

**ENTS 689i**

**Lecture 9:  
Network Security Introduction**

**Part III: Network Security**

# Part III: Goals

---

- Review how networks work
- Explore why networks break
- Understand tools and techniques used by attackers (threats)
- Study network defense mechanisms (protocols, systems, etc)
- Broaden our security mindset to networks

# Part II: Organization

---

- Section format
  - Three lectures
  - In-class exam
  - Mini-lecture: “Network Attacks in the News”
- Assignments (2)
  - (1) Written assignment
  - (1) Lab assignment
- Office hours
  - Following class

# Ethical Computing

---

- We are going to discuss concepts and techniques that work against real systems
- Isolated virtual machines will be used for all experimentation
- **YOU** are responsible for knowing what is acceptable and what is not!

# Part II: Outline

---

- November 6: Network Security Intro
  - November 7: HW 4 out
- November 13: Secure Protocols
  - November 16: HW 4 due
- November 20: Infrastructure Defense
  - November 21: HW 5 out
  - November 30: HW 5 due
- December 4: Network Attacks in the News
  - EXAM 3

# Part II: Outline

---

- November 6 (Today)
  - Networking Basics
  - Network Security Services (Goals)
  - Threat Model
  - Anatomy of an Attack
  - Attacks on Internet Services
  - Denial of Service Attacks

# Network Security: Motivation

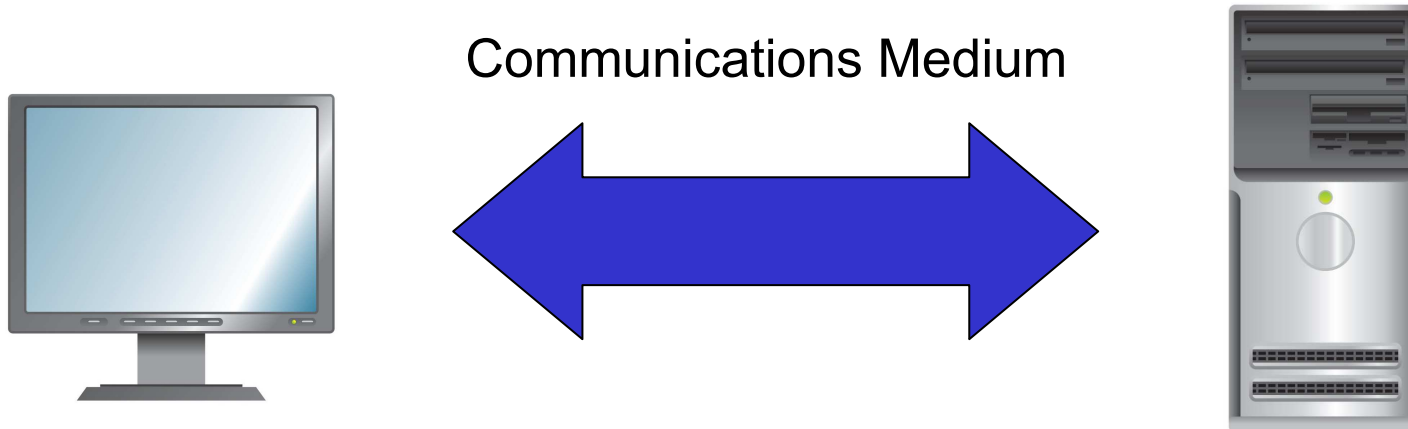
---

- Network Security
  - Information
  - Computers (Applications, OS, Database)
  - Connections between them
- Why study network security?
  - Banking/credit card transactions
  - Communications (telephone, Internet, etc)
  - Personal information (PII, CCNs)
  - Network enabled!

# Networks

---

- Two devices connected across a communications medium (simplified)
  - Devices: laptop, workstation, cell phone, printers, scanners
  - Medium: cables, wireless, cellular, satellite



