Operating System Concepts

Lecture 10: Distributed Systems

Omid Ardakanian
oardakan@ualberta.ca
University of Alberta

MWF 12:00-12:50 VVC 2 215

Today's class

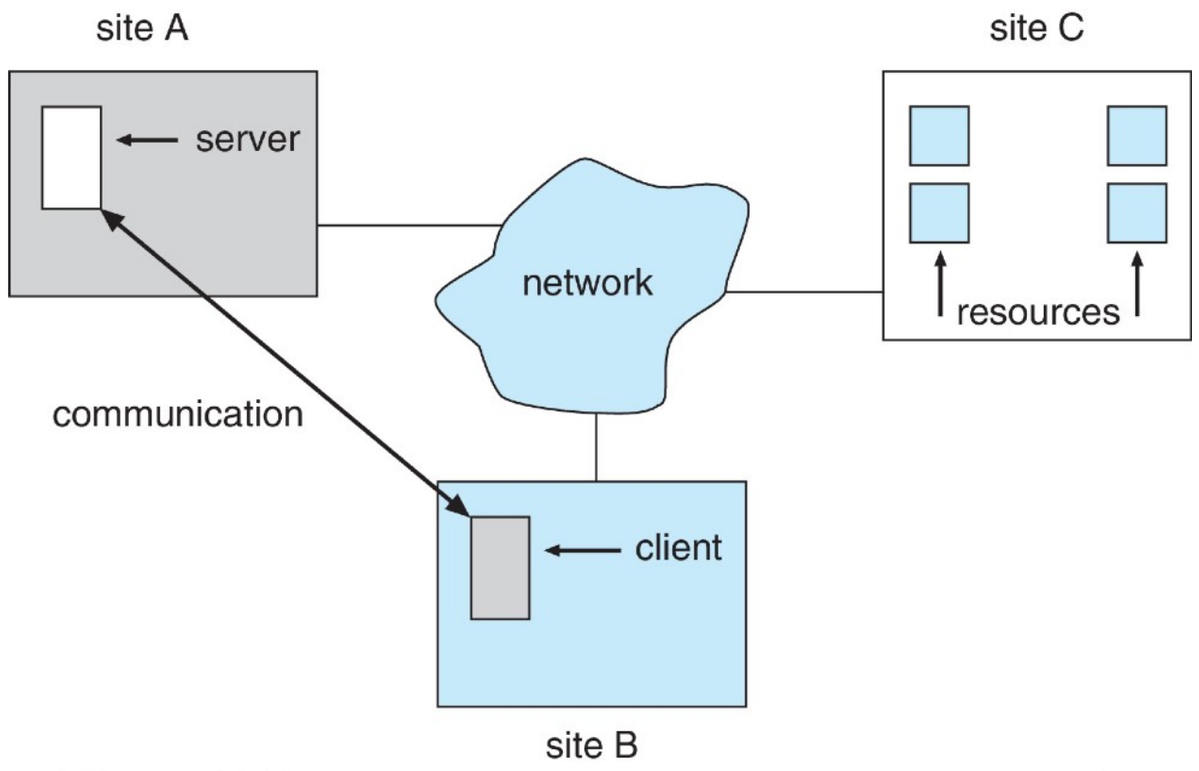
- Distributed systems
  - Motivation
  - Design issues
- Communication basics
  - Network structure
  - Network protocol

Distributed systems

- Definition: a set of physically separate, **loosely coupled** nodes connected by a communication network (high-speed buses or the Internet)
  - each node has an independent OS along with its own memory and other resources
  - nodes are variously called processors, machines, computers, and hosts

Distributed systems

- Definition: a set of physically separate, **loosely coupled** nodes connected by a communication network (high-speed buses or the Internet)
  - each node has an independent OS along with its own memory and other resources
  - nodes are variously called processors, machines, computers, and hosts
- communication over a network occurs through **message passing**
  - inherently unreliable: messages sometimes do not reach their destination



Distributed systems

- Definition: a set of physically separate, **loosely coupled** nodes connected by a communication network (high-speed buses or the Internet)
  - each node has an independent OS along with its own memory and other resources
  - nodes are variously called processors, machines, computers, and hosts
- communication over a network occurs through **message passing**
  - inherently unreliable: messages sometimes do not reach their destination
- nodes may exist in a client-server, peer-to-peer, or hybrid configuration
  - in client server model the server has a resource that the client (at a different site) wants to use
  - in peer-to-peer model each node shares equal responsibilities and can act as both clients and servers

Distributed systems

- Definition: a set of physically separate, **loosely coupled** nodes connected by a communication network (high-speed buses or the Internet)
  - each node has an independent OS along with its own memory and other resources
  - nodes are variously called processors, machines, computers, and hosts
- communication over a network occurs through **message passing**
  - inherently unreliable: messages sometimes do not reach their destination
- nodes may exist in a client-server, peer-to-peer, or hybrid configuration
  - in client server model the server has a resource that the client (at a different site) wants to use
  - in peer-to-peer model each node shares equal responsibilities and can act as both clients and servers
- nearly all systems today are distributed in some way
  - complex services are built from a large collection of machines cooperating to provide the particular service of the site
  - networks hook them together

Why distributed systems are popular?

- resource sharing
  - resources need not be replicated at each processor, e.g., shared files
  - expensive and/or scarce resources can be shared, e.g., printers, GPUs
  - user can use specialized and licensed software on another machine

Why distributed systems are popular?

- resource sharing
  - resources need not be replicated at each processor, e.g., shared files
  - expensive and/or scarce resources can be shared, e.g., printers, GPUs
  - user can use specialized and licensed software on another machine
- computational speedup
  - distribute tasks among various sites to run concurrently
    - they do not compete with each other for a single CPU core
  - load balancing (moving jobs to more lightly-loaded sites) would help
  - coordination and communication are necessary yet quite challenging

Why distributed systems are popular?

- resource sharing
  - resources need not be replicated at each processor, e.g., shared files
  - expensive and/or scarce resources can be shared, e.g., printers, GPUs
  - user can use specialized and licensed software on another machine
- computational speedup
  - distribute tasks among various sites to run concurrently
    - they do not compete with each other for a single CPU core
  - load balancing (moving jobs to more lightly-loaded sites) would help
  - coordination and communication are necessary yet quite challenging
- reliability and availability
  - resource replication yields fault tolerance (no single point of failure)
    - performance will degrade but system remains operational
  - must be able to detect and recover from site failure (function transfer, reintegrate failed sites, etc.)

Design issues of distributed systems

- robustness — the system should withstand failures
  - including failure of a link, failure of a site, and loss of messages
  - a fault-tolerant system can tolerate a certain level of failure
  - the degree of fault tolerance depends on the system design and the specific fault
  - detecting hardware failure is difficult (can use a heartbeat protocol)

Design issues of distributed systems

- robustness — the system should withstand failures
  - including failure of a link, failure of a site, and loss of messages
  - a fault-tolerant system can tolerate a certain level of failure
  - the degree of fault tolerance depends on the system design and the specific fault
  - detecting hardware failure is difficult (can use a heartbeat protocol)
- transparency to user — in terms of where files are stored and user mobility
  - the distributed system should appear as a conventional, centralized system to the user
    - user interface should not distinguish between local and remote resources
    - user mobility allows users to log into any machine in the environment and see his/her environment

Design issues of distributed systems

- scalability — the system should react gracefully to increased load (by accepting new resources)
  - data **compression** and **deduplication** can cut down on storage and network resources used

Design issues of distributed systems

- scalability — the system should react gracefully to increased load (by accepting new resources)
  - data **compression** and **deduplication** can cut down on storage and network resources used
- consistency — the cached copy of the data must be consistent with the master copy
  - consistency checks must be performed periodically
  - nodes must keep track of the cached data to detect potential inconsistencies

Network OS

- all general-purpose operating systems today are network operating systems
  - allow the user to login remotely and transfer files
  - users are aware of multiplicity of machines

Network OS

- all general-purpose operating systems today are network operating systems
  - allow the user to login remotely and transfer files
  - users are aware of multiplicity of machines
- access to resources of various machines is done explicitly by
  - remote login via Secure Shell (SSH) protocol
    `ssh ohaton.cs.ualberta.edu`
  - remote file transfer via File Transfer Protocol (FTP) and Secure File Transfer Protocol (SFTP)
    `get, put, ls, cd` commands are used for locating and transferring files

