Introduction to Network Security

Chapter 7

Transport Layer Protocols

Topics

- TCP Layer
  - Responsible for reliable end-to-end transfer of application data.
- TCP vulnerabilities
- UDP
- UDP vulnerabilities
- DNS

TCP Services

Multiplexing:

- A process within a host using TCP service is identified with a **port**. A port, when concatenated with an internet address, forms a **Socket**, which is unique throughout the internet. Service provided by TCP is provided by means of a logical connection between a pair of sockets.

Multiplexing service

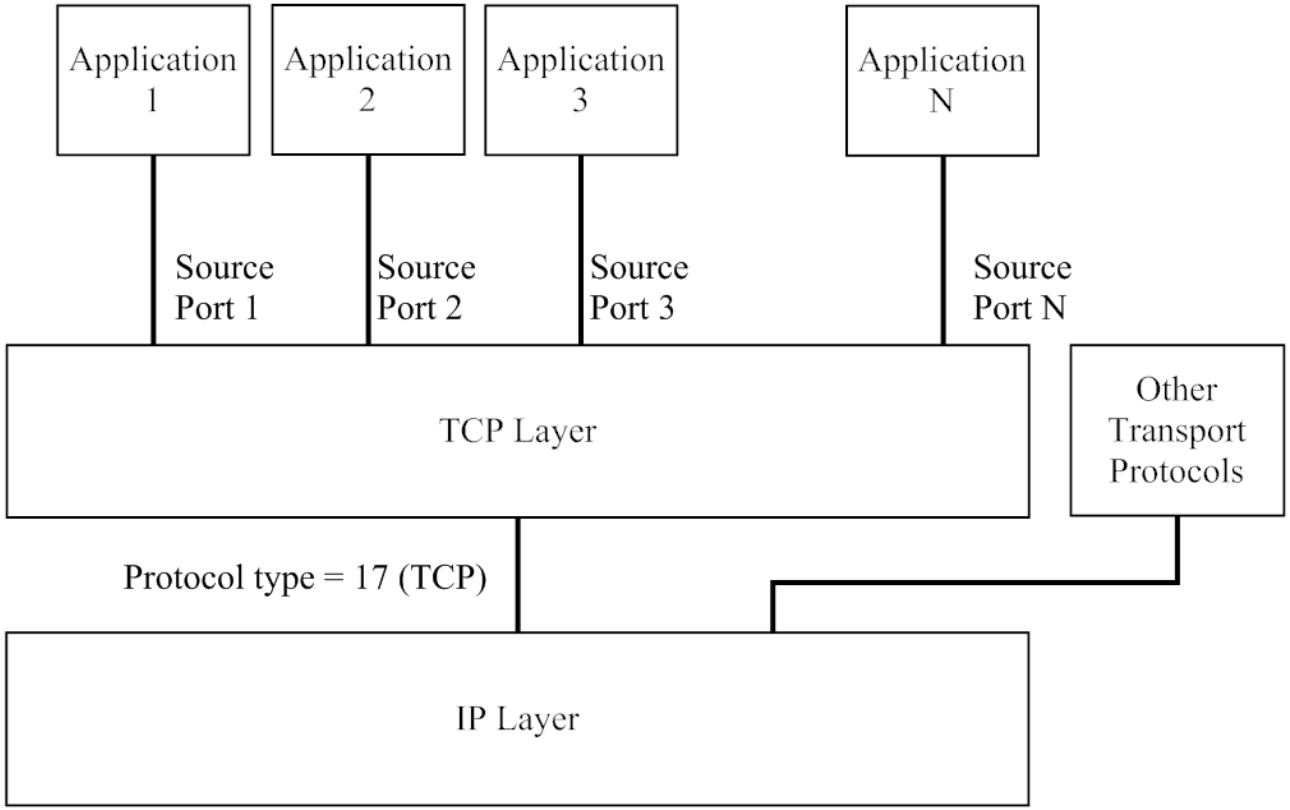


Figure 7.1 TCP Multiplexing

TCP port numbers

5	RJE
7	echo
9	Discard
11	Active Users
13	daytime
15	Who is up
17	Quote of the day
19	Character Generator
20	FTP (default data)
21	FTP (control)
23	TELNET
25	SMTP
37	Time
42	Host name service
53	Domain name server
67	BOOTP
68	Bootstrap Protocol Client
69	Trivial FTP
75	any private dialout service
77	any Private RJE service
79	FINGER
101	NIC host name server
102	ISO-TSAP
103	X.400
104	X.400-SND
105	CSnet Name server
109	Post Office Protocol Ver 2
113	Authentication Service
115	Simple FTP
119	NNTP
123	NTP
161	SNMP agent
162	SNMP management station

TCP Connection Management

Consists of three services:

- Connection Establishment: Allow two TCP users to setup a logical connection between their respective sockets. A connection may be setup if:
- No connection between the two sockets currently exists. From a given socket, it is possible to simultaneously maintain more than one connection, but only one connection to any specific remote socket at a time is permitted.
  - Internal TCP resources are sufficient.
  - Both users have agreed to the connection.

TCP Connection Management

- Connection Maintenance service provides for the exchange of data between the two sockets and supports the data transport (described in the next slide).
- Connection Termination may be either abrupt or graceful. With abrupt termination, data in transit may be lost. A graceful termination prevents either side from shutting down until all data have been received.

TCP Data Transport

- Full Duplex: Both users may transmit at once.

- Timely: The user may request timely delivery of data by associating a timeout with data submitted for transmission. If TCP detects a timeout the connection is abruptly terminated.
- Ordered: TCP is stream oriented. TCP guaranteed that the stream of data presented by one user to TCP will be delivered in the same order to the destination user.
- Labeled: TCP establishes a connection only if the security designation provided by both users match.
- Flow Control: Used to prevent internal TCP congestion
- Error Control: TCP uses a simple checksum.

TCP

- **Stream Orientation -** When two application processes transfer large volumes of data, we can think of the as a stream of bits divided into 8-bit bytes The stream service on the destination passes the same sequence of octets to the receiver that the sender passed to the source machine. Data are not treated as packets but as a stream of data that is passed to the transport entity. The transport entity will divide the data into packets for transmission to the destination. The destination transport entity will pass the data to the user as a stream.

TCP Stream

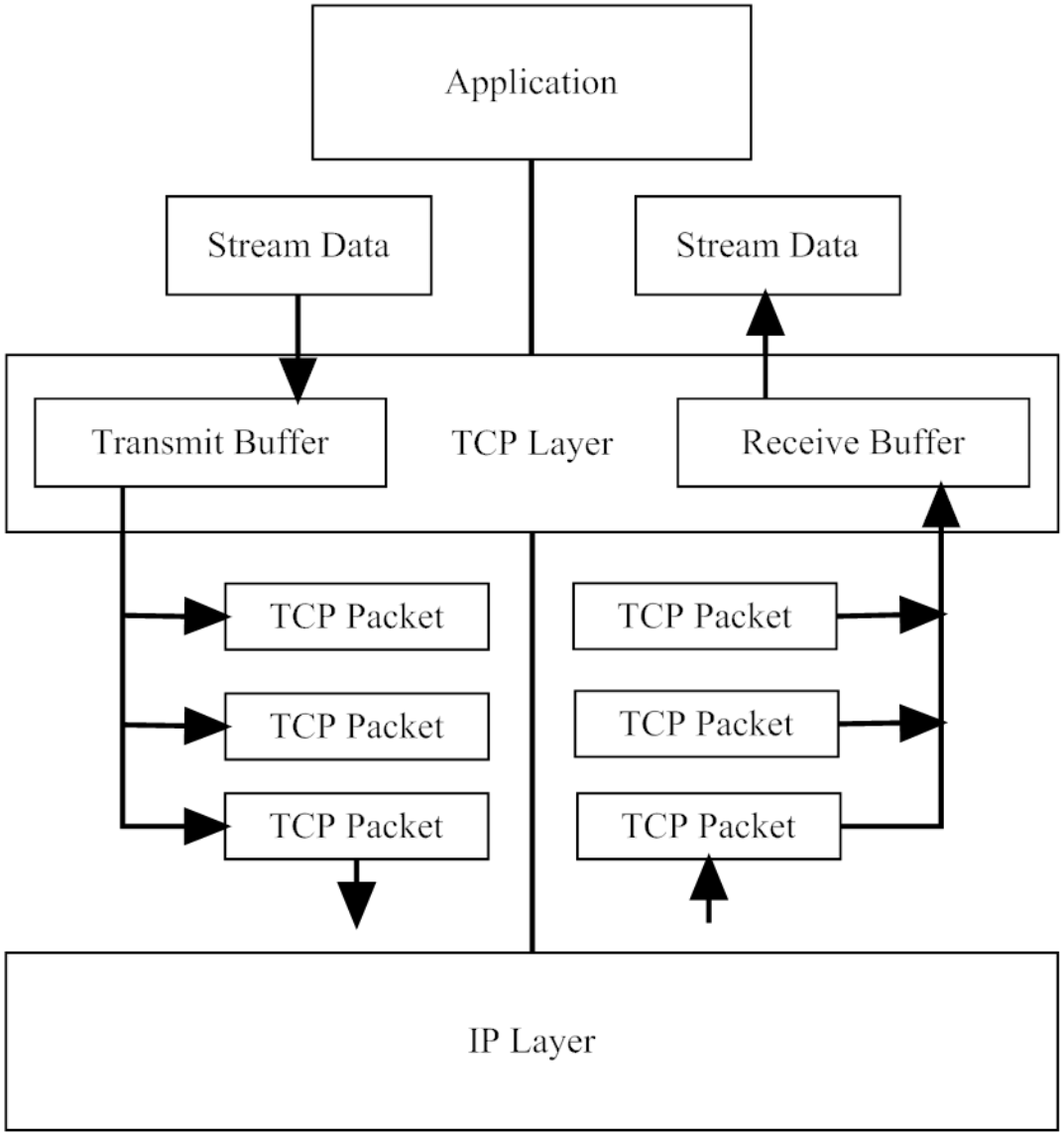


Figure 7.2 TCP Stream Service

TCP Special Capabilities

TCP supports two special capabilities associated with the transfer of data

- Data Stream Push: Used to force the delivery of all data waiting to be sent.
- Urgent Data Signaling: Provides a means of informing the destination TCP user that urgent data is in the incoming data stream.