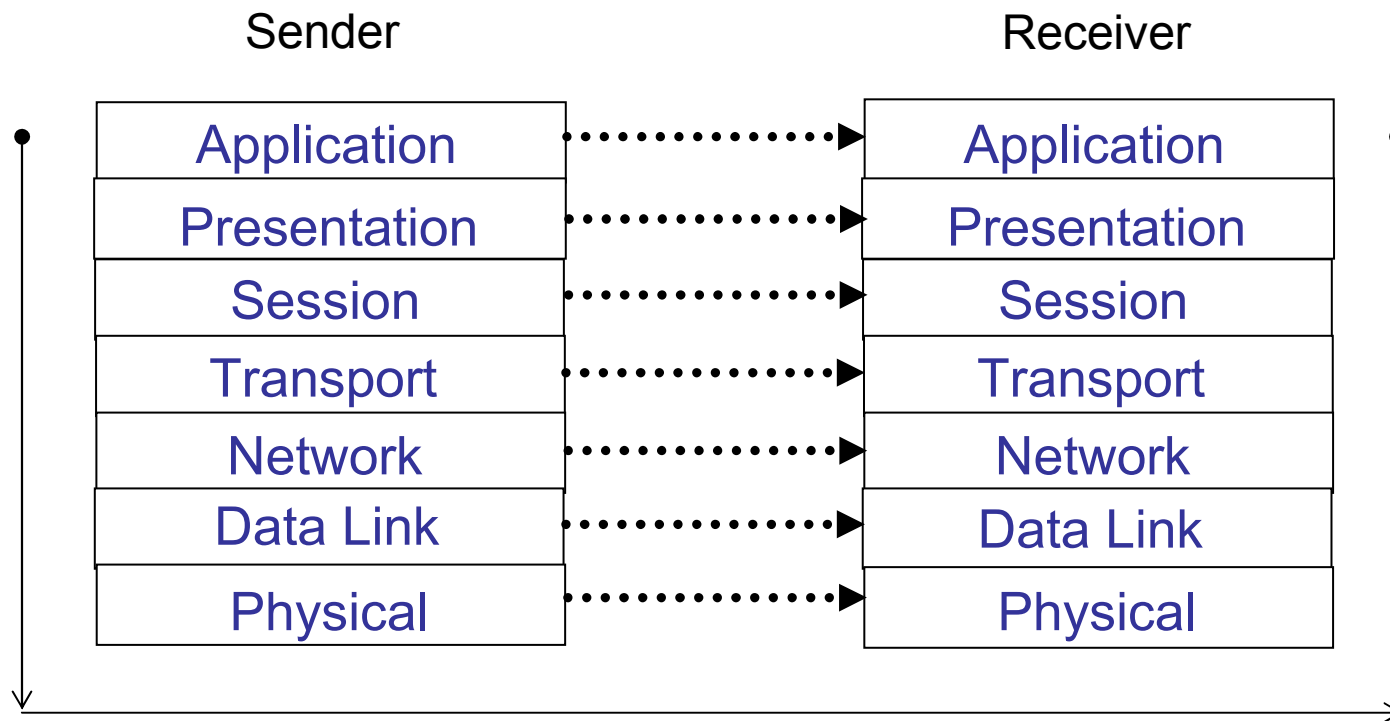
# Protocols

---

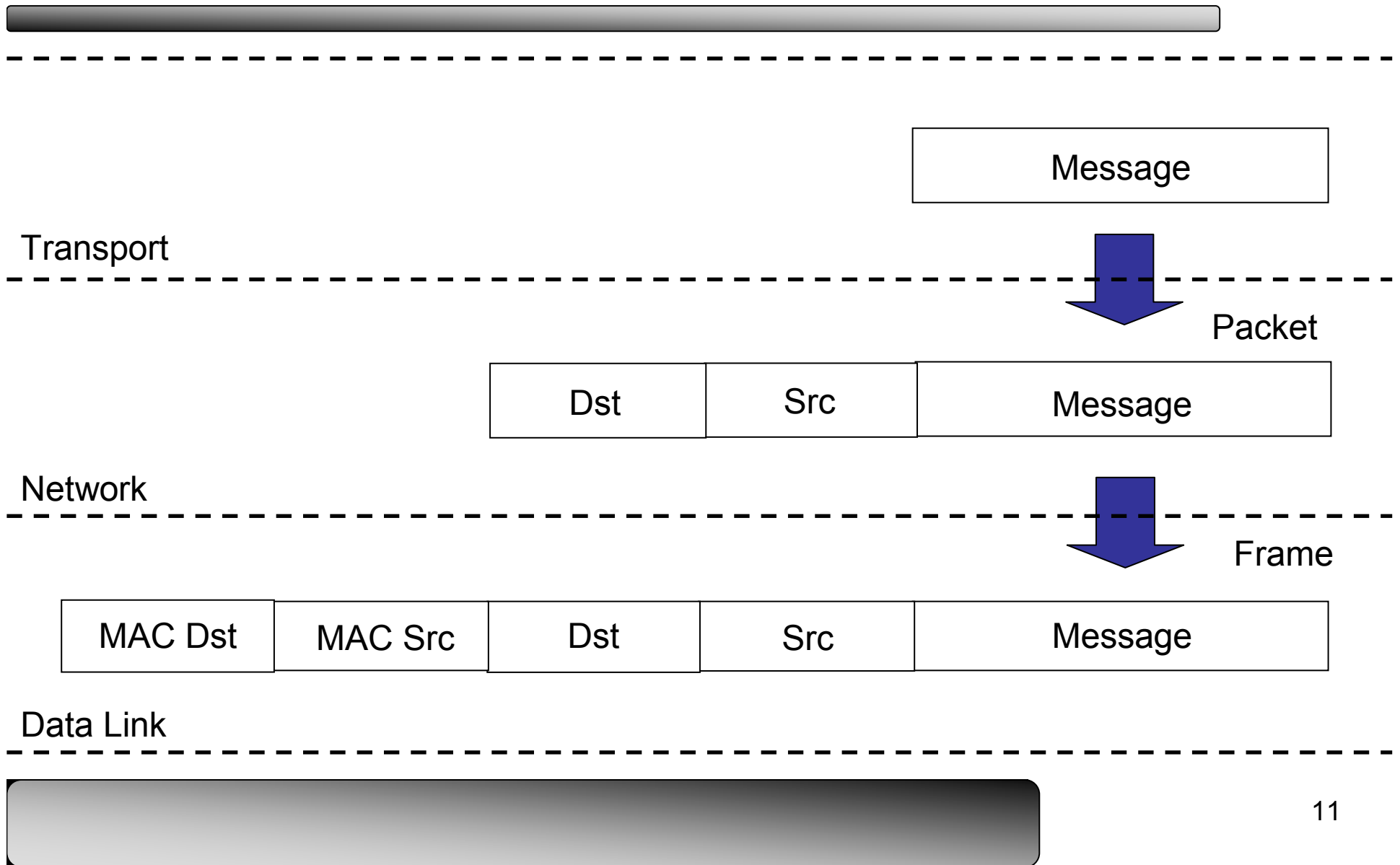
- Networks rely on protocols
  - Communication medium is transparent
    - Copper wire, wireless, cellular
  - Rules/message formats (PL)
- Protocol Stacks
  - Layers of abstraction
    - Provides (above)/use (below) service
  - ISO Open Systems Interconnection reference model
  - TCP/IP Suite

# OSI Reference Model

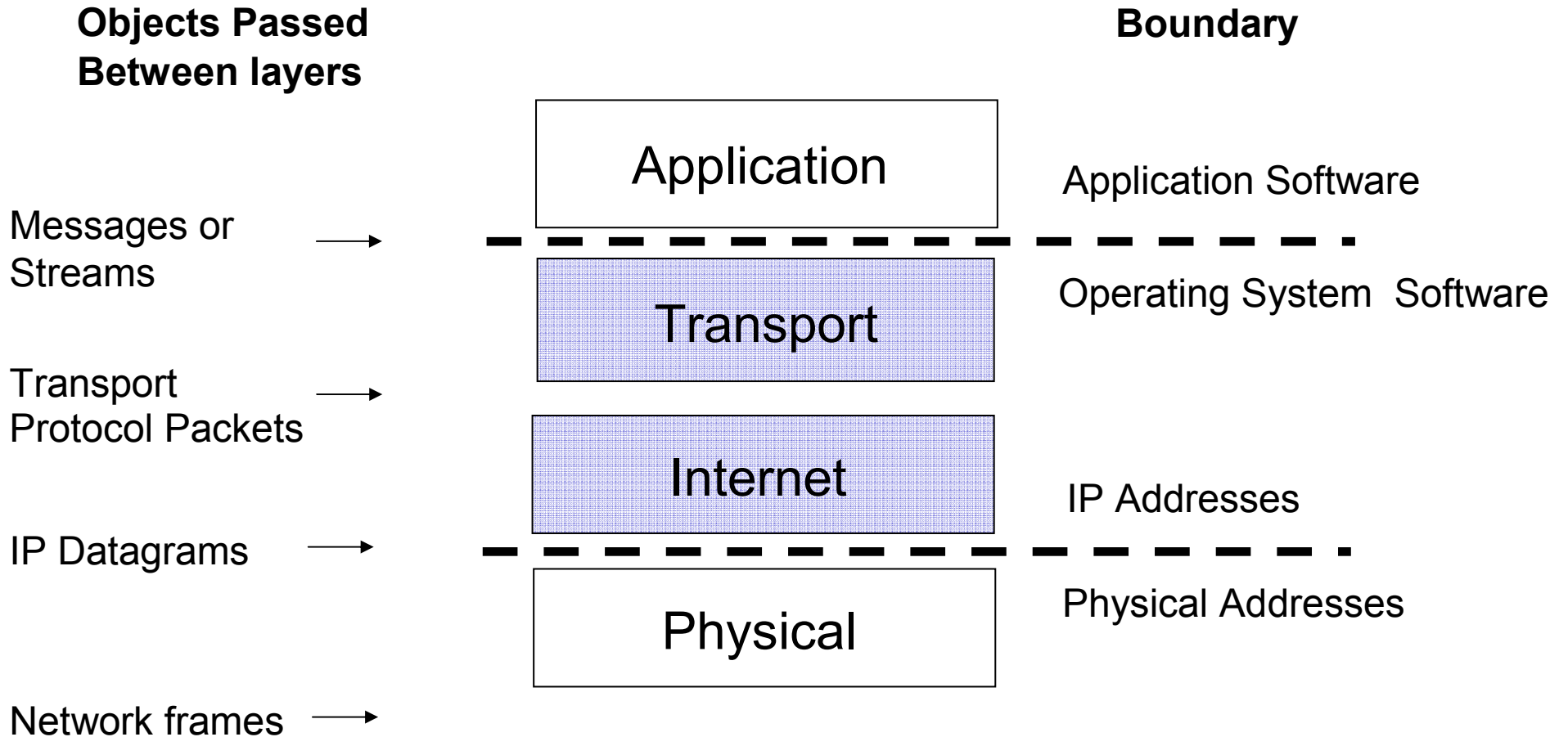
---



# Layering



# TCP/IP Reference Model



# Transport Layer

---

- Provides communication between application programs (end-to-end)
  - Messages → Packets
- TCP (Transmission Control Protocol)
  - Reliable transport: ordering, integrity
  - Flow control
- UDP (User Datagram Protocol)
  - Unreliable/Connection-less (user datagram)

