TCP/IP Networking
An Example

Introductory material.
This module illustrates the interactions of the protocols of the TCP/IP protocol suite with the help of an example. The example intends to motivate the study of the TCP/IP protocols.

A simple TCP/IP Example

• A user on host argon.tcpip-lab.edu (“Argon”) makes a web access to URL


• What actually happens in the network?
HTTP Request and HTTP response

- Web browser runs an HTTP client program
- Web server runs an HTTP server program
- HTTP client sends an HTTP request to HTTP server
- HTTP server responds with HTTP response

![Diagram showing HTTP request and response](image)

HTTP Request

```
GET /example.html HTTP/1.1
Accept: image/gif, */*
Accept-Language: en-us
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0
Host: 192.168.123.144
Connection: Keep-Alive
```
HTTP Response

HTTP/1.1 200 OK
Date: Sat, 25 May 2002 21:10:32 GMT
Server: Apache/1.3.19 (Unix)
Last-Modified: Sat, 25 May 2002 20:51:33 GMT
ETag: "56497-51-3ceff955"
Accept-Ranges: bytes
Content-Length: 81
Keep-Alive: timeout=15, max=100
Connection: Keep-Alive
Content-Type: text/html

<html>
<body>
<h1>Internet Lab</h1>
Click <a href="http://www.tcpip-lab.net/index.html">here</a> for the Internet Lab webpage.
</body>
</html>

- How does the HTTP request get from Argon to Neon?

From HTTP to TCP

- To send request, HTTP client program establishes a TCP connection to the HTTP server Neon.
- The HTTP server at Neon has a TCP server running

![HTTP to TCP diagram]
Resolving hostnames and port numbers

• Since TCP does not work with hostnames and also would not know how to find the HTTP server program at Neon, two things must happen:

  1. The name “neon.tcpip-lab.edu” must be translated into a 32-bit IP address.
  2. The HTTP server at Neon must be identified by a 16-bit port number.

Translating a hostname into an IP address

• The translation of the hostname neon.tcpip-lab.edu into an IP address is done via a database lookup

• The distributed database used is called the Domain Name System (DNS)
• All machines on the Internet have an IP address:

  - argon.tcpip-lab.edu 128.143.137.144
  - neon.tcpip-lab.edu 128.143.71.21
Finding the port number

- **Note:** Most services on the Internet are reachable via well-known ports. E.g. All HTTP servers on the Internet can be reached at port number “80”.
- **So:** Argon simply knows the port number of the HTTP server at a remote machine.

- On most Unix systems, the well-known ports are listed in a file with name `/etc/services`. The well-known port numbers of some of the most popular services are:
  - ftp 21          finger 79
  - telnet 23       http 80
  - smtp 25         nntp 119

---

Requesting a TCP Connection

- The HTTP client at `argon.tcpip-lab.edu` requests the TCP client to establish a connection to port 80 of the machine with address `128.141.71.21`
Invoking the IP Protocol

- The TCP client at Argon sends a request to establish a connection to port 80 at Neon.

- This is done by asking its local IP module to send an IP datagram to 128.143.71.21.

- (The data portion of the IP datagram contains the request to open a connection.

Sending the IP datagram to an IP router

- Argon (128.143.137.144) can deliver the IP datagram directly to Neon (128.143.71.21), only if it is on the same local network ("subnet").

- But Argon and Neon are not on the same local network (Q: How does Argon know this?)

- So, Argon sends the IP datagram to its default gateway.

- The default gateway is an IP router.

- The default gateway for Argon is Router137.tcpip-lab.edu (128.143.137.1).
The route from *Argon* to *Neon*

- Note that the gateway has a different name for each of its interfaces.

Finding the MAC address of the gateway

- To send an IP datagram to Router137, *Argon* puts the IP datagram in an Ethernet frame, and transmits the frame.
- However, Ethernet uses different addresses, so-called Media Access Control (MAC) addresses (also called: physical address, hardware address)
- Therefore, *Argon* must first translate the IP address 128.143.137.1 into a MAC address.
- The translation of addressed is performed via the Address Resolution Protocol (ARP)
Address resolution with ARP

• IP address 128.143.137.1 belongs to MAC address 00:e0:f9:23:a8:20

Invoking the device driver

• The IP module at Argon, tells its Ethernet device driver to send an Ethernet frame to address 00:e0:f9:23:a8:20
Sending an Ethernet frame

- The Ethernet device driver of Argon sends the Ethernet frame to the Ethernet network interface card (NIC)
- The NIC sends the frame onto the wire

Forwarding the IP datagram

- The IP router receives the Ethernet frame at interface 128.143.137.1, recovers the IP datagram and determines that the IP datagram should be forwarded to the interface with name 128.143.71.1
- The IP router determines that it can deliver the IP datagram directly
Another lookup of a MAC address

- The router needs to find the MAC address of Neon.
- Again, ARP is invoked, to translate the IP address of Neon (128.143.71.21) into the MAC address of neon (00:20:af:03:98:28).

Invoking the device driver at the router

- The IP protocol at Router71, tells its Ethernet device driver to send an Ethernet frame to address 00:20:af:03:98:28.
Sending another Ethernet frame

- The Ethernet device driver of Router71 sends the Ethernet frame to the Ethernet NIC, which transmits the frame onto the wire.

Data has arrived at Neon

- *Neon* receives the Ethernet frame
- The payload of the Ethernet frame is an IP datagram which is passed to the IP protocol.
- The payload of the IP datagram is a TCP segment, which is passed to the TCP server

- **Note**: Since the TCP segment is a connection request (SYN), the TCP protocol does not pass data to the HTTP program for this packet. Instead, the TCP protocol at neon will respond with a SYN segment to *Argon.*
Wrapping-up the example

- So far, Neon has only obtained a single packet
- Much more work is required to establish an actual TCP connection and the transfer of the HTTP Request

The example was simplified in several ways:
- No transmission errors
- The route between Argon and Neon is short (only one IP router)
- Argon knew how to contact the DNS server (without routing or address resolution)
- ….

How many packets were really sent?

tcpdump: listening on fxp0
16:54:51.341749 128.143.137.11.53 > 128.143.137.144.1555: 1 NXDomain* 0/1/0 (98) (DF)
16:54:51.342539 128.143.137.144.1556 > 128.143.137.11.53: 2+ (41)
16:54:51.343436 128.143.137.11.53 > 128.143.137.144.1556: 2 NXDomain* 0/1/0 (109) (DF)
16:54:51.344147 128.143.137.144.1557 > 128.143.137.11.53: 3+ (38)
16:54:51.345220 128.143.137.11.53 > 128.143.137.144.1557: 3+ 1/1/2 (122) (DF)

16:54:51.350996 arp who-has 128.143.137.1 tell 128.143.137.144
16:54:51.351614 arp reply 128.143.137.1 is-at 0:e0:f9:23:a8:20

16:54:51.351712 128.143.137.144.1558 > 128.143.71.21.21: S 607568:607568(0) win 8192
<mss 1460> (DF)
16:54:51.352895 128.143.71.21.80 > 128.143.137.144.1558: S 3964010655:3964010655(0)
ack 607569 win 17520 <mss 1460> (DF)
16:54:51.353007 128.143.137.144.1558 > 128.143.71.21.80: . ack 1 win 8760 (DF)
16:54:51.365603 128.143.71.21.80 > 128.143.137.144.1558: P 1:60(59)
ack 1 win 17520 (DF) [tos 0x10]
16:54:51.507399 128.143.137.144.1558 > 128.143.71.21.80: . ack 60 win 8701 (DF)