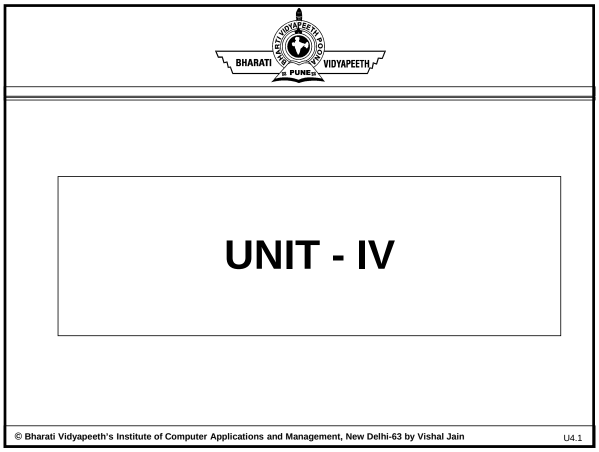
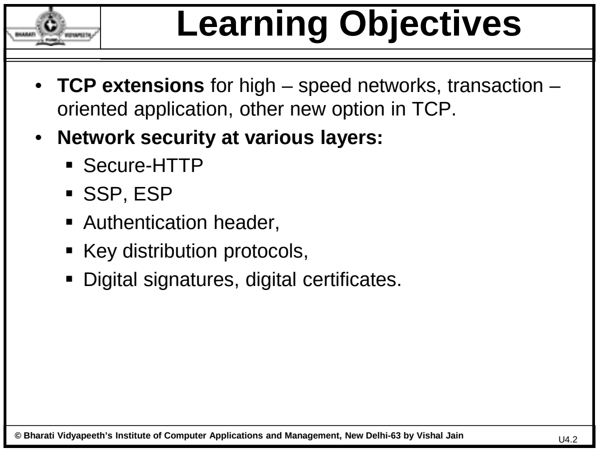
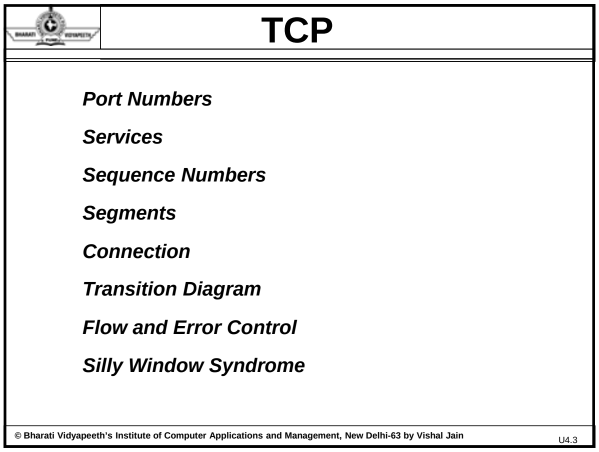


UNIT - IV



Learning Objectives

- **TCP extensions** for high – speed networks, transaction – oriented application, other new option in TCP.
- **Network security at various layers:**
  - Secure-HTTP
  - SSP, ESP
  - Authentication header,
  - Key distribution protocols,
  - Digital signatures, digital certificates.



TCP

Port Numbers

Services

Sequence Numbers

Segments

Connection

Transition Diagram

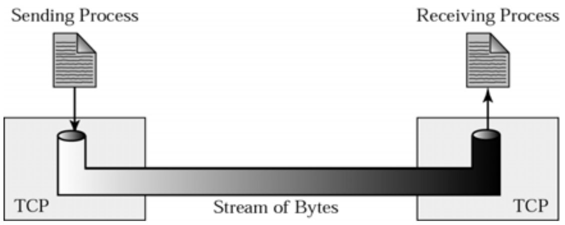
Flow and Error Control

Silly Window Syndrome

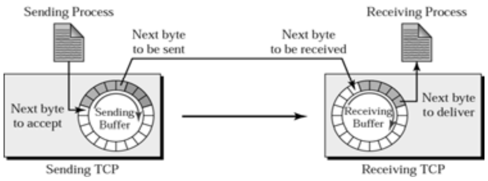
Well-Known Ports Used by TCP

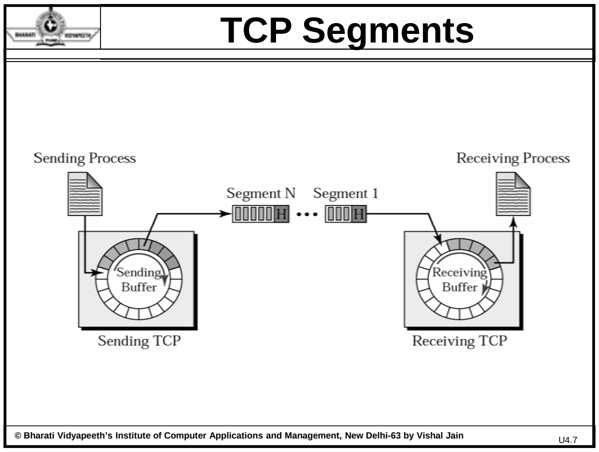
Port	Protocol	Description
7	Echo	Echoes a received datagram back to the sender
9	Discard	Discards any datagram that is received
11	Users	Active users
13	Daytime	Returns the date and the time
17	Quote	Returns a quote of the day
19	Chargen	Returns a string of characters
20	FTP, Data	File Transfer Protocol (data connection)
21	FTP, Control	File Transfer Protocol (control connection)
23	TELNET	Terminal Network
25	SMTP	Simple Mail Transfer Protocol
53	DNS	Domain Name Server
67	BOOTP	Bootstrap Protocol
79	Finger	Finger
80	HTTP	Hypertext Transfer Protocol

