Intro to IT Security

CS306C—Fall 2022

Prof. Antonio R. Nicolosi

Antonio.Nicolosi@stevens.edu

STEVENS INSTITUTE of TECHNOLOGY
THE INNOVATION UNIVERSITY

Networking Background

Networking Background

- What is a network
- Types of networks
- Network models/network operations

What is a Network

At least two points of view:

- End-system centric view
- Infrastructure centric view

End-System Centric View

Network is seen as communication medium

- What differentiates different type of networks
  - Latency
  - Bandwidth
  - Loss rate
  - Number of end systems
  - Interface
- Security issues
  - Authentication, Confidentiality, Anonymity, Integrity

Infrastructure Centric View

Network is seen as system of several components

- Network components
  - Links: cable, optical fiber, wireless (802.11), microwave (802.16), infrared, satellite
  - Interfaces
    - hardware / software
  - Hosts/End-Points
    - PCs, PDAs, cellphones, laptops
  - Everything else
    - Routers, switches, HUB, bridges, modem, . . .
- Protocols: rules governing data movements
  - TCP/IP, HTTP, SMTP, IMAP
  - Operate at different level of abstractions (layers)
- Security issues
  - Authentication, Confidentiality, Integrity

Infrastructure Centric View

Network is seen as system of several components

- Network components
  - Links: cable, optical fiber, wireless (802.11), microwave (802.16), infrared, satellite
  - Interfaces
    - hardware / software
  - Hosts/End-Points
    - PCs, PDAs, cellphones, laptops
  - Everything else
    - Routers, switches, HUB, bridges, modem, . . .
- Protocols: rules governing data movements
  - TCP/IP, HTTP, SMTP, IMAP
  - Operate at different level of abstractions (layers)
- Security issues
  - Authentication, Confidentiality, Integrity

Infrastructure Centric View

Network is seen as system of several components

- Network components
  - Links: cable, optical fiber, wireless (802.11), microwave (802.16), infrared, satellite
  - Interfaces
    - hardware / software
  - Hosts/End-Points
    - PCs, PDAs, cellphones, laptops
  - Everything else
    - Routers, switches, HUB, bridges, modem, . . .
- Protocols: rules governing data movements
  - TCP/IP, HTTP, SMTP, IMAP
  - Operate at different level of abstractions (layers)
- Security issues
  - Authentication, Confidentiality, Integrity

Types of Networks

- Local Area Network (LAN)
  - Small
  - Locally Controlled
  - Physically protected
  - Limited scope (supports a single group)
- Wide Area Network (WAN)
  - Single control (many subscribers, one organization)
  - Covers a significant distance
  - Physically exposed
- Internet
  - Federation of networks
  - Controlled by the Internet Society (ISOC) and the Internet Corporation for Assigned Names and Numbers (ICANN)
  - Heterogeneous
  - Enormous
  - Physically and logically exposed

Types of Networks

- Local Area Network (LAN)
  - Small
  - Locally Controlled
  - Physically protected
  - Limited scope (supports a single group)
- Wide Area Network (WAN)
  - Single control (many subscribers, one organization)
  - Covers a significant distance
  - Physically exposed
- Internet
  - Federation of networks
  - Controlled by the Internet Society (ISOC) and the Internet Corporation for Assigned Names and Numbers (ICANN)
  - Heterogeneous
  - Enormous
  - Physically and logically exposed

Types of Networks

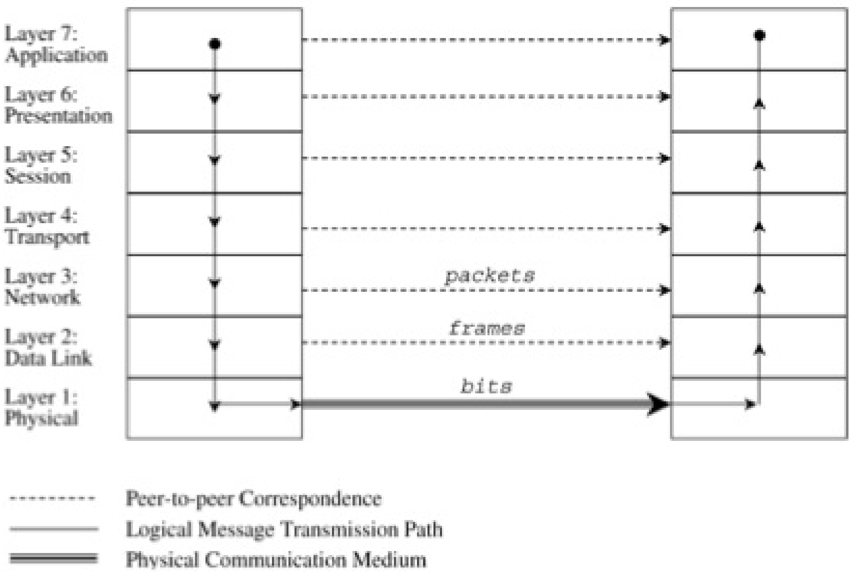
- Local Area Network (LAN)
  - Small
  - Locally Controlled
  - Physically protected
  - Limited scope (supports a single group)
- Wide Area Network (WAN)
  - Single control (many subscribers, one organization)
  - Covers a significant distance
  - Physically exposed
- Internet
  - Federation of networks
  - Controlled by the Internet Society (ISOC) and the Internet Corporation for Assigned Names and Numbers (ICANN)
  - Heterogeneous
  - Enormous
  - Physically and logically exposed

