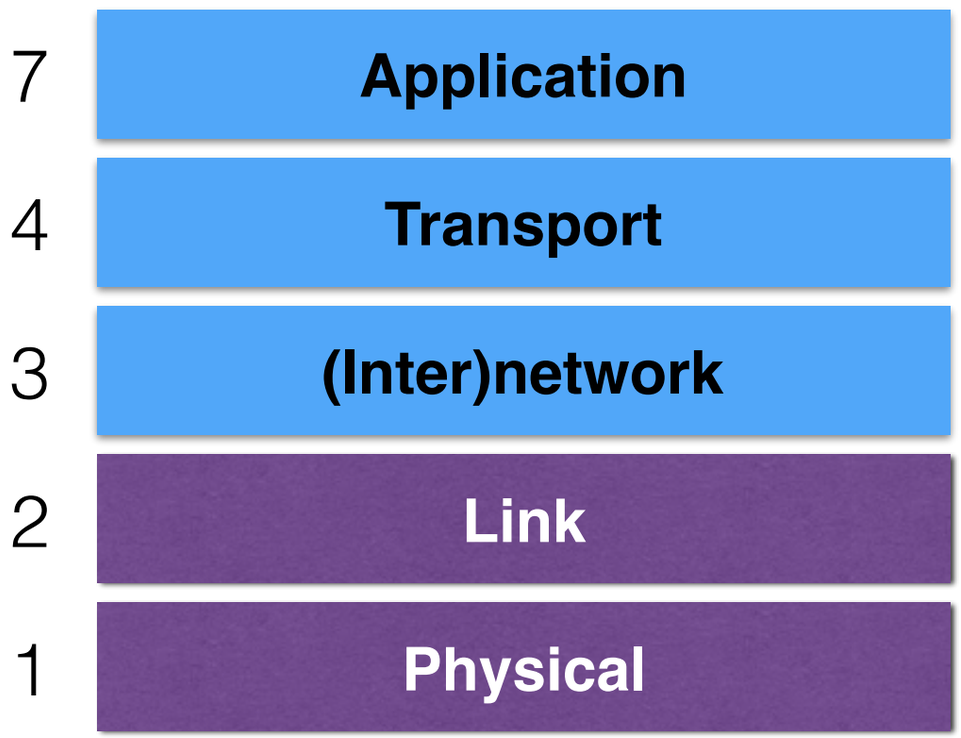
Network layer attacks

Slides from

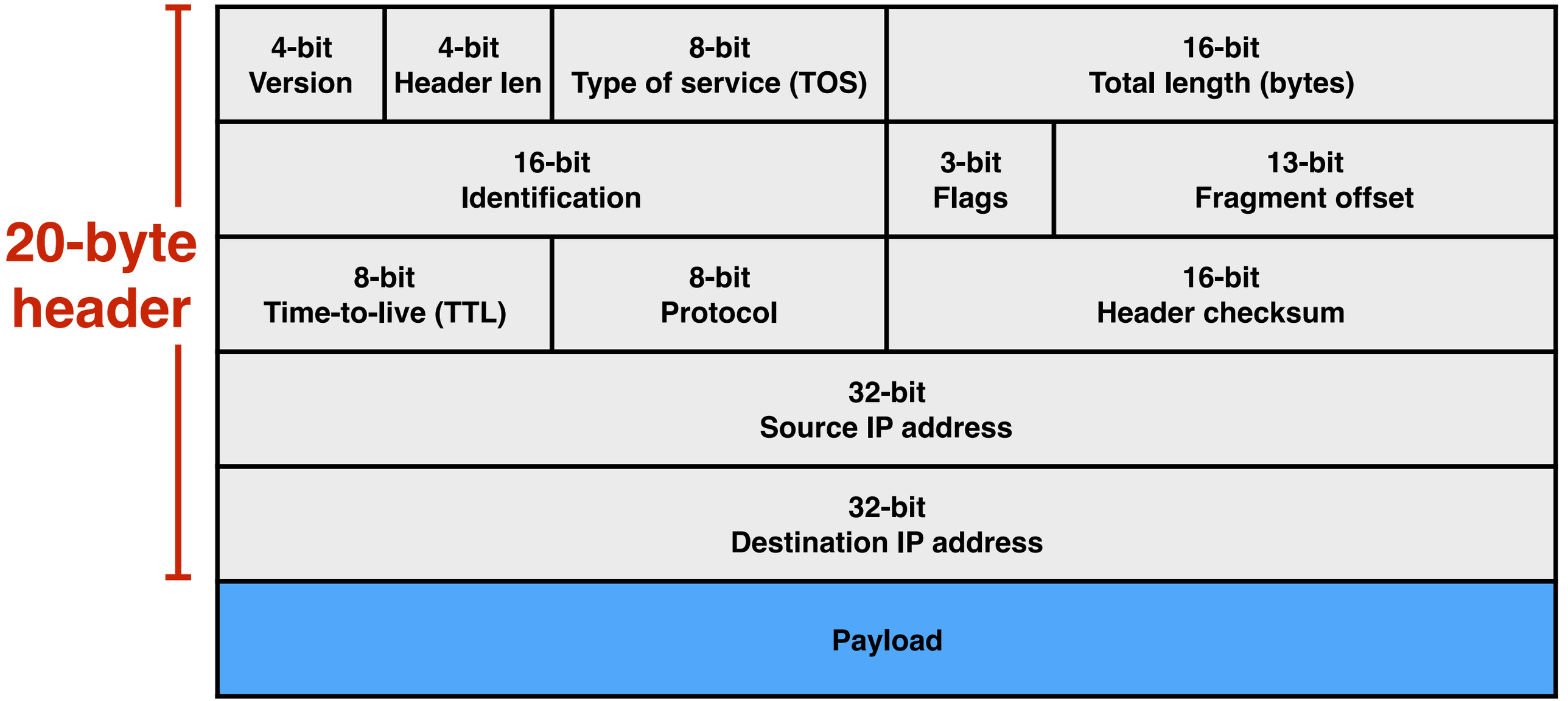
- Dave Levin 414-spring2016

Layer 3: (Inter)network layer



- Bridges multiple “subnets” to provide *end-to-end* internet connectivity between nodes
- Provides **global** addressing (IP addresses)
- Only provides **best-effort** delivery of data (i.e., no retransmissions, etc.)
- Works across different link technologies

IP packet “header”



IP Packet Header Fields (1)

- Version number (4 bits)
  - Indicates the version of the IP protocol
  - Necessary for knowing what fields follow
  - “4” (for IPv4) or “6” (for IPv6)

- Header length (4 bits)
  - How many 32-bit words (rows) in the header
  - Typically 5
  - Can provide IP options, too

- Type-of-service (8 bits)
  - Allow packets to be treated differently based on different needs
  - Low delay for audio, high bandwidth for bulk transfer, etc.

IP Packet Header Fields (2)

- Two IP addresses
  - Source (32 bits)
  - Destination (32 bits)

- Destination address
  - *Unique* identifier/locator for the receiving host
  - Allows each node (end-host and router) to make forwarding decisions

- Source address
  - Unique identifier/locator for the sending host
  - Recipient can decide whether to accept the packet
  - Allows destination to *reply* to the source

