CS4450

Computer Networks:
Architecture and Protocols

Lecture 19
Putting ALL the Pieces Together

Rachit Agarwal
Discovery protocols: ARP and DHCP

Suppose Host A wants to communicate with Host B
Discovery

• Suppose I am host A

• I want to communicate with B (say, www.google.com)

• I was “born” knowing only my name — my MAC address :-)

• Must discover some information before I can communicate with B
  • What is my IP address?
  • What is B’s IP address?
    • Using DNS
  • Is B within my LAN?
  • If yes, what is B’s MAC address?
  • If not, what is the address of my first-hop router to B?
  • ...
DHCP and ARP

- Link layer discovery protocols
  - DHCP — Dynamic Host Configuration Protocol
  - ARP — Address Resolution Protocol
- Configured to a single LAN
- Rely on broadcast capability
DHCP and ARP

• Link layer discovery protocols
• Serve two functions
  1. Discovery of local end-hosts
     • For communication between hosts on the same LAN
  2. Bootstrap communication with remote hosts
     • What’s my IP address?
     • Who/where is my local DNS server?
     • Who/where is my first hop router?
DHCP

• Dynamic Host Configuration Protocol
  • Defined in RFC 2131

• A host uses DHCP to discover
  • Its own IP address
  • Subnet masks — allows to test whether an IP address is local or not
  • IP address(es) for its local DNS name server(s)
  • IP address(es) for its first-hop “default” router(s)
DHCP: operation

1. One or more local DHCP servers maintain required information
   - IP address pool, netmask, DNS servers, etc.
   - Application that listens on UDP port 67
DHCP: operation

1. One or more local DHCP servers maintain required information

2. Client broadcasts a DHCP discovery message
   - L2 broadcast, to MAC address FF:FF:FF:FF:FF:FF
DHCP: operation

1. One or more local DHCP servers maintain required information

2. Client broadcasts a DHCP discovery message

3. One or more DHCP servers respond with a DHCP “offer” message
   • Proposed IP address for client, lease time
   • Other parameters
DHCP: operation

1. One or more local DHCP servers maintain required information

2. Client broadcasts a DHCP discovery message

3. One or more DHCP servers respond with a DHCP “offer” message

4. Client broadcasts a DHCP request message
   • Specifies which offer it wants
   • Echoes accepted parameters
   • Other DHCP servers learn they were not chosen
DHCP: operation

1. One or more local DHCP servers maintain required information

2. Client broadcasts a DHCP discovery message

3. One or more DHCP servers respond with a DHCP “offer” message

4. Client broadcasts a DHCP request message

5. Selected DHCP server responds with an ACK
Are we there yet?

What I learnt from DHCP
My IP: 1.2.3.48
Netmask: 1.2.3.0/24
Local DNS: 1.2.3.156
Router: 1.2.3.9
ARP: Address Resolution Protocol

• Every host maintains an ARP table
  • List of (IP address — MAC address) pairs
  • For IP addresses within the same LAN

• Consult the table when sending a packet
  • Map destination IP address to destination MAC address

• But: what if IP address not in the table?
  • Either its not local (detected using DHCP)
  • If its local:
    • Sender broadcasts: “Who has IP address 1.2.3.156?”
    • Caches the answer in ARP table
Key Ideas in Both ARP and DHCP

• Broadcasting: can use broadcast to make contact
  • Scalable because of limited size

• Caching: remember the past for a while
  • Store the information you learn to reduce overhead
# Taking Stock: Discovery

<table>
<thead>
<tr>
<th>Layer</th>
<th>Examples</th>
<th>Structure</th>
<th>Configuration</th>
<th>Resolution Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>App Layer</td>
<td><a href="http://www.cs.cornell.edu">www.cs.cornell.edu</a></td>
<td>Organizational hierarchy</td>
<td>~ manual</td>
<td>DNS</td>
</tr>
<tr>
<td>Network Layer</td>
<td>123.45.6.78</td>
<td>Topological hierarchy</td>
<td>DHCP</td>
<td>ARP</td>
</tr>
<tr>
<td>Link Layer</td>
<td>45-CC-4E-12-F0-97</td>
<td>Vendor(flat)</td>
<td>Hard-coded</td>
<td></td>
</tr>
</tbody>
</table>
Where are we?
What protocols have we learnt on LAN?