Protocol Stack

- Protocols allow to view the network at an abstract level of communication
  - Details of how the communication is accomplished are hidden with software and hardware at both ends
- Protocol stack: a layered architecture for communication
  - Open Systems Interconnection (OSI)
  - Transmission Control Protocol and Internet Protocol (TCP/IP)

ISO OSI Protocol Layer Levels

- ISO: International Standard Organization
- The ISO OSI model consists of 7 layers



ISO OSI Protocol Layer Levels

- Layers represents the different activities that must be performed for actual transmission of a message
- Equivalent layers perform similar functions for the sender and the receiver
- Each layer passes data
  - Above: with a layer communicating more abstractly
  - Across (conceptually): to the same layer at another host
  - Below: with a layer handling less abstract data

ISO OSI Protocol Layer Levels

- Application: User-level data
- Presentation: Standardized data appearance, separation in blocks, text compression
- Session: Bundling of logically connected upper-layer units
- Transport
  - Flow control
  - End-to-end error detection and correction
- Network
  - Routing
  - Breaks message into uniformly sized packets
- Data Link
  - Reliable data delivery
  - Transmission error recovery
  - Breaks packets into uniformly sized frames
- Physical
  - Actual communication across physical medium
  - Individual bit transmission

ISO OSI Protocol Layer Levels

- Application: User-level data
- Presentation: Standardized data appearance, separation in blocks, text compression
- Session: Bundling of logically connected upper-layer units
- Transport
  - Flow control
  - End-to-end error detection and correction
- Network
  - Routing
  - Breaks message into uniformly sized packets
- Data Link
  - Reliable data delivery
  - Transmission error recovery
  - Breaks packets into uniformly sized frames
- Physical
  - Actual communication across physical medium
  - Individual bit transmission

ISO OSI Protocol Layer Levels

- Application: User-level data
- Presentation: Standardized data appearance, separation in blocks, text compression
- Session: Bundling of logically connected upper-layer units
- Transport
  - Flow control
  - End-to-end error detection and correction
- Network
  - Routing
  - Breaks message into uniformly sized packets
- Data Link
  - Reliable data delivery
  - Transmission error recovery
  - Breaks packets into uniformly sized frames
- Physical
  - Actual communication across physical medium
  - Individual bit transmission

ISO OSI Protocol Layer Levels

- Application: User-level data
- Presentation: Standardized data appearance, separation in blocks, text compression
- Session: Bundling of logically connected upper-layer units
- Transport
  - Flow control
  - End-to-end error detection and correction
- Network
  - Routing
  - Breaks message into uniformly sized packets
- Data Link
  - Reliable data delivery
  - Transmission error recovery
  - Breaks packets into uniformly sized frames
- Physical
  - Actual communication across physical medium
  - Individual bit transmission

ISO OSI Protocol Layer Levels

- Application: User-level data
- Presentation: Standardized data appearance, separation in blocks, text compression
- Session: Bundling of logically connected upper-layer units
- Transport
  - Flow control
  - End-to-end error detection and correction
- Network
  - Routing
  - Breaks message into uniformly sized packets
- Data Link
  - Reliable data delivery
  - Transmission error recovery
  - Breaks packets into uniformly sized frames
- Physical
  - Actual communication across physical medium
  - Individual bit transmission

ISO OSI Protocol Layer Levels

- Application: User-level data
- Presentation: Standardized data appearance, separation in blocks, text compression
- Session: Bundling of logically connected upper-layer units
- Transport
  - Flow control
  - End-to-end error detection and correction
- Network
  - Routing
  - Breaks message into uniformly sized packets
- Data Link
  - Reliable data delivery
  - Transmission error recovery
  - Breaks packets into uniformly sized frames
- Physical
  - Actual communication across physical medium
  - Individual bit transmission

