Network Layer
Topics

• Network service models
  • Datagrams (packets), virtual circuits

• IP (Internet Protocol)
  • Internetworking
  • Forwarding (Longest Matching Prefix)
    • Helpers: ARP and DHCP
    • Fragmentation and MTU discovery
    • Errors: ICMP (traceroute!)
    • IPv6, scaling IP to the world
    • NAT, and “middleboxes”

• Routing Algorithms
Dynamic Host Configuration Protocol (DHCP)
Bootstrapping

• Problem:
  • A node wakes up for the first time ...
  • What is its IP address? What’s the IP address of its router?
  • At least Ethernet address is on NIC
Bootstrapping

1. Manual configuration (old days)
   • Can’t be factory set, depends on use

2. DHCP: Automatically configure addresses
   • Shifts burden from users to IT folk
DHCP

• DHCP (Dynamic Host Configuration Protocol), from 1993, widely used
• It leases IP address to nodes
• Provides other parameters too
  • Network prefix
  • Address of local router
  • DNS server, time server, etc.
DHCP Protocol Stack

- DHCP is a client-server application
  - Uses UDP ports 67, 68
DHCP Addressing

• Bootstrap issue:
  • How does node send a message to DHCP server before it is configured?

• Answer:
  • Node sends **broadcast** messages that delivered to all nodes on the network
  • **Broadcast address** is all 1s
  • IP (32 bit): 255.255.255.255
  • Ethernet/MAC (48 bit): ff:ff:ff:ff:ff:ff
DHCP Messages

Client

One link

Server
DHCP Messages

Client

Server

DISCOVER

All Broadcast (255.255.255.255)

OFFER

REQUEST

ACK
DHCP Messages

• To renew an existing lease, an abbreviated sequence is used:
  • REQUEST, followed by ACK

• Protocol also supports replicated servers for reliability
Address Resolution Protocol (ARP)
Sending an IP Packet

• Problem:
  • A node needs Link layer addresses to send a frame over the local link
  • How does it get the destination link address from a destination IP address?

Uh oh ...
My IP is 1.2.3.4
ARP (Address Resolution Protocol)

- Node uses to map a local IP address to its Link layer addresses
ARP Protocol Stack

- ARP sits right on top of link layer
  - No servers, just asks node with target IP to identify itself
  - Uses broadcast to reach all nodes
ARP Messages

Node — One link — Target
ARP Messages (2)

Node

REQUEST
Who has IP 1.2.3.4?

Target

Broadcast

REPLY
I do at 1:2:3:4:5:6
ARP Table

# arp -an | grep 10

? (10.241.1.114) at 00:25:90:3e:dc:fc [ether] on vlan241
? (10.252.1.8) at 00:c0:b7:76:ac:19 [ether] on vlan244
? (10.252.1.9) at 00:c0:b7:76:ae:56 [ether] on vlan244
? (10.252.1.6) at 00:c0:b7:74:fb:9a [ether] on vlan244
? (10.241.1.121) at 00:25:90:2c:d4:f7 [ether] on vlan241
[...]
Discovery Protocols

• Help nodes find each other
  • There are more of them!
    • E.g., DLNA, Bonjour

• Often involve broadcast
  • Since nodes aren’t introduced
  • Very handy glue
Fragmentation
Fragmentation

• Problem: How do we connect networks with different maximum packet sizes?
  • Need to split up packets, or discover the largest size to use

Take that

It’s too big!
Packet Size Problem

• Different networks have different max packet sizes
  • Or MTU (Maximum Transmission Unit)
  • E.g., Ethernet 1.5K, WiFi 2.3K

• Prefer large packets for efficiency
  • But what size is too large?
  • Difficult as node doesn’t know complete network path
Packet Size Solutions

• Fragmentation (now)
  • Split up large packets in if they are too big to send
  • Classic method, dated

• MTU Discovery (next)
  • Find the largest packet that fits on the network path
  • IP uses today instead of fragmentation
IPv4 Fragmentation

- Routers fragment packets too large to forward
- Receiving host reassembles to reduce load on routers

Fits on first link
IPv4 Fragmentation Fields

• Header fields used to handle packet size differences
  • Identification, Fragment offset, MF/DF control bits
IPv4 Fragmentation Procedure