# Protocol Ports

---

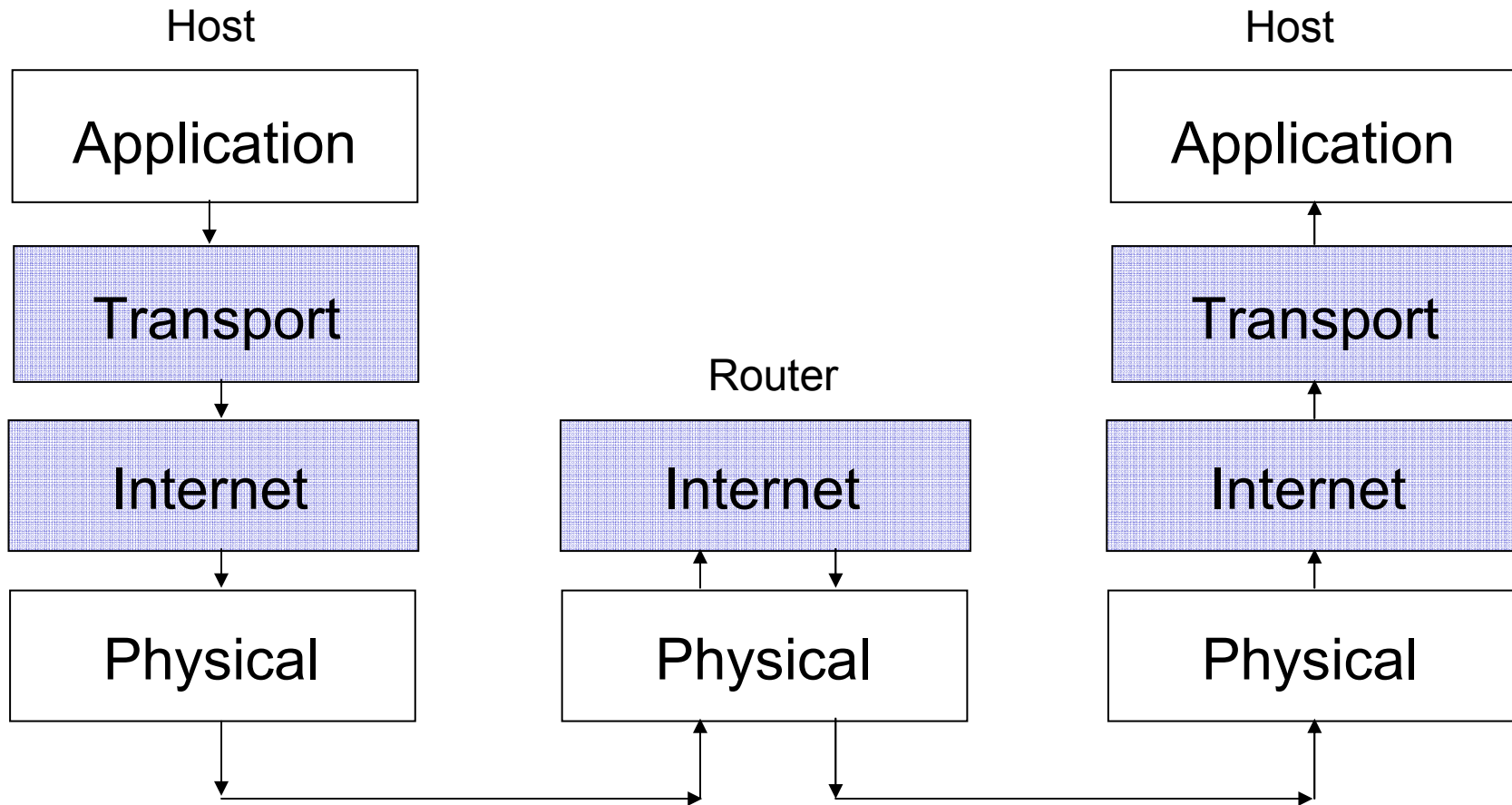
- Communication endpoint (port number)
- Demultiplexes traffic within a machine
  - Process “binds” socket to a port (snd/rcv)
- Port numbers (16-bits: 0-65535)
  - Services: 0-1023 (assigned, superuser), well-known
  - Clients: 1024-65535 (dynamic, ephemeral)
- Common Ports (IANA)
  - TCP: 80 (http), 443 (https), 25 (SMTP)
  - UDP: 123 (NTP), 69 (TFTP), 67:68 (BOOTP, DHCP)