ISO OSI Protocol Layer Levels

- Application: User-level data
- Presentation: Standardized data appearance, separation in blocks, text compression
- Session: Bundling of logically connected upper-layer units
- Transport
  - Flow control
  - End-to-end error detection and correction
- Network
  - Routing
  - Breaks message into uniformly sized packets
- Data Link
  - Reliable data delivery
  - Transmission error recovery
  - Breaks packets into uniformly sized frames
- Physical
  - Actual communication across physical medium
  - Individual bit transmission

Example: Sending an Email

L7 Application

- Composing an email happens at the application layer (`bob@example.com`)
- At this level of abstraction, network is the medium that allow to reach Bob at his email address.

L6 Presentation

- Raw text modified to prepare it for transmission (*e.g.*, compression, encryption)

L5 Session: N/A (no interaction)

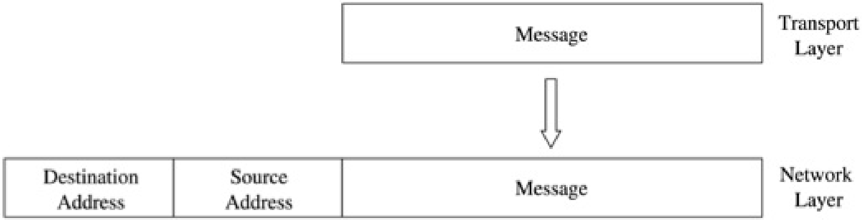
L4 Transport

- Add error detection and correction coding

Example: Sending an Email

L3 Network: Router sends the message from Alice’s network to Bob’s network

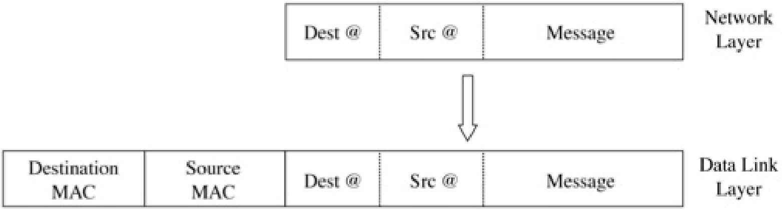
- Packet: Data + Source Address + Destination Address



Example: Sending an Email

L2 Data Link: Move message from Alice’s computer to Alice’s router

- NIC: Network Interface Card
- Every computer connected to the network has a NIC with a unique physical address: Media Access Control (MAC)
- Frame: Data (Packet) + Source MAC + Destination MAC



L1 Physical: Actual bits are sent (over e.g. WiFi, optic fibers, copper wire, . . .)

TCP/IP protocol

TCP/IP (Transmission Control Protocol/Internet Protocol)

Four layers (a.k.a. PITA)

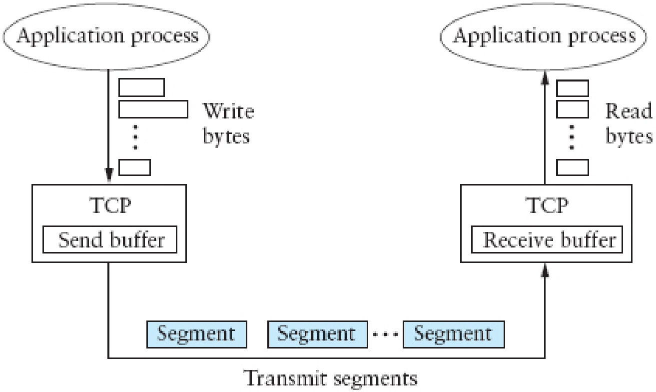
L4 **Application**: Prepare msg from user interaction

L3 **Transport**: Convert msg to packets

L2 **Internet**: Convert packet to datagrams

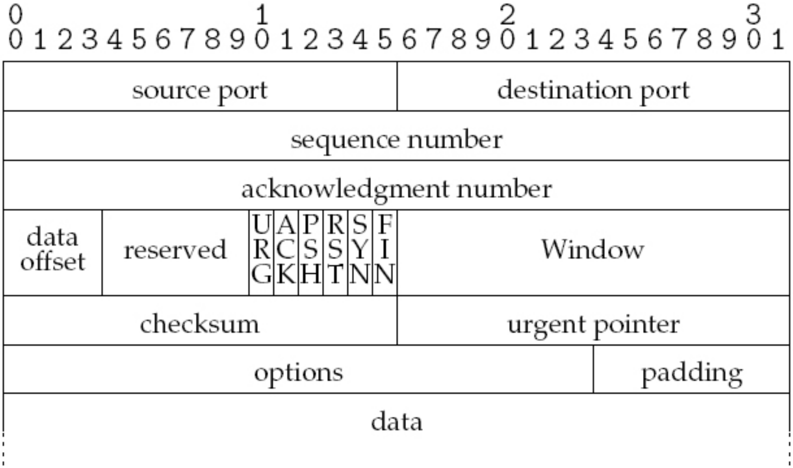
L1 **Physical**: Transmit datagrams as individual bits