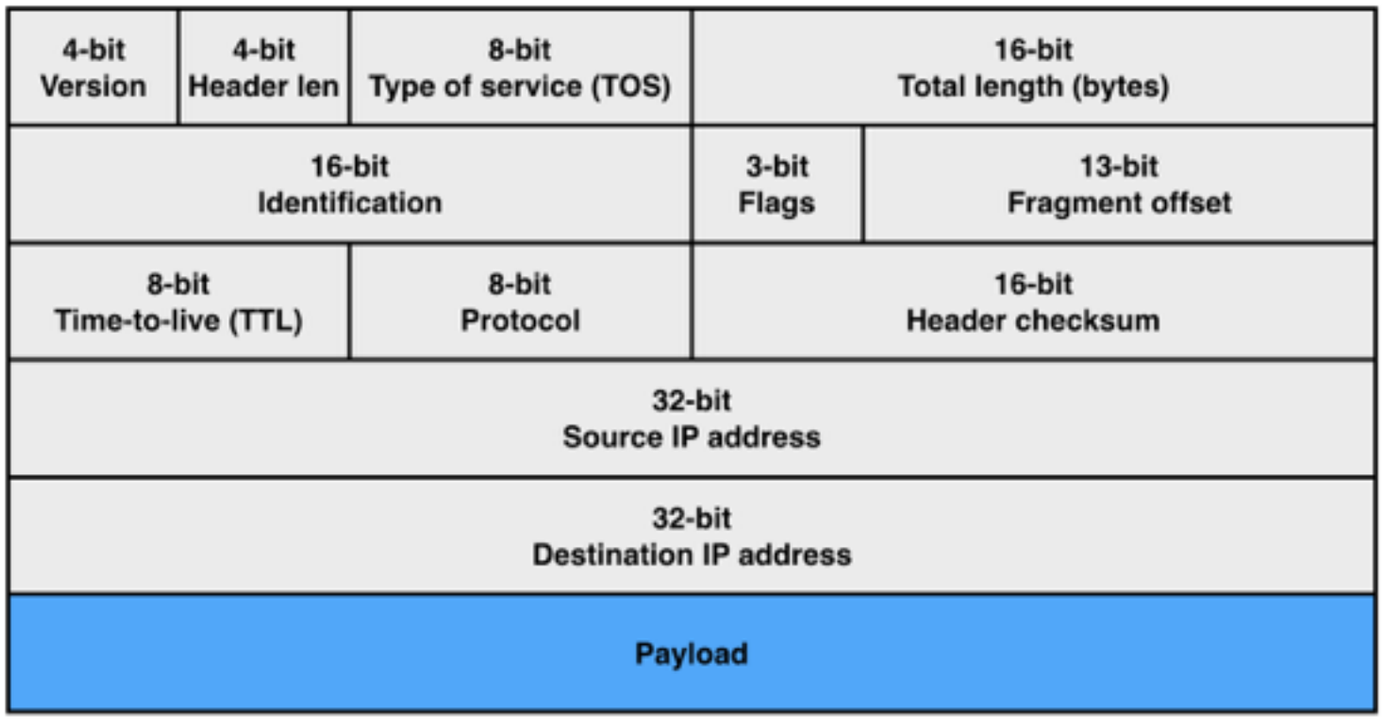
IP: “Best effort” packet delivery

- Routers inspect destination address, determine “next hop” in the forwarding table

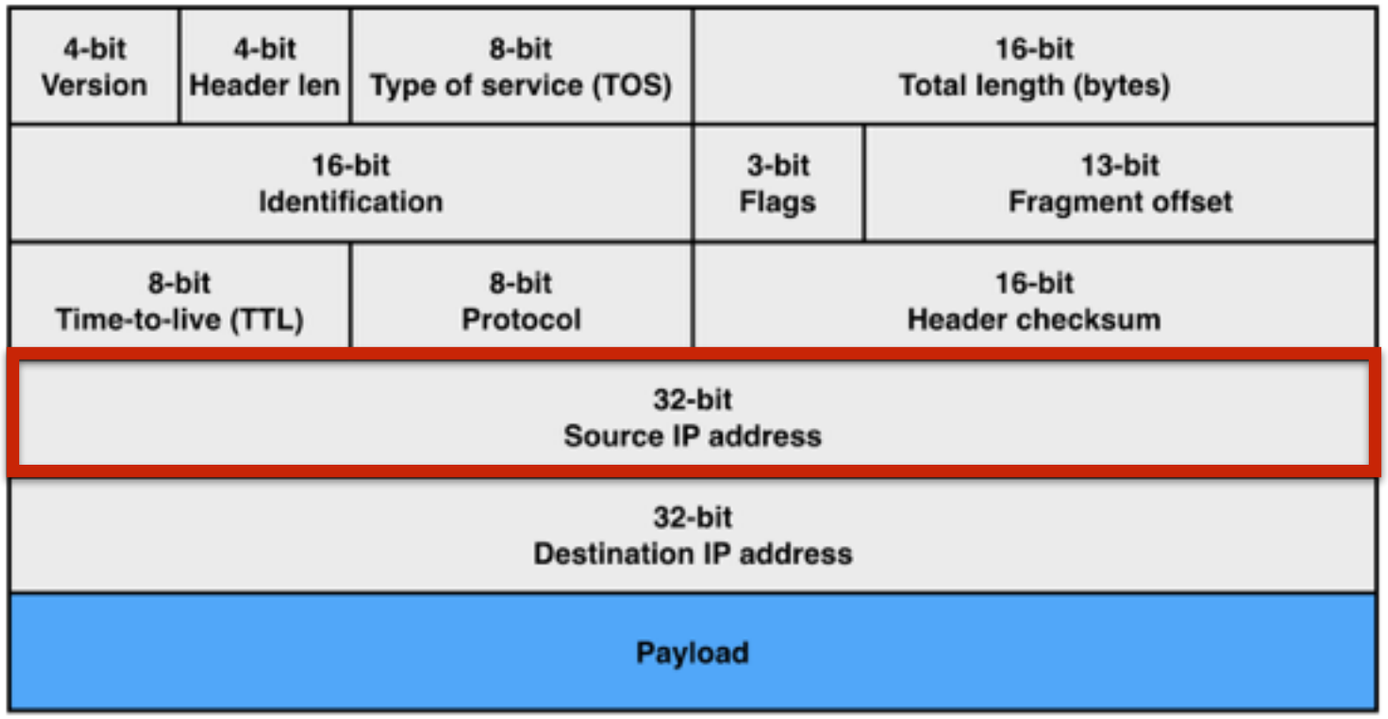
- Best effort = “I’ll give it a try”
  - Packets may be lost
  - Packets may be corrupted
  - Packets may be delivered out of order