# Internet Layer

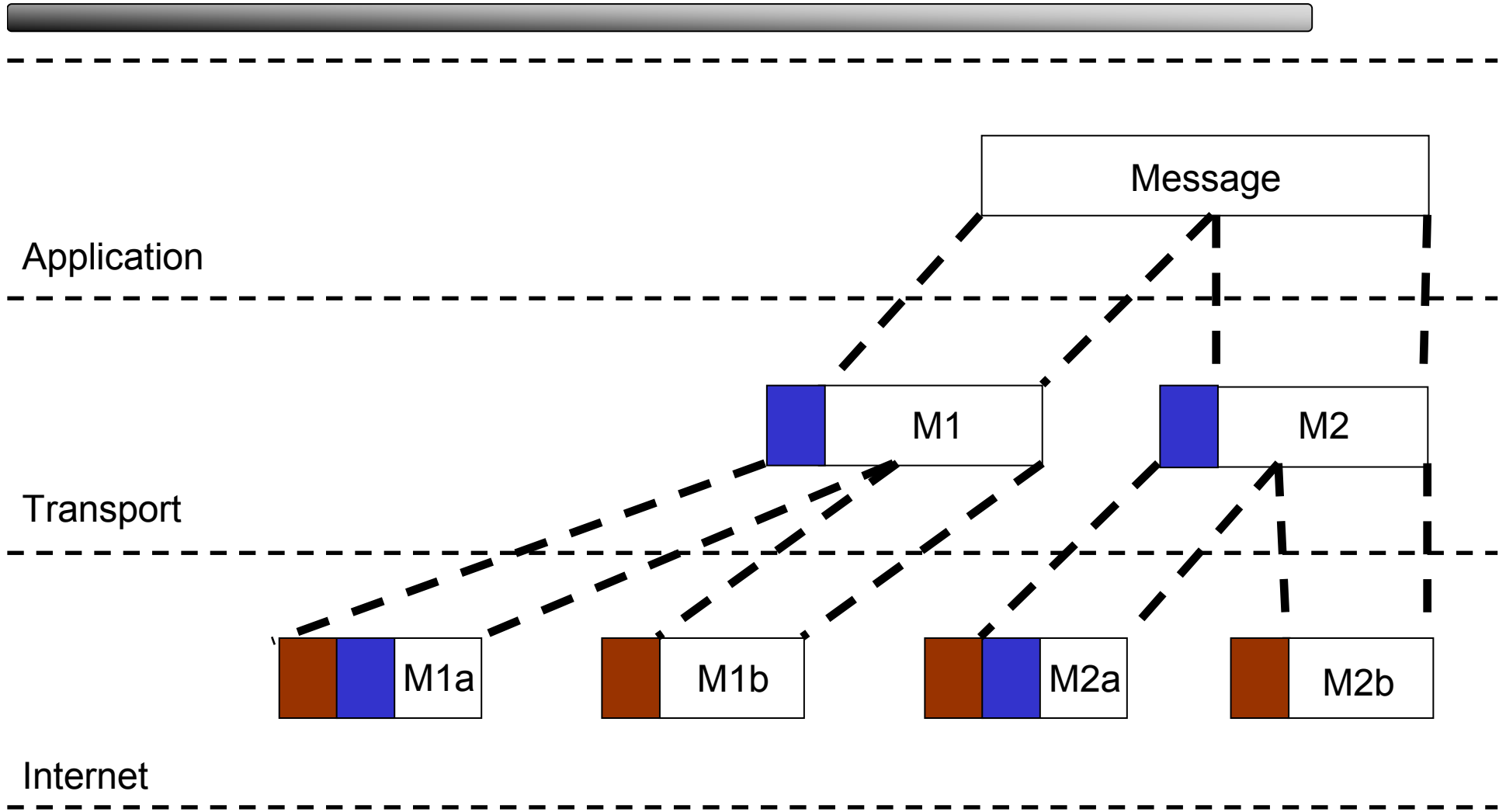
---

- Provides communication between machines
  - Packets → Datagram
- Internet Protocol (IP)
  - Unreliable/Connectionless
- IP address (IANA)
  - IPv4: 32-bit address (127.0.0.1)
  - IPv6: 128-bit address
- Responsible for routing

# Router



# Segmentation/Fragmentation



# Programming Example (client)

---

```
int main()
{
    int sock, bytes_recvd;
    char recvdata[4096];
    struct hostent *host;
    struct sockaddr_in server_addr;

    host = gethostbyname("127.0.0.1");

    // AF_INET          IPv4 Internet protocols
    // SOCK_STREAM      TCP
    // SOCK_DGRAM       UDP

    sock = socket(AF_INET, SOCK_STREAM, 0);
    server_addr.sin_family = AF_INET;
    server_addr.sin_port = htons(4444);
    server_addr.sin_addr = *((struct in_addr *)host->h_addr);
    bzero(&(server_addr.sin_zero),8);

    connect(sock, (struct sockaddr *)&server_addr,
            sizeof(struct sockaddr));

    bytes_recvd=recv(sock,recvdata,4096,0);
    close(sock);
}
```

# Network Security Services (Goals)

---

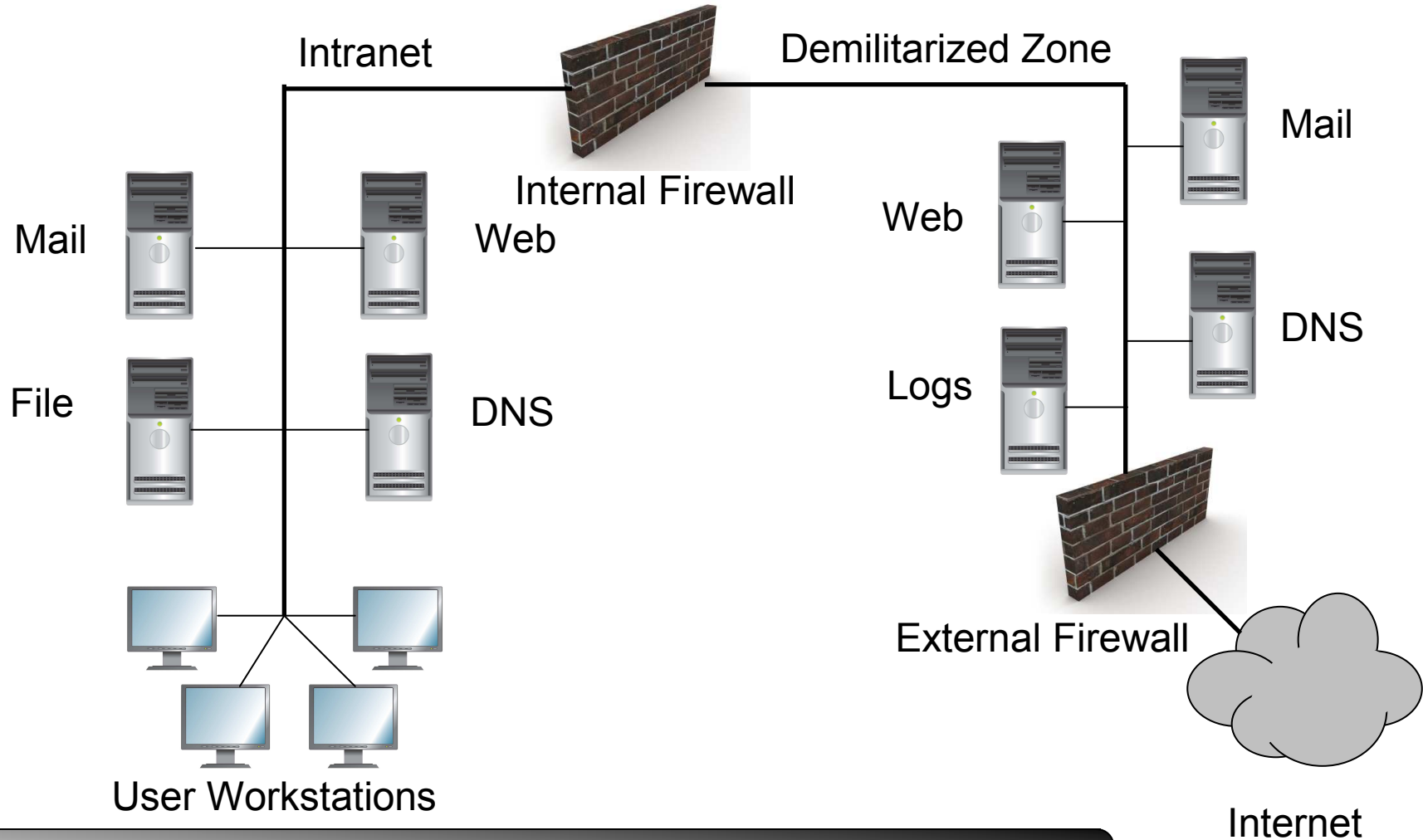
- Confidentiality
  - Only authorized people or systems can access protected data
- Integrity
  - Data was not modified in transit
- Availability
  - Authorized parties have the ability to access/use resource

# Network Security Services (Goals)

---

- **Authentication**
  - Data source: From authorized party
  - Entity: Are who they claim to be
- **Access Control/Authorization**
  - Only authorized parties can access/use resources
- **Non-repudiation**
  - Prove message transmission (sndr/rcvr)
- **Anonymity**
  - Protect identity/data association

# “Standard” Corporate Topology



# Why are Networks Vulnerable?

---

- Anonymity
  - Who and where is the attacker?
- Unknown perimeter
  - Where is my networks boundary?
- Unknown path
  - Who controls the systems? (shared medium)
- System complexity
  - What should be happening on the network?
- Sharing
  - Who needs access to what and why?
- Points of failure
  - What is the weakest link?

