Internet Layering

Internet Layering

- Internet design is partitioned into layers
  - each layer relies on services provided by the layer below
  - each layer provides services to the layer above
- analogy: software you write
  - code you write
  - run-time library
  - system calls
  - device drivers
  - voltage levels

When you call “fopen”
you are not thinking about voltages.

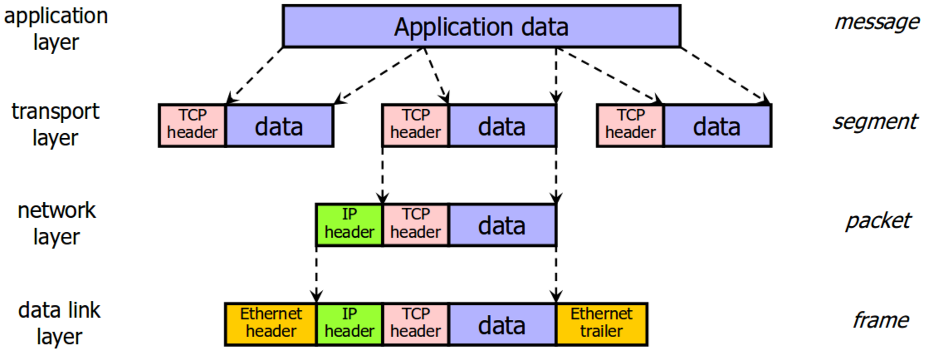
Internet Layering

- AKA protocol stack
- has the layers
  - application
  - transport (e.g., TCP)
  - (inter)network (e.g., IP)
  - link (e.g., ethernet)
  - physical (e.g., radio waves)

The stack is always drawn with low level layers below.
But for the packets the low-layer headers
come before and are drawn on top.

Vertical View of a Packet

- link layer header (first bit transmitted)
- (inter)network layer header
- transport layer header
- application data (depends on the application)


application layer
transport layer
network layer
data link layer
Application data
TCP header
data
TCP header
data
TCP header
data
IP header
TCP header
data
Ethernet header
IP header
TCP header
data
Ethernet trailer
message
segment
packet
frame

Physical Layer

- encodes bits to send over a single physical link
  - voltage levels
  - photon intensities
  - RF modulations

Link Layer

- framing and transmission of a collection of bits
  - into individual **messages** sent a single **subnetwork**
  - this may involve multiple physical links
    - e.g., Ethernet
  - often supports **broadcast** transmission
    - every **node** connected to subnet receives

Network Layer

- bridges multiple subnets
  - provides **end-to-end** Internet connectivity between nodes
  - provides global addressing
  - works across **different** link technologies
- the link and physical layers can change for each “hop”
  - the data for the network layer and above stays the same

Transport Layer

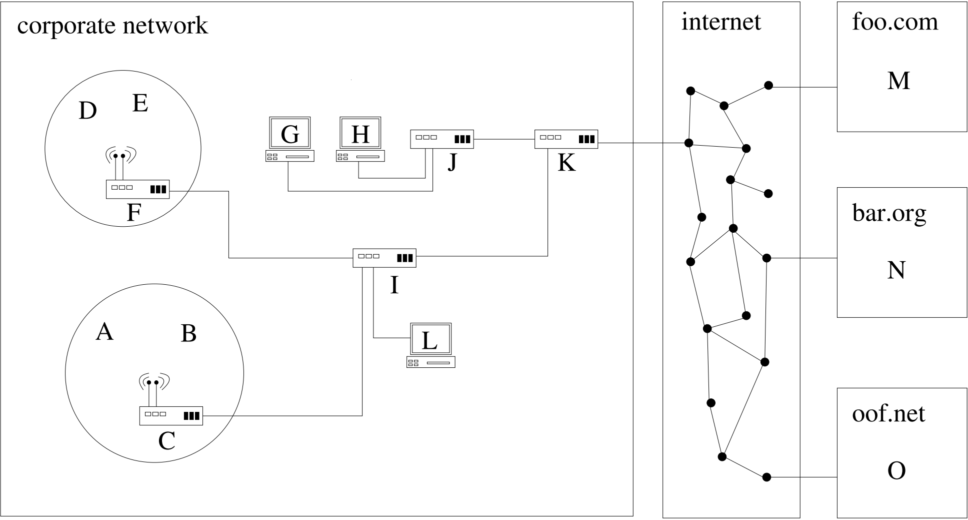
- end-to-end communication between processes
  - TCP: reliable byte stream
    - provides guaranteed in-order delivery
    - provides congestion control
  - UDP: unreliable datagrams
    - datagram is a single packet message

Application Layer

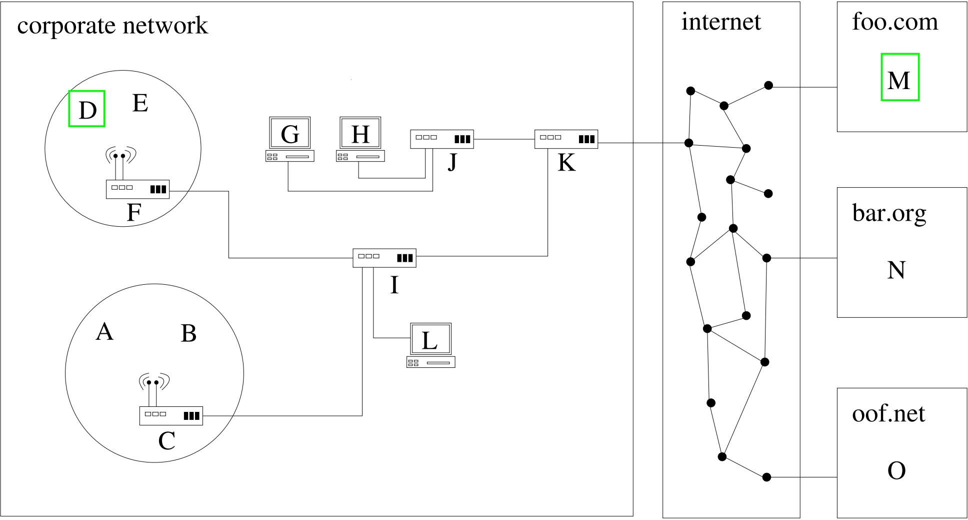
- communication of whatever you want
  - write to a stream at one end
  - read from a stream at the other
- freely structured
  - e.g., SMTP, HTTP, BitTorrent