Network OS

- all general-purpose operating systems today are network operating systems
  - allow the user to login remotely and transfer files
  - users are aware of multiplicity of machines
- access to resources of various machines is done explicitly by
  - remote login via Secure Shell (SSH) protocol
    `ssh ohaton.cs.ualberta.edu`
  - remote file transfer via File Transfer Protocol (FTP) and Secure File Transfer Protocol (SFTP)
    `get, put, ls, cd` commands are used for locating and transferring files
- users must change paradigms – establish a **session**, give network-based commands (might be different from the normal OS commands)
  - session is a complete round of communication

Distributed OS

- users are not aware of multiplicity of machines
  - access to remote resources similar to access to local resources
  - data migration and process migration are handled seamlessly by the distributed operating system
    - data translation may be required (when they have different character-code representations, different order of bits)

Distributed OS

- users are not aware of multiplicity of machines
  - access to remote resources similar to access to local resources
  - data migration and process migration are handled seamlessly by the distributed operating system
    - data translation may be required (when they have different character-code representations, different order of bits)
- data migration
  - transfer data by transferring the entire file, or transferring only those portions of the file necessary for the immediate task

Distributed OS

- users are not aware of multiplicity of machines
  - access to remote resources similar to access to local resources
  - data migration and process migration are handled seamlessly by the distributed operating system
    - data translation may be required (when they have different character-code representations, different order of bits)
- data migration
  - transfer data by transferring the entire file, or transferring only those portions of the file necessary for the immediate task
- computation migration
  - transfer the computation, rather than the data, across the system via remote procedure calls (RPCs) for example

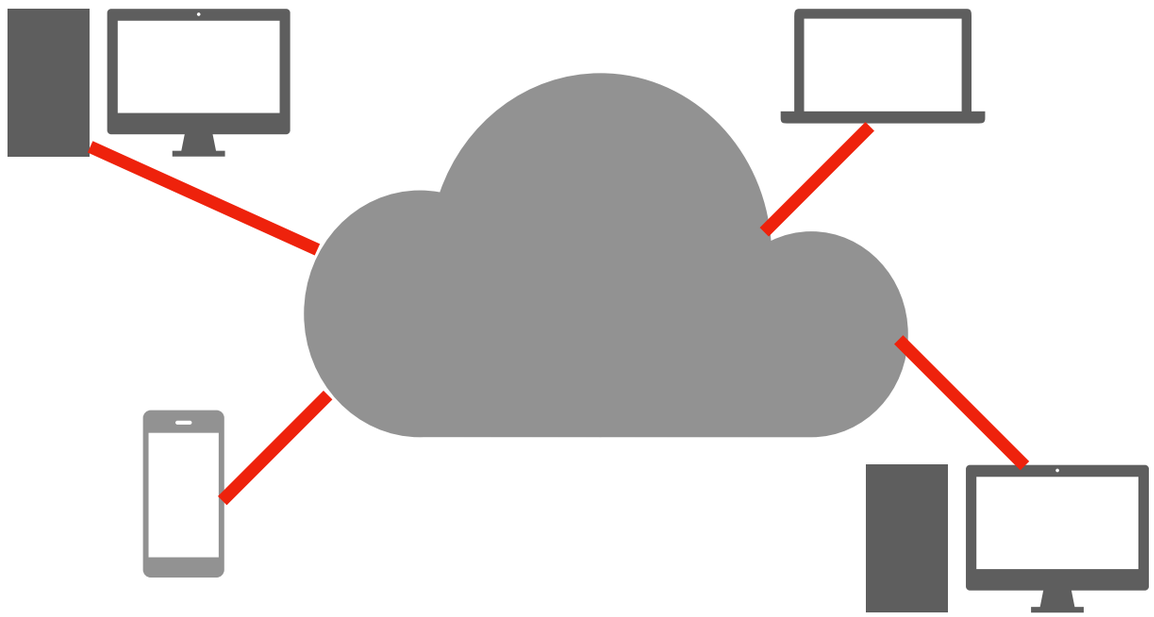
Distributed OS

- users are not aware of multiplicity of machines
  - access to remote resources similar to access to local resources
  - data migration and process migration are handled seamlessly by the distributed operating system
    - data translation may be required (when they have different character-code representations, different order of bits)
- data migration
  - transfer data by transferring the entire file, or transferring only those portions of the file necessary for the immediate task
- computation migration
  - transfer the computation, rather than the data, across the system via remote procedure calls (RPCs) for example
- process migration
  - execute the entire process (computation and data), or parts of it, at different sites

Communication Basics

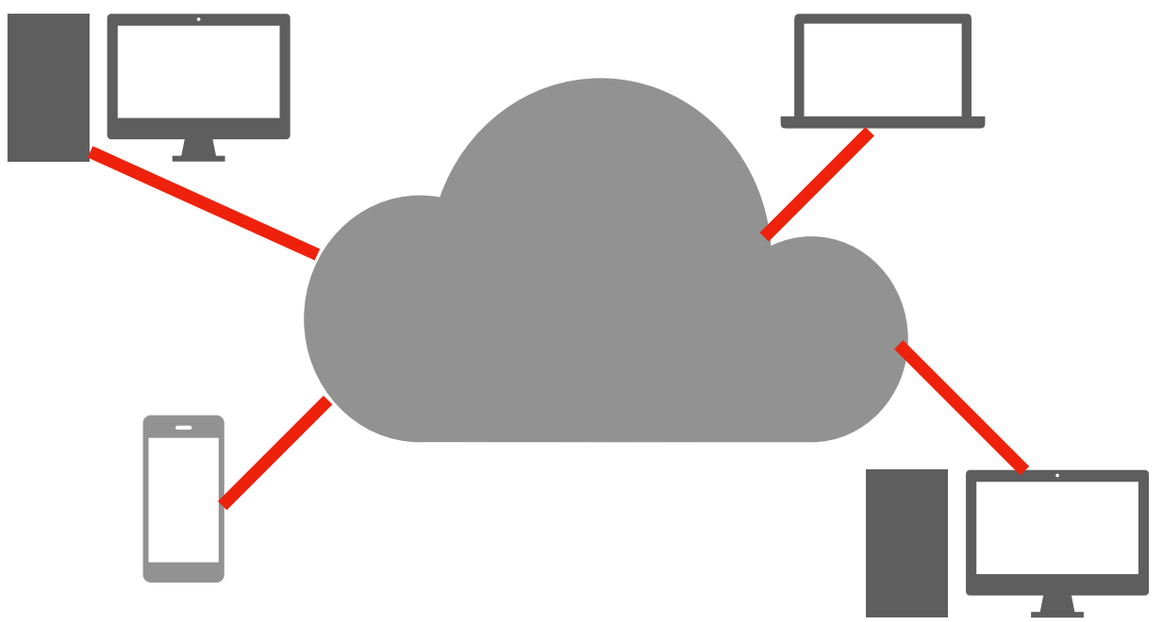
Definitions

- network: one or more communication links allowing two computers to communicate
  - each computer has a network address



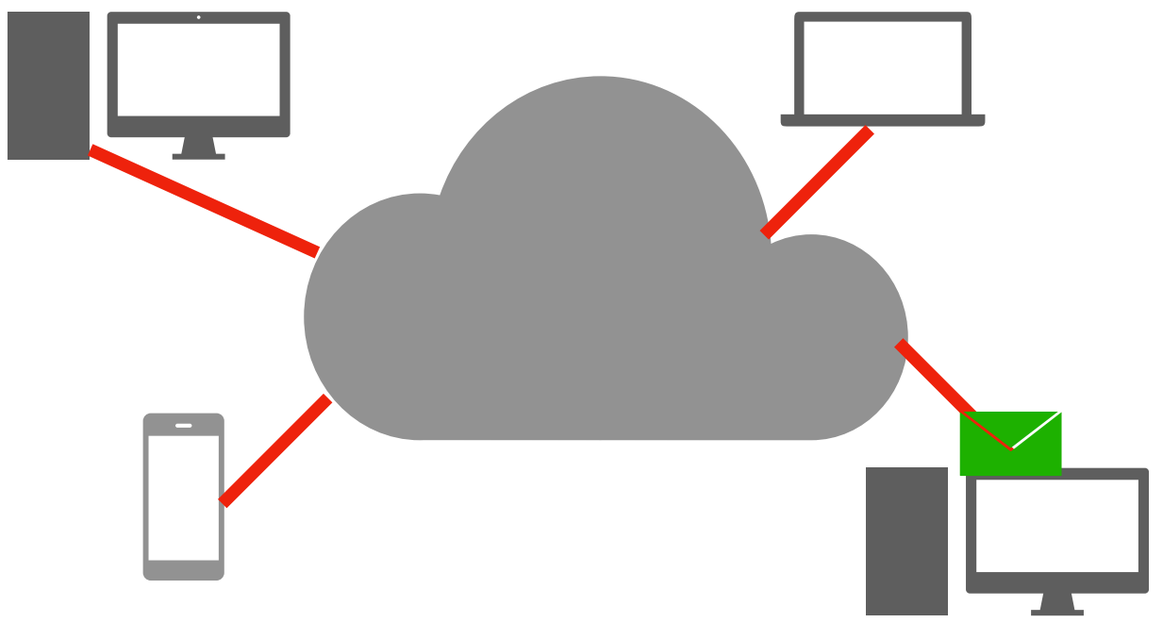
Definitions

- network: one or more communication links allowing two computers to communicate
  - each computer has a network address
- network interface: computer's interface to the network
  - each network interface card (NIC) has a unique hardware address