TCP Error Reporting

- TCP will report service failure stemming from catastrophic conditions

TCP Services

- Unspecified Passive open
- Fully Specified Passive Open
- Active Open
- Active Open with data
- Send
- Deliver
- Allocate
- Close
- Abort
- Terminate
- Error

TCP Protocol

Connection Establishment:

- TCP uses a three handshake for connection establishment. We will see TCP defines only one packet format that contains flags to indicate what type of packet it is. The connection packets have the SYN flag set.

TCP 3-way Handshake

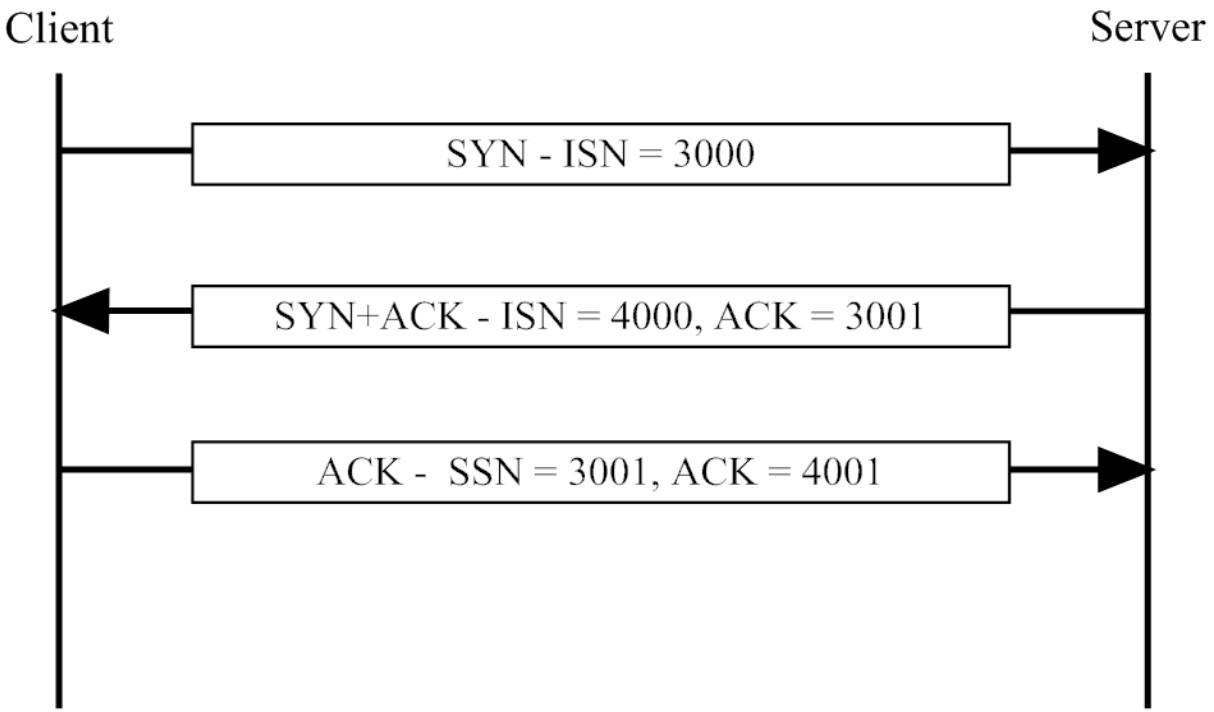


Figure 7.3 TCP Connection Establishment

TCP Protocol

Data Transfer:

- Sequence numbers are used for data transfer. The sequence numbers represent the number of bytes not the number of packets. Flow control is handled by using a credit allocation scheme as describe earlier.

TCP Data Transfer

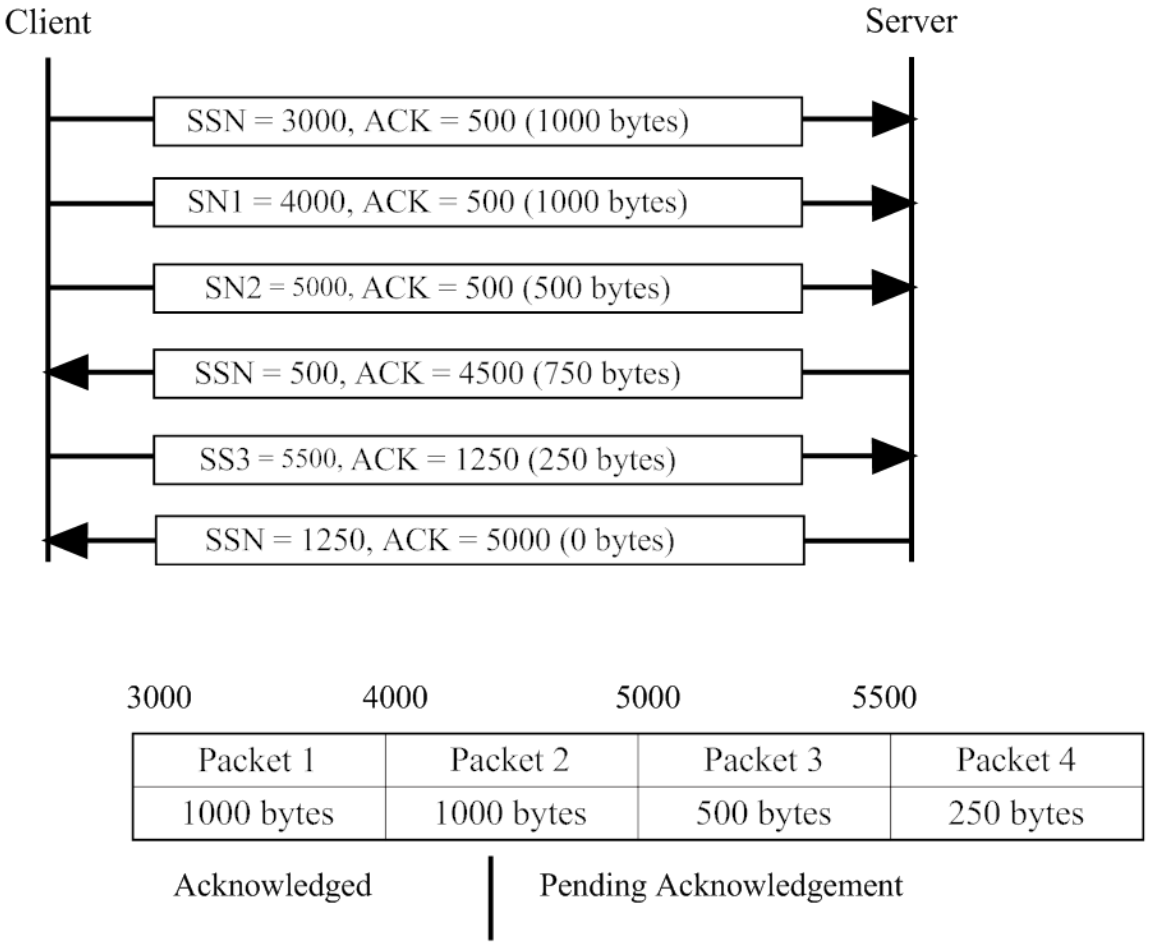


Figure 7.4 TCP Data Transfer

TCP Connection Termination

Connection Termination:

- The connection is terminated by sending a packet with the FIN flag set. This packet contains the number of the last packet sent.

TCP Connection termination

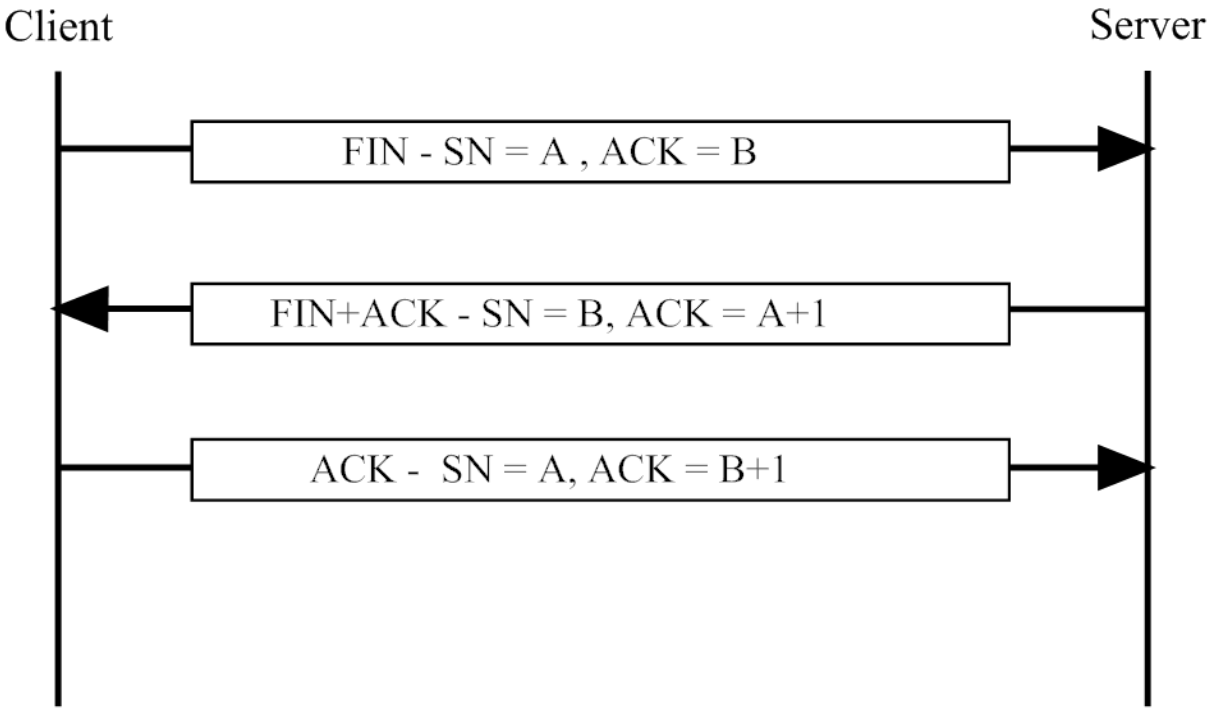


Figure 7.5 TCP Graceful Termination

TCP Header Format

Source Port			Destination Port
Sequence Number			
Acknowledgement Number			
Hdr-Len	Reserved	Flags	Window Size
Checksum			Urgent Pointer
Options			