Overview of TCP



- Full duplex, connection-oriented byte stream
- Flow control
  - If one end stops reading, writes at other eventually block/fail
- Congestion control
  - Keeps sender from overrunning network

TCP Segment


0
0 1 2 3 4 5 6 7 8 9
1
0 1 2 3 4 5 6 7 8 9
2
0 1 2 3 4 5 6 7 8 9
3
0 1
source port
destination port
sequence number
acknowledgment number
data offset
reserved
URG
ACK
PSH
RST
SYN
FIN
Window
checksum
urgent pointer
options
padding
data

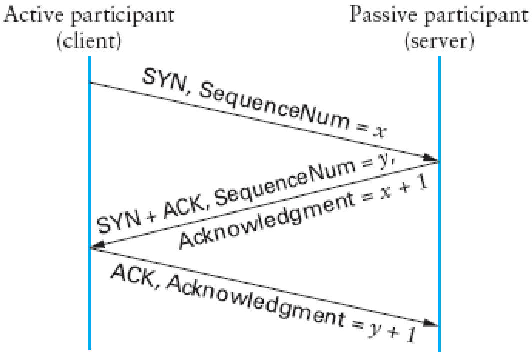
TCP Fields

- Ports
- Seq no.: segment position in byte stream
- Ack no.: seq no. sender expects to receive next
- Data offset: num. of 4-byte header & option words
- Window: willing to receive (flow control)
- Checksum
- Urgent pointer

TCP Flags

- URG: urgent data present
- ACK: ack no. valid (all but first segment)
- PSH: push data up to application immediately
- RST: reset connection
- SYN: “synchronize” establishes connection
- FIN: close connection

Connection Establishment

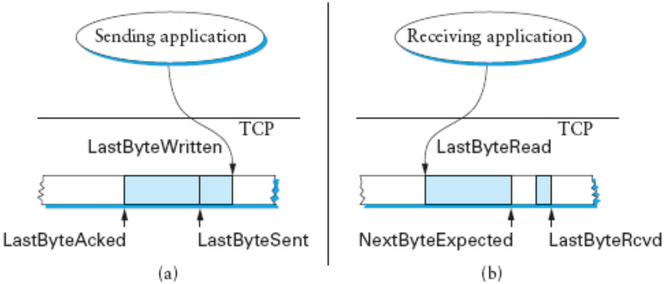


- Need SYN packet in each direction
  - Typically second SYN also acknowledges first
  - Supports “simultaneous open”, seldom used in practice
- If no program listening: server sends RST
- If server backlog exceeded: ignore SYN
- If no SYN-ACK received: retry, timeout

Sending Data

- Segments may arrive out of order
  - Sequence number used to reassemble in order
- Window achieves flow control
  - If window 0 and sender’s buffer full, write will block

Sliding Window



- Used to guarantee reliable & in-order delivery
- Also used for flow control
  - Instead of fixed window size, receiver sends `AdvertisedWindow`

Retransmission

- TCP dynamically estimates round trip time
- If segment goes unacknowledged, must retransmit
- Use exponential backoff (in case loss from congestion)
  - Additive increase and multiplicative decrease
- After ≈10 minutes, give up and reset connection
- Many optimizations in TCP
  - E.g., Don't necessarily halt everything for one lost packet
  - Just reduce window by half, then slowly augment

Security Issues in TCP/IP

- Network packets pass by untrusted hosts
  - Eavesdropping (packet sniffing/snooping)
- IP addresses are public
  - Smurf attacks
- TCP connection requires state
  - SYN flooding
- TCP state is easy to guess
  - TCP spoofing and connection hijacking

Packet Sniffing

- Many applications send data unencrypted
  - ftp, telnet, SMTP with `auth=plain` send passwords in the clear
- Network interface card (NIC) in “promiscuous mode” reads all passing data
- Solution: encryption (e.g., IPSec), improved routing