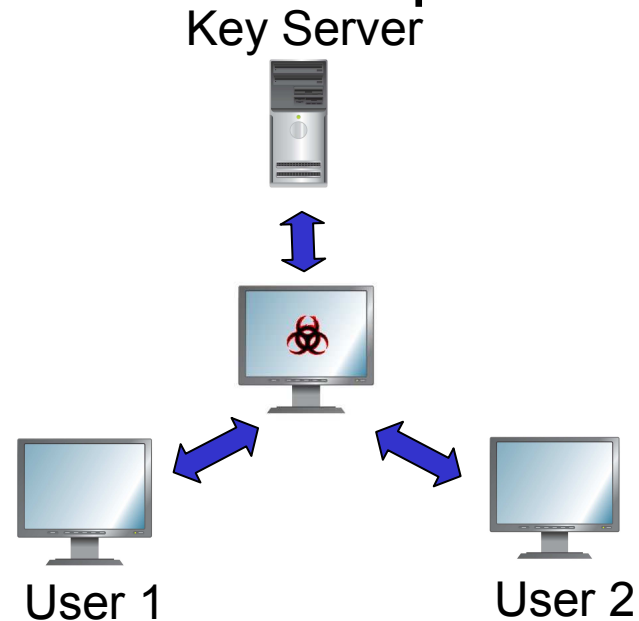
# Passive vs. Active Attacks

---

- **Passive Attack**
  - Eavesdrops on communication (covertly)
    - Data in transit
  - Does not modify the message stream
  - Sniffer (Wireshark), Passive OS Fingerprinting (PoF)
- **Active Attack**
  - Manipulate messages
    - Create, replay, modify, delete
  - Impersonation, Man-in-the-middle

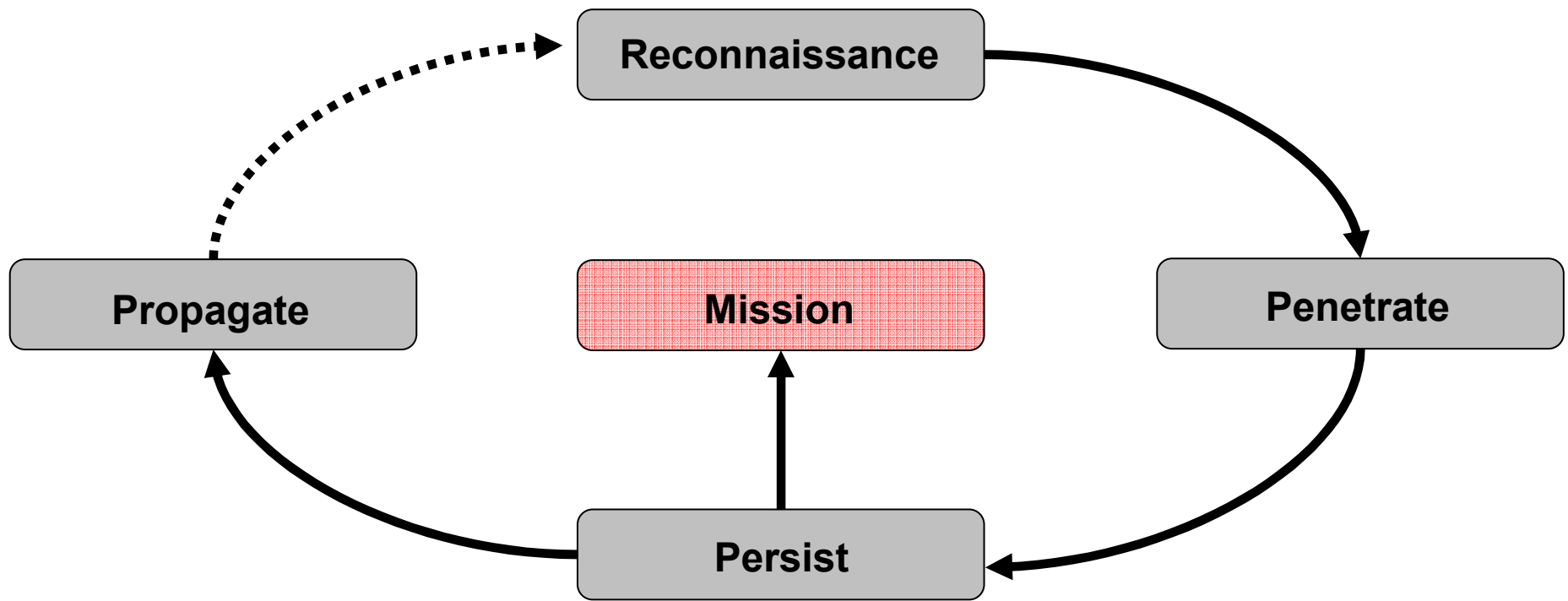
# Man-in-the-Middle

- Active third party
  - Participates from the start of session
  - Mediates communications between parties
- Security protocols
  - Implementation
  - No identity binding
    - Authentication
    - Who? Trusted?



# Anatomy of an Attack

---



# Reconnaissance

---

- **Network Scanning**
  - Port scanning
  - Operating system/application fingerprinting
  - Nmap scanner (Fyodor)
- **Social Engineering (HUMINT)**
  - Social skills/personal interaction
- **Open Source Intelligence (OSINT)**
  - Information from publicly available sources
  - Maltego (email addresses, relationships, documents)

# Web Attacks

---

- Servers
  - Brute force (authentication)
  - Software vulnerabilities (buffer overflows)
  - User input validation
    - SQL injection, application boundaries (dot-dot-slash)
  - Remote timing attack (OpenSSL)
- Clients
  - Browser vulnerabilities
  - Masquerading web sites (man-in-the-middle)
  - Domain typo squatting
  - Malicious content: Trojan (“Security software”)
  - Hidden functionality (IFRAMES, JavaScript)

# Email Attacks

---

- Message interception (Confidentiality)
  - Eavesdropping
  - Typo-squatting
- Message Spoofing (Authentication)
- Message Modification (Integrity)
- Deliver malicious content
  - Malicious software (Trojan)
  - Malicious documents (Office Documents)
  - Malicious links: Phishing (Click this link!)