Flags

URG	ACK	PSH	RST	SYN	FIN

Flag	Function
URG	Packet contains urgent data
ACK	Acknowledgment number is valid
PSH	Data should be pushed to the application
RST	Reset Packet
SYN	Synchronize packet
FIN	Finish packet

Figure 7.6 TCP Header Format

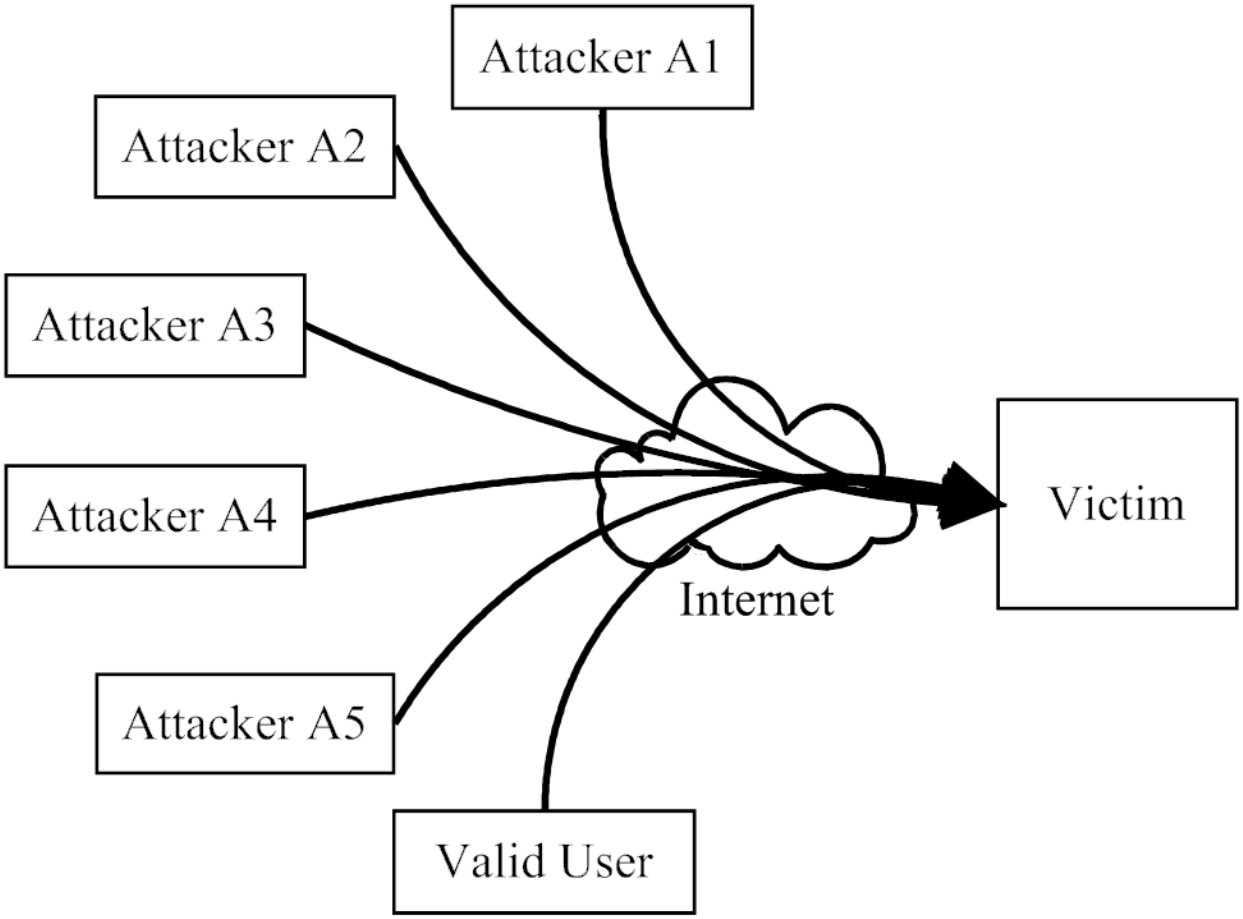
Header Based

- There have been several attacks using invalid flag combinations.
- Most have been fixed, however this is now used to help determine the type of operating system
  - Probing attacks
    - Invalid header responses
    - Initial values
      - sequence numbers
      - Window size

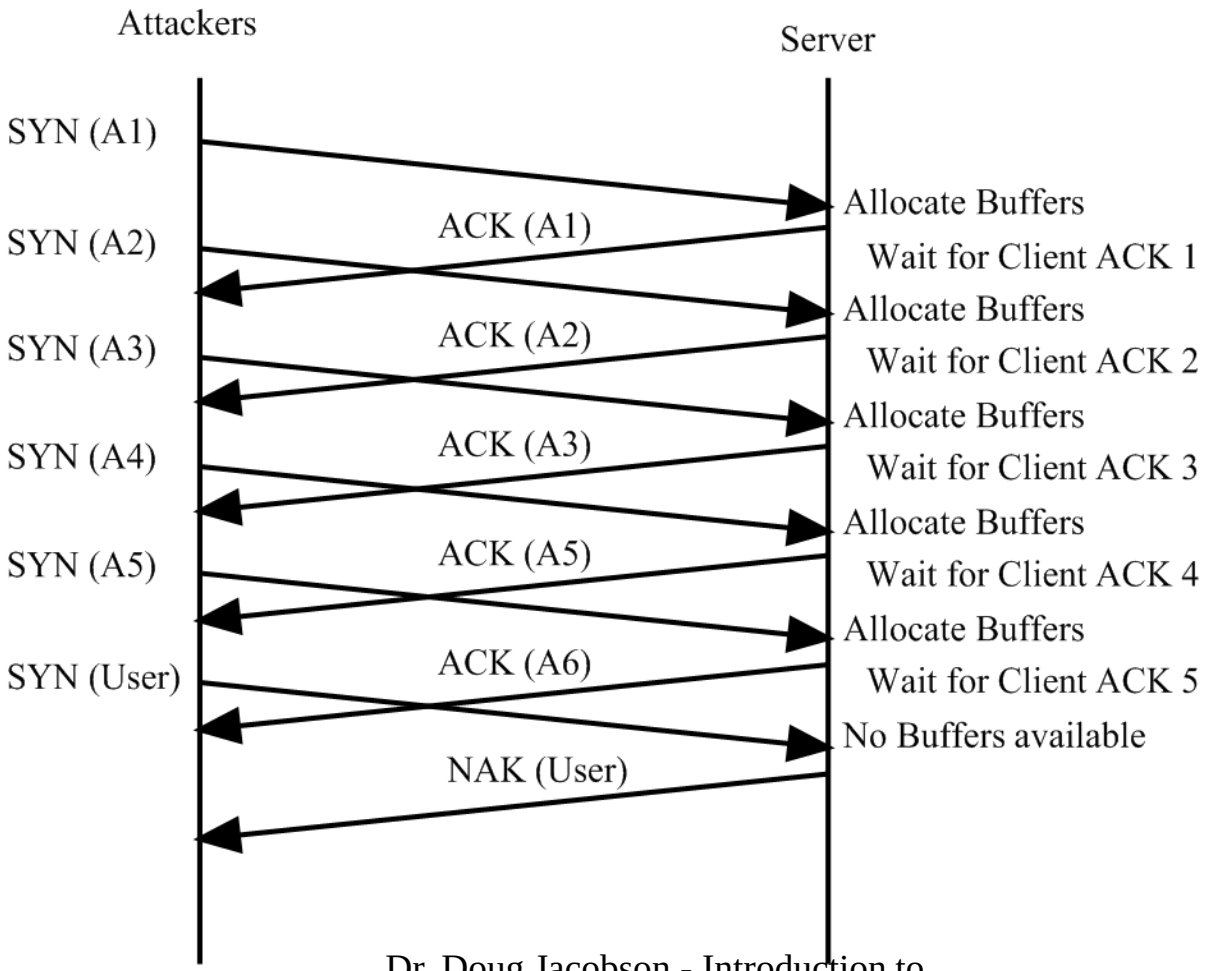
Protocol Based

- Syn flood
- Reset Packets
- Session Hijacking

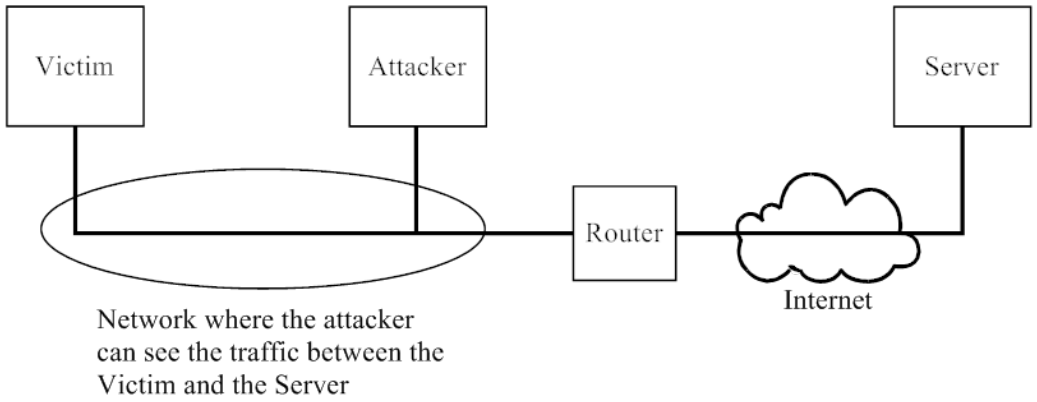
SYN Flood


Attacker A1
Attacker A2
Attacker A3
Attacker A4
Attacker A5
Valid User
Internet
Victim

