

Wireshark + Layers

Open the sample trace at https://teaching.pschmitt.net/EE449_Spring2023/exercises/http.pcap with Wireshark.

- Is this browser running HTTP version 1.0, 1.1, or 2? What version of HTTP is the server running?
- What languages (if any) does the browser indicate that it can accept to the server?
- What is the IP address of the client computer? What is the IP address of the gaia.cs.umass.edu server?
- What is the status code returned from the server to your browser?
- When was the HTML file that you are retrieving last modified at the server?
- How many bytes of content are being returned to the client browser?
- By inspecting the raw data in the packet content window, do you see any headers within the data that are not displayed in the packet-listing window? If so, name one.

Wireshark + Layers

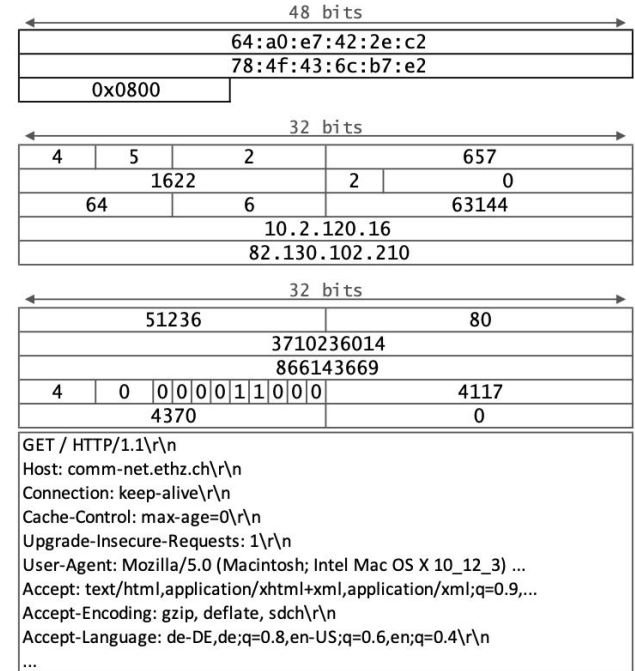
Open the sample trace at https://teaching.pschmitt.net/EE449_Spring2023/exercises/http.pcap with Wireshark.

- Is this browser running HTTP version 1.0, 1.1, or 2? What version of HTTP is the server running?
Solution: Both the client and server are running HTTP/1.1
- What languages (if any) does the browser indicate that it can accept to the server?
Solution: English (en-US) is sent in the "Accept Language" HTTP header
- What is the IP address of the client computer? What is the IP address of the gaia.cs.umass.edu server?
Solution: Client is 192.168.86.81, Server is 128.119.245.12
- What is the status code returned from the server to your browser?
Solution: 200 OK for the main body, 404 Not Found for favicon.ico
- When was the HTML file that you are retrieving last modified at the server?
Solution: Last-Modified: Sun, 22 Jan 2023 06:59:01 GMT
- How many bytes of content are being returned to the client browser?
Solution: 128 bytes
- By inspecting the raw data in the packet content window, do you see any headers within the data that are not displayed in the packet-listing window? If so, name one.
Solution: <html>

Layers are Distributed on Every Network Device

The figure on the right shows a (simplified version of a) packet that was recorded on a machine's network interface using Wireshark.

- What kind of activity does this packet belong to?
- You know that a packet is composed of data in different layers. The fields in the right-hand figure are already grouped to four blocks representing the link, network, transport and application layer. Do you know which protocol is used in each of the four layers?
- After being sent out at your machine's network interface, the packet will traverse a switch which will forward it based on the destination address. On which layer does the switch operate and which is the destination address it will look at?
- After leaving your local network, the packet will traverse a router. On which layer does the router operate and which is the destination address it will look at?



Packet consisting of data in four layers (link, network, transport and application layer).