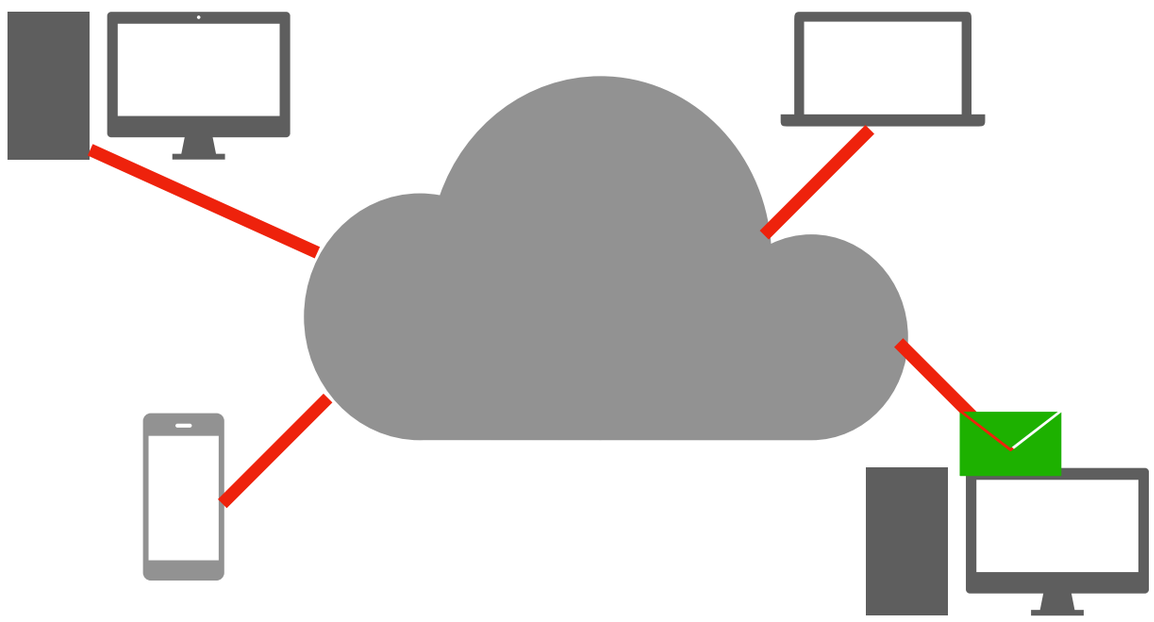
Definitions

- network: one or more communication links allowing two computers to communicate
  - each computer has a network address
- network interface: computer’s interface to the network
  - each network interface card (NIC) has a unique hardware address
- packet: network's basic transmission unit; a sequence of bits



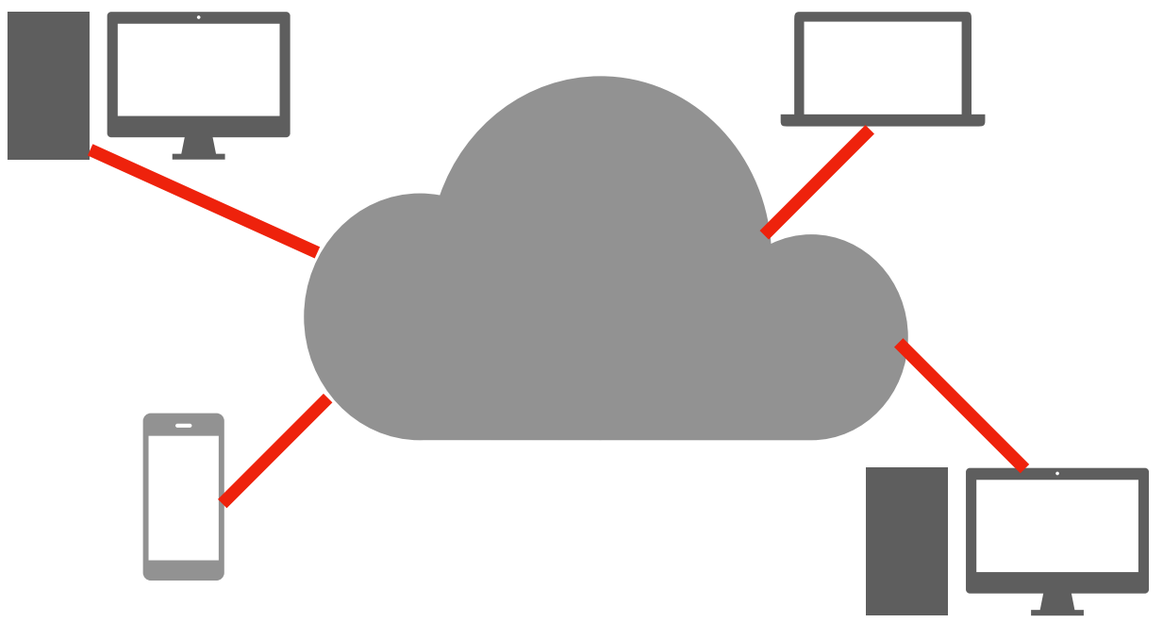
Definitions

- network: one or more communication links allowing two computers to communicate
  - each computer has a network address
- network interface: computer’s interface to the network
  - each network interface card (NIC) has a unique hardware address
- packet: network's basic transmission unit; a sequence of bits



Definitions

- network: one or more communication links allowing two computers to communicate
  - each computer has a network address
- network interface: computer's interface to the network
  - each network interface card (NIC) has a unique hardware address
- packet: network's basic transmission unit; a sequence of bits
- protocol: a set of rules for communication that are agreed to by all parties



Network structure

- Local Area Network (LAN) covers a small geographical area (e.g., a building)
  - consists of multiple computers, peripherals (printers, storage arrays), and routers providing access to other networks
  - needs to be fast and reliable

Network structure

- Local Area Network (LAN) covers a small geographical area (e.g., a building)
  - consists of multiple computers, peripherals (printers, storage arrays), and routers providing access to other networks
  - needs to be fast and reliable
- Ethernet and/or Wireless (WiFi) most common way to construct LANs
  - Ethernet defined by IEEE 802.3 standard with speeds typically varying from 10Mbps to over 10Gbps
    - everyone taps into a single wire
    - everyone gets packets and discards them if it is not the target
  - WiFi defined by IEEE 802.11 standard with speeds typically varying from 11Mbps to over 400Mbps

Network structure

- Wide Area Network links geographically separated sites (across the country or planet)
  - WAN is slower and less reliable than LAN
  - Internet WAN enables hosts world wide to communicate

Network structure

- Wide Area Network links geographically separated sites (across the country or planet)
  - WAN is slower and less reliable than LAN
  - Internet WAN enables hosts world wide to communicate
- point-to-point connections via links
  - telephone lines, leased (dedicated data) lines, optical cable, microwave links, radio waves, and satellite channels
  - speeds vary
    - many backbone providers have speeds at 40-100Gbps
    - local Internet Service Providers (ISPs) may be slower

Network structure

- Wide Area Network links geographically separated sites (across the country or planet)
  - WAN is slower and less reliable than LAN
  - Internet WAN enables hosts world wide to communicate
- point-to-point connections via links
  - telephone lines, leased (dedicated data) lines, optical cable, microwave links, radio waves, and satellite channels
  - speeds vary
    - many backbone providers have speeds at 40-100Gbps
    - local Internet Service Providers (ISPs) may be slower
- WANs and LANs interconnect

Principles of network communication

- data sent through the network is chopped into **packets**

Principles of network communication

- data sent through the network is chopped into **packets**
- computers at the switching points control the packet flow
  - analogy: cars/road/police - packets/network link/computer