Fixing these is the job of the transport layer!

Attacks on IP



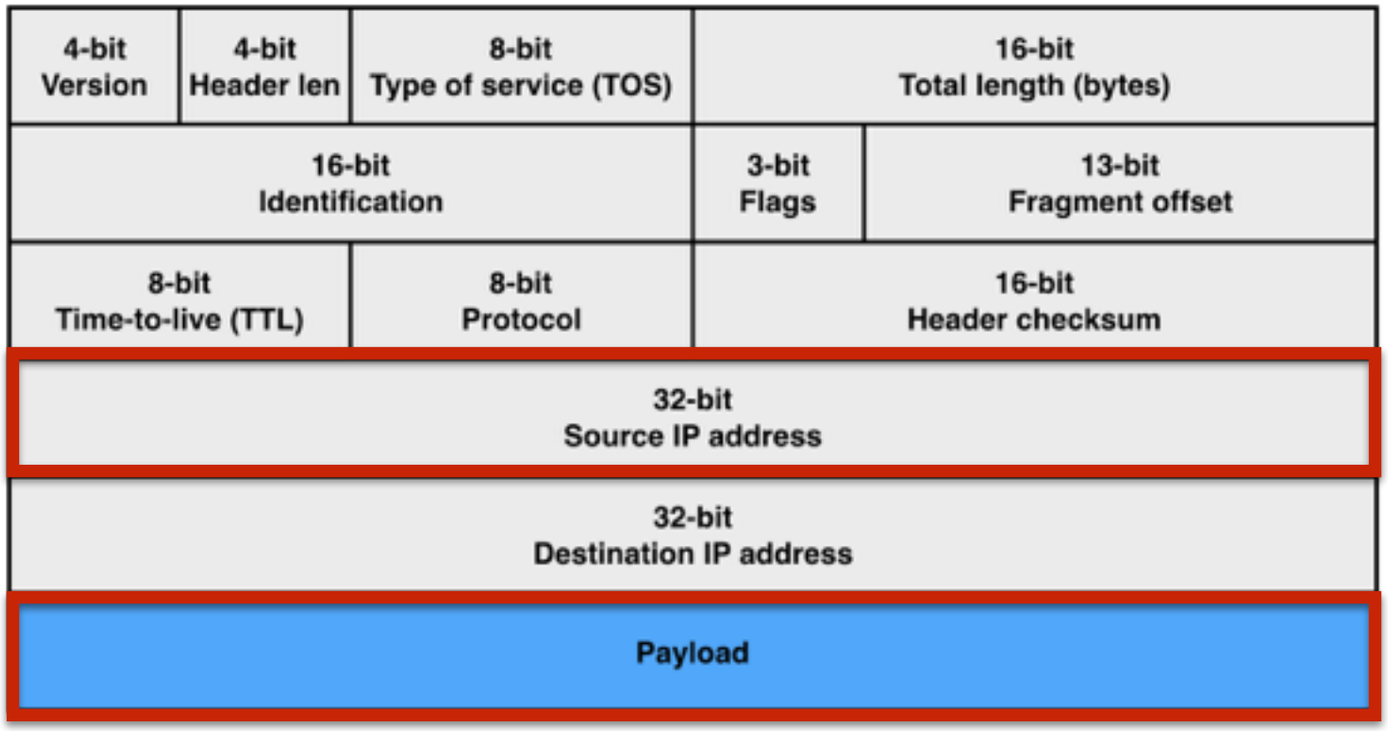
Attacks on IP



Source-spoof

There is nothing in IP that enforces that your source IP address is really “yours”