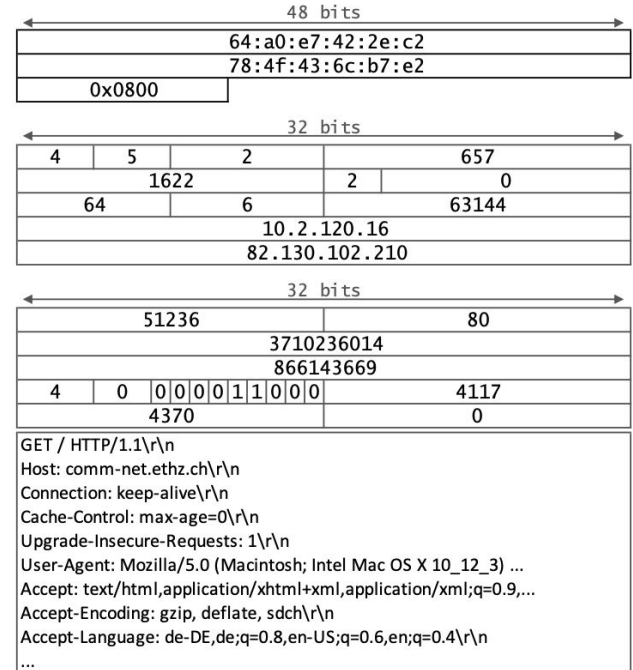
Layers are Distributed on Every Network Device

The figure on the right shows a (simplified version of a) packet that was recorded on a machine's network interface using Wireshark.

- What kind of activity does this packet belong to?
Solution: Web browsing. The easiest way to see this is by looking at the payload of the application layer (the lowermost box in the figure). There, one can see that the packet contains a HTTP-GET request to comm-net.ethz.ch.
- You know that a packet is composed of data in different layers. The fields in the right-hand figure are already grouped to four blocks representing the link, network, transport and application layer. Do you know which protocol is used in each of the four layers?

Solution:

- Link layer: Ethernet (indicated by the address- format 00:00:00:00:00:00). The EtherType tells the parser which protocol is used for the next layer. In this case, the 0x800 stands for IPv4.
 - Network layer: IP version 4 (indicated by the address format 000.000.000.000). The protocol field identifies the next protocol type: here, the 6 stands for TCP.
 - Transport layer: TCP (identifiable by comparing with the format of the TCP-header). The destination port tells the networking stack to which application the payload should be delivered. Port 80 is the standard port for HTTP.
 - Application layer: HTTP (HTTP-Request in plain text)
- After being sent out at your machine's network interface, the packet will traverse a switch which will forward it based on the destination address. On which layer does the switch operate and which is the destination address it will look at?
Solution: Switches operate at the link layer. The destination address in this layer is 64:a0:e7:42:2e:c2.
 - After leaving your local network, the packet will traverse a router. On which layer does the router operate and which is the destination address it will look at?
Solution: Routers operate at the network layer. The destination address in this layer is 82.130.102.210.



Packet consisting of data in four layers (link, network, transport and application layer).

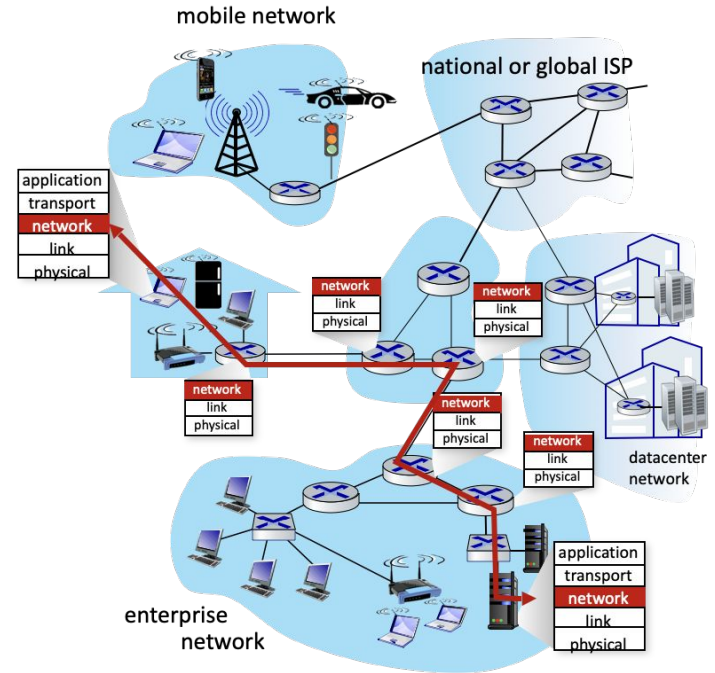
Routing Foundations

Network Layer Functions

- **Forwarding**: move packet from router input interface to correct output interface (data plane)
- **Routing**: determine route taken by packets from source to destination (control plane)
- There are two approaches to structuring the control plane:
 - Local view
 - Global view