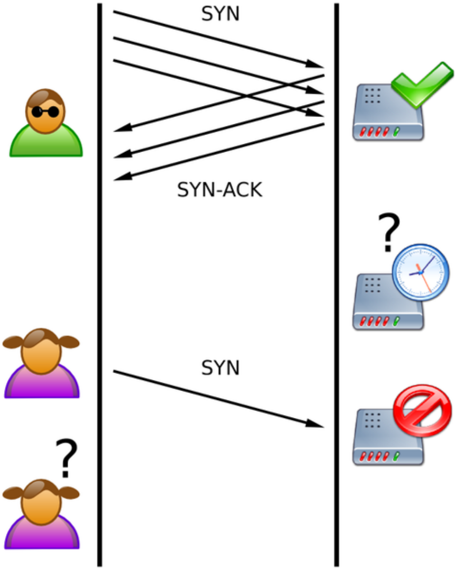
DoS Attacks

- In Feb. 2000, Yahoo's router kept crashing
  - Engineers had problems with it before, but this was worse
  - Turned out they were being flooded with ICMP echo replies
  - Many DDoS attacks followed against high-profile sites
- Basic Denial of Service attack
  - Overload a server or network with too many packets
  - Maximize cost of each packet to server in CPU and memory
- Distributed DoS (DDos) particularly effective:
  - Penetrate many machines in semi-automatic fashion
  - Make hosts into "zombies" that will attack on command
  - Later start simultaneous widespread attacks on a victim

Smurf Attack

- Yahoo attack was a “smurf attack”
  - Penetrated hosts on well-connected networks
  - Flooded LAN with broadcast pings “from” Yahoo
  - Every host on LAN then replied to Yahoo
  - Attack was amplified through uncompromised hosts
- Can tolerate above by filtering packets
  - Ignore all ICMP echo replies from particular addresses
  - Attack still had to be traced to stop waste
  - But attack packets could be distinguished from most legitimate traffic

The SYN-Flooding Attack



The SYN-Flooding Attack

- TCP handshake:
  - $C \to S$: SYN, $S \to C$: SYN-ACK, $C \to S$: ACK
- How to implement:
  - Server inserts connection state in a table
  - Waits for 3rd packet (times out after a minute)
  - Compares each new ack packet to existing connections
- OS can't handle arbitrary # partial connections
- Attack: Send SYN packets from bogus addresses
  - SYN-ACKs will go off into the void
  - Server's tables fill up, stops accepting connections
  - A few hundred pkts/sec completely disables most servers

Other Attacks

- IP Fragment flooding
  - Kernel must keep IP fragments around for partial packets
  - Flood it with bogus fragments, as with TCP SYN bomb
- UDP echo port 7 replies to all packets
  - Forge packet from port 7, two hosts echo each other
  - Has been fixed in most implementations
- Standard flooding attacks
  - Just flood-ping any site
  - Or bombard DNS server with requests

TCP Connection Spoofing

- TCP connection has an associated state
  - Sequence number, port number
- TCP state is easy to guess
  - Port numbers are standard, sequence numbers are often predictable
  - Can inject packets into existing connections
- If attacker knows initial sequence number and amount of traffic, can guess a likely current number
  - Send a flood of packets with likely sequence numbers

Making Attacks Hard to Stop

- Make DoS traffic indistinguishable from legitimate traffic
  - SYN-bomb ideal, DNS service good
  - Flood-ping at least can be filtered anywhere upstream
- Make source of attack hard to trace
  - Victims need to trace attack and pull the plug
  - Can forge source IP address so packet origin not obvious
  - Most DoS tools use a random address for each packet

Coping with Denial of Service

- Engineering OSes to tolerate attacks
  - Reduce state required for embryonic TCP connections
  - Increase size of hash table for protocol control blocks
  - Use “SYN-cookies” which don’t fill up the queue until the last ack returns
  - Cryptographic puzzles—make the client perform a certain amount of work before starting the connection
- Network monitoring box
  - Passively monitors network
  - Uses heuristics to detect SYN bomb attacks (e.g., traffic patterns w. invalid source addresses)
  - Monitor engineered to keep little state
  - Send out forged RST packets to free resources on victim

Egress Filtering

- Forged addresses complicate shutting off DoS
  - Where is flood of packets coming from?
- Filter forged outgoing packets
  - Sites should block outgoing packets not from their network
  - ISPs should block packets not from customer's network
- But still need to detect and shut down attacks
- And most attackers can find non-filtered networks anyway

Input Debugging

- Some routers can trace output to input
  - Develop attack signature to classify bad packets
  - Router tells you which input port they are from
- Of course, only router administrator can do this
- Must continue on to upstream routers, in other realms
- Not all routers have this capability