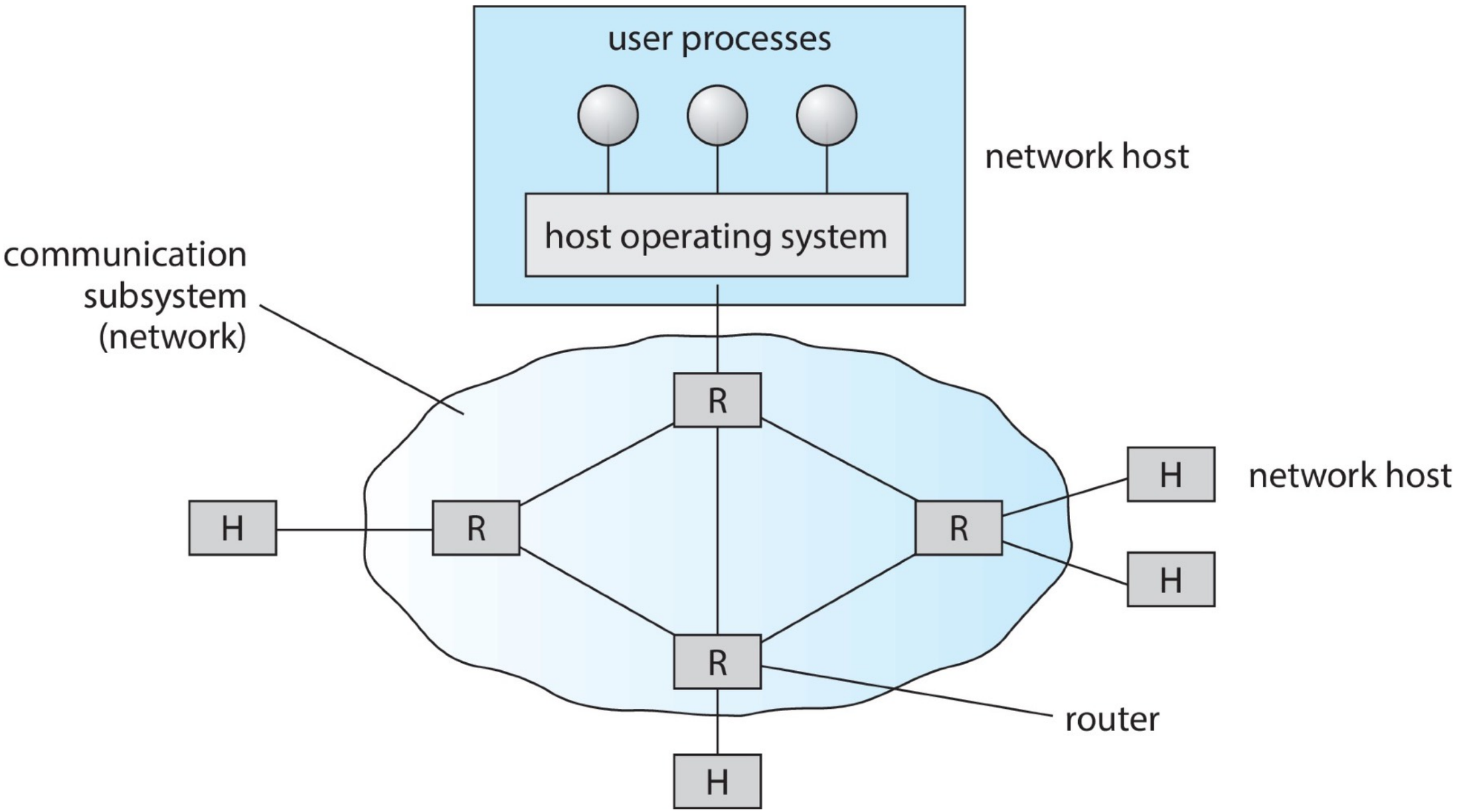
Principles of network communication

- data sent through the network is chopped into **packets**
- computers at the switching points control the packet flow
  - analogy: cars/road/police - packets/network link/computer
- shared resources can lead to contention (just like traffic jams)

Principles of network communication

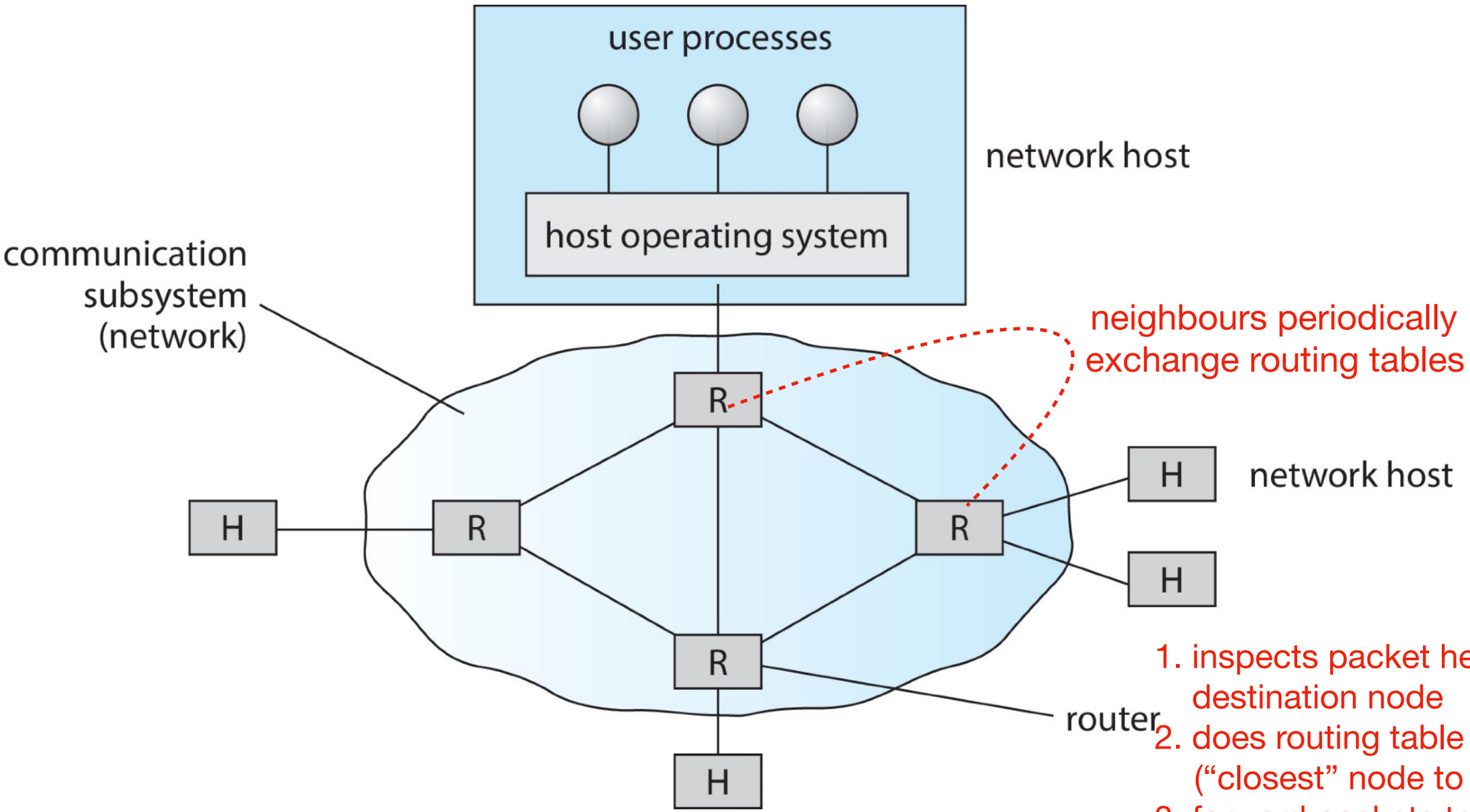
- data sent through the network is chopped into **packets**
- computers at the switching points control the packet flow
  - analogy: cars/road/police - packets/network link/computer
- shared resources can lead to contention (just like traffic jams)
- the destination computer is interrupted when a packet arrives

How a packet is delivered to destination if the hosts are located on separate networks?



WAN is implemented via routers to direct traffic from one network to another

How a packet is delivered to destination if the hosts are located on separate networks?



WAN is implemented via routers to direct traffic from one network to another

How to identify the destination system/process?

- a process on a remote system is identified by `<host-name, identifier>` pair

How to identify the destination system/process?