Attacks on IP



Source-spoof

There is nothing in IP that enforces that your source IP address is really “yours”

Eavesdrop / Tamper

IP provides no protection of the *payload* or *header*

Source-spoofing

- Why source-spoof?
  - Consider spam: send many emails from one computer
  - Easy defense: block many emails from a given (source) IP address
  - Easy countermeasure: spoof the source IP address
  - Counter-countermeasure?

- How do you know if a packet you receive has a spoofed source?

Salient network features

- Recall: The Internet operates via *destination-based routing*

- attacker: pkt (spoofed source) -> destination
  destination: pkt -> spoofed source

- In other words, the response goes to the spoofed source, *not* the attacker

Defending against source-spoofing

- How do you know if a packet you receive has a spoofed source?
  - Send a challenge packet to the (possibly spoofed) source (e.g., a difficult to guess, random nonce)
  - If the recipient can answer the challenge, then likely that the source was not spoofed

- So do you have to do this with every packet??
  - Every packet should have something that's difficult to guess
  - Recall the query ID in the DNS queries! Easy to predict => Kaminsky attack

Source spoofing

- Why source-spoof?
  - Consider DoS attacks: generate as much traffic as possible to congest the victim's network
  - Easy defense: block all traffic from a given source near the edge of your network
    - Easy countermeasure: spoof the source address
- Challenges won't help here; the damage has been done by the time the packets reach the core of our network
- Ideally, detect such spoofing *near the source*

Egress filtering

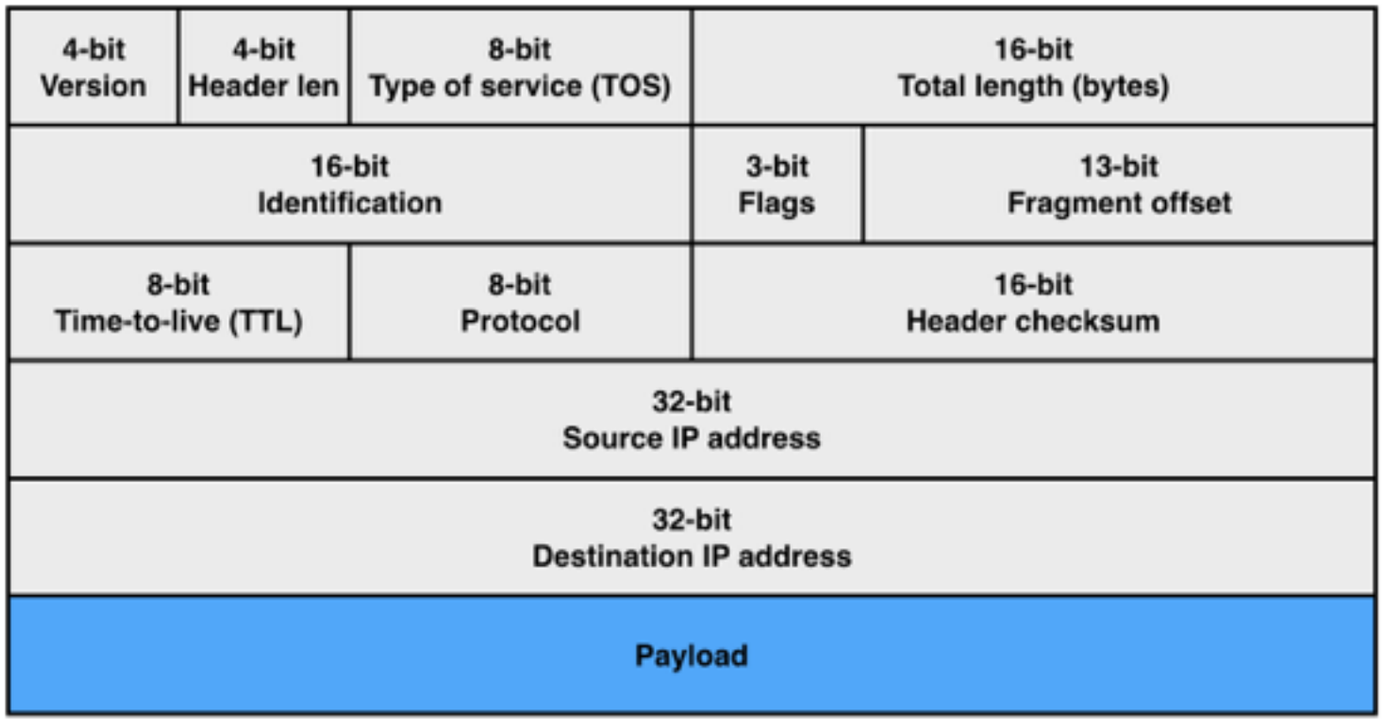
- The point (router/switch) at which traffic *enters* your network is the *ingress point*