SYN Flood


Attackers
Server
SYN (A1)
Allocate Buffers
ACK (A1)
SYN (A2)
Wait for Client ACK 1
Allocate Buffers
ACK (A2)
SYN (A3)
Wait for Client ACK 2
Allocate Buffers
ACK (A3)
SYN (A4)
Wait for Client ACK 3
Allocate Buffers
ACK (A5)
SYN (A5)
Wait for Client ACK 4
Allocate Buffers
ACK (A6)
SYN (User)
Wait for Client ACK 5
No Buffers available
NAK (User)

Reset Shutdown



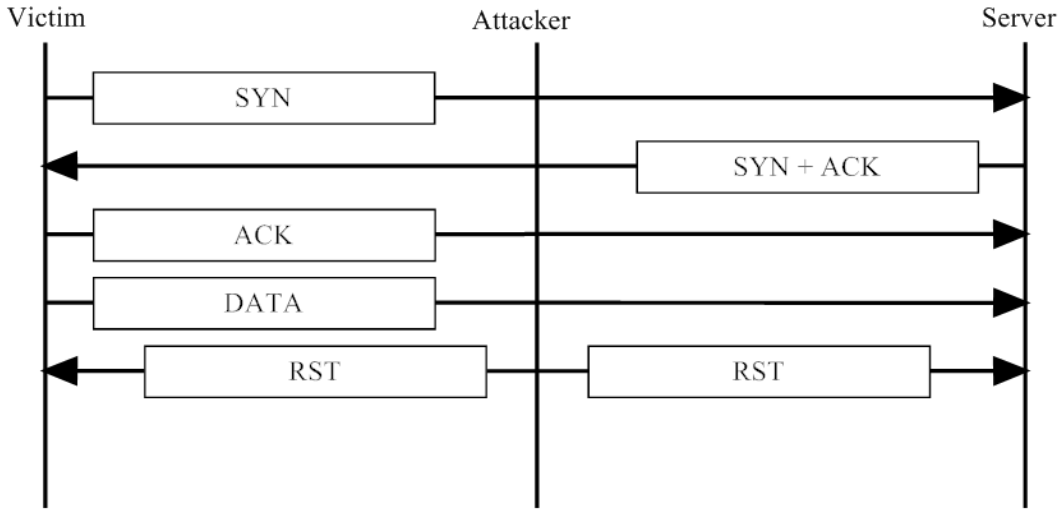
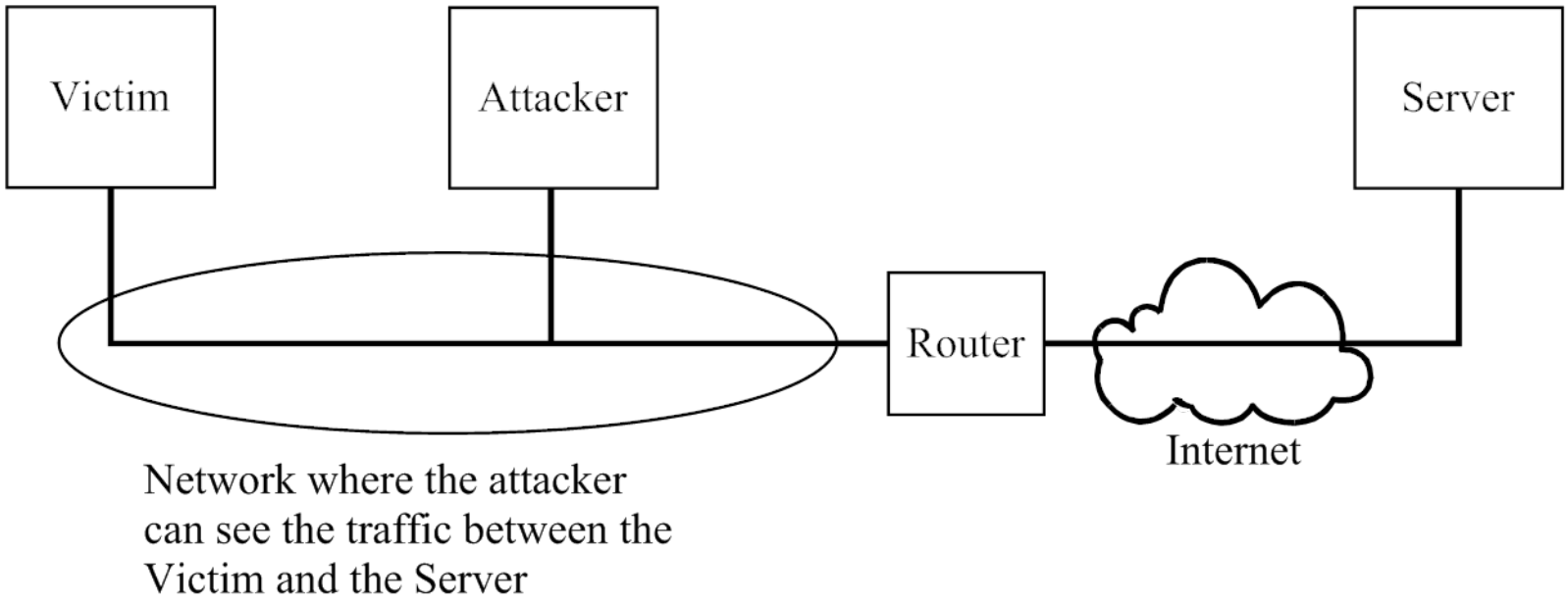
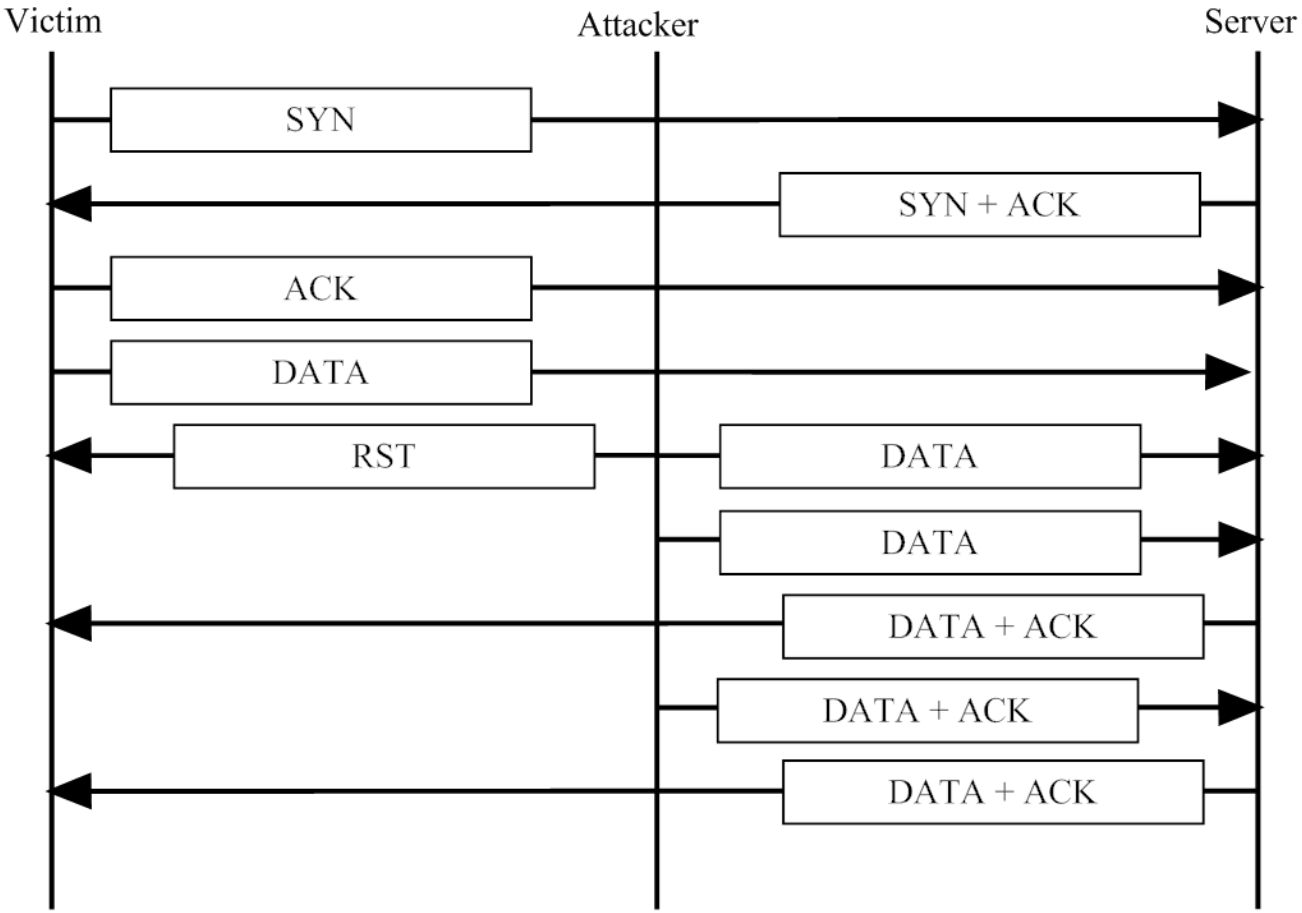


