CS2105 Lecture 12
Putting Everything Together

14 April, 2014
Turn on modem
Turn on router
Turn on laptop
Visit IVLE
Turn on Modem
Modulate between Digital (Manchester Coding) and Analog (QAM/QPSK)
Turn on Router
Router broadcasts DHCP Discover
04:42:00:31:98:12

ff:ff:ff:ff:ff:ff:ff:ff

IP
Manchester
DOCSIS

←

←

←

→
64-QAM
Router receives DHCP Offer
Router broadcasts ARP Request
04:42:00:31:98:12    ARP
ff:ff:ff:ff:ff:ff:ff:ff
Router receives ARP Respond
Router sends DHCP Request
<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>16</th>
<th>32</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>68</td>
<td>67</td>
<td></td>
</tr>
</tbody>
</table>
255  UDP

0.0.0.0

218.186.2.1
Router receives DHCP ACK
Turn on Laptop and connects to WiFi
Turn on Laptop and connects to WiFi

There are packet exchanges when the laptop associates itself and authenticate itself with the WiFi AP, which we skip here.
Laptop runs DHCP client
Router runs DHCP server
Configure IPv4: Using DHCP
IPv4 Address: 10.0.1.7
Subnet Mask: 255.255.255.0
Router: 10.0.1.1

Configure IPv6: Automatically
IPv6 Address:
Prefix Length:
Visit https://ivle.nus.edu.sg/
DNS lookup for ivle.nus.edu.sg
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connect Using</td>
<td>DHCP</td>
</tr>
<tr>
<td>IPv4 Address</td>
<td>218.186.176.42</td>
</tr>
<tr>
<td>Subnet Mask</td>
<td>255.255.252.0</td>
</tr>
<tr>
<td>Router Address</td>
<td>218.186.176.1</td>
</tr>
<tr>
<td>DNS Servers</td>
<td>218.186.2.16</td>
</tr>
<tr>
<td></td>
<td>218.186.2.6</td>
</tr>
<tr>
<td>IPv6 DNS Servers</td>
<td></td>
</tr>
<tr>
<td>Domain Name</td>
<td></td>
</tr>
<tr>
<td>IPv6 Address</td>
<td></td>
</tr>
</tbody>
</table>
laptop:$ ooiwt$ arp -an
? (10.0.1.1) at 70:56:81:c7:11:e2 on en0
? (10.0.1.8) at d0:23:db:47:64:6f on en0
? (10.0.1.255) at ff:ff:ff:ff:ff:ff on en0
? (10.0.1.255) at ff:ff:ff:ff:ff:ff on en1
laptop

Internet

router  modem
traceroute to 218.186.2.6 (218.186.2.6)
1  10.0.1.1  1.354 ms  0.826 ms  0.913 ms
2  218.186.176.1  14.015 ms  7.879 ms  12.394 ms
3  172.20.12.1  11.866 ms  11.813 ms  7.667 ms
4  172.20.7.82  10.794 ms  15.408 ms  11.775 ms
5  172.20.7.38  9.464 ms  8.032 ms  11.391 ms
6  172.20.8.5  12.319 ms  13.101 ms  55.906 ms
7  218.186.2.6  12.111 ms  11.196 ms  9.436 ms
Laptop receives DNS response
QUESTION SECTION:
;ivle.nus.edu.sg. IN A

ANSWER SECTION:
ivle.nus.edu.sg. 294 IN A 137.132.39.20
Laptop initiates HTTPS connection (sends TCP SYN to port 443)
TCP

10.0.1.7

137.132.39.20
Router creates NAT table entry
<table>
<thead>
<tr>
<th>10000</th>
<th>443</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
TCP

218.186.176.42

137.132.39.20
GET / HTTP/1.1
Host: ivle.nus.edu.sg
User-Agent: Mozilla/5.0
Keep-Alive: 300
Connection: keep-alive
Cookie: __utma=145952555.278120873
Transfer data $m$ with sequence number $s$ as $m'$

$$h = H(m + M_B + s)$$
$$m' = E_B(m, h)$$
Receives $m'$ with expected sequence number $s'$

$$m, h = E_A(m')$$

check if $h = H(m + M_A + s')$?
General Lessons from CS2105
How to build a complex system
Many issues to consider, to support different applications running on large number of hosts through different access technology and physical media.
1. Layering
Lower layers hide details from upper layer

Upper layers make minimum assumptions about lower layer

Protocols in layers are interchangeable

Keep lower layer simple, more features at higher layer.
2. Decouple name from address

hostname, cname, IP addr, MAC addr
- Rotate hosts for load balancing
- Change hosts based on geo-location, network conditions
- Replace or move hosts without affecting names
3. Hierarchical Naming/Routing

www.nus.edu.sg
137.132/16
► More organized
► Allow distribution of responsibility
► Allow aggregation of addresses for forwarding
► Other examples: files, zip code
4. Periodically Forget

DNS caches, switching table,
DHCP leases, RIP
- Systems adapt quickly
- Prevent stale information automatically
5. Randomization

nonce, TCP sequence numbers, BEB
► Avoid expensive coordination

► Avoid "coincidence" (collision, TCP sequence numbers)

► Avoid malicious party from guessing
6. Learn by observation

NAT, Switch, BEB
Avoid expensive manual configuration

Adapt to changes automatically
What's Next?
CS2105R 1MC extension
CS3103 Computer Networks and Protocols
CS4222 Wireless and Sensor Networks
CS4274 Mobile and Multimedia Networking
CS4344 Networked and Mobile Gaming
CS5229 Advanced Computer Networks
CS5248 Systems Support for Continuous Media
Acknowledgement
The End