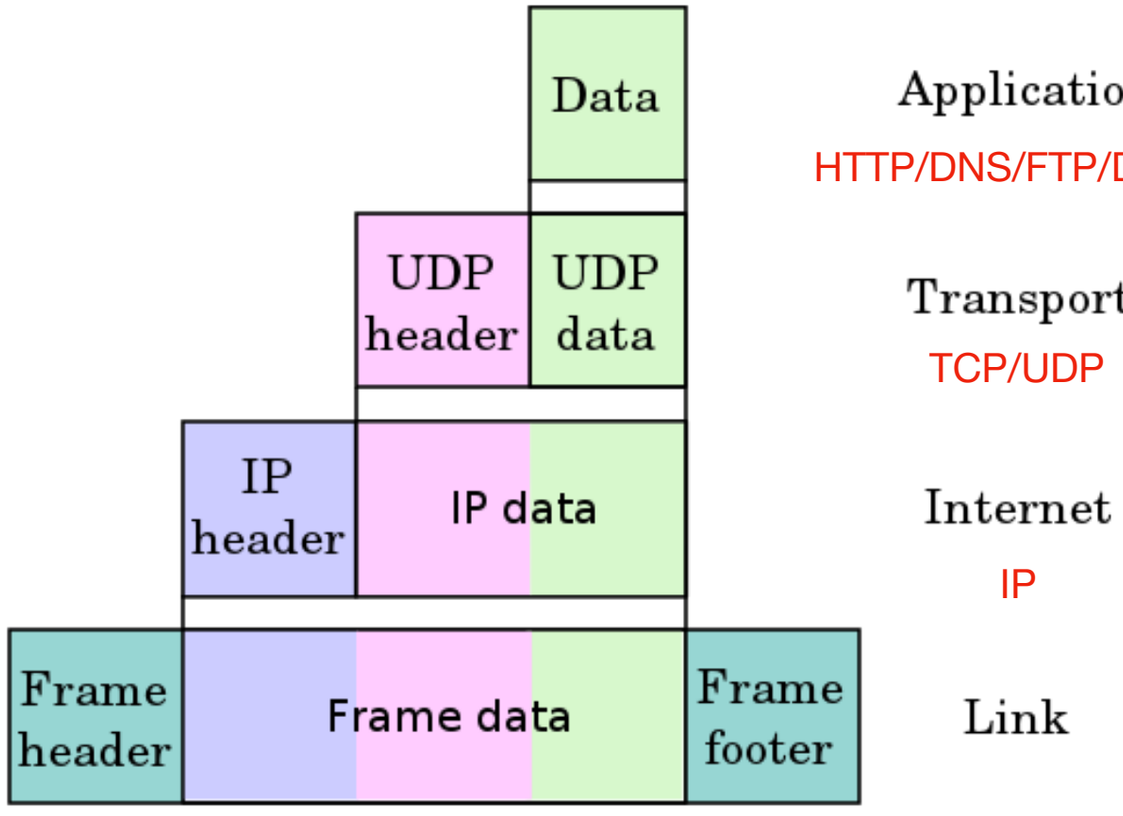
- a process on a remote system is identified by `<host-name, identifier>` pair
- each process in a given system has a unique name (process-id)

How to identify the destination system/process?

- a process on a remote system is identified by `<host-name, identifier>` pair
- each process in a given system has a unique name (process-id)
- each computer system in the network has a unique name
  - Domain Name System (DNS): a global distributed database system for resolving hostname-IP address mappings
    - 32-bit (IPv4) address: e.g., 129.128.5.180
    - humane readable names: e.g., gpu.srv.ualberta.ca

TCP/IP protocol stack (Internet protocol suite)

enables end-to-end communications using functions organized into four abstraction layers

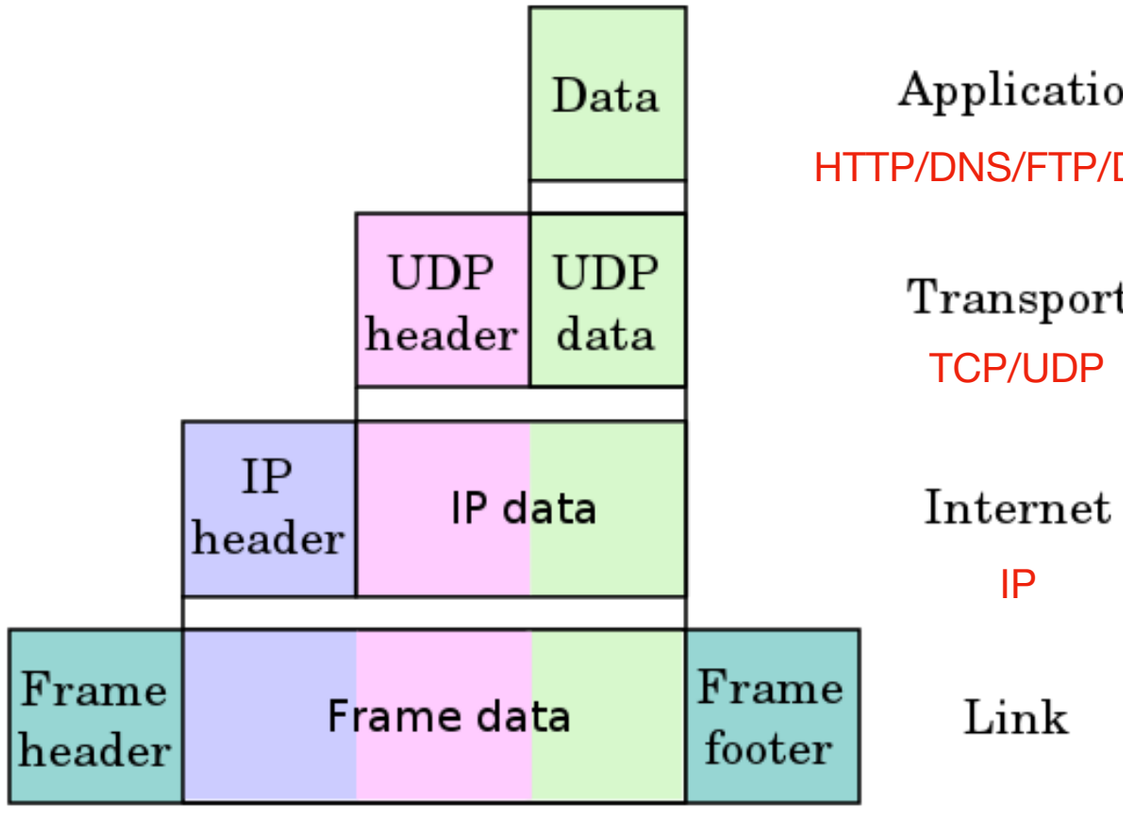


routers only have the last two layers

TCP/IP protocol stack (Internet protocol suite)

enables end-to-end communications using functions organized into four abstraction layers

- Application layer
  - provides process-to-process data exchange for applications

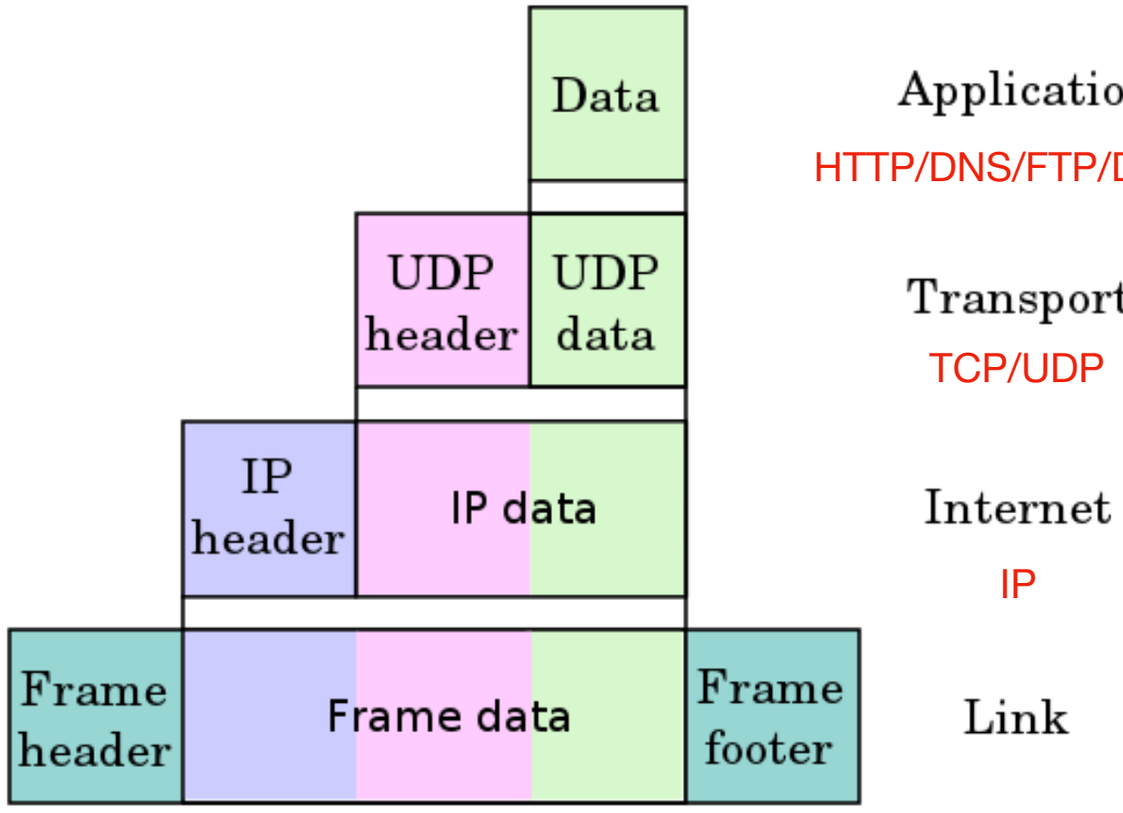


routers only have the last two layers

TCP/IP protocol stack (Internet protocol suite)

enables end-to-end communications using functions organized into four abstraction layers