Figure 7.8 RST Connection Shutdown

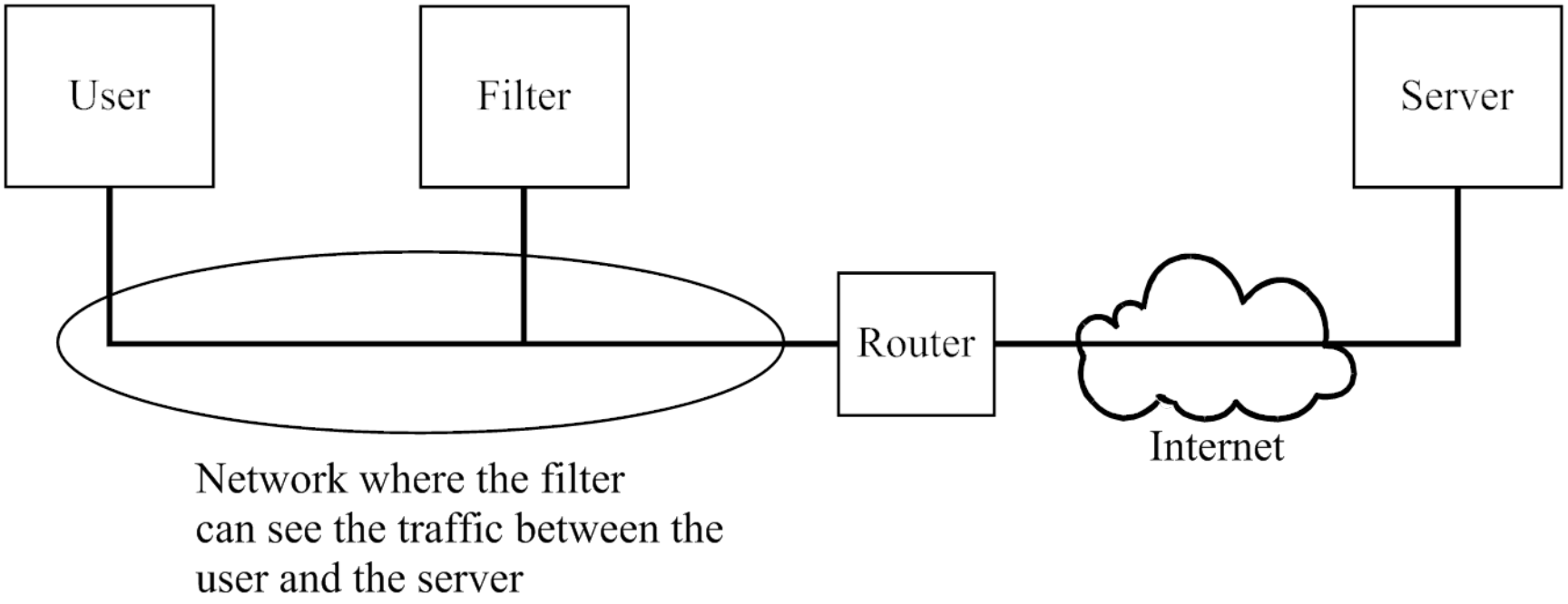
Session Hijacking



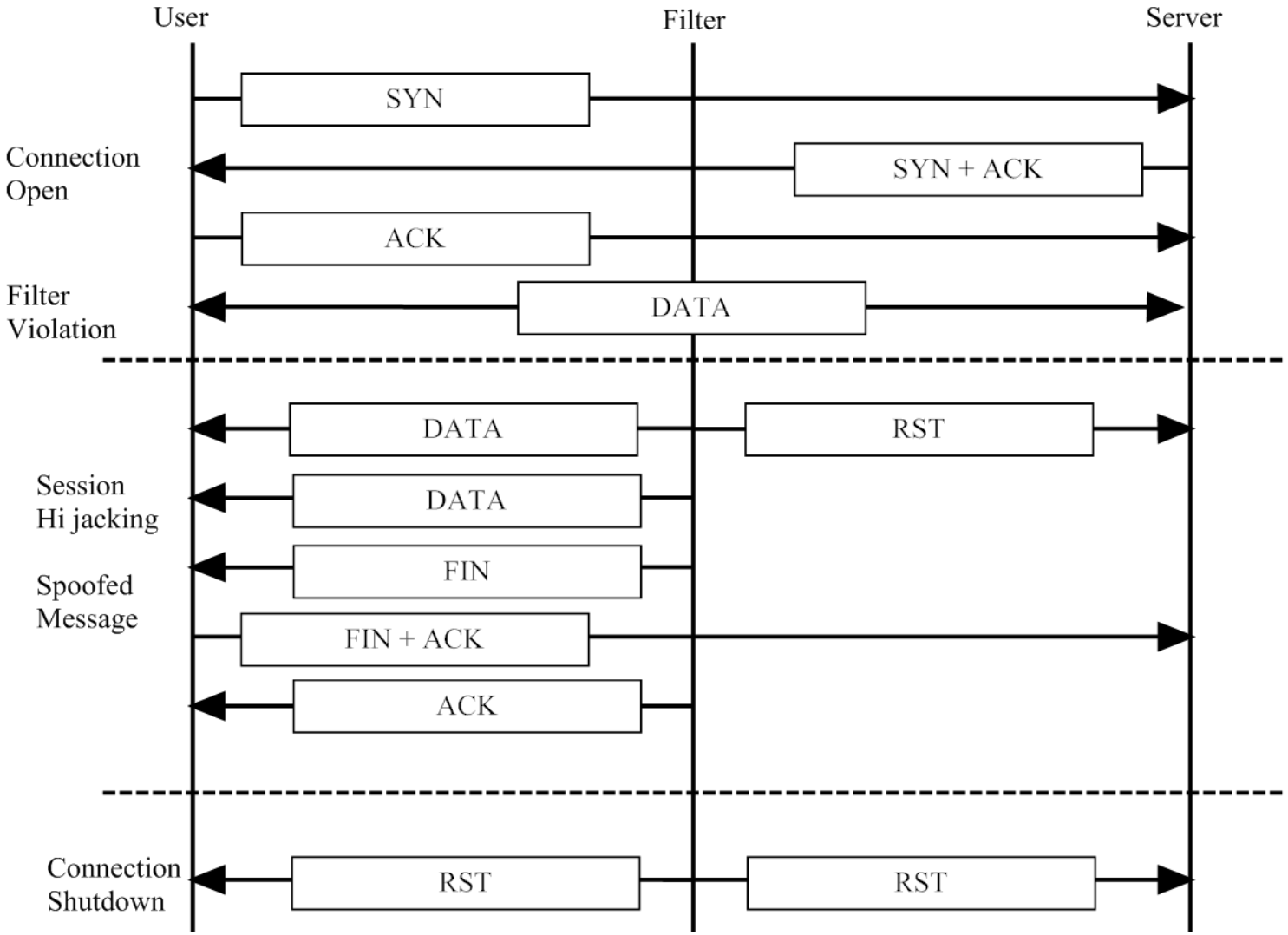
Session Hijacking



Passive Network Filter



Passive Network Filter



Mitigation

- Encryption can fix Session hijacking
- Reset is harder
- Syn flood is hard

Authentication Based

- No authentication in TCP
- Ports might be considered an authentication of the application

Traffic Based

- Flooding (using all of the TCP resources)
- QOS
- Sniffing

User Datagram Protocol

- Designed to allow connectionless protocols
- Typical applications will send one packet and wait for a single response.

Source Port	Destination Port
UDP Total Length	Checksum

UDP Attacks

- Header & Protocol: None since there is no protocol and very simple header
- Authentication: same as TCP
- Traffic: typically not a problem. Sniffing is a potential problem, but most UDP protocols don’t try to hide data. Flooding is hard with UDP.
- Mitigation: Most organizations block all UDP except port 53 (DNS)