# Domain Name Service (DNS)

---

- Associates host names and IP addresses
  - Forward record: host name → IP address
  - Reverse record: IP address → host name
  - Data is distributed (hierarchical: root servers)
  - Caching
- Vulnerabilities:
  - Software vulnerabilities (buffer overflows)
  - Information disclosure (zone transfers)
  - DNS cache poisoning
  - DNS Spoofing (authoritative)
  - Denial of Service (UDP)

# Network Security Defenses

---

- Confidentiality/Integrity/Authentication
  - IPSec/VPN
  - SSL
- Authentication/Access Control
  - Kerberos
  - Firewalls
- Availability
  - Redundancy/Sensors
  - Significant threat to networks
  - More than security protocols.....

# Denial of Service (DoS)

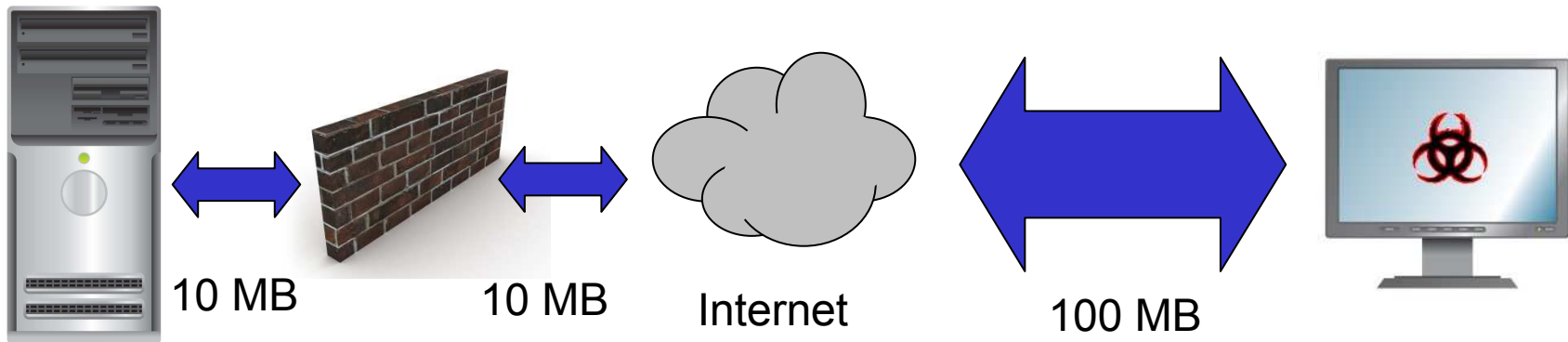
---

- Limit or disrupt authorized access to resources
- Flooding attacks (overwhelm resources)
  - Single source
  - Multiple source (DDoS)
  - Reflector (DDoS)
- Protocol vulnerability
- Protocol design

# Single Source Flooding

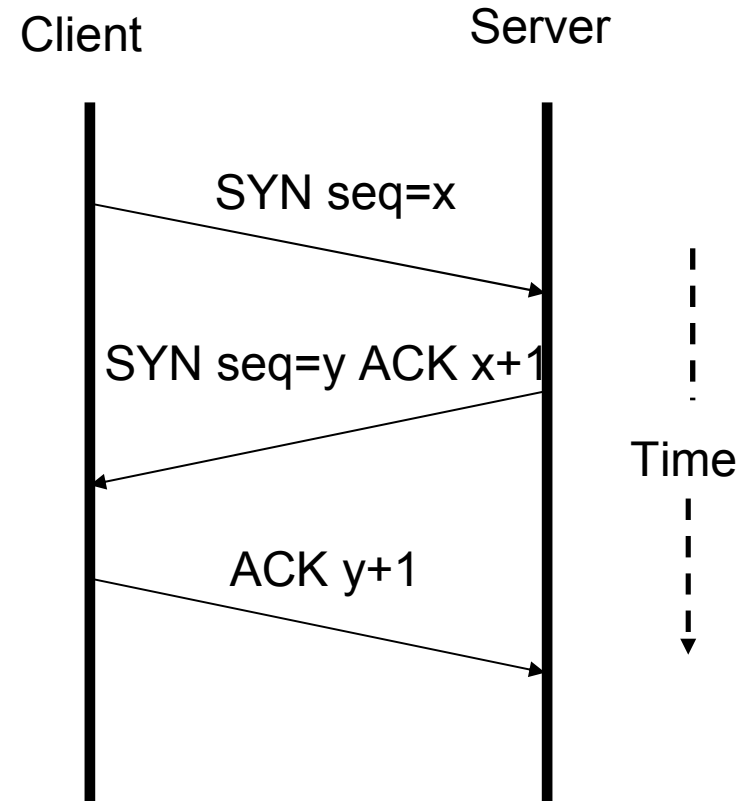
---

- Relative resource allocation (asymmetric)



# TCP Handshake (3-way)

- Client initiates connection (SYN)
- Server acknowledges connection request (SYN/ACK). Allocates resources.
- Client sends Final message (ACK) (connection is established)