- Application layer
  - provides process-to-process data exchange for applications
- Transport layer
  - responsible for message transfer between hosts, including partitioning messages into TCP/UDP packets, maintaining packet order, and controlling flow

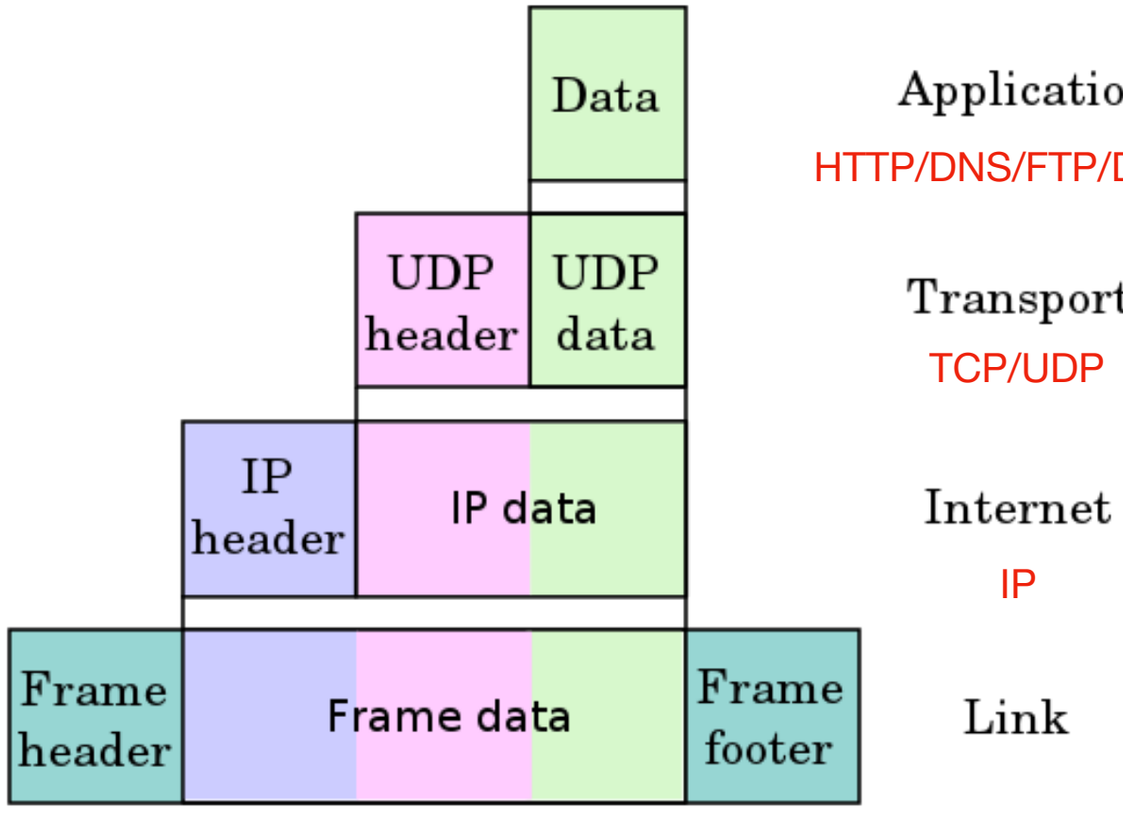


routers only have the last two layers

TCP/IP protocol stack (Internet protocol suite)

enables end-to-end communications using functions organized into four abstraction layers

- Application layer
  - provides process-to-process data exchange for applications
- Transport layer
  - responsible for message transfer between hosts, including partitioning messages into TCP/UDP packets, maintaining packet order, and controlling flow
- Internet layer
  - responsible for routing IP packets through the network, and encoding/decoding addresses

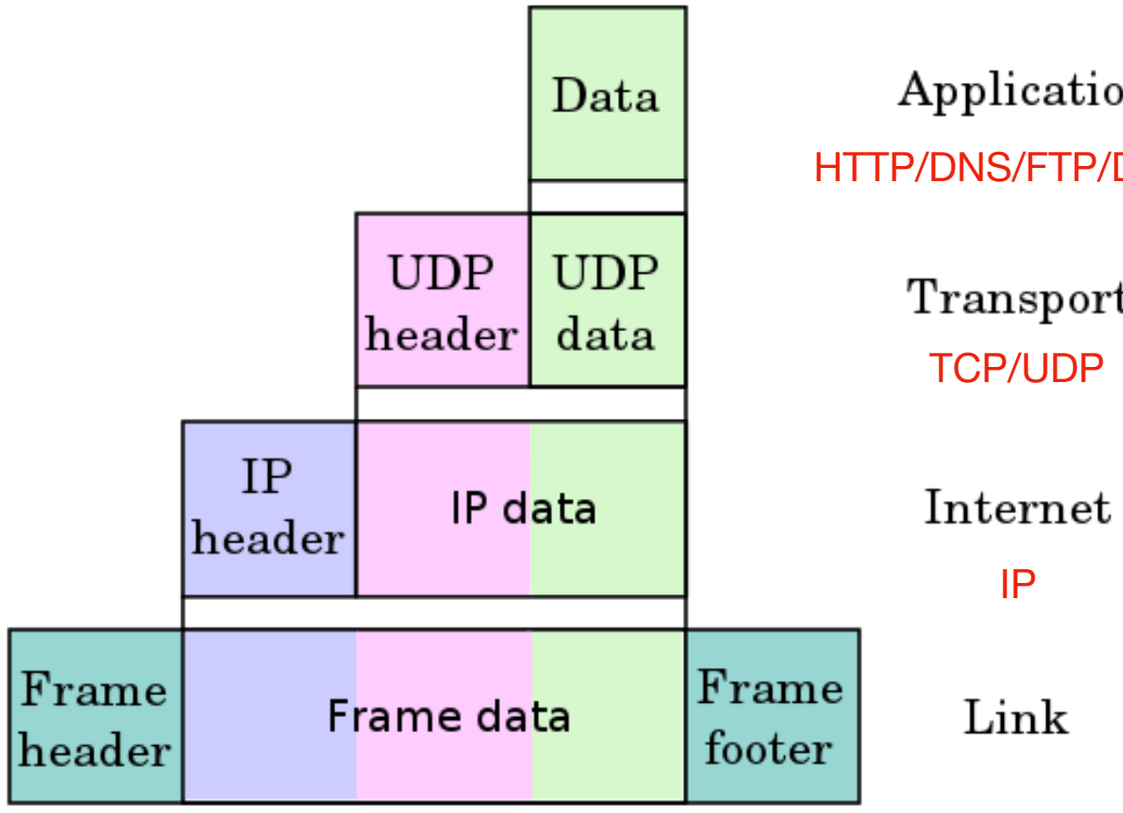


routers only have the last two layers

TCP/IP protocol stack (Internet protocol suite)

enables end-to-end communications using functions organized into four abstraction layers

- Application layer
  - provides process-to-process data exchange for applications
- Transport layer
  - responsible for message transfer between hosts, including partitioning messages into TCP/UDP packets, maintaining packet order, and controlling flow
- Internet layer
  - responsible for routing IP packets through the network, and encoding/decoding addresses
- Link layer
  - handles the frames, or fixed-length parts of packets, including transmission over over a physical medium, and any error detection and recovery that occurred in the physical layer



routers only have the last two layers

Media Access Control (MAC) address

How does a packet move from sender (host or router) to receiver on the same LAN?

Media Access Control (MAC) address

How does a packet move from sender (host or router) to receiver on the same LAN?

- every Ethernet/WiFi device has a unique medium access control (MAC) address

Media Access Control (MAC) address

How does a packet move from sender (host or router) to receiver on the same LAN?