- The point (router/switch) at which traffic *leaves* your network is the *egress point*

- You don't know who owns all IP addresses in the world, but you *do* know who in *your own network* gets what IP addresses
  - If you see a packet with a source IP address that doesn't belong to your network trying to cross your egress point, then *drop it*

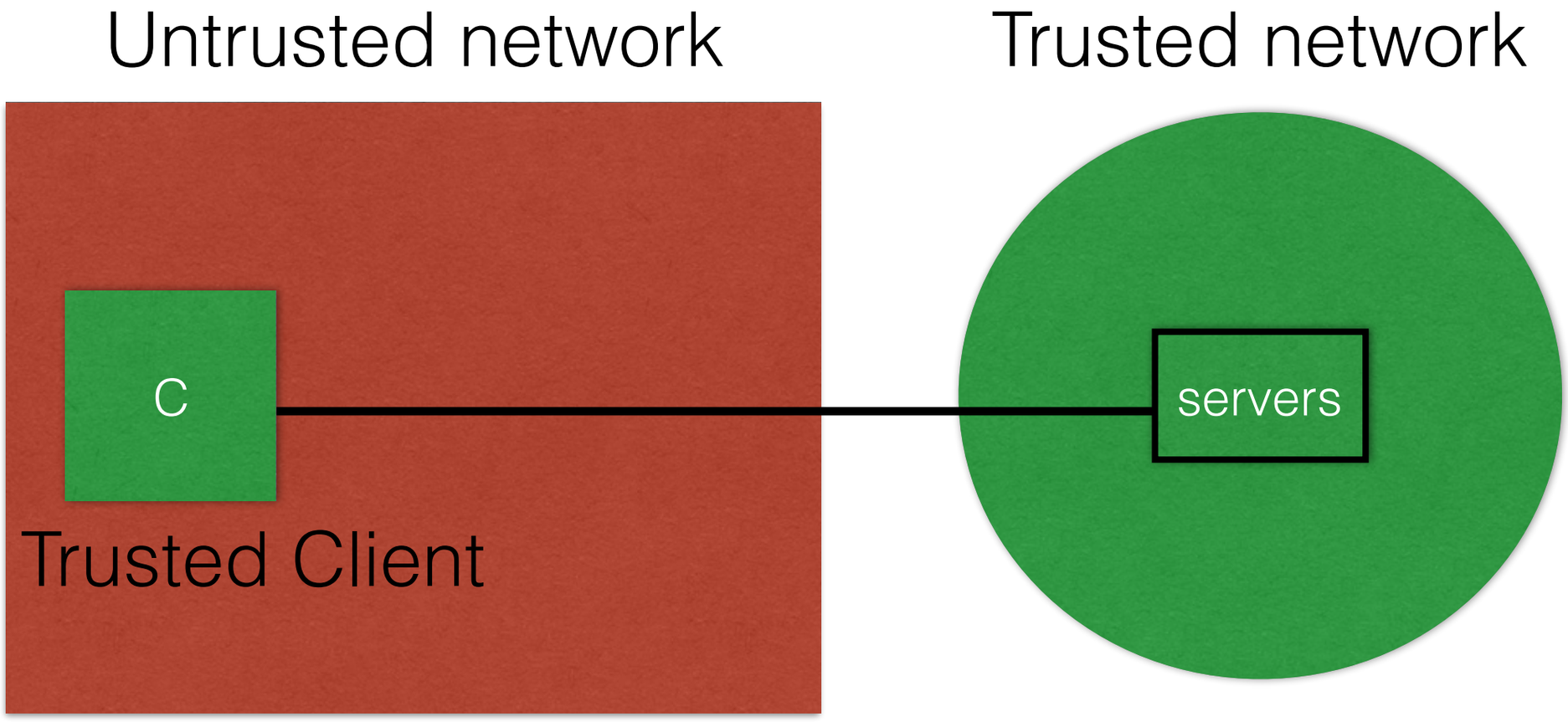
Egress filtering is not widely deployed

Eavesdropping / Tampering



- No security built into IP
- => Deploy secure IP *over IP*

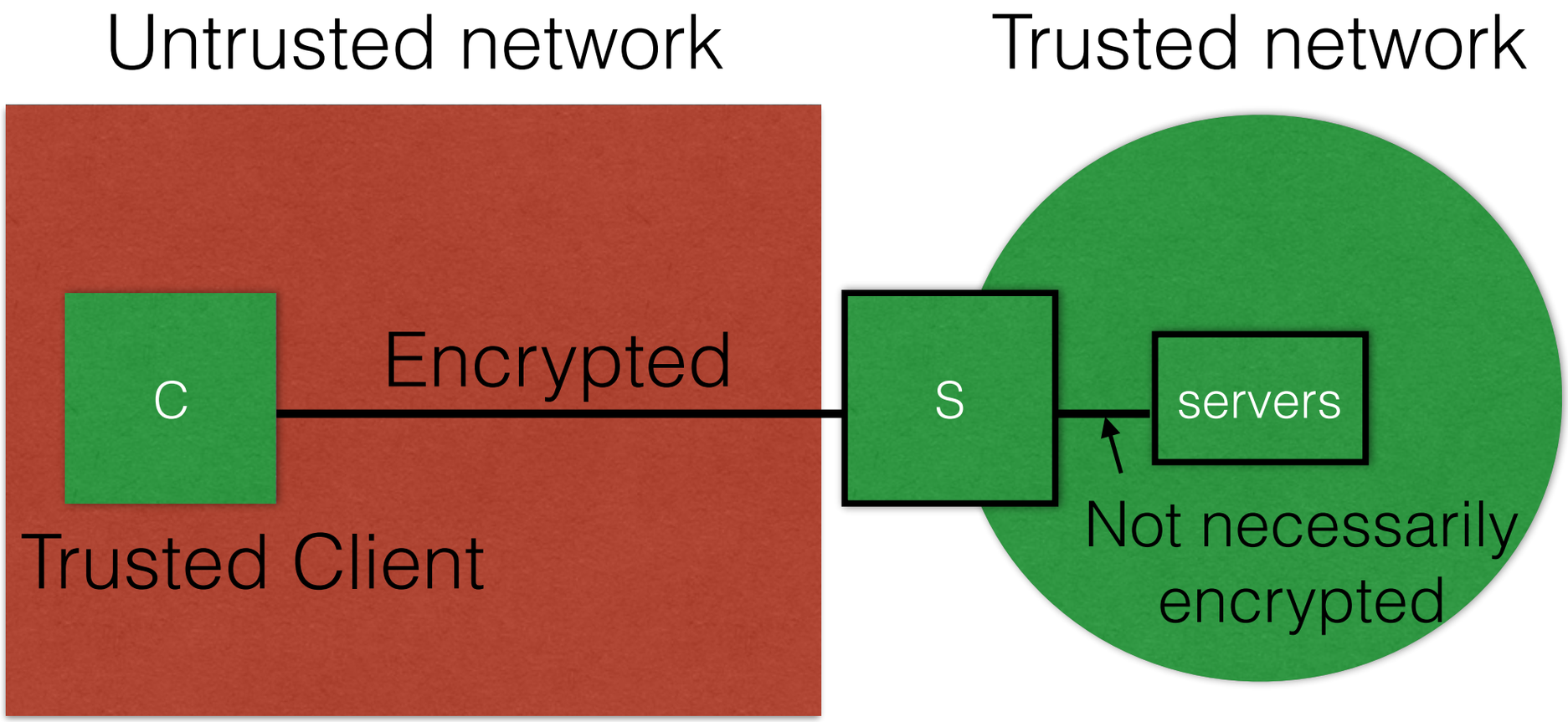
Virtual Private Networks (VPNs)