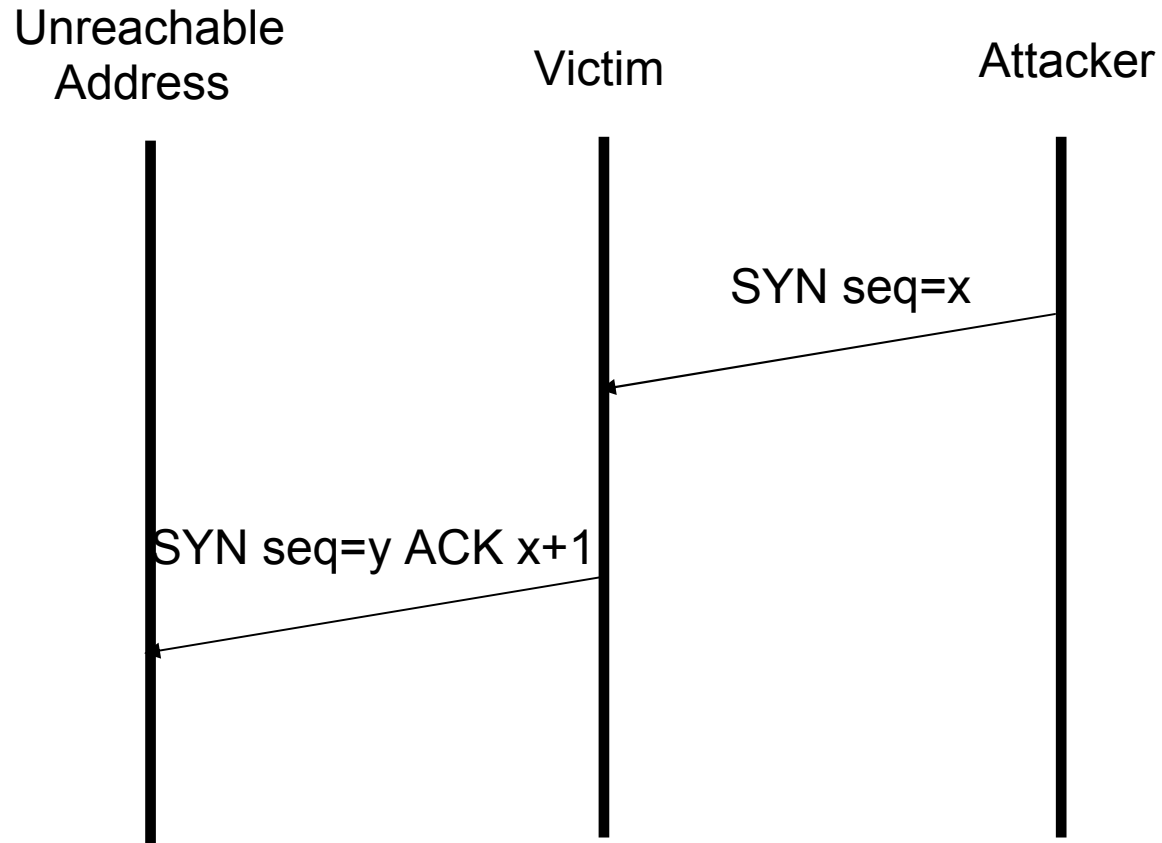


# SYN Flood Attack

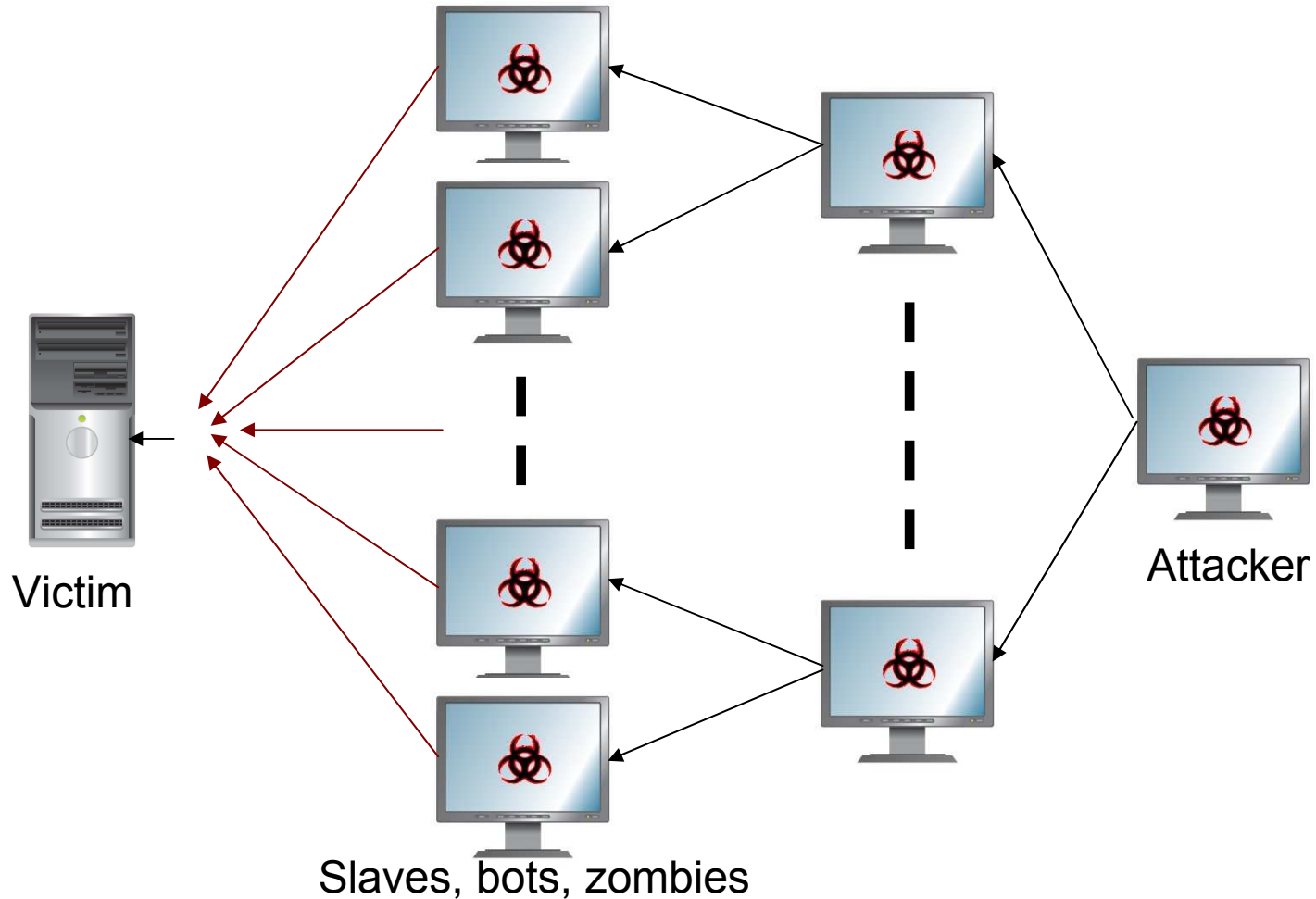
---

- Attacker floods victim with TCP SYN packets
  - Source address is spoofed (unreachable host)
- Victim allocates resources (memory) to open connections (SYN/ACK)
- Handshakes never complete
  - Connections are pending, eventually time out
- Victim unable to service legitimate requests
- Why use an unreachable host?