- every Ethernet/WiFi device has a unique medium access control (MAC) address
- if a system wants to send data to another system, it needs to perform the IP to MAC address mapping
  - using **address resolution protocol (ARP)**
  - run `arp -a` to see the content of your arp table

Internet Protocol (IP) address

- every host has a name and an associated **IP address**
  - 32-bit (IPv4) address: approximately 4.3 billion addresses
  - 128-bit (IPv6) address: approximately 3.4×10^{38} addresses
  - a special address is reserved for local host: 127.0.0.1

Internet Protocol (IP) address

- every host has a name and an associated **IP address**
  - 32-bit (IPv4) address: approximately 4.3 billion addresses
  - 128-bit (IPv6) address: approximately 3.4×10^{38} addresses
  - a special address is reserved for local host: 127.0.0.1
- the sending system checks routing tables and locates a router to send packet

Internet Protocol (IP) address

- every host has a name and an associated **IP address**
  - 32-bit (IPv4) address: approximately 4.3 billion addresses
  - 128-bit (IPv6) address: approximately 3.4×10^{38} addresses
  - a special address is reserved for local host: 127.0.0.1

- the sending system checks routing tables and locates a router to send packet
- each router uses the network part of host-id to determine where to transfer packet

Internet Protocol (IP) address

- every host has a name and an associated **IP address**
  - 32-bit (IPv4) address: approximately 4.3 billion addresses
  - 128-bit (IPv6) address: approximately 3.4×10^{38} addresses
  - a special address is reserved for local host: 127.0.0.1
- the sending system checks routing tables and locates a router to send packet
- each router uses the network part of host-id to determine where to transfer packet
- the destination system receives the packet
  - it may be complete message, or it may need to be reassembled into larger message spanning multiple packets

Transport layer address

- once a host with a specific IP address receives a packet, it must somehow pass it to the correct waiting process

Transport layer address

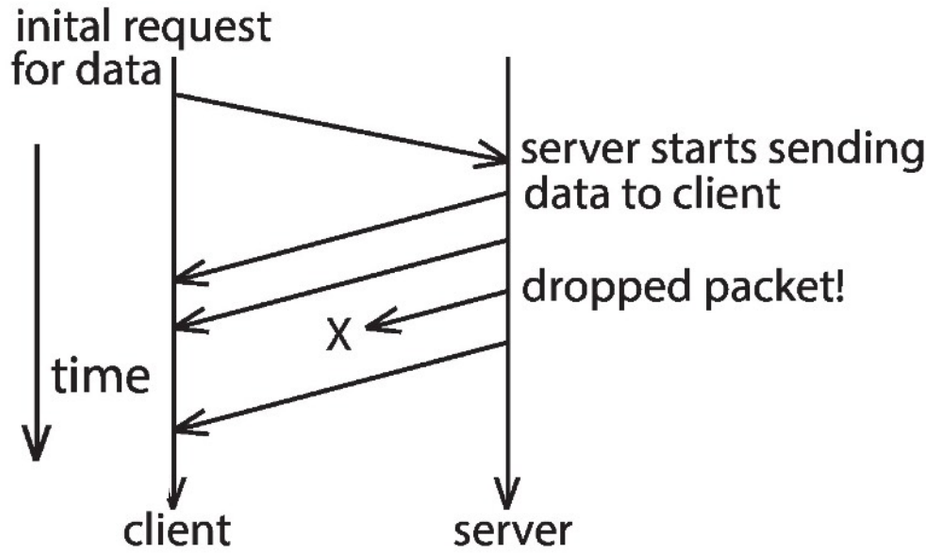
- once a host with a specific IP address receives a packet, it must somehow pass it to the correct waiting process
- transport protocols, TCP and UDP, identify receiving and sending processes through the use of a port number (16 bits)
  - allows a host with a single IP address to have multiple processes sending and receiving packets
  - system or well-known ports are used to implement **standard** services: 0 through 1023
    - FTP – port 21/TCP; ssh – port 22/TCP; SMTP – port 25/TCP; HTTP – port 80/TCP; NTP – port 123/UDP
  - registered ports: 1024 through 49151
    - registered for specific services
  - dynamic or private ports: 49152 to 65535
    - available for use by any application communicating using TCP/UDP

Transport layer address

- once a host with a specific IP address receives a packet, it must somehow pass it to the correct waiting process
- transport protocols, TCP and UDP, identify receiving and sending processes through the use of a port number (16 bits)
  - allows a host with a single IP address to have multiple processes sending and receiving packets
  - system or well-known ports are used to implement **standard** services: 0 through 1023
    - FTP – port 21/TCP; ssh – port 22/TCP; SMTP – port 25/TCP; HTTP – port 80/TCP; NTP – port 123/UDP
  - registered ports: 1024 through 49151
    - registered for specific services
  - dynamic or private ports: 49152 to 65535
    - available for use by any application communicating using TCP/UDP
- transport protocol can be simple or can add reliability to network packet stream

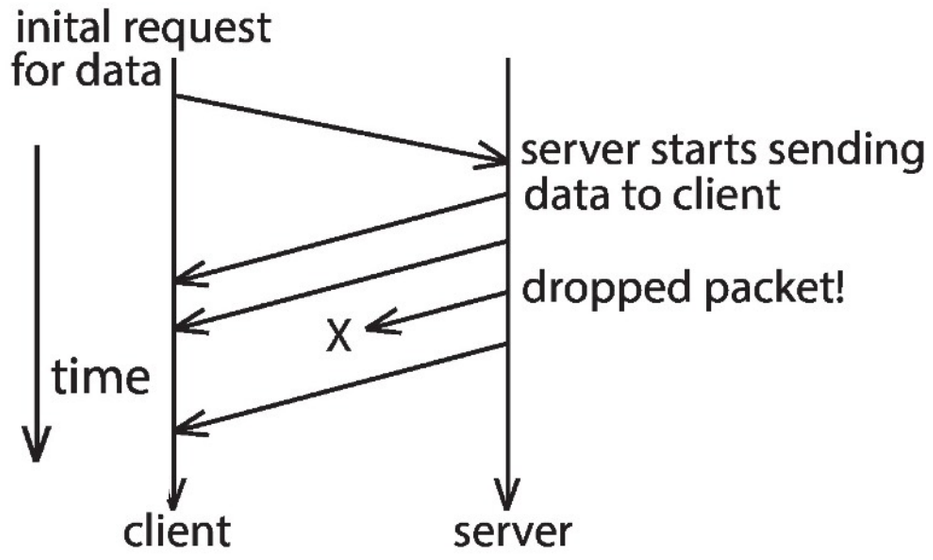
User Datagram Protocol (UDP)

- UDP packets are also called **datagrams**
  - **messages** up to some maximum size



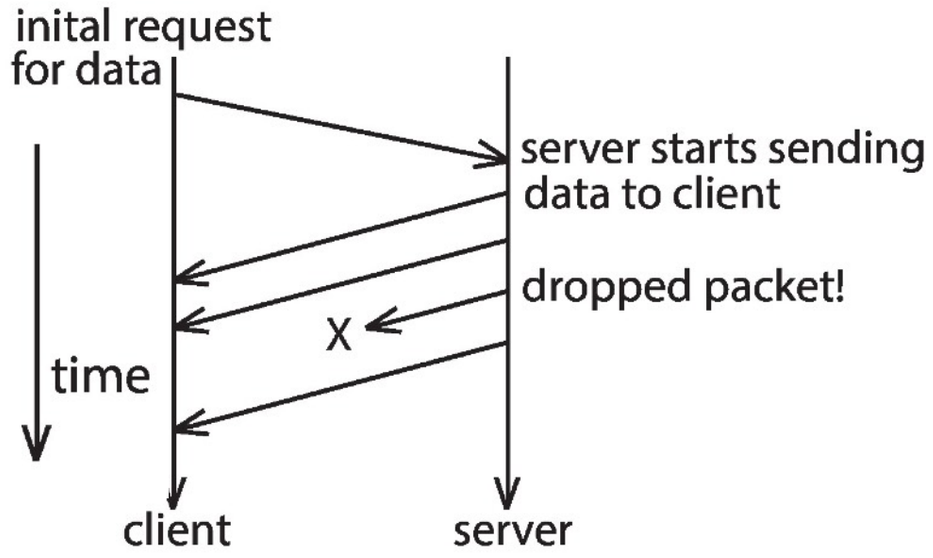
User Datagram Protocol (UDP)

- UDP packets are also called **datagrams**
  - **messages** up to some maximum size
- UDP is unreliable
  - bare-bones extension to IP with the addition of port number
  - packets may be lost or received out-out-order