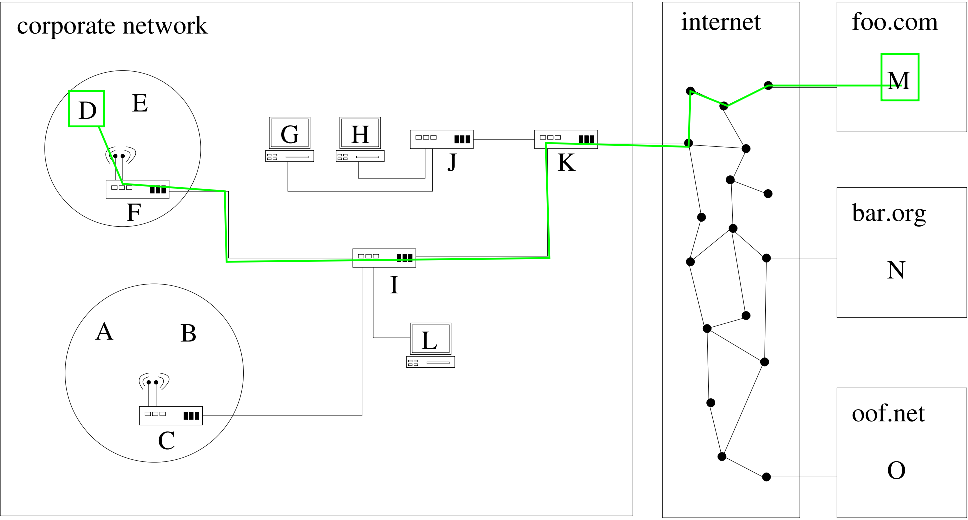
Application and Transport only implemented at hosts, not at interior routers.

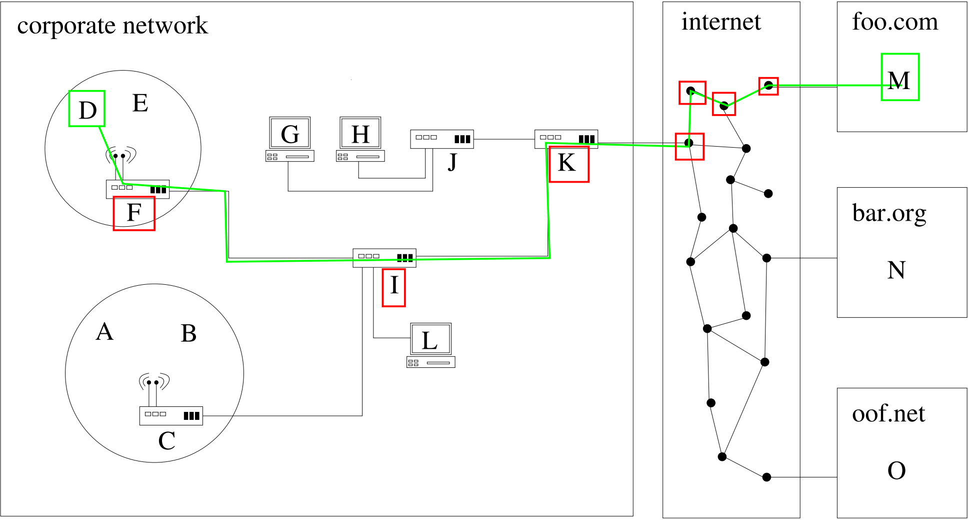
Application and Transport only implemented at hosts, not at interior routers.
Physical, link, and network implemented everywhere.

