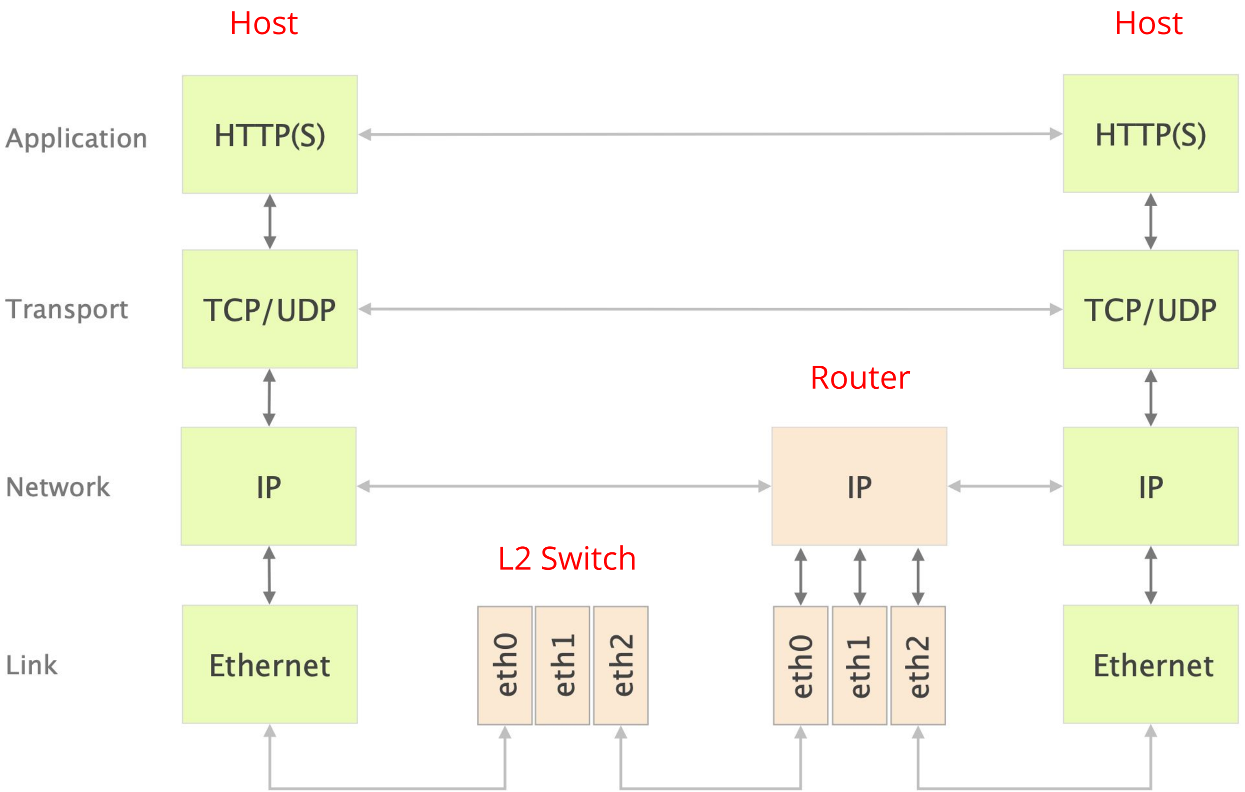
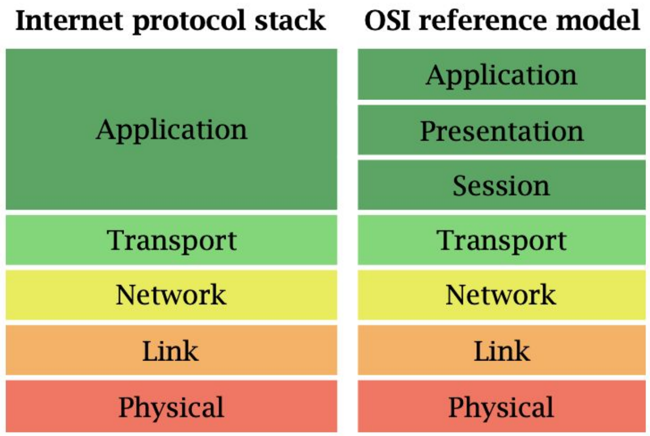
Layers are Distributed on Every Network Device



Network Protocol Stack

Communication over the Internet can be decomposed into independent layers. Find the best matching layer for the following operations/devices. You can use the already known 5-layer model.
Hint: Some of the operations could be implemented in different layers.

- Bit-to-bit transmission between two devices.
- Encryption of a message.
- A switch in a network.
- Routing path search.
- A router in a network.
- A middlebox in a network performing deep packet inspection (DPI) to find malware in Web traffic.