Stream Delivery

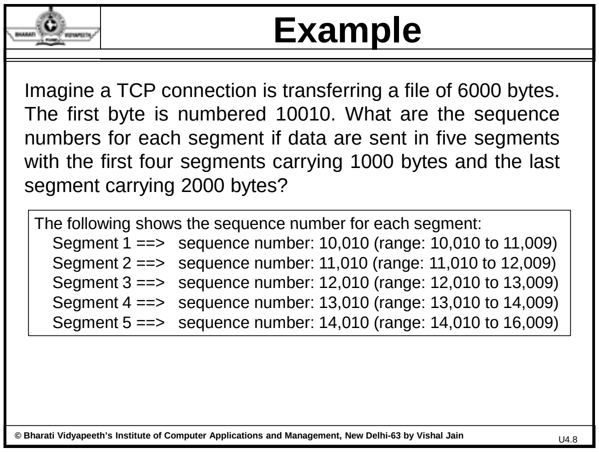


Sending and Receiving Buffers





TCP Segments

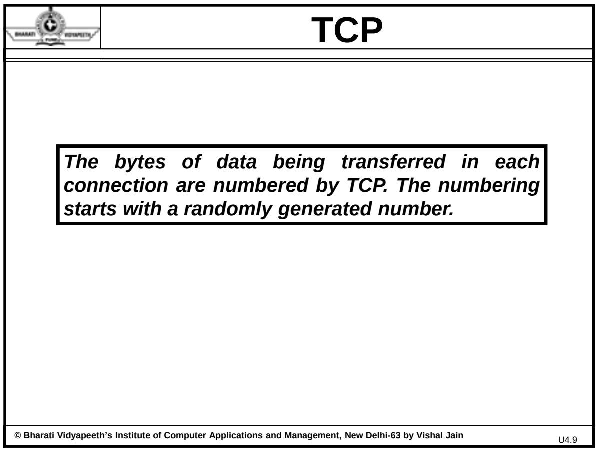


Example

Imagine a TCP connection is transferring a file of 6000 bytes. The first byte is numbered 10010. What are the sequence numbers for each segment if data are sent in five segments with the first four segments carrying 1000 bytes and the last segment carrying 2000 bytes?

The following shows the sequence number for each segment:

Segment 1 ==> sequence number: 10,010 (range: 10,010 to 11,009)
Segment 2 ==> sequence number: 11,010 (range: 11,010 to 12,009)
Segment 3 ==> sequence number: 12,010 (range: 12,010 to 13,009)
Segment 4 ==> sequence number: 13,010 (range: 13,010 to 14,009)
Segment 5 ==> sequence number: 14,010 (range: 14,010 to 16,009)



TCP

The bytes of data being transferred in each connection are numbered by TCP. The numbering starts with a randomly generated number.

TCP

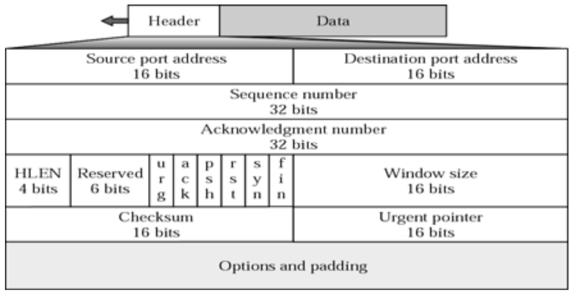
The value of the sequence number field in a segment defines the number of the first data byte contained in that segment.

TCP

The value of the acknowledgment field in a segment defines the number of the next byte a party expects to receive.

The acknowledgment number is cumulative.

TCP Segment Format



Header Data

Source port address 16 bits | Destination port address 16 bits

Sequence number 32 bits

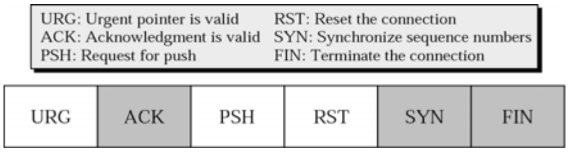
Acknowledgment number 32 bits

HLEN 4 bits | Reserved 6 bits | urg | ack | psh | rst | syn | fin | Window size 16 bits

