CS 43: Computer Networks

29: End-to-end Example

Dec 7, 2018
Putting it all together…

• What happens when a user shows up to a new network and wants to access a web site?
Scenario

Network: 1.0.0.0/24
24 bits: network
8 bits: host

Network: 5.0.0.0/16
16 bits: network
16 bits: host
Network: 1.0.0.0/24
24 bits: network
8 bits: host:
- all host addresses are 1.0.0.*
- 256 possible addresses

Network: 5.0.0.0/16
16 bits: network
16 bits: host:
- all host addresses are 5.0.*
- 65,536 possible addresses!
Before anyone starts sending data, we’ll assume the routers have run a routing protocol (BGP) to learn about each other.
Before anyone starts sending data, we’ll assume the routers have run a routing protocol (BGP) to learn about each other.
Step 1: User Joins the Network

User arrives and needs an IP address. They bring MAC address with them (built in to hardware).
Step 1: User Joins the Network

User broadcasts DHCP DISCOVER message to acquire IP address. (Alternative, they manually enter IP config details.)
Step 1: User Joins the Network

DHCP responds with: IP address (1.0.0.15), subnet mask (255.255.255.0), gateway (1.0.0.1), and DNS server (1.0.0.2).
Step 1: User Joins the Network

DHCP responds with: IP address (1.0.0.15), subnet mask (255.255.255.0), gateway (1.0.0.1), and DNS server (1.0.0.2).
Step 2: User Resolves Name

Suppose user tries to access website: www.xkcd.com
Must resolve name using DNS. Query local resolver.
Step 2: User Resolves Name

User’s PC must answer: is the DNS resolver (1.0.0.2) I was given by DHCP on my subnet? (Local vs. Internet)

Match! It’s local. Send directly, no need to go through Internet gateway (router).
Step 2: User Resolves Name

User’s PC does NOT know DNS server’s MAC address! Broadcast ARP request looking for 1.0.0.2!

<table>
<thead>
<tr>
<th>IP Address</th>
<th>MAC Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0.0.15</td>
<td>00:01:02:03:04:05</td>
</tr>
<tr>
<td>1.0.0.2</td>
<td>?</td>
</tr>
</tbody>
</table>
Step 2: User Resolves Name

DNS server responds with MAC address.

<table>
<thead>
<tr>
<th>IP Address</th>
<th>MAC Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0.0.15</td>
<td></td>
</tr>
<tr>
<td>1.0.0.2</td>
<td>00:AA:BB:CC:DD:FF</td>
</tr>
</tbody>
</table>
Step 2: User Resolves Name

User queries local DNS resolver for www.xkcd.com. Resolver runs necessary queries (root, TLD, etc.)
Step 2: User Resolves Name

DNS reply says that www.xkcd.com is 5.0.9.25.
Step 3: Establish a TCP Connection

User’s PC must answer: is the destination (5.0.9.25) on my subnet? (Local vs. Internet)

1.0.0.2

1.0.0.15: 00000001 00000000 00000000 00001111
255.255.255.0: 11111111 11111111 11111111 00000000

ANDed tog. ether: my network prefix

my address 00000001 00000000 00000000
subnet mask 11111111 11111111 11111111 00000000

target address 00000001 00000000 00000000
5.0.9.25 00000101 00000000 00001001 00011101

No Match! Send it to the default gateway (router that connects to the Internet) that DHCP gave us (1.0.0.1).
Step 3: Establish a TCP Connection

User’s PC does NOT know router’s MAC address!
Broadcast ARP request looking for 1.0.0.1!

<table>
<thead>
<tr>
<th>IP Address</th>
<th>MAC Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0.0.2</td>
<td>00:AA:BB:CC:DD:FF</td>
</tr>
<tr>
<td>1.0.0.1</td>
<td>?</td>
</tr>
</tbody>
</table>
Step 3: Establish a TCP Connection

Router responds with MAC address.

<table>
<thead>
<tr>
<th>IP Address</th>
<th>MAC Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0.0.1</td>
<td>00:00:11:11:22:22</td>
</tr>
<tr>
<td>1.0.0.2</td>
<td>00:AA:BB:CC:DD:FF</td>
</tr>
</tbody>
</table>

www.xkcd.com
Step 3: Establish a TCP Connection

Send TCP SYN to the destination, start 3-way handshake.