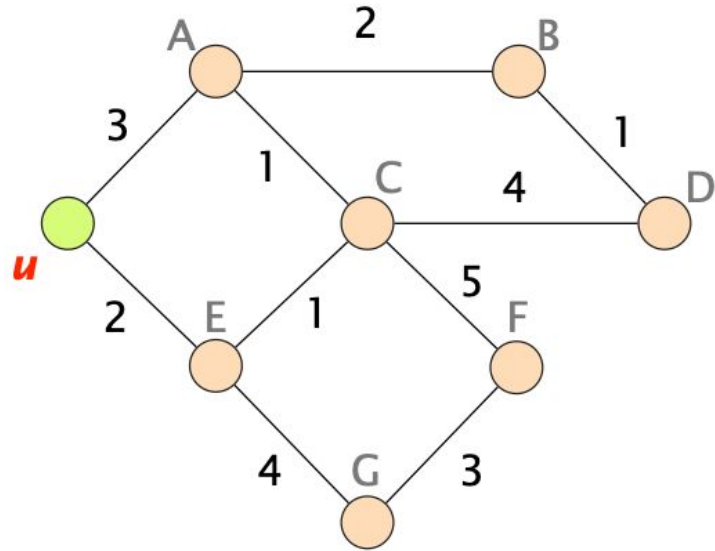
Routing Protocols

- **Routing protocol goal**: determine “good” paths (i.e., routes), from sending hosts to receiving host, through network of routers
- **path**: sequence of routers packets traverse from given initial source host to final destination host
- **“good”**: least “cost”, “fastest”, “least congested”