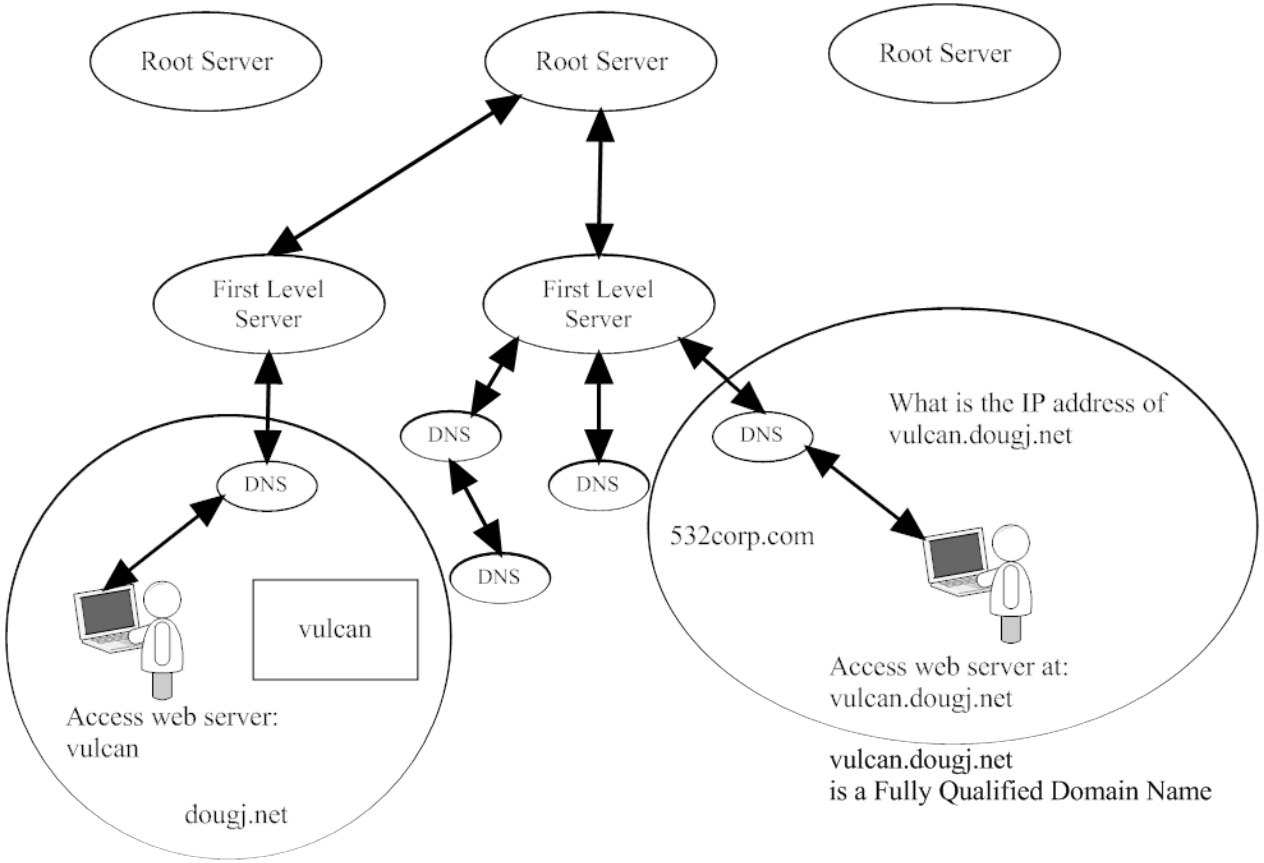
Domain Name Service

- Designed to give organizations a way of controlling their name space
- Distributed control over computer name to IP address mapping
- DNS normally uses UDP and port 53
  - If the answer is bigger than 512 bytes, can use TCP

Domain Names

- Tree Structure - max 128 levels, root = level 0
- Domain name: www.iastate.edu
  - Each name between the dots is called a **label**
  - Label <= 63 characters
- Fully qualified domain name: www.iastate.edu.
  - Adds “.” at the end
- Partially qualified domain name
  - Supported by the client
  - The leftmost part of a domain name
  - E.g., www. Gets filled in to www.iastate.edu by the client

DNS Name Space



vulcan	.	dougj	.	net
Label	.	Label	.	Label

Server Types

- Server Types
  - Root Server
  - Primary Server
  - Secondary Server
- Can only push data from Primary to Secondary (not Secondary to Primary)

DNS Queries

- DNS Queries
  - Name to Address
  - Address to Name
- Resolver: Client code that queries DNS using two lookup methods:
  - Recursive
  - Iterative

Reverse Query

- IP to Name
- 129.186.5.100 – what is its name
- Query is made to:
  - 100.5.186.129.in-addr.arpa.
- This way it can be parsed just like a name
  - 129 then 186 then 5 then 100