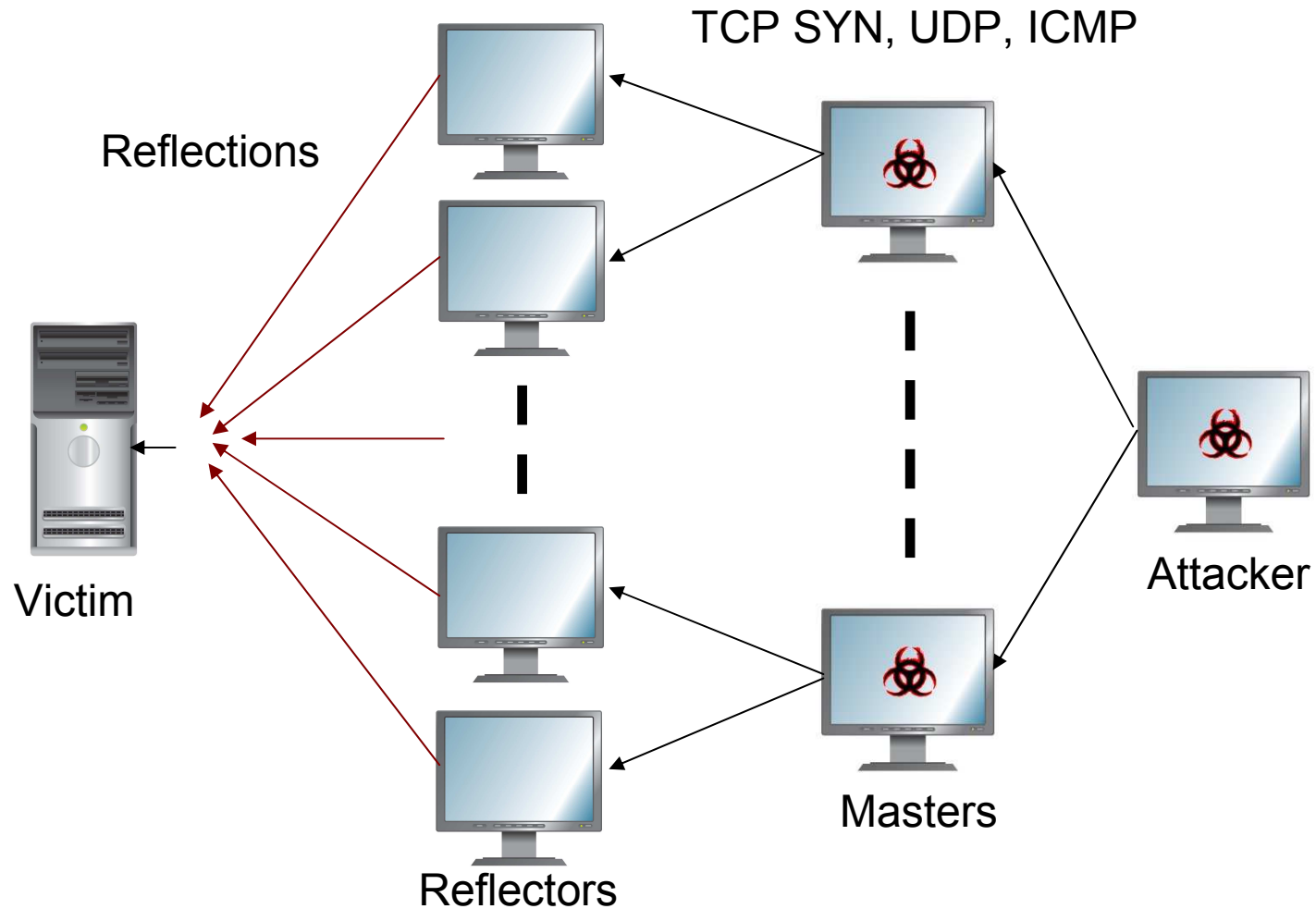
# SYN Flood Attack



# Multiple Source DDoS Attack



# DDoS Reflector Attack



# Defending DDoS

---

- **Entities**
  - Multiple source networks (globally distributed)
  - Intermediate networks (service providers)
  - Victim network
- **Challenges**
  - “Slashdot effect”
  - Detected close to the victim but needs to be stopped close to the sources
    - Who controls the sources?

# DDoS Defenses

---

- **Victim Network**
  - Firewalls/IDS/Proxies?
  - Bandwidth Defense
    - Large pipes and large distributed networks
    - Load balancing
- **Intermediate Network**
  - Upstream packet filtering
    - ISPs
  - IP traceback (probabilistic marking)
    - Try to find source networks

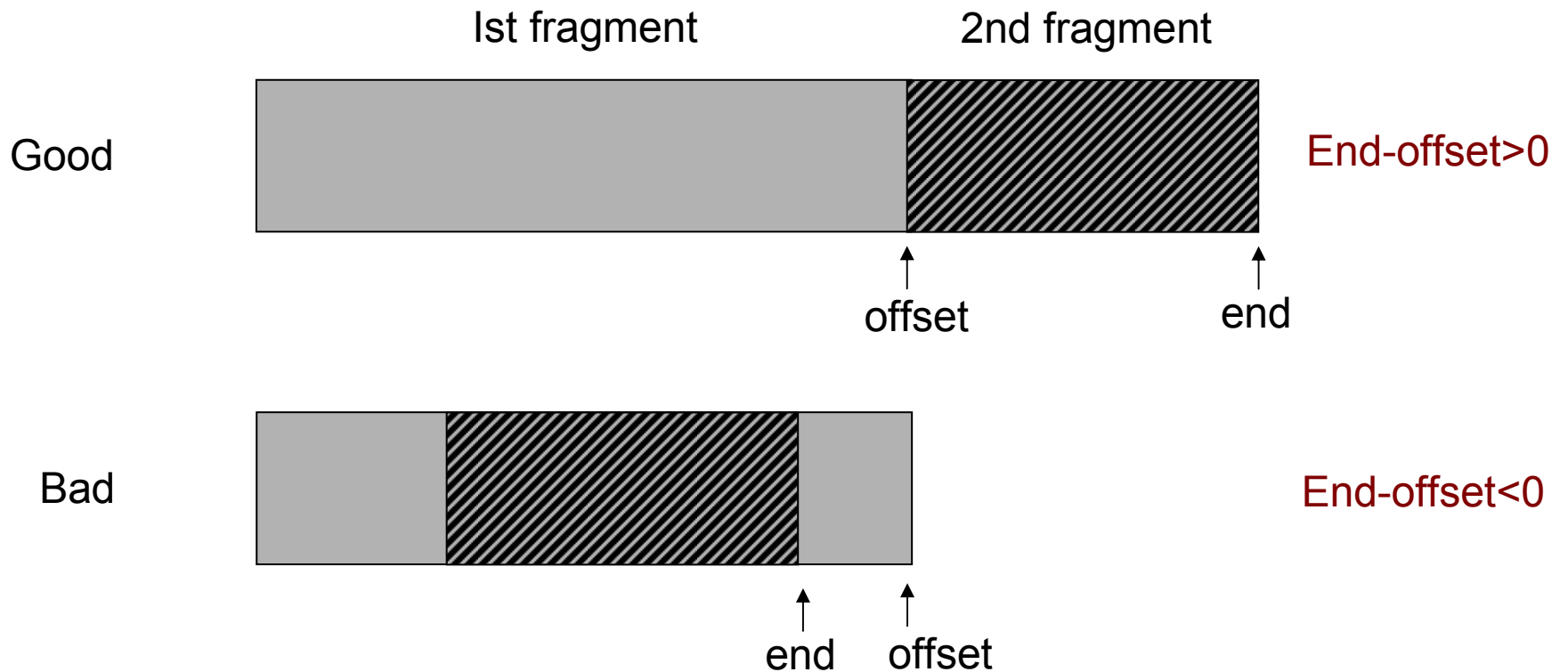
# Protocol Vulnerabilities

---

- Vulnerabilities in the network stack
  - Network code runs in the kernel (crash)
- Examples
  - Land: SYN packet with same src/dest addrs/ports
  - Reassembly: Teardrop (UDP), Ping of Death (ICMP)
- Still relevant?
  - Vulnerabilities resurface: MS 1997 (March,2005)
  - New Protocols: IPv6
- Tools: IP Stack Integrity Checker (ISIC)

# Teardrop (Example)

<http://users.tkk.fi/~lhuovine/study/hacker98/dos.html>



# Protocol Design

---

- Protocols often not designed for malicious adversary
  - Adaptation (network conditions)
  - Determinism
- Low-rate (average) TCP denial of service
  - Congestion control mechanism (packet drops)
  - Sample round trip time
  - Retransmission time-out (RTO)
  - Reduce transmission rates/volume exponentially

# References

---

- **Computer Security: Art and Science**
  - Matt Bishop
- **Information Security**
  - Cristina Nita-Rotaru
- **Security in Computing**
  - Charles P. Pfleeger and Shari Lawrence Pfleeger
- **Internetworking with TCP/IP**
  - Douglas E. Comer
- **Low-Rate TCP-Targeted Denial of Service Attacks**
  - Aleksandar Kuzmanovic and Edward W. Knightly