Goal: Allow the client to connect to the trusted network from within an untrusted network

Example: Connect to your company's network (for payroll, file access, etc.) while visiting a competitor's office

Virtual Private Networks (VPNs)



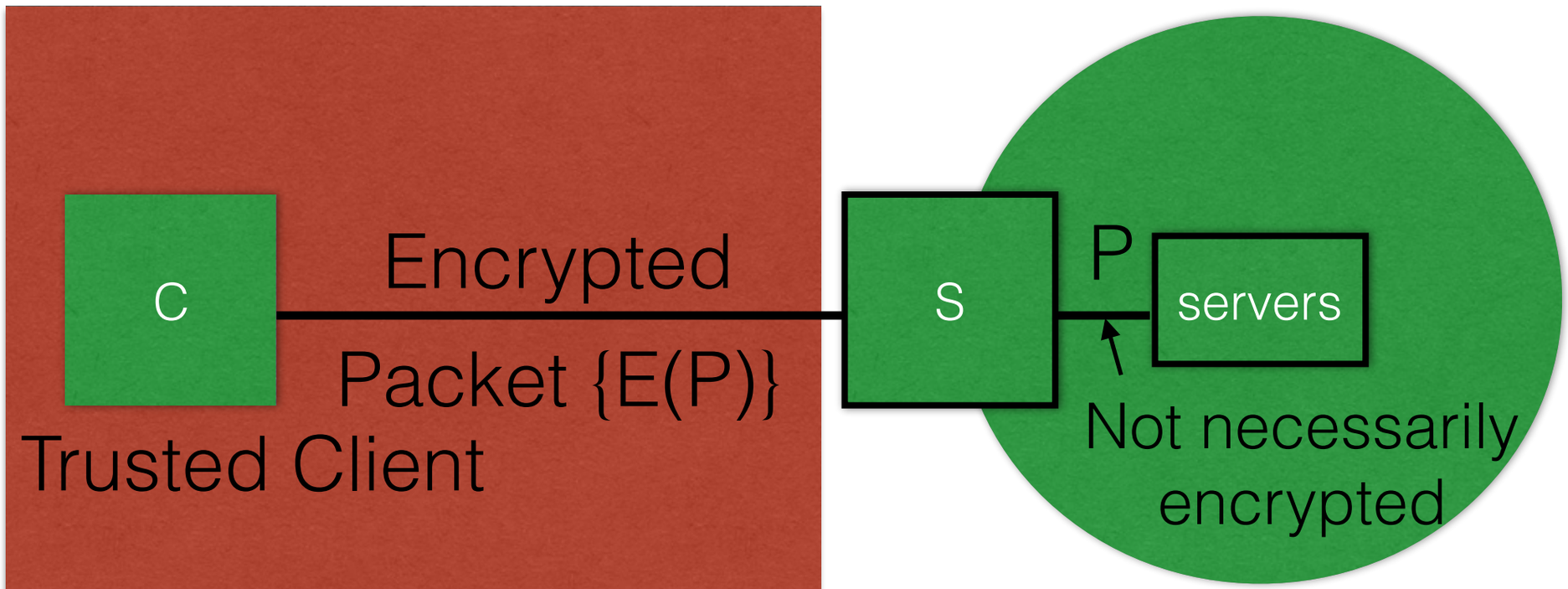
Idea: A VPN “client” and “server” together create end-to-end encryption/authentication

Predominate way of doing this: IPSec

IPSec

- Operates in a few different modes
  - Transport mode: Simply encrypt the payload but not the headers
  - Tunnel mode: Encrypt the payload *and* the headers
- But how do you encrypt the headers? How does routing work?
  - Encrypt the entire IP packet and make that the payload of another IP packet

Tunnel mode



The VPN server decrypts and then sends the payload (itself a full IP packet) as if it had just received it from the network

From the client/servers' perspective:
Looks like the client is physically connected to the network!