP allocations

route: 10.0.0.0/8
descr: University of Blogging
descr: Anytown, USA
origin: AS65099
mnt-by: MNT-UNIVERSITY
notify: person@example.com
changed: person@example.com 20180101
source: RADB

But, database not verified and often incomplete/wrong
What can be done?

$whois -h whois.radb.net AS14325
aut-num:   AS14325
as-name:   ASN-OSHEAN
descr:     OSHEAN, Inc.
import:    from AS14325:AS-MBRS  accept PeerAS
mp-import: from AS14325:AS-MBRS  accept PeerAS
export:    to AS-ANY  announce AS14325:AS-MBRS
mp-export: to AS-ANY  announce AS14325:AS-MBRS
admin-c:   Tim Rue
tech-c:    Ventsislav Gotov
notify:    vgotov@oshean.org
mnt-by:    MAINT-AS14325
changed:   vgotov@oshean.org 20210512
source:    RADB
Proposed Solution: RPKI

- Based on a public key infrastructure
- Address attestations
  - Claims the right to originate a prefix
  - Signed and distributed out of band, checked on BGP updates
  - Checked through delegation chain from ICANN
- Can avoid
  - Prefix hijacking
  - Addition, removal, or reordering of intermediate ASes

Two Reg’s:
- Each AS needs to add signatures to DB
- ASes accepting routes need to validate against DB
RPKI deployment

RPKI-ROV Analysis of Unique Prefix-Origin Pairs (IPv4)

Valid: 35.12%
Invalid: 0.74%
Not-Found: 64.13%

Unique P-O
TOTAL: 996,018
Not-Found: 638,780
64.13%

Valid: 349,820
Not-Found: 638,780
Invalid: 7,418
Your ISP (Verizon, AS701) does not implement BGP safely. It should be using RPKI to protect the Internet from BGP hijacks. Tweet this →

Details

- Correctly accepted valid prefixes
- Incorrectly accepted invalid prefixes
Data Plane Attacks

• Routers/ASes can advertise one route, but not necessarily follow it!
• May drop packets
  – Or a fraction of packets
  – What if you just slow down some traffic?
• Can send packets in a different direction
  – Impersonation attack
  – Snooping attack
• How to detect?
  – Congestion or an attack?
  – Can let ping/traceroute packets go through
  – End-to-end checks?
• Harder to pull off, as you need control of a router
Different IP Service Models

- Unicast: send packet to one node
Different IP Service Models

- **Unicast**: send packet to one node
- **Broadcast**: send a packet to all nodes in some subnet. "One to all"
  - 255.255.255.255: all hosts within any subnet, never forwarded by a router
  - Last address in a prefix (host part is all 1’s):
    eg. 192.168.1.0/24 => Bcast: 192.168.1.255
  - Example use: DHCP
Anycast

- Multiple hosts may share the same IP address
- "One to one of many" routing
- Example uses: load balancing, nearby servers
  - DNS Root Servers (e.g. f.root-servers.net)
  - Google Public DNS (8.8.8.8)
  - IPv6 6-to-4 Gateway (192.88.99.1)
Anycast Implementation

• Anycast addresses are /32s
• At the BGP level
  – Multiple ASs can advertise the same prefixes
  – Normal BGP rules choose one route
Anycast Implementation

- Anycast addresses are /32s
- At the BGP level
  - Multiple ASs can advertise the same prefixes
  - Normal BGP rules choose one route
- At the Router level
  - Router can have multiple entries for the same prefix
  - Can choose among many
Anycast Implementation

- Anycast addresses are /32s
- At the BGP level
  - Multiple ASs can advertise the same prefixes
  - Normal BGP rules choose one route
- At the Router level
  - Router can have multiple entries for the same prefix
  - Can choose among many
- Each packet can go to a different server
  - Best for services that are fine with that (connectionless, stateless)
Multicast

- Send messages to many nodes: “one to many”
Multicast

- Send messages to many nodes: “one to many”
- Why?
  - Snowcast, Internet Radio, IPTV
  - Stock quote information
  - Multi-way chat / video conferencing
Multicast

- Send messages to many nodes: “one to many”
- Why?
  - Snowcast, Internet Radio, IPTV
  - Stock quote information
  - Multi-way chat / video conferencing
- What’s wrong with sending data to each recipient?
  - Link stress
  - Have to know address of all destinations
Multicast Service Model

- Receivers join a multicast group G
- Senders send packets to address G
- Network routes and delivers packets to all members of G
- Multicast addresses: class D (start 1110)
  - 224.x.x.x to 229.x.x.x
    - 28 bits left for group address
LAN Multicast

- Easy on a shared medium
- Ethernet multicast address range:
  - 01:00:5E:00:00:00 to 01:00:5E:7f:ff:ff
- Set low 23 bits of Ethernet address to low bits of IP address
  - (Small problem: 28-bit group address -> 23 bits)

How about on the Internet?

In general, not adopted over Internet
IP Tunneling

- Encapsulate an IP packet inside another IP packet
- Makes an end-to-end path look like a single IP hop

[Diagram showing the process of encapsulating an IPv6 packet inside another IP packet]
Other uses for tunneling

- Virtual Private Networks (VPN)
- Use case: access CS network from the outside
  - Set up an encrypted TCP connection between your computer and Brown’s OpenVPN server
  - Configure routes to Brown’s internal addresses to go through this connection
- Can connect two remote sites securely
A VPN: OVERVIEW

HOST CONNECTION

PKETS SENT TO VPN SERVER ARE DECRYPTED AND FORWARDED THROUGH CS NETWORK.
Next time: Transport Layer

- UDP, TCP, Congestion Control
- Application protocols
- ...
Extension Headers

- Two types: hop-by-hop and end-to-end
- Both have a next header byte
- Last next header also denotes transport protocol
- Destination header: intended for IP endpoint
  - Fragment header
  - Routing header (loose source routing)
- Hop-by-hop headers: processed at each hop
  - Jumbogram: packet is up to $2^{32}$ bytes long!
IPv6 in IPv4 Tunneling

- Key issues: configuring the tunnels
  - Determining addresses
  - Determining routes
  - Deploying relays to encapsulate/forward/decapsulate

- Several proposals, not very successful
  - 6to4, Teredo, ISATAP, 6rd
  - E.g., 6to4
    - Deterministic address generation
    - Anycast 192.88.99.1 to find gateway into IPv6 network
    - Drawbacks: voluntary relays, requires public endpoint address
Example Next Header Values

- 0: Hop by hop header
- 1: ICMPv4
- 4: IPv4
- 6: TCP
- 17: UDP
- 41: IPv6
- 43: Routing Header
- 44: Fragmentation Header
- 58: ICMPv6
Fragmentation and MTU

• Fragmentation is supported only on end hosts!
• Hosts should do MTU discovery
• Routers will not fragment: just send ICMP saying packet was too big
• Minimum MTU is 1280-bytes
  – If some link layer has smaller MTU, must interpose fragmentation reassembly underneath