- **Link Layer MAC names/addresses**: come with the hardware
- **CSMA/CD Protocol**: To transmit frames on broadcast Ethernet
- **Spanning Tree Protocol**: To transmit frames on switched Ethernet
- **Domain Name System**: To map application-layer names to IP addresses
- **DHCP**: To learn
  - Its own IP address
  - Whether an IP address is local or not
  - IP address(es) for its local DNS name server(s)
  - IP address(es) for its first-hop “default” router(s)
- **ARP**: To map IP addresses to MAC names (for IP addresses on the same LAN)
What have we learnt beyond LAN?

- **Link-state and Distance-vector Protocols:**
  - For finding routes (and a next-hop) to an IP address within an ISP

- **Border Gateway Protocol:**
  - For finding routes to an IP address range

- **Forwarding at routers**
  - Store **routing tables** *(map destination prefixes to outgoing port)*
  - Longest prefix match for destination address lookup
How does the Internet work?

Let's first see what it actually looks like!
What is a computer network?

A set of network elements connected together, that implement a set of protocols for the purpose of sharing resources at the end hosts.
What does Internet actually look like?

• The smallest component:
  • A Network Interface Card (NIC), or a machine, or a server
  • Has a Link Layer MAC name/address

• Multiple NICs connected in a Local Area Network (LAN) via
  • Broadcast Ethernet,
  • Or, Switched Ethernet

• Switches in LAN
  • Connected to larger routers
What does Internet actually look like?
What does Internet actually look like?

“Autonomous System (AS)” or “Domain”
Region of a network under a single administrative entity
What does Internet actually look like?

Multiple “Autonomous Systems (AS)” or “Domains” connect together using Border Routers
This entire infrastructure is a part of the INTERNET :-)

[Diagram of interconnected networks]
How does the Internet work?

Are you ready?

(Count the number of protocols used for each packet)
How does Internet work — end-to-end?

• Network stack receives the packet from the application (roughly speaking)
• What is my IP address? (using DHCP)
• What is the destination IP address? (using DNS)
• Is destination IP address within my LAN? (using DHCP)

• If destination IP address local:
  • What is destination MAC address (using ARP)?
  • Convert packet into frames with correct source/destination address
  • Convert frames into bits
  • Forward the bits to the wire ...

• Each switch:
  • Forwards to destination (using STP/CSMA/CD)
End-to-End I

Source IP

Destination within my LAN?

Destination IP

Destination MAC

DHCP

DNS

ARP

T N L

S

T N L

D

T N L

L L

T N L

L L

T N L
How does Internet work — end-to-end?

• Network stack receives the packet from the application (roughly speaking)

• What is my IP address? (using DHCP)

• What is the destination IP address? (using DNS)

• Is destination IP address within my LAN? (using DHCP)

• If destination IP address remote:
  • What is my next-hop router IP address? (using DHCP)
  • What is my next-hop router MAC address? (using ARP)
  • Convert packet into frames with correct source/destination address
  • Convert frames into bits
  • Forward the bits to the wire ...

• Each router ....
End-to-End II

- Source IP
- Destination within my LAN?
- Destination IP
- Destination MAC
- First Hop Router MAC
- First Hop Router IP
- DHCP
- DNS
- ARP

S

T

N

L

First Hop Router IP

LL IP

LL IP

LL IP

LL IP

T

N

L

IP

IP

IP

IP

IP

IP

IP

IP

IP

IP

First Hop Router MAC

DNS

ARP

DHCP

Destination within my LAN?
How does Internet work — end-to-end?

A router upon receiving a packet (implicit questions)

• **Is the destination in a LAN connected to me?**
  • Forward the packet to the destination
  • Using STP/CSMA/CD

• **Is the destination not in my LAN but in my ISP?**
  • Forward the packet to the next-hop router towards the destination
  • Using next-hops computed via distance-vector routing algorithm

• **Is the destination in a different ISP?**
  • Forward the packet to the next-hop router towards the destination
  • Using next-hops computed via BGP routing algorithm
Are We There Yet?

• Yes!
• How can we be sure?
• Let's go back to where we started ....
Recall the end-to-end story from our fifth lecture :-) 

- Application opens a **socket** that allows it to connect to the **network stack**
- Maps **name** of the web site to its **address** using **DNS**
- The network stack at the source embeds the address and **port** for both the source and the destination in **packet header**
- Each **router** constructs a **routing table** using a distributed algorithm
- Each router uses destination address in the packet header to look up the **outgoing link** in the routing table
  - And when the link is free, forwards the packet
- When a packet arrives the destination:
  - The network stack at the destination uses the port to forward the packet to the right application
You now know how the Internet works!!!!

All that is remaining:

Reliability.