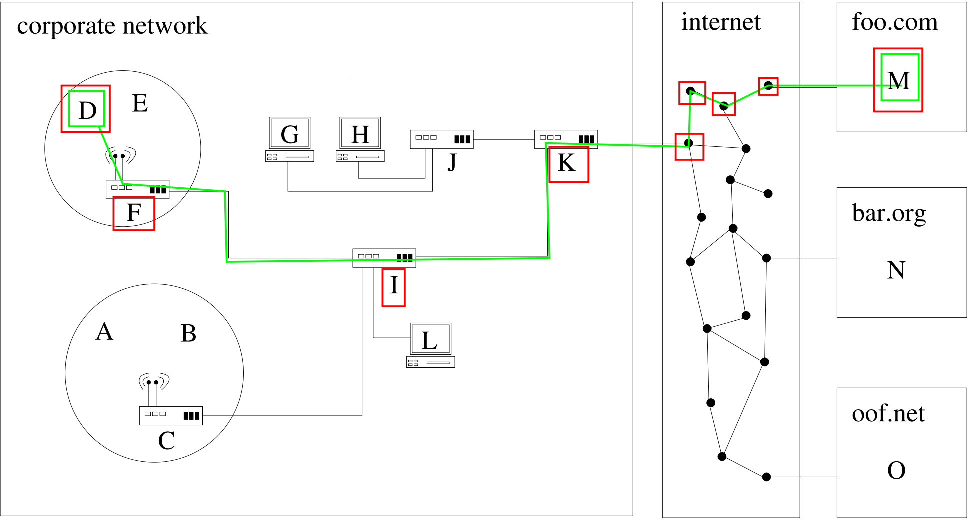


corporate network
D
E
F
G
H
J
K
I
L
A
B
C
internet
foo.com
M
bar.org
N
oof.net
O

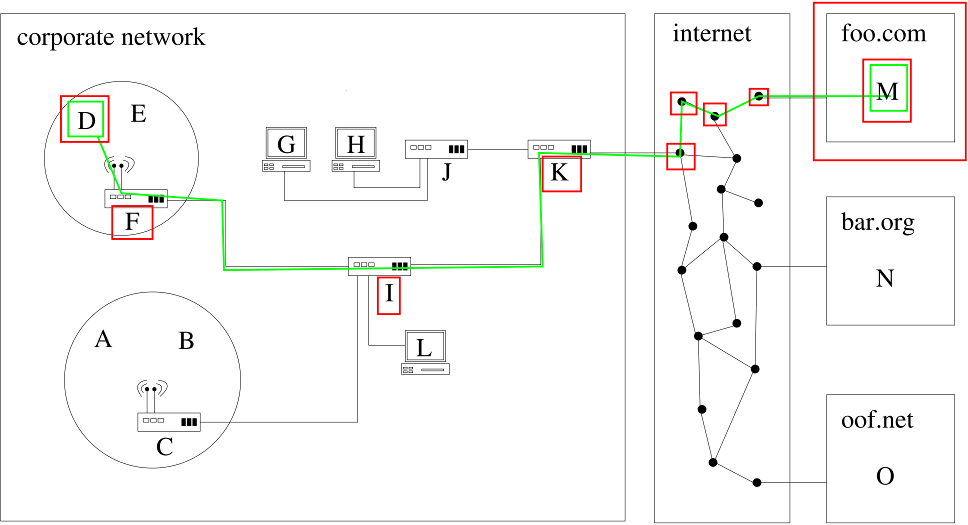

corporate network
D
E
F
A
B
C
G
H
J
K
I
L
internet
foo.com
M
bar.org
N
oof.net
O

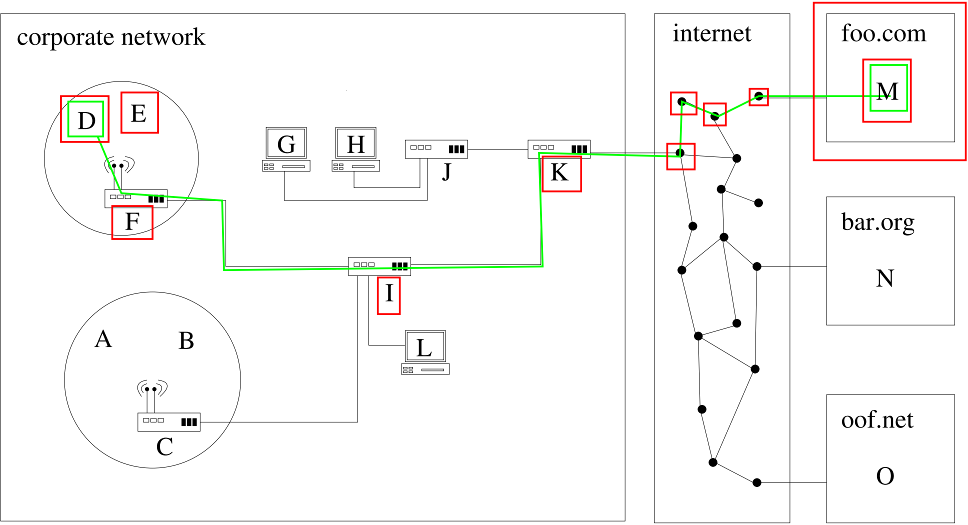

corporate network
D
E
F
G
H
J
K
I
L
A
B
C
internet
foo.com
M
bar.org
N
oof.net
O

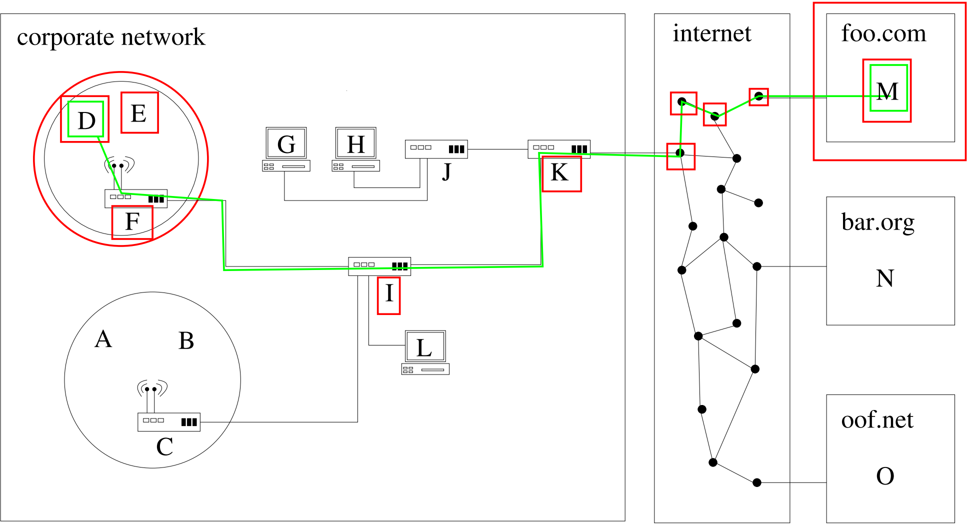

corporate network
D
E
F
G
H
J
K
I
L
A
B
C
internet
foo.com
M
bar.org
N
oof.net
O



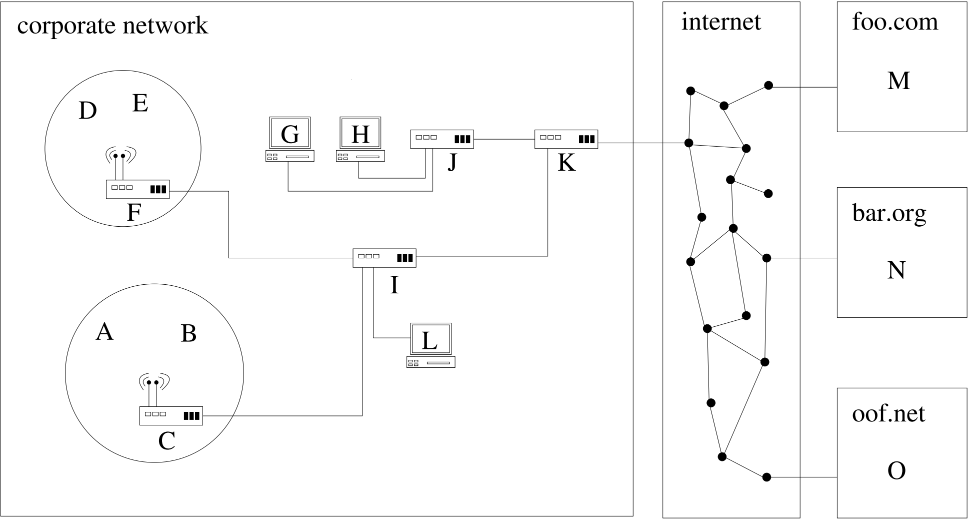
corporate network
D
E
F
G
H
J
K
I
L
A
B
C
internet
foo.com
M
bar.org
N
oof.net
O