We Use Graph Abstractions to Solve Path Selection

- **Nodes** are routers
- **Edges** are links
- **Weights** are costs

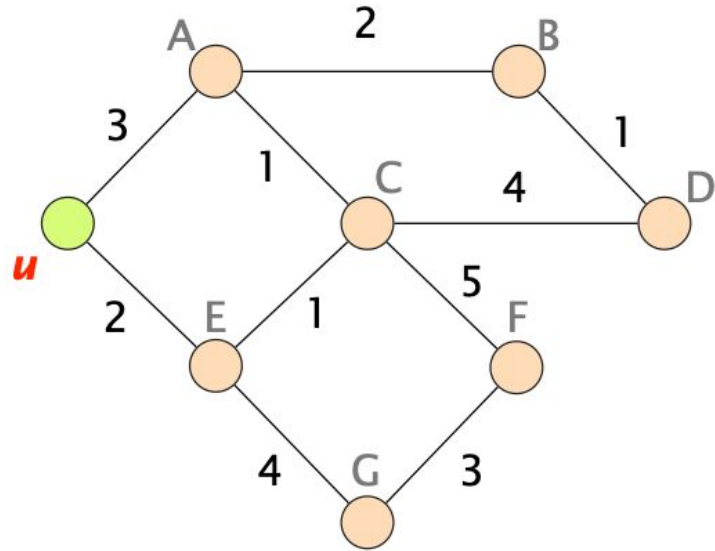


Dijkstra's Algorithm for Shortest Path Search

$S = \{u\}$ — set of nodes for which we know the shortest-path

$D(v)$ — the smallest distance currently known by u to reach v

$c(u, v)$ — the weight of the link connecting u and v



Dijkstra's Algorithm for Shortest Path Search

Initialization

$S = \{u\}$

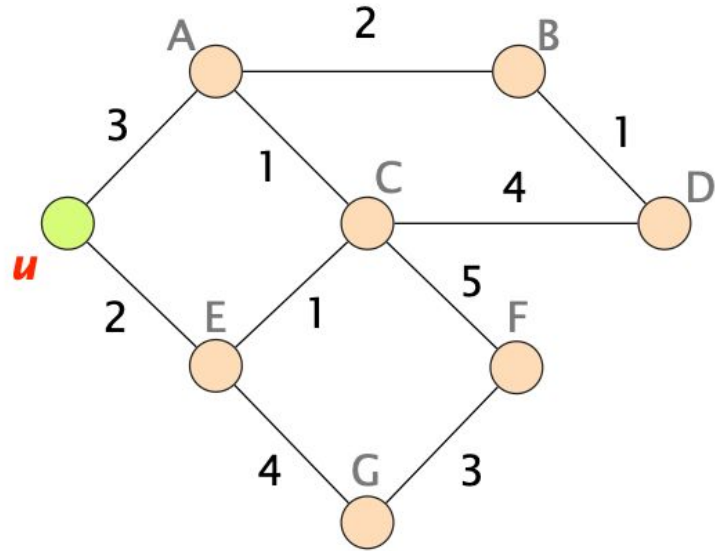
for all nodes v :

if (v is adjacent to u):

$$D(v) = c(u, v)$$

else:

$$D(v) = \infty$$

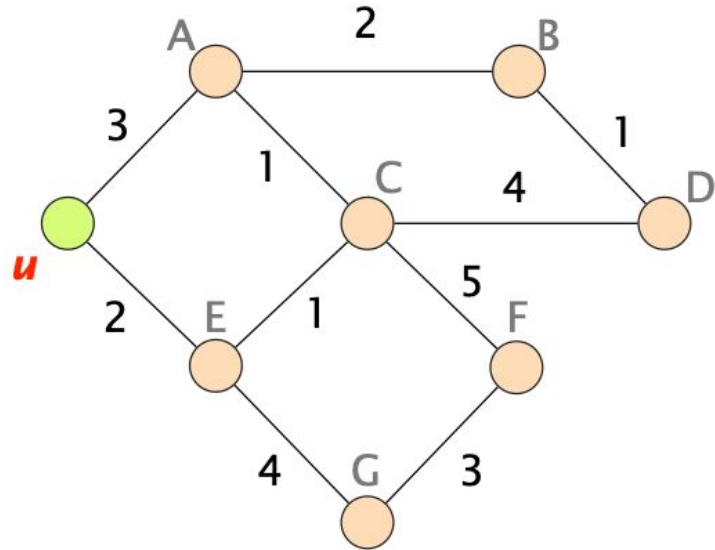


Dijkstra's Algorithm for Shortest Path Search

$D(.) =$

$S = \{u\}$

A	3
B	∞
C	∞
D	∞
E	2
F	∞
G	∞



Dijkstra's Algorithm for Shortest Path Search

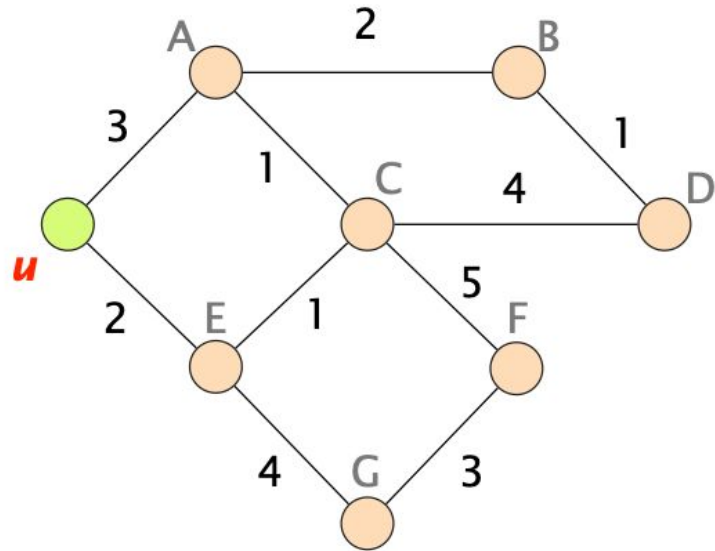
Loop

while *not* all nodes in S:

 add w with the smallest $D(w)$ to S

 update $D(v)$ for all adjacent v not in S:

$$D(v) = \min\{D(v), D(w) + c(w,v)\}$$



Dijkstra's Algorithm for Shortest Path Search