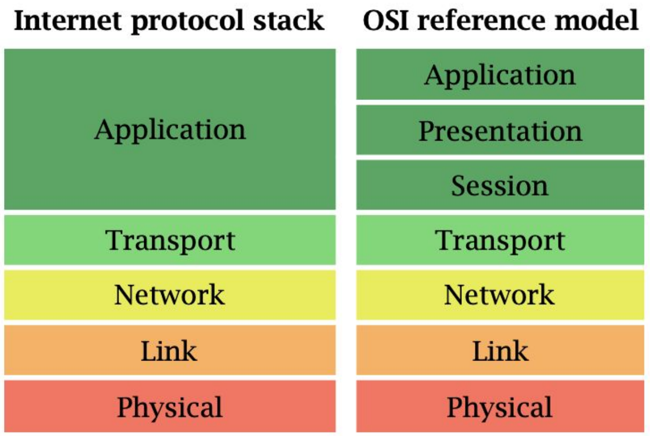


Internet communication layers: Internet protocol stack and the OSI reference model.

Network Protocol Stack

Communication over the Internet can be decomposed into independent layers. Find the best matching layer for the following operations/devices. You can use the already known 5-layer model. Hint: Some of the operations could be implemented in different layers.

- Bit-to-bit transmission between two devices.
  Solution: Physical layer
- Encryption of a message.
  Solution: Application layer
- A switch in a network.
  Solution: Link layer
- Routing path search.
  Solution: Network layer
- A router in a network.
  Solution: Network layer
- A middlebox in a network performing deep packet inspection (DPI) to find malware in Web traffic.
  Solution: To analyze the payload of packets the middlebox is operating in the application layer. Most likely, it will also use information from other layers, e.g. IP addresses from the network layer.



Internet communication layers: Internet protocol stack and the OSI reference model.

Applications

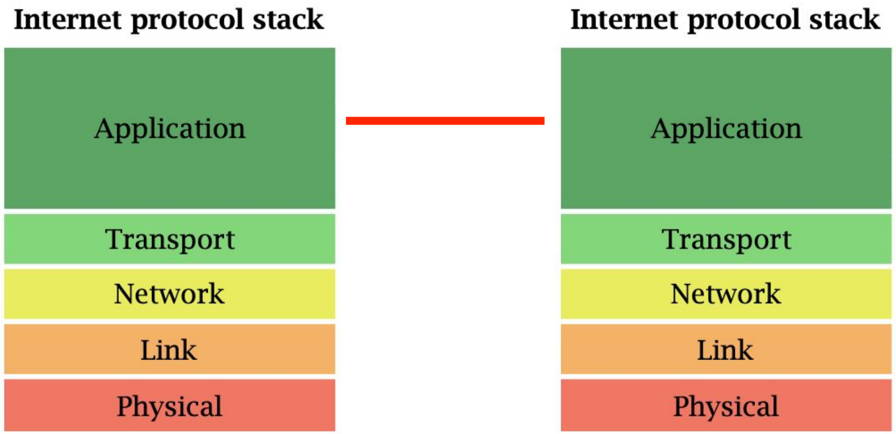
Recall that stack layers communicate “horizontally”

write programs that:

- run on (different) end systems
- communicate over network
- e.g., web server software communicates with browser software

no need to write software for network-core devices

- network-core devices do not run user applications
- applications on end systems allows for rapid app development, propagation



Client-Server Communications

- server:
  - always-on host
  - permanent IP address
  - often in data centers, for scaling

- clients:
  - contact, communicate with server
  - may be intermittently connected
  - may have dynamic IP addresses
  - do *not* communicate directly with each other
  - examples: HTTP, IMAP, FTP

Peer-to-Peer Communications

- *no* always-on server
- arbitrary end systems directly communicate
- peers request service from other peers, provide service in return to other peers
  - self scalability – new peers bring new service capacity, as well as new service demands
- peers are intermittently connected and change IP addresses
  - complex management
- example: P2P file sharing [BitTorrent]

Client-Server vs Peer-to-Peer

- Why choose one over the other?
    - Client-server is dominant for most services you use
        - Predictable performance
    - Client-server is more expensive

    - P2P leads to some peers exhausting limited resources (your home Internet connection is asymmetric - not meant to serve a lot of data)
    - P2P peers churn (they go offline without notice)
        - More complexity in managing the p2p network

Processes Talking

process: program running within a host

- within same host, two processes communicate using inter-process communication (defined by OS)
- processes in different hosts communicate by exchanging messages

clients, servers

client process: process that initiates communication

server process: process that waits to be contacted

- note: applications with P2P architectures have client processes & server processes

Sockets

- process sends/receives messages to/from its socket
- socket analogous to door
  - sending process shoves message out door
  - sending process relies on transport infrastructure on other side of door to deliver message to socket at receiving process
  - two sockets involved: one on each side