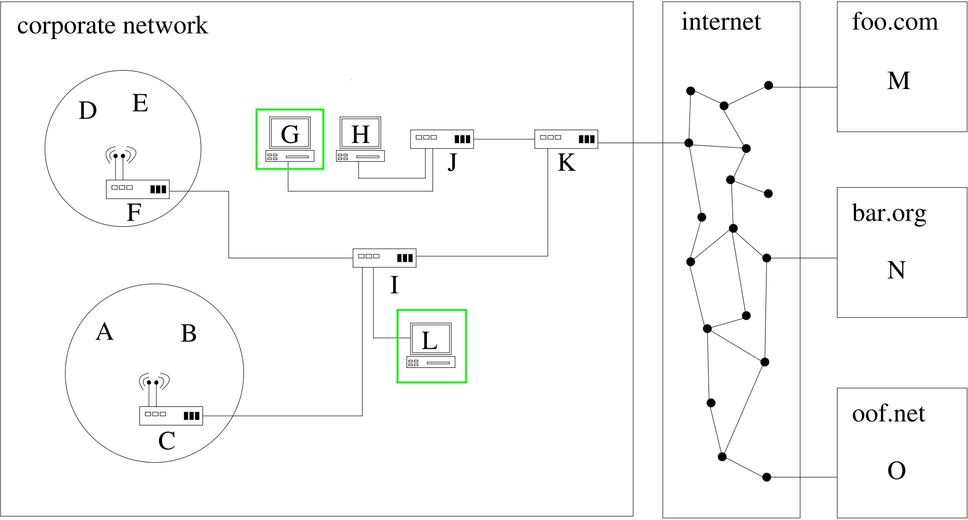

corporate network
D
E
F
G
H
J
K
I
L
A
B
C
internet
foo.com
M
bar.org
N
oof.net
O


corporate network
D
E
F
A
B
C
G
H
J
K
I
L
internet
foo.com
M
bar.org
N
oof.net
O

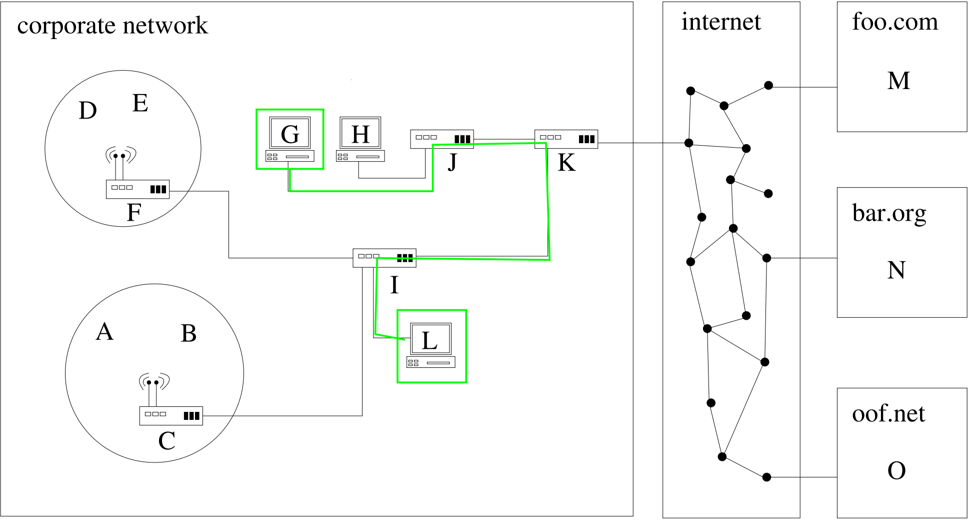


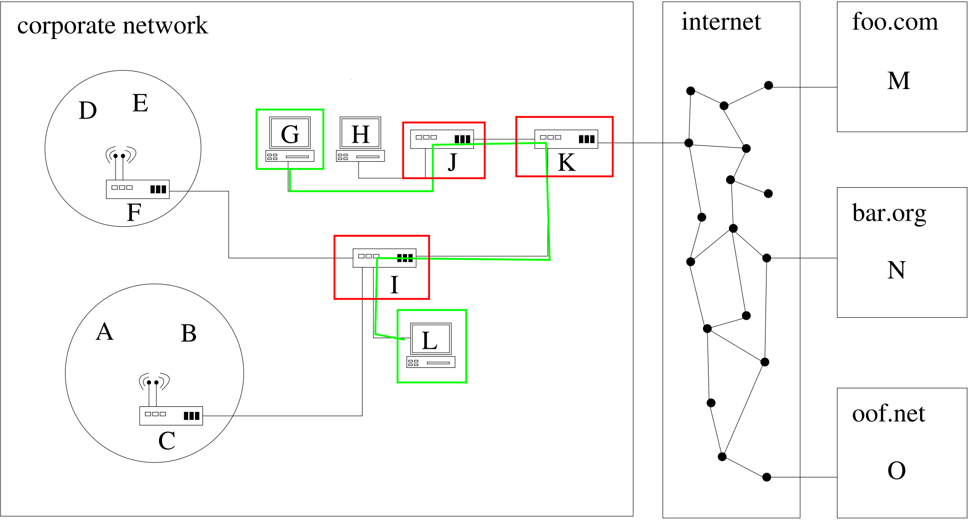
corporate network
D
E
F
G
H
J
K
I
L
A
B
C
internet
foo.com
M
bar.org
N
oof.net
O

