



First we define the different components of the 2 node network.

C1=Client Computer; C1IP= IP address for C1; C1M= Mac Address for C1

C2=Server Computer; C2IP=IP address for C2; C2M= MAC address for C2

N1= Local Network for C1; N1IP= IP address table for N1

N2= Local Network for C2; N2IP= IP address table for N2

R1= Router; R1M= MAC address for Router; R1N1IP= IP for R1>N1 connection; R1N2IP= IP for R1>N2 connection

Next we define preconditions:

C2 is an Apache server set to LISTEN on Port 80

R1 contains C1N1IP and C2N2IP in its routing table

There is a physical connection between C1>N1; N1>R1; R1>N2; N2>C2

The initial TCP connection will be requested from the client C1 to the server C2 using a web browser

Next we describe the steps that take place while establishing a TCP connection between C1 and C2

1. We type the URL or IP address for server C2 into the web browser residing on the Application Layer; "C2IP" (xx.xx.xx.xx:80)

1.1. The web browser communicates with the OS that it would like to make a connection with C2IP:80

1.1.1. The OS recognizes that the browser is trying to establish a TCP connection and reserves port 50000 on C1 for outbound communication

1.2. C1 searches N1 for the mac address of C2IP

1.2.1. Result of search is negative

1.2.2. Since C2IP is not on N1, C1 must use the N1 gateway to connect to R1 since the C2IP mac address is not located on the local network

1.3. C1 send an ARP request for R1N1IP through the N1 switch which forwards the request to R1

1.3.1. R1 responds to the ARP request by sending R1M for R1N1IP back to C1

1.4. C1 now has and is ready to build the TCP/IP datagrams and ethernet frame

1.4.1. C1 constructs TCP datagram- Transport Layer

1.4.1.1. Source Port:50000

- 1.4.1.2. Destination Port:80
- 1.4.1.3. SYN FLAG for establishing connection
- 1.4.1.4. Sets sequence # for SYN/ACK handshake
- 1.4.1.5. Checksum Calculation
- 1.4.2. C1 constructs IP Datagram- Network Layer
  - 1.4.2.1. Source IP: C1IP
  - 1.4.2.2. Destination IP: C2IP
  - 1.4.2.3. Sets TTL(64)
  - 1.4.2.4. Datapayload: TCP datagram
  - 1.4.2.5. Checksum Calculation
- 1.4.3. C1 constructs Ethernet Frame to send the IP datagram to R1- DataLink Layer
  - 1.4.3.1. Source MAC: C1M
  - 1.4.3.2. Destination MAC: R1M
  - 1.4.3.3. Datapayload: IP datagram
  - 1.4.3.4. Checksum Calculation
- 1.5. C1 sends Ethernet frame to N1 using the physical layer of wires through electronic modulations which N1 uses to forward it to the R1N1IP associated with the R1M mac address
- 1.6. R1 receives the Ethernet frame from the N1 switch on its R1N1IP connection
  - 1.6.1. R1 calculates checksum and checks to verify validity of data
    - 1.6.1.1. R1 analyzes the Ethernet frame and confirms that the destination MAC R1M matches its own MAC address R1M
    - 1.6.1.2. R1 unwraps the Ethernet frame and examines the IP datagram
      - 1.6.1.2.1. R1 calculates checksum and checks to verify validity of data
      - 1.6.1.2.2. R1 checks destination IP address: C2IP
        - 1.6.1.2.2.1. R1 looks up C2IP in its routing table located in the N2IP network
      - 1.6.1.2.3. R1 decreases the TTL from 64 to 63
      - 1.6.1.2.4. R1 calculates new checksum for TCP datagram
    - 1.6.1.3. R1 creates a new Ethernet frame with the TCP datagram in the data payload
    - 1.6.1.4. Source MAC; R1M
    - 1.6.1.5. Destination MAC: C2M
  - 1.6.2. Calculates new check sum
  - 1.6.3. Forwards new ethernet frame to N2
- 1.7. N2 switch receives the Ethernet frame from R1 and recognize C2M on its network and forwards the frame to C2
- 1.8. C2 receives Ethernet frame
  - 1.8.1. C2 calculates checksum of ethernet frame to verify data
    - 1.8.1.1. C2 verifies destination MAC as C2M and confirms match
    - 1.8.1.2. C2 unwraps Ethernet frame
  - 1.8.2. C2 calculates checksum of IP datagram to verify data
    - 1.8.2.1. C2 confirms IP Destination IP C2IP matches its own
    - 1.8.2.2. C2 Unwraps IP datagram
  - 1.8.3. C2 calculates checksum of TCP datagram to verify data
    - 1.8.3.1. Checks destination port in TCP datagram :80
      - 1.8.3.1.1. verifies port 80 is open
      - 1.8.3.1.2. Recieves the SYNFLAG for SYN/ACK handshake
      - 1.8.3.1.3. examines and stores sequence number to receive more data
      - 1.8.3.1.4. packages ACKFLAGS into the TCP datagram
    - 1.8.3.2. Switches the source and destination ports in the TCP datagram
  - 1.8.4. Calculates Checksum for TCP datagram

- 1.8.5. Packages TCP datagram in a new IP Datagram switch the Source and destination Ips
  - 1.8.5.1. Source: C2IP
  - 1.8.5.2. Destination: C1IP
  - 1.8.5.3. Sets TTL(64)
- 1.8.6. Calculates checksum for IP datagram
- 1.8.7. Packages IP datagram in new Ethernet Frame
  - 1.8.7.1. Source MAC: C2M
  - 1.8.7.2. Destination MAC: R1M
- 1.9. Sends Ethernet Frame with TCP ACK back to C1 through router R1 until the entire sequence of data has been successfully delivered.