User Datagram Protocol (UDP)

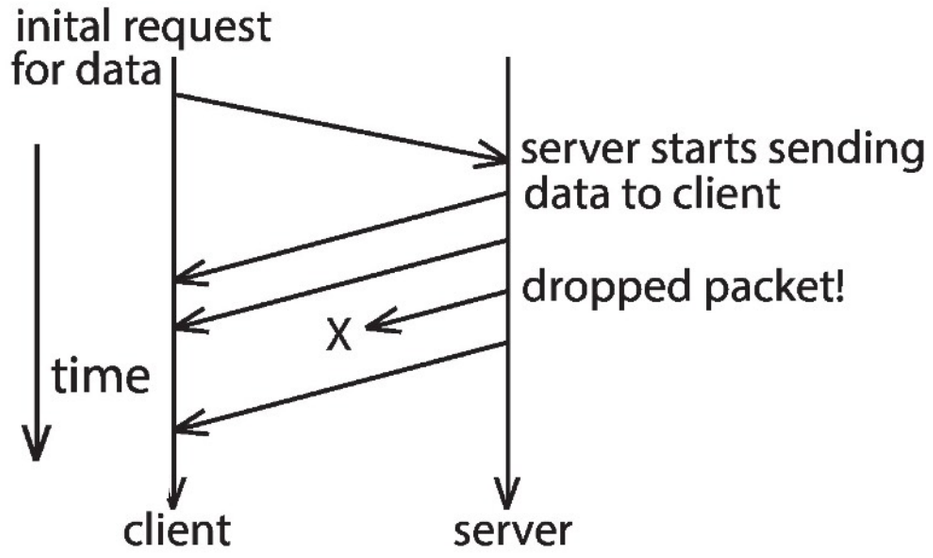
- UDP packets are also called **datagrams**
  - **messages** up to some maximum size
- UDP is unreliable
  - bare-bones extension to IP with the addition of port number
  - packets may be lost or received out-out-order
- UDP is also connectionless
  - no connection setup at the beginning of the transmission to set up state
  - also no connection tear-down at the end of transmission



User Datagram Protocol (UDP)

- UDP packets are also called **datagrams**
  - **messages** up to some maximum size
- UDP is unreliable
  - bare-bones extension to IP with the addition of port number
  - packets may be lost or received out-out-order
- UDP is also connectionless
  - no connection setup at the beginning of the transmission to set up state
  - also no connection tear-down at the end of transmission

Why to use unreliable communication?
many applications simply want to send data
to a destination and not worry about packet loss



Transmission Control Protocol (TCP)

- TCP is both reliable and connection-oriented

Transmission Control Protocol (TCP)

- TCP is both reliable and connection-oriented
- in addition to port number, TCP provides abstraction to allow in-order, uninterrupted byte-stream across an unreliable network
  - whenever host sends packet, the receiver must send an acknowledgement packet (ACK)
  - if ACK is not received before a timer expires, the sender will timeout and retransmit the packet
    - requires keeping a copy of messages sent and not yet acknowledged
  - sequence counters in TCP header allow the receiver to put packets in order and notice duplicate packets (`ack` was lost)

Transmission Control Protocol (TCP)

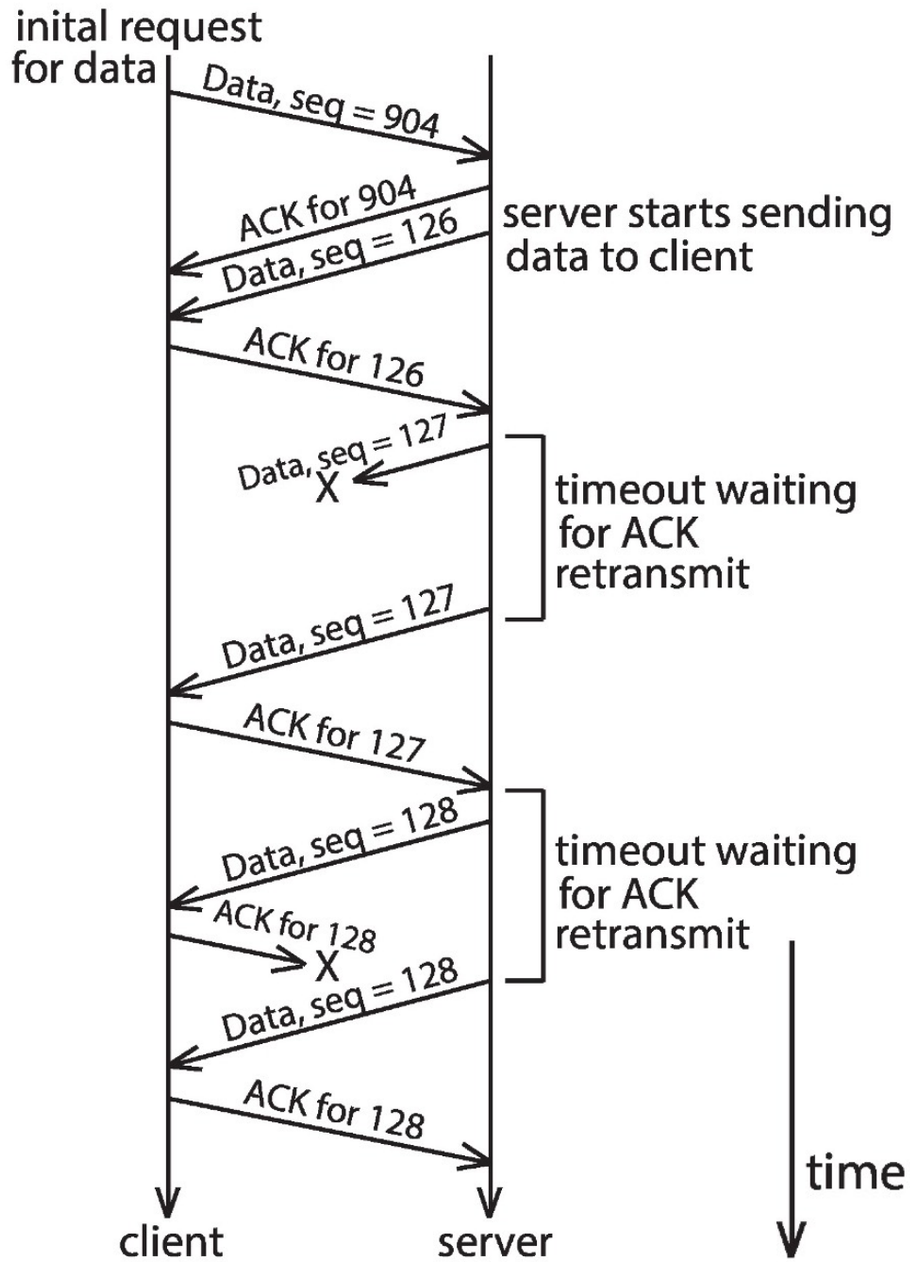
- TCP is both reliable and connection-oriented
- in addition to port number, TCP provides abstraction to allow in-order, uninterrupted byte-stream across an unreliable network
  - whenever host sends packet, the receiver must send an acknowledgement packet (ACK)
  - if ACK is not received before a timer expires, the sender will timeout and retransmit the packet
    - requires keeping a copy of messages sent and not yet acknowledged
  - sequence counters in TCP header allow the receiver to put packets in order and notice duplicate packets (`ack` was lost)
- connections are initiated with series of control packets
  - three-way handshake (SYN, SYN+ACK, ACK)

Transmission Control Protocol (TCP)

- TCP is both reliable and connection-oriented
- in addition to port number, TCP provides abstraction to allow in-order, uninterrupted byte-stream across an unreliable network
  - whenever host sends packet, the receiver must send an acknowledgement packet (ACK)
  - if ACK is not received before a timer expires, the sender will timeout and retransmit the packet
    - requires keeping a copy of messages sent and not yet acknowledged
  - sequence counters in TCP header allow the receiver to put packets in order and notice duplicate packets (`ack` was lost)
- connections are initiated with series of control packets
  - three-way handshake (SYN, SYN+ACK, ACK)
- connections also closed with series of control packets

TCP data transfer

- receiver can send a cumulative ACK to acknowledge series of packets
  - server can also send multiple packets before waiting for ACKs
  - takes advantage of network throughput



TCP data transfer

- receiver can send a cumulative ACK to acknowledge series of packets
  - server can also send multiple packets before waiting for ACKs
  - takes advantage of network throughput
- flow of packets regulated through flow control and congestion control
  - **flow control** – prevents sender from overrunning capacity of receiver
  - **congestion control** – approximates congestion of the network to slow down or speed up packet sending rate