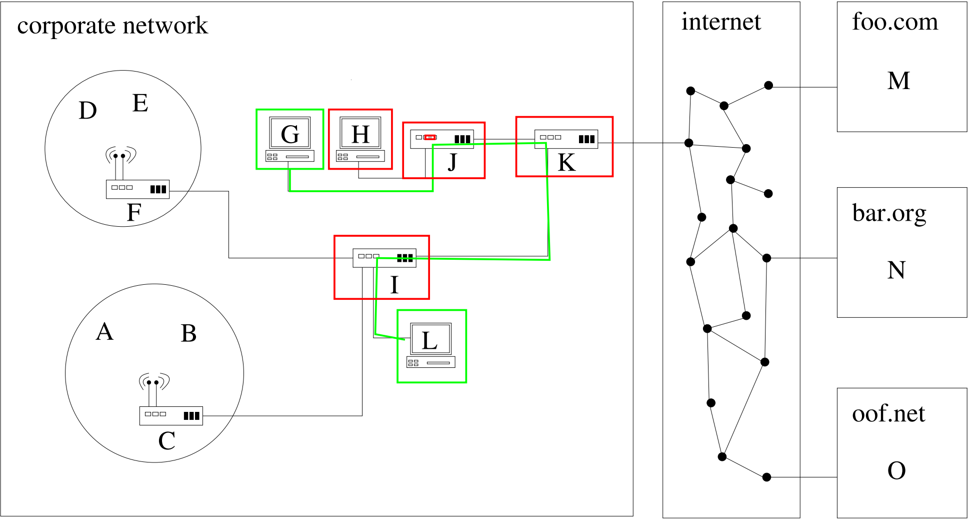

corporate network
D
E
F
G
H
J
K
I
L
A
B
C
internet
foo.com
M
bar.org
N
oof.net
O

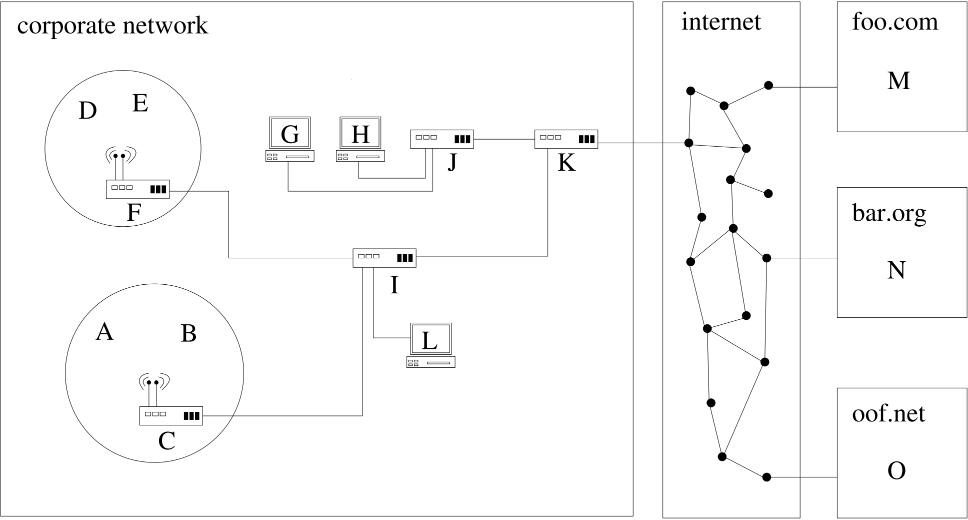


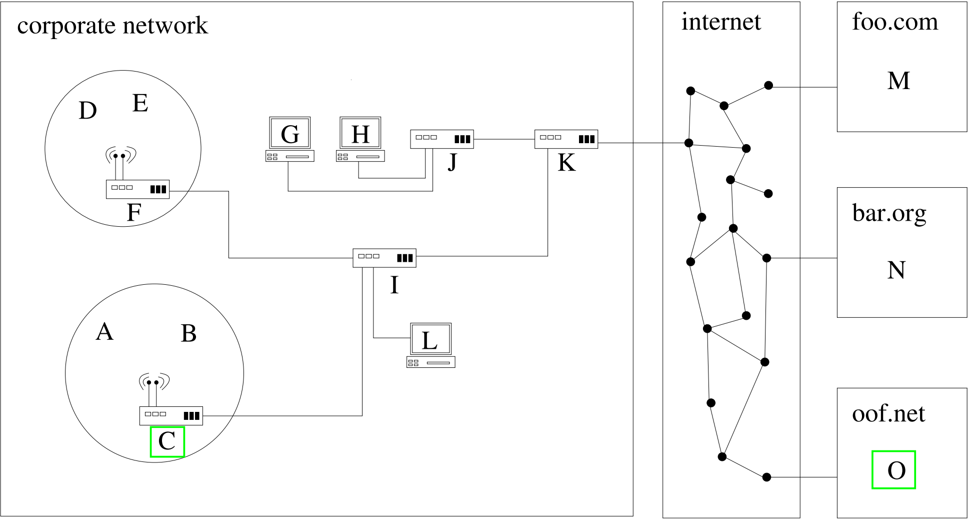
corporate network
D
E
F
G
H
J
K
I
A
B
C
L
internet
foo.com
M
bar.org
N
oof.net
O