Reverse Lookups

- IP to Name conversion
- Not all IP addresses will resolve to a name

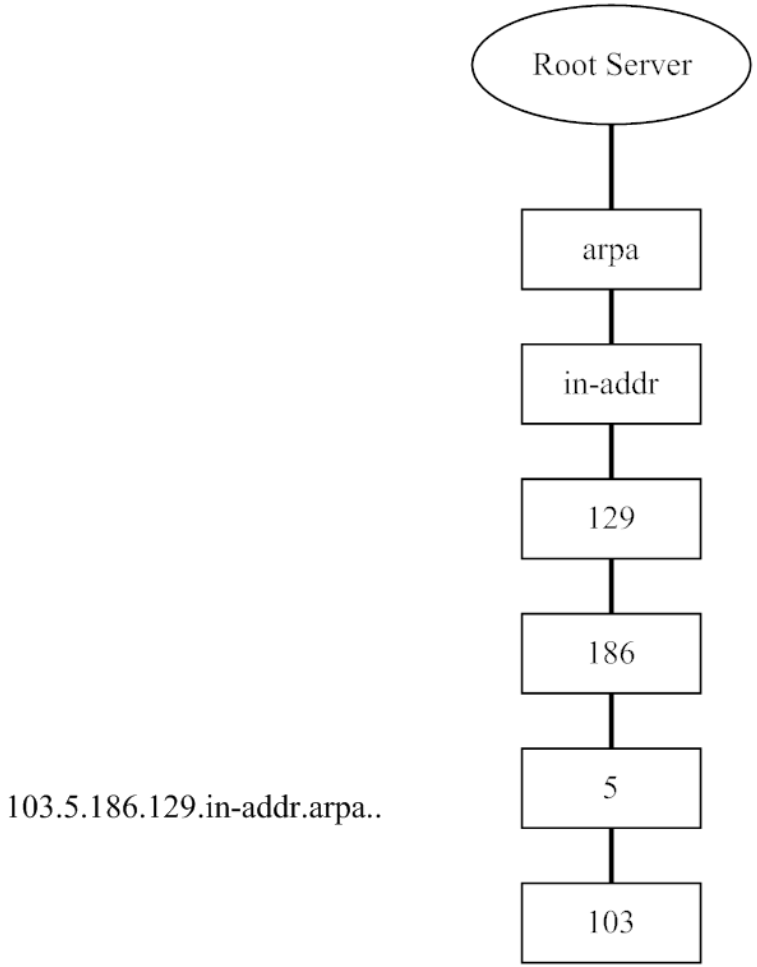


Figure 7-13 DNS Reverse Name Hierarchy

DNS System

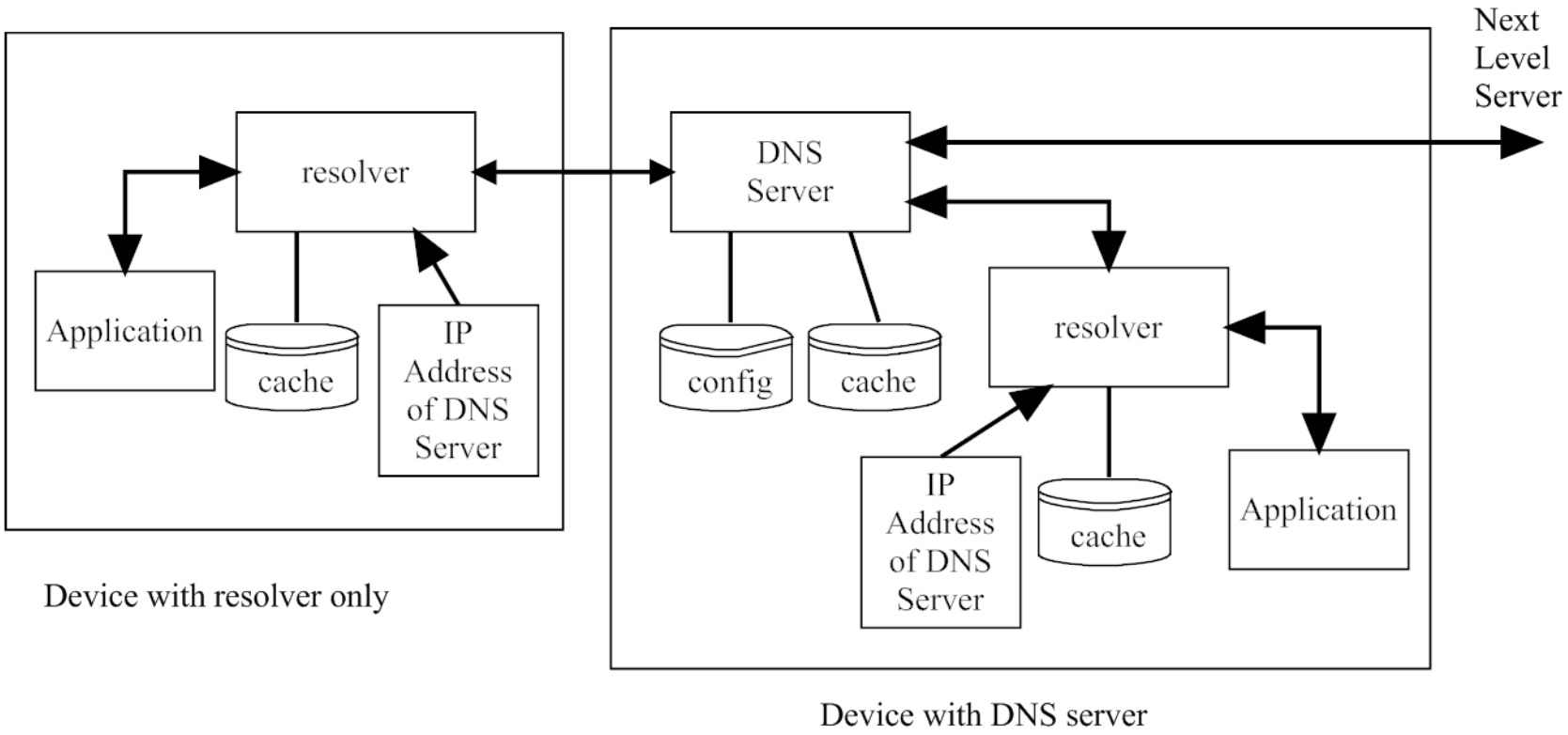


Figure 7.14 DNS System

Recursive Query Method

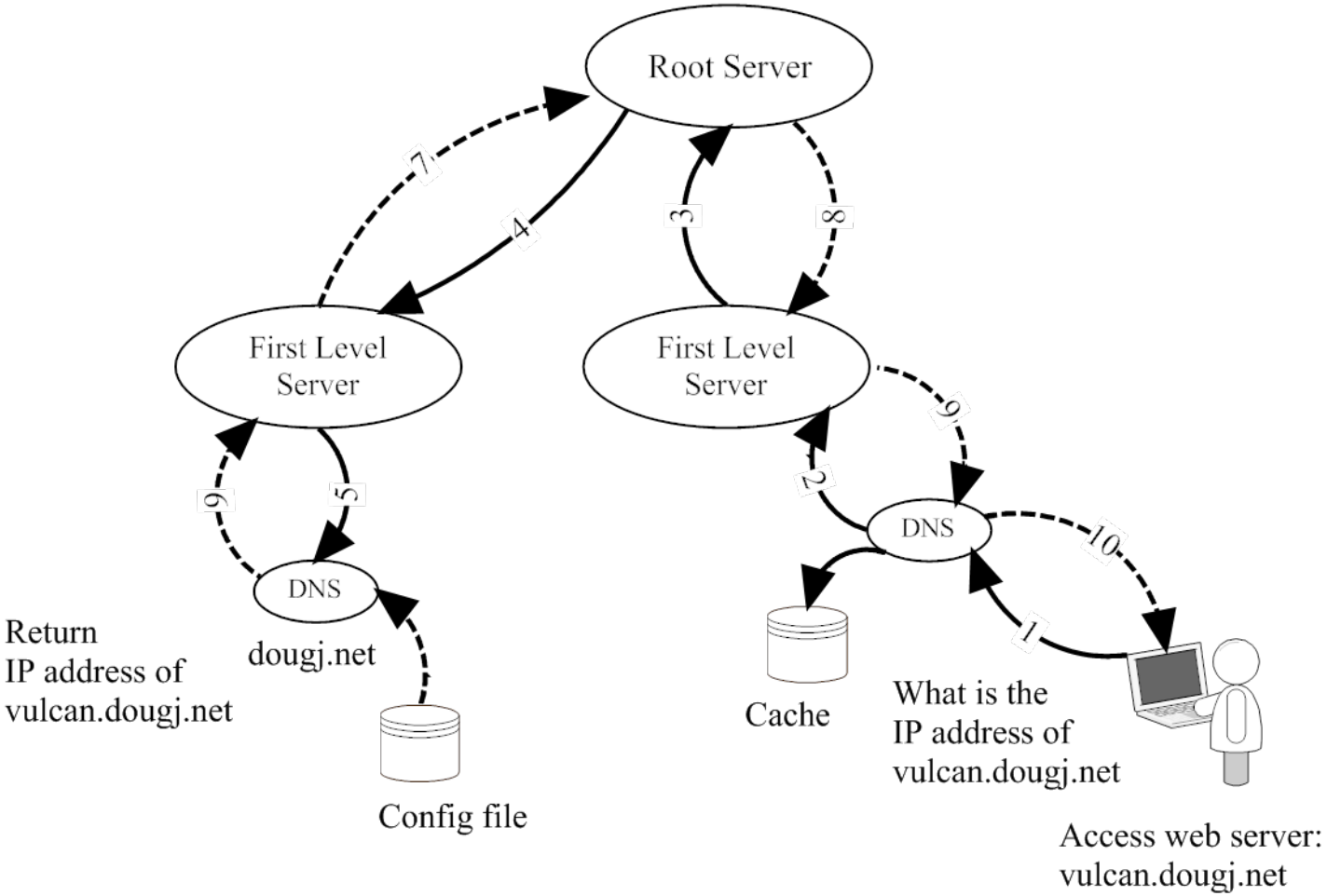


Figure 7.15 DNS Recursive Mode

Iterative Query Method

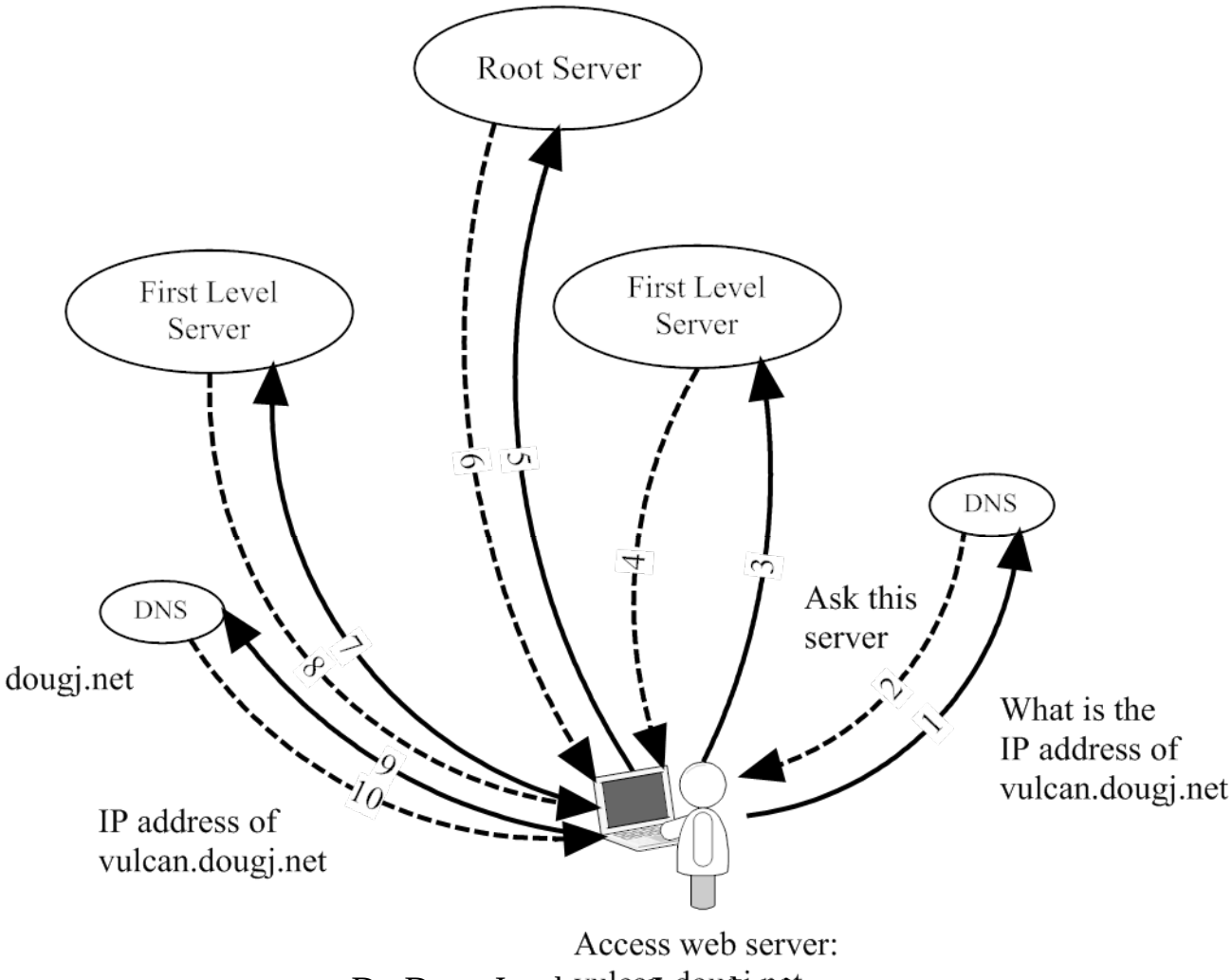


Figure 7.16 DNS Iterative Mode

Responses

- If the answer comes back from any DNS server that has the answer cached it is called unauthoritative
- To handle the stale cache issue there is a time to live for each response.

DNS Uses Two Messages

- Query := two fields
  - header | question
- Response := five fields
  - header | question | answer | authoritative | additional

DNS Packet Format

ID	Flags	
Number of Questions	Number of Answers	Fixed Header
Number of Authoritative Answers	Number of Additional Records	
Question		Question Section
Query Type	Query Class	

Query Packet

QR	Opcode	AA	TC	RD	RA	0	0	0	rCode

Flags

DNS Packet Format

ID	Flags	
Number of Questions	Number of Answers	Fixed Header
Number of Authoritative Answers	Number of Additional Records	
Question		Question Section
Query Type	Query Class	
Answer(s)		
Authoritative Answer(s)		
Additional Records		

Response Packet

QR	Opcode	AA	TC	RD	RA	0	0	0	rCode

Flags

DNS Message Header

- Header = 12 bytes
  - Id = 2 bytes
  - Flags = 2 bytes (see next slide)
  - # of questions = 2 bytes
  - # of answers = 2 bytes (0 in query)
  - # of authoritative answers = 2 bytes (0 in query)
  - # of additional answers = 2 bytes (0 in query)

Flags Field

- 1 bit – Q/R 0=query, 1= response
- 4 bits – opcode
  - 0 = standard
  - 1 = inverse
  - 2 = server status request
- 1 bit AA – 1 = Authoritative answer
- 1 bit TC – 1 = answer > 512 bytes
- 1 bit RA – 1 = recursion available
- 3 bits of zero
- 4 bits – response code (see next slide)

Response codes

- 0 No Error
- 1 format error
- 2 problem at name server
- 3 domain reference problem
- 4 query type not supported
- 5 administratively prohibited

DNS Question section

- Variable length – Query name
- 16 bits – query type
- 16 bits – query class

DNS Query Name

- 6vulcan2ee7iastate3edu0
- Numbers are the count fields, they are in binary
- The count fields are only 6 bits to tell the difference between a count value and a offset pointer used for compression

DNS Types

- 1- A – Address
- 2 – NS – Name server
- 5 – CNAME – Alias
- 6 – SOA – Start of Authority
- 11 – WKS – Well known services
- 12 – PTR – IP to name conversion
- 13 – HINFO – Host info
- 15 – MX – Mail exchange
- 28 – AAAA – IPV6 address
- 252 – AXFR – Request a zones transfer
- 255 – ANY – Request all records

DNS Resource Record

- Domain name – Variable length (pointer to the name in the query section
- Domain type (16 bits) same as query
- Domain class (16 bits) same as query
- Time to Live (32 bits) number of seconds, 0 = don't cache
- Resource data length (16 bits)
- Resource data (variable length)

Resource data

- Number (4 bytes – V4)
- Domain name (variable length)
- Offset pointer (upper two bits of first byte = 11
- Char string – 1 byte length followed by characters