• Routers split a packet that is too large:
  • Typically break into large pieces
  • Copy IP header to pieces
  • Adjust length on pieces
  • Set offset to indicate position
  • Set MF (More Fragments) on all pieces except last
    • Recomputes checksum

• Receiving hosts reassembles the pieces:
  • Identification field links pieces together, MF tells receiver when complete
IPv4 Fragmentation

Before
MTU = 2300

ID = 0x12ef
Data Len = 2300
Offset = 0
MF = 0

After
MTU = 1500

ID =
Data Len =
Offset =
MF =

(Ignore length of headers)
IPv4 Fragmentation

**Before**
- MTU = 2300
- ID = 0x12ef
- Data Len = 2300
- Offset = 0
- MF = 0

**After**
- MTU = 1500
- ID = 0x12ef
- Data Len = 1500
- Offset = 0
- MF = 1

- ID = 0x12ef
- Data Len = 800
- Offset = 1500
- MF = 0
IPv4 Fragmentation

• It works!
  • Allows repeated fragmentation

• But fragmentation is undesirable
  • More work for routers, hosts
  • Tends to magnify loss rate
  • Security vulnerabilities too
Path MTU Discovery

• Discover the MTU that will fit
  • So we can avoid fragmentation
  • The method in use today

• Host tests path with large packet
  • Routers provide feedback if too large; they tell host what size would have fit
Path MTU Discovery

Source

Packet (with length)

MTU=1400

MTU=1200 bytes

MTU=900

Destination

MTU=1200 bytes
Path MTU Discovery

MTU=1400

Test #1

MTU=1200 bytes

Try 1200

Test #2

MTU=900

Try 900

Source

Packet (with length)

Destination

Test #3

MTU=1400

Try 1200

MTU=900

MTU=1200
Path MTU Discovery

• Process may seem involved
  • But usually quick to find right size
  • MTUs smaller on edges of network

• Path MTU depends on the path and can change
  • Search is ongoing

• Implemented with ICMP (next)
  • Set DF (Don’t Fragment) bit in IP header to get feedback
Internet Control Message Protocol (ICMP)
Topic

• Problem: What happens when something goes wrong during forwarding?
  • Need to be able to find the problem
Internet Control Message Protocol

- ICMP is a companion protocol to IP
  - They are implemented together
  - Sits on top of IP (IP Protocol=1)

- Provides error report and testing
  - Error is at router while forwarding
  - Also testing that hosts can use
ICMP Errors

• When router encounters an error while forwarding:
  • It sends an ICMP error report back to the IP source
  • It discards the problematic packet; host needs to rectify

Oh, now I see ...
ICMP Message Format

- Each ICMP message has a Type, Code, and Checksum
- Often carry the start of the offending packet as payload
- Each message is carried in an IP packet

```
Src=router, Dst=A
Protocol = 1
Type=X, Code=Y
Src=A, Dst=B
XXXXXXXXXXXXXXXXXX
```

Portion of offending packet, starting with its IP header

<table>
<thead>
<tr>
<th>IP header</th>
<th>ICMP header</th>
<th>ICMP data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Src=router, Dst=A Protocol = 1</td>
<td>Type=X, Code=Y</td>
<td>Src=A, Dst=B XXXXXXXXXXXXXXXXX</td>
</tr>
</tbody>
</table>
## Example ICMP Messages

<table>
<thead>
<tr>
<th>Name</th>
<th>Type / Code</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dest. Unreachable (Net or Host)</td>
<td>3 / 0 or 1</td>
<td>Lack of connectivity</td>
</tr>
<tr>
<td>Dest. Unreachable (Fragment)</td>
<td>3 / 4</td>
<td>Path MTU Discovery</td>
</tr>
<tr>
<td>Time Exceeded (Transit)</td>
<td>11 / 0</td>
<td>Traceroute</td>
</tr>
<tr>
<td>Echo Request or Reply</td>
<td>8 or 0 / 0</td>
<td>Ping</td>
</tr>
</tbody>
</table>

Testing, not a forwarding error: Host sends Echo Request, and destination responds with an Echo Reply.
Traceroute

• IP header contains TTL (Time to live) field
  • Decremented every router hop, with ICMP error at zero
  • Protects against forwarding loops
Traceroute (2)

- Traceroute repurposes TTL and ICMP functionality
- Sends probe packets increasing TTL starting from 1
- ICMP errors identify routers on the path