corporate network
D
E
F
G
H
J
K
I
L
A
B
C
internet
foo.com
M
bar.org
N
oof.net
O

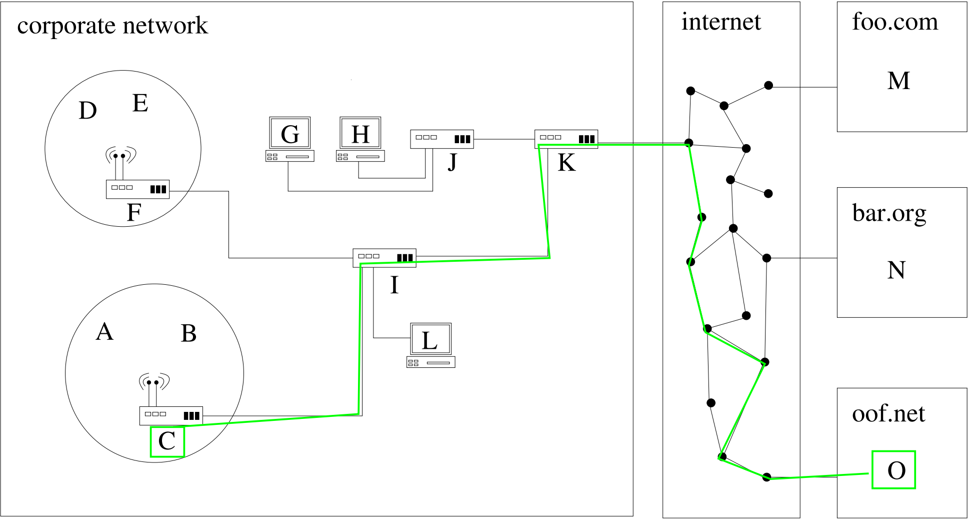

corporate network
D
E
F
G
H
J
K
I
L
A
B
C
internet
foo.com
M
bar.org
N
oof.net
O

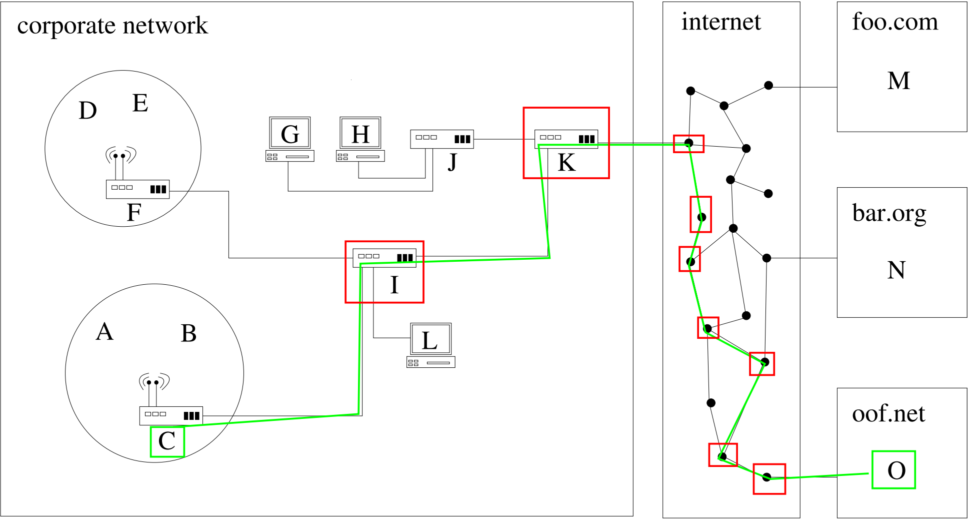

corporate network
D
E
F
G
H
J
K
I
L
A
B
C
internet
foo.com
M
bar.org
N
oof.net
O

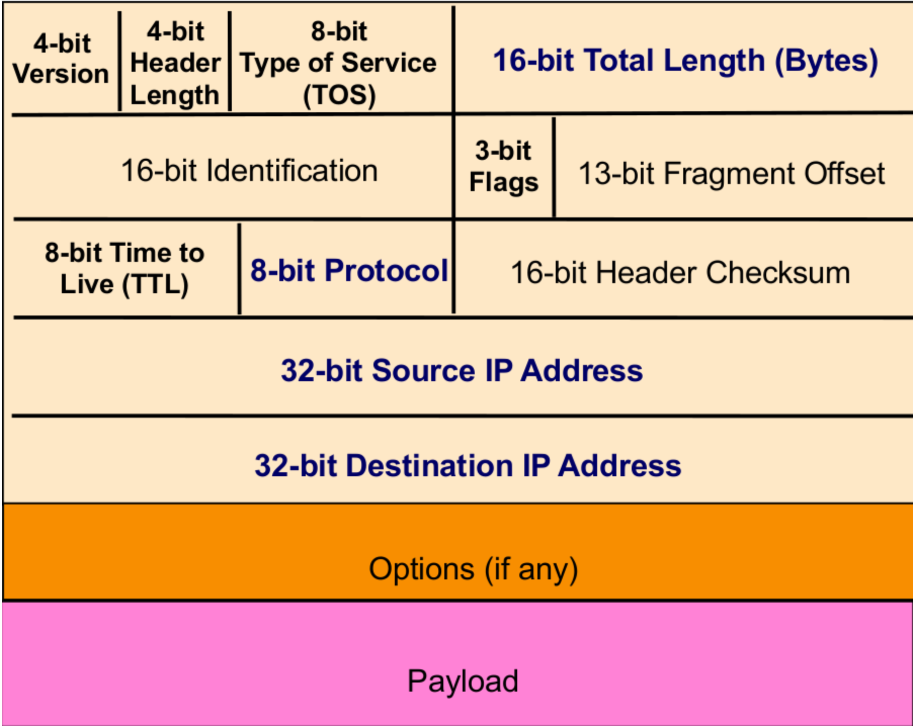

corporate network
D
E
F
G
H
J
K
I
L
A
B
C
internet
foo.com
M
bar.org
N
oof.net
O



corporate network
D
E
F
G
H
J
K
I
L
A
B
C
internet
foo.com
M
bar.org
N
oof.net
O


corporate network
D
E
F
G
H
J
K
I
L
A
B
C
internet
foo.com
M
bar.org
N
oof.net
O


corporate network
D
E
F
G
H
J
K
I
L
A
B
C
internet
foo.com
M
bar.org
N
oof.net
O


4-bit Version
4-bit Header Length
8-bit Type of Service (TOS)
16-bit Total Length (Bytes)
16-bit Identification
3-bit Flags
13-bit Fragment Offset
8-bit Time to Live (TTL)
8-bit Protocol
16-bit Header Checksum
32-bit Source IP Address
32-bit Destination IP Address
Options (if any)
Payload

Which of these fields can an
on-path non-blinder attacker change?

IP Packet

- has two IP addresses
  - source IP
  - destination IP
- destination address
  - unique identifier (locator) for the receiving host
  - allows each node (router) to make forwarding decision
- source address
  - unique identifier (locator) for the sending host
  - recipient can decide whether to accept packet
  - allows recipient to send a reply back to source

IP “best effort” Packet Delivery

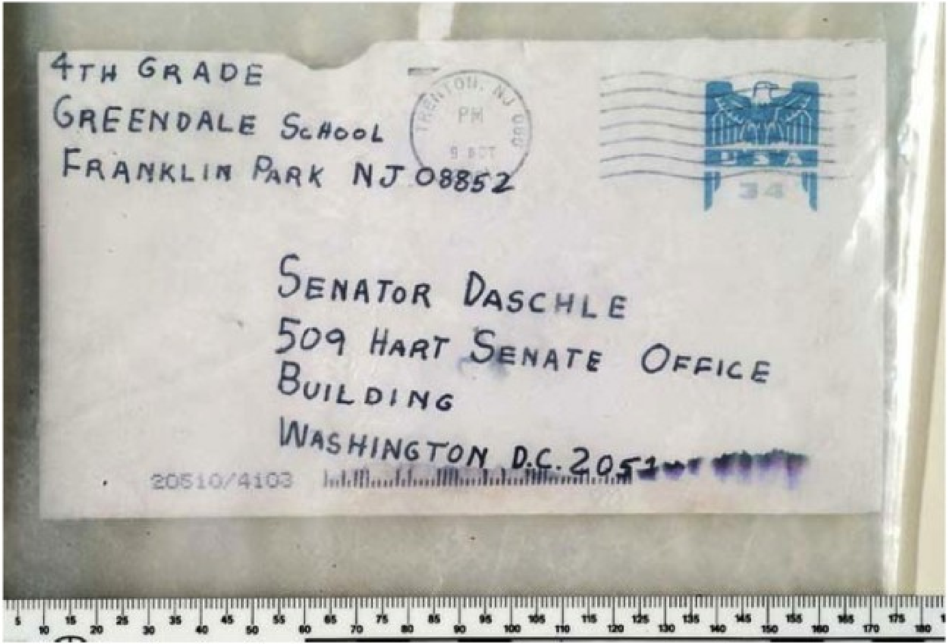
- router looks at destination address
- locates “next hop” in forwarding table
- only gives a “I’ll give it a try” delivery service
    - packets may be lost
    - packets may be corrupted
    - packets may be delivered out of order