Compression

- 11 [address of the beginning byte]
- 12 is the first byte of the question section

Header & Protocol attacks

- Header
  - Not many attacks, bad headers are rejected.
  - Can be used to leak data through a firewall
- Protocol
  - Simple protocol
  - Can use the DNS port number to communicate through a firewall

Authentication

- Bad DNS Entries
  - Break in DNS server
  - Rouge DNS server
  - DNS cache poisoning
  - Bogus DNS replies
- Scope of Damage

DNS attack damage scope

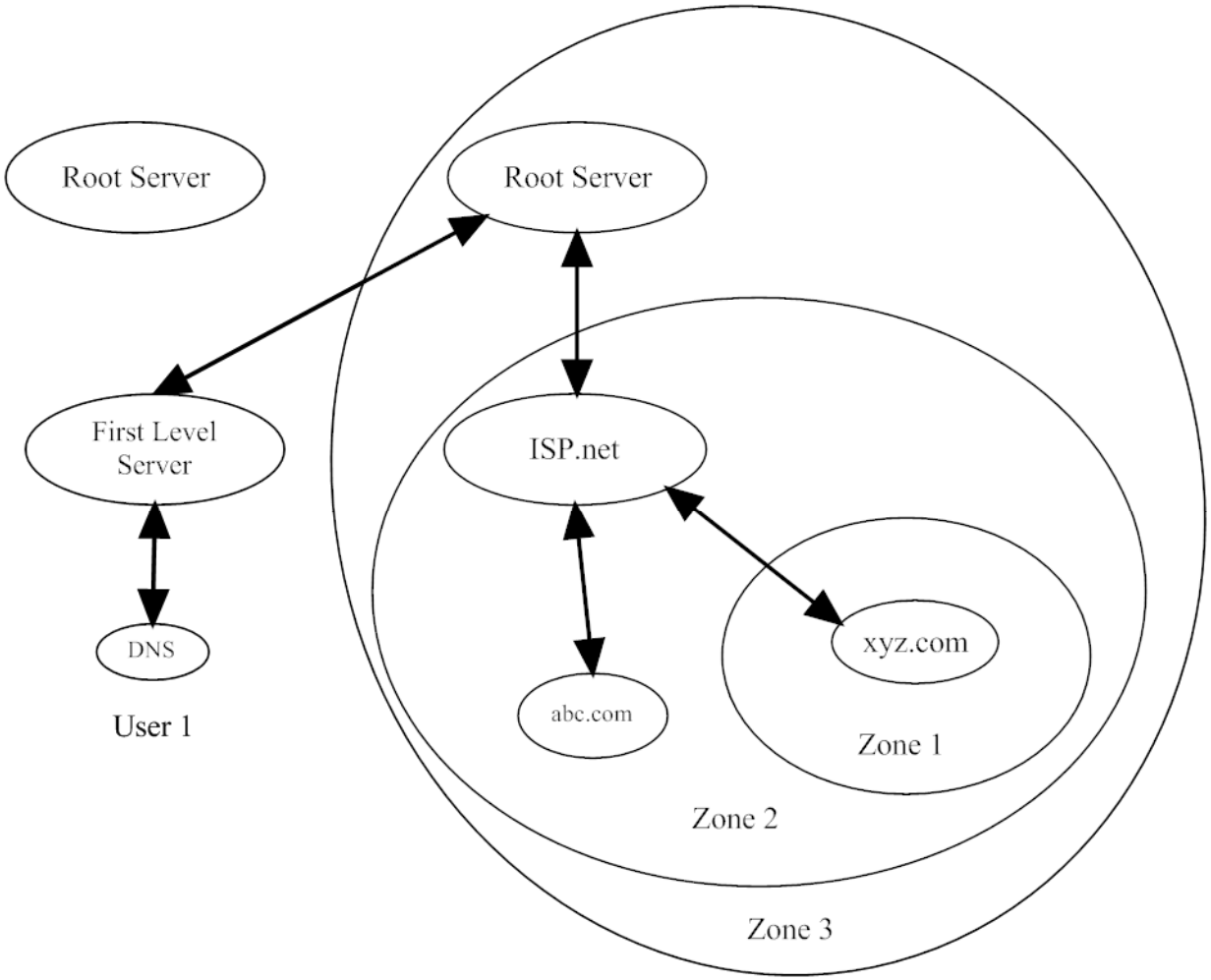


Figure 7.18 DNS Attack Damage Scope

Traffic

- DNS server flooding can cause delayed to dropped responses. DNS client will try 4 times so they often will get an answer
- Sniffing is not a problem

DNS

- DNSSEC is a new protocol and server that offers authenticated DNS with certificates.
  - Not widely adopted
- DNS is a major weak point in the Internet. Taking down the DNS system can take down the entire Internet.

Transport Layer Security

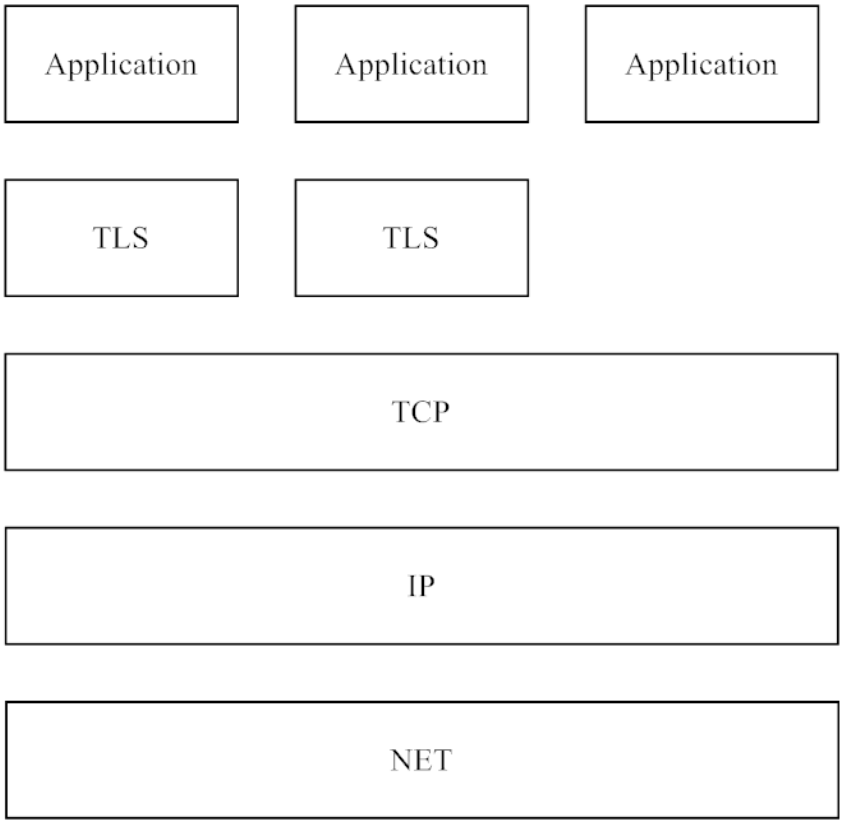


Figure 7.19 TLS Stack

TLS Protocol

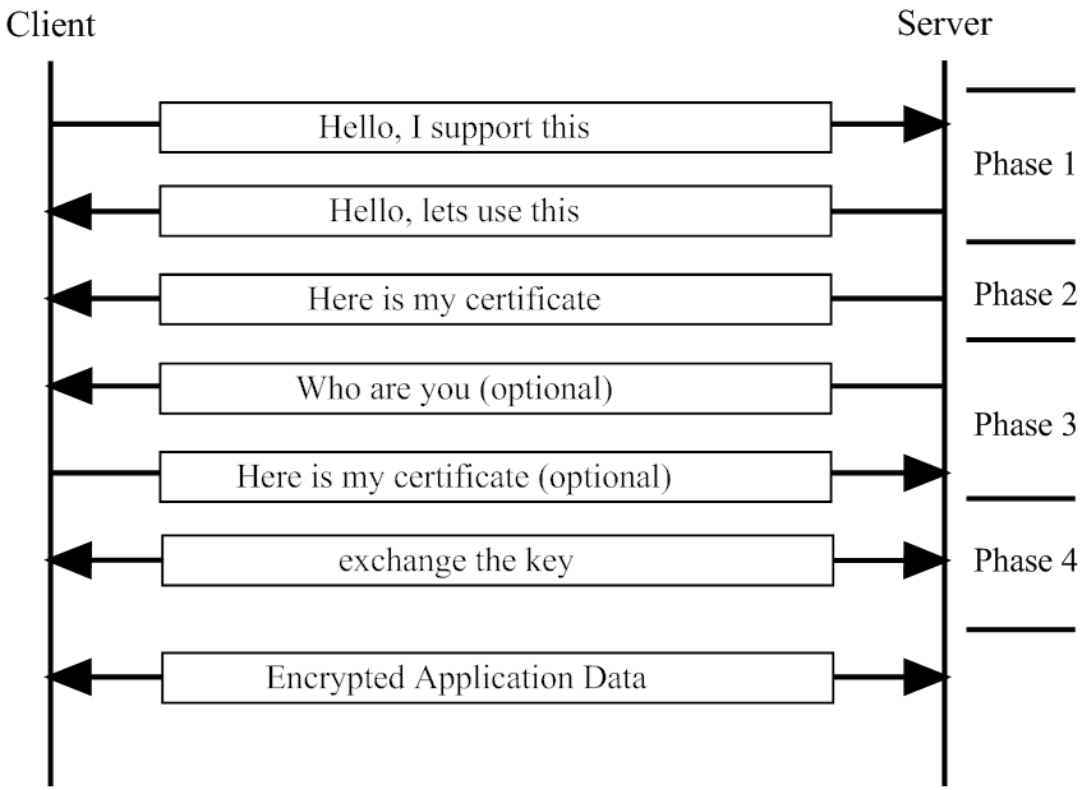


Figure 7.20 TLS Protocol