Checksum 16 bits | Urgent pointer 16 bits

Options and padding

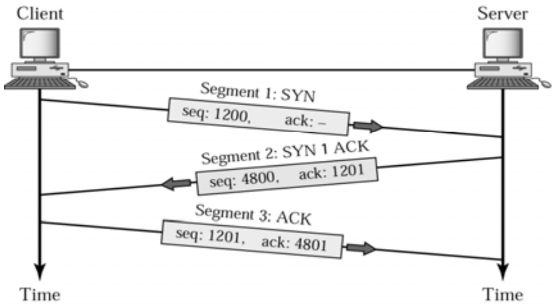
Control Field


URG: Urgent pointer is valid
ACK: Acknowledgment is valid
PSH: Request for push
RST: Reset the connection
SYN: Synchronize sequence numbers
FIN: Terminate the connection
URG
ACK
PSH
RST
SYN
FIN

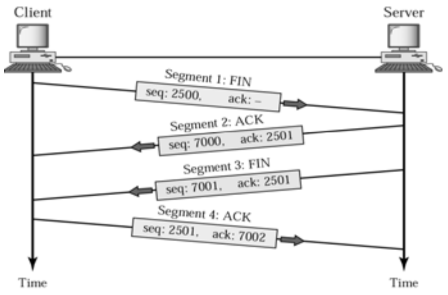
Description of Flags in the Control Field

Flag	Description
URG	The value of the urgent pointer field is valid.
ACK	The value of the acknowledgment field is valid.
PSH	Push the data.
RST	The connection must be reset.
SYN	Synchronize sequence numbers during connection.
FIN	Terminate the connection.

Three-Step Connection Establishment



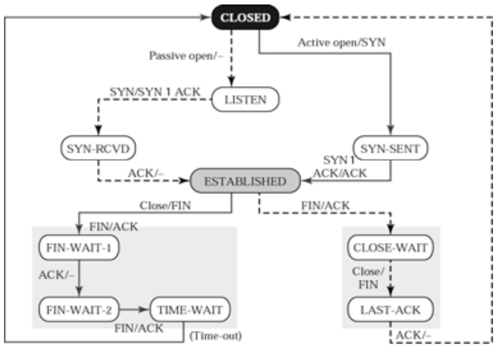
Four-Step Connection Termination



States for TCP

State	Description
CLOSED	There is no connection.
LISTEN	The server is waiting for calls from the client.
SYN-SENT	A connection request is sent; waiting for acknowledgment.
SYN-RCVD	A connection request is received.
ESTABLISHED	Connection is established.
FIN-WAIT-1	The application has requested the closing of the connection.
FIN-WAIT-2	The other side has accepted the closing of the connection.
TIME-WAIT	Waiting for retransmitted segments to die.
CLOSE-WAIT	The server is waiting for the application to close.
LAST-ACK	The server is waiting for the last acknowledgment.

State Transition Diagram



TCP

Flow Control

Flow control regulates the amount of data a source can send before receiving an acknowledgment from the destination. TCP defines a window that is imposed on the buffer of data delivered from the application program.

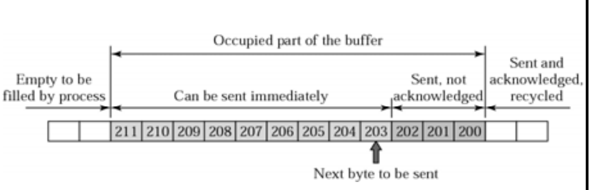
1. Sliding Window Protocol
2. Silly Window Syndrome

TCP

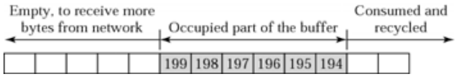
A sliding window is used to make transmission more efficient as well as to control the flow of data so that the destination does not become overwhelmed with data.

TCP's sliding windows are byte-oriented.

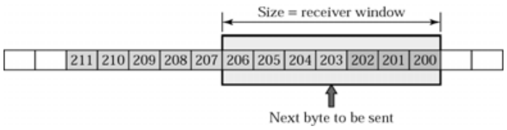
Sender Buffer



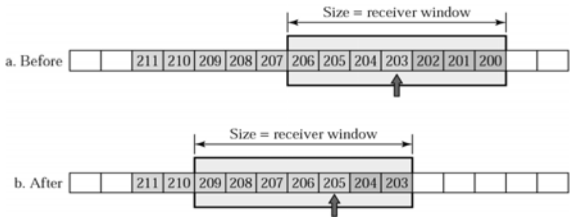
Receiver Window

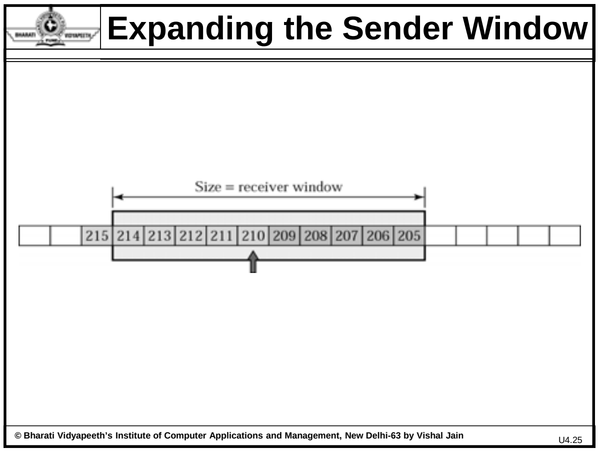