IP Address Spoofing

- sender can put whatever they want for IP source address
  - i.e., not their actual IP address
  - called spoofing, imposturing, masquerading
- destination address controled by socket API
  - e.g., connect()
- source address can only be set with a raw socket
  - instead of OS creating TCP/IP headers, the program writes them all
  - a privileged operation
    - requires root or cap_net_raw

IP Address Spoofing

- sender can make a packet appear as though it came from elsewhere
  - sender won't likely get reply
  - if attacker doesn't need reply then this is enough to attack
- if attacker can also eavesdrop on reply then this is a powerful attack
- **blind spoofing**: spoofing without eavesdropping
- **on path**: traffic goes through attacker
- **off path**: traffic does not go through attacker


4TH GRADE
GREENDALE SCHOOL
FRANKLIN PARK NJ 08852
TRENTON, NJ 085
PM
9 OCT
USA
34
SENATOR DASCHLE
509 HART SENATE OFFICE
BUILDING
WASHINGTON D.C. 2051
20510/4103

Anti-Spoofing Mechanism

- for end-users spoofing is easy to detect
  - your ISP assigns you an IP address
  - your ISP sees all your network traffic
  - your ISP can check if your IP matches
- symmetry principle
  - if I wouldn't **send** you this packet were src/dst swapped then I won't believe the source is correct
  - reject the packet instead of sending it on network
  - analogy: letter dropped in Calgary mail box with return address in Edmonton
- this is harder to do between ASes in general, however
  - analogy: letter received on some airmail flight to a sorting facility
    - maybe someone didn't do a check earlier
    - maybe they would send you mail but you wouldn't send it to them

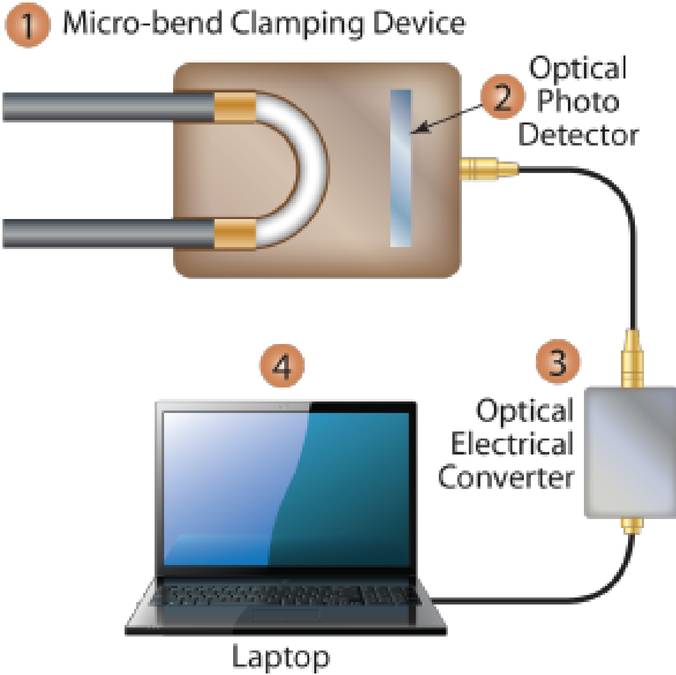
Threats Due to the Lower Layers

Physical and Link Layer Threats

- eavesdropping (also called sniffing)
  - subnets using broadcast (e.g., Wi-Fi) it's "free"
  - any attached NIC (network interface card) can capture communication on the subnet
- `tcpdump` is a handy tool to do that
- `wireshark` is a GUI that does protocol analysis
- any router **on-path** can look at or export traffic
- anyone **on-path** can "tap" a link

The divers found the cable and installed a 20-foot long listening device on the cable. designed to attach to the cable without piercing the casing, the device recorded all communications that occurred. If the cable malfunctioned and the Soviets raised it for repair, the bug, by design, would fall to the bottom of the ocean. Each month Navy divers retrieved the recordings and installed a new set of tapes.

Upon their return to the United States, intelligence agents from the NSA analyzed the recordings and tried to decipher any encrypted information. The Soviets apparently were confident in the security of their communications lines, as a surprising amount of sensitive information traveled through the lines without encryption.


1 Micro-bend Clamping Device
2 Optical Photo Detector
3 Optical Electrical Converter
4
Laptop

Internet Bootstrapping

Dynamic Host Configuration Protocol (DHCP)

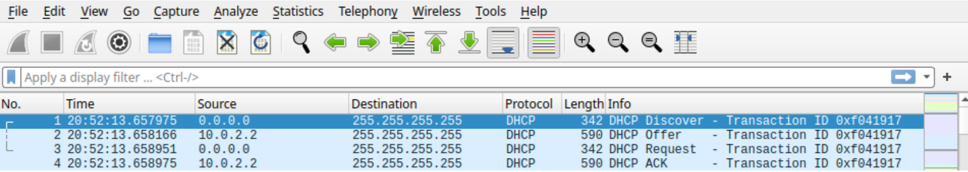
- a new host doesn't have an IP yet
  - doesn't know what source address to use
- host doesn't know who to ask for IP address?
  - doesn't know what destination address to use

DHCP

- host broadcasts a server-discovery message (link layer)
- "Does anyone know what basic config should I use?"
- server(s) sends a reply offering an address

DHCP

- new client → DHCP server: DHCP discovery (broadcast)
- DHCP server → new client: DHCP offer
  - offer contains IP, DNS, gateway router
  - how long client can use them (lease time)
  - DNS resolves hostnames (like gmail.com) to IPs
  - gateway is router that acts as first hop to reach out to the Internet
- new client → DHCP server: DHCP request (broadcast)
- DHCP server → new client: DHCP acknowledgement
  - last two are to confirm that's the IP the client will use