<table>
<thead>
<tr>
<th>IP Address</th>
<th>MAC Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0.0.2</td>
<td>00:AA:BB:CC:DD:FF</td>
</tr>
<tr>
<td>1.0.0.1</td>
<td>00:00:11:11:22:22</td>
</tr>
</tbody>
</table>

Eth Header
Dest MAC: 00:00:11:11:22:22
IP Header
Dest IP: 5.0.9.25
TCP Header
Dest port: 80
SYN

No data in SYN

Network layer header: final destination!

Link layer header: next hop destination!
Step 3: Establish a TCP Connection

Send SYN to router. NOTE: while the switch moves the frame to router, it is not ever addressed directly.

DHCP  DNS

Eth Header
Dest MAC: 00:00:11:11:22:22

IP Header
Dest IP: 5.0.9.25

TCP Header
Dest port: 80, SYN

Internet

1.0.0.2

1.0.0.1

5.0.0.1

www.xkcd.com

1.0.0.15
00:01:02:03:04:05
Step 3: Establish a TCP Connection

Router removes Ethernet header.

DHCP
DNS

Internet

1.0.0.1
5.0.0.1

1.0.0.2

Eth Header
Dest MAC:
00:00:11:11:22:22

IP Header
Dest IP:
5.0.9.25

TCP Header
Dest port: 80
SYN

www.xkcd.com

1.0.0.15
00:01:02:03:04:05
Step 3: Establish a TCP Connection

Router removes Ethernet header.

DHCP  DNS

1.0.0.2

1.0.0.1

IP Header
Dest IP: 5.0.9.25

TCP Header
Dest port: 80
SYN

5.0.0.1

www.xkcd.com

00:01:02:03:04:05
Step 3: Establish a TCP Connection

Router R1 compares destination IP with its forwarding table, looks for longest prefix match.

Best match: 5.0.0.0/16 -> Output port C
Step 3: Establish a TCP Connection

Router R1 constructs frame and forwards it to R2.

```
TCP Header
Dest port: 80

IP Header
Dest IP: 5.0.9.25

Link Layer Header
Dest MAC: 55:44:33:22:11:00
```

```
DNS

DHCP
```

```
www.xkcd.com
```

```
1.0.0.1
00:01:02:03:04:05
```

```
1.0.0.2
```

```
1.0.0.15
```

```
1.0.0.2
```

```
Internet
```

```
1
```

```
www.xkcd.com
```
Step 3: Establish a TCP Connection

Router R2 compares destination IP with its forwarding table, looks for longest prefix match.

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Output Port</th>
<th>Next Router’s Link Layer Addr</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0.0.0/24</td>
<td>A</td>
<td>R₁’s Address</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>5.0.0.0/8</td>
<td>B</td>
<td>Some Internet router’s address</td>
</tr>
<tr>
<td>5.0.0.0/16</td>
<td>C</td>
<td>(N/A - no router there)</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Best match: 5.0.0.0/16 -> Output port C
Destination MAC: ?
Step 3: Establish a TCP Connection

R2 does NOT know destination’s MAC address!
Broadcast ARP request looking for 5.0.9.25!
Data packet is queued while waiting for ARP to resolve.
Step 3: Establish a TCP Connection

Host replies with MAC address.

<table>
<thead>
<tr>
<th>IP Address</th>
<th>MAC Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.0.0.25</td>
<td>AA:BB:CC:DD:EE:FF</td>
</tr>
<tr>
<td>00:01:02:03:04:05</td>
<td><a href="http://www.xkcd.com">www.xkcd.com</a></td>
</tr>
</tbody>
</table>
Step 3: Establish a TCP Connection

R2 constructs frame, forwards it to destination.

Ethernet Header

IP Header
Dest IP: 5.0.9.25

TCP Header
Dest port: 80
SYN

www.xkcd.com
Mission Accomplished!

Destination peels off headers, generates reply (SYN+ACK).
Mission Accomplished!

Process repeats in the opposite direction, without the ARPs this time. (MAC addresses were recently used, thus cached.)
Steady State

• With DNS cached and ARP entries cached, host encapsulates data in TCP, IP, Eth headers and sends to router. Router forwards.

• Even with all the DNS/ARP, all that stuff happens in < 1 second. (besides step 0: routing protocol)