File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
No. Time Source Destination Protocol Length Info
1 20:52:13.657975 0.0.0.0 255.255.255.255 DHCP 342 DHCP Discover - Transaction ID 0xf041917
2 20:52:13.658166 10.0.2.2 255.255.255.255 DHCP 590 DHCP Offer - Transaction ID 0xf041917
3 20:52:13.658951 0.0.0.0 255.255.255.255 DHCP 342 DHCP Request - Transaction ID 0xf041917
4 20:52:13.658975 10.0.2.2 255.255.255.255 DHCP 590 DHCP ACK - Transaction ID 0xf041917

```
▸ User Datagram Protocol, Src Port: 68, Dst Port: 67
▾ Dynamic Host Configuration Protocol (Discover)
    Message type: Boot Request (1)
    Hardware type: Ethernet (0x01)
    Hardware address length: 6
    Hops: 0
    Transaction ID: 0x0f041917
    Seconds elapsed: 3
  ▸ Bootp flags: 0x0000 (Unicast)
    Client IP address: 0.0.0.0
    Your (client) IP address: 0.0.0.0
    Next server IP address: 0.0.0.0
    Relay agent IP address: 0.0.0.0
    Client MAC address: 52:54:00:12:34:56 (52:54:00:12:34:56)
    Client hardware address padding: 00000000000000000000
    Server host name not given
    Boot file name not given
    Magic cookie: DHCP
  ▾ Option: (53) DHCP Message Type (Discover)
      Length: 1
      DHCP: Discover (1)
  ▾ Option: (61) Client identifier
      Length: 7
      Hardware type: Ethernet (0x01)
      Client MAC address: 52:54:00:12:34:56 (52:54:00:12:34:56)
```

```
▸ User Datagram Protocol, Src Port: 67, Dst Port: 68
▾ Dynamic Host Configuration Protocol (Offer)
    Message type: Boot Reply (2)
    Hardware type: Ethernet (0x01)
    Hardware address length: 6
    Hops: 0
    Transaction ID: 0x0f041917
    Seconds elapsed: 0
  ▸ Bootp flags: 0x0000 (Unicast)
    Client IP address: 0.0.0.0
    Your (client) IP address: 10.0.2.15
    Next server IP address: 10.0.2.2
    Relay agent IP address: 0.0.0.0
    Client MAC address: 52:54:00:12:34:56 (52:54:00:12:34:56)
    Client hardware address padding: 00000000000000000000
    Server host name not given
    Boot file name not given
    Magic cookie: DHCP
  ▸ Option: (53) DHCP Message Type (Offer)
  ▸ Option: (54) DHCP Server Identifier (10.0.2.2)
  ▸ Option: (1) Subnet Mask (255.255.255.0)
  ▾ Option: (3) Router
      Length: 4
      Router: 10.0.2.2
  ▾ Option: (6) Domain Name Server
      Length: 4
      Domain Name Server: 10.0.2.3
  ▾ Option: (51) IP Address Lease Time
      Length: 4
      IP Address Lease Time: 1 day (86400)
```

```
▾ Dynamic Host Configuration Protocol (Request)
    Message type: Boot Request (1)
    Hardware type: Ethernet (0x01)
    Hardware address length: 6
    Hops: 0
    Transaction ID: 0x0f041917
    Seconds elapsed: 3
  ▸ Bootp flags: 0x0000 (Unicast)
    Client IP address: 0.0.0.0
    Your (client) IP address: 0.0.0.0
    Next server IP address: 0.0.0.0
    Relay agent IP address: 0.0.0.0
    Client MAC address: 52:54:00:12:34:56 (52:54:00:12:34:56)
    Client hardware address padding: 00000000000000000000
    Server host name not given
    Boot file name not given
    Magic cookie: DHCP
  ▸ Option: (53) DHCP Message Type (Request)
  ▾ Option: (61) Client identifier
        Length: 7
        Hardware type: Ethernet (0x01)
        Client MAC address: 52:54:00:12:34:56 (52:54:00:12:34:56)
  ▸ Option: (50) Requested IP Address (10.0.2.15)
  ▸ Option: (54) DHCP Server Identifier (10.0.2.2)
  ▸ Option: (57) Maximum DHCP Message Size
```

```
▸ User Datagram Protocol, Src Port: 67, Dst Port: 68
▾ Dynamic Host Configuration Protocol (ACK)
    Message type: Boot Reply (2)
    Hardware type: Ethernet (0x01)
    Hardware address length: 6
    Hops: 0
    Transaction ID: 0x0f041917
    Seconds elapsed: 0
  ▸ Bootp flags: 0x0000 (Unicast)
    Client IP address: 0.0.0.0
    Your (client) IP address: 10.0.2.15
    Next server IP address: 10.0.2.2
    Relay agent IP address: 0.0.0.0
    Client MAC address: 52:54:00:12:34:56 (52:54:00:12:34:56)
    Client hardware address padding: 00000000000000000000
    Server host name not given
    Boot file name not given
    Magic cookie: DHCP
  ▸ Option: (53) DHCP Message Type (ACK)
  ▸ Option: (54) DHCP Server Identifier (10.0.2.2)
  ▸ Option: (1) Subnet Mask (255.255.255.0)
  ▾ Option: (3) Router
      Length: 4
      Router: 10.0.2.2
  ▾ Option: (6) Domain Name Server
      Length: 4
      Domain Name Server: 10.0.2.3
  ▸ Option: (51) IP Address Lease Time
  ▸ Option: (255) End
```

DHCP Threats

- broadcast is done allowing any local attacker to race to reply
- attacker reply can give a bad DNS server
  - redirect any domain searches to attacker's choice
  - more on this later
- attacker reply can give a bad gateway router
  - puts attacker **on path** thenceforth
  - MITM to sniff or modify traffic
- victim has no idea it is happening
  - DHCP offer looks legitimate
  - multiple replies can happen benignly

What DHCP Shows

- broadcast protocols inherently at risk of local attacker spoofing
    - can spoof and eavesdrop
- when initializing, systems are particularly vulnerable
    - they lack a trusted foundation to build upon
- tension between wiring in trust versus flexibility and convenience
    - configuring your DNS and gateway is more secure
- MITM attacks can exist without any indicators they're occurring
    - on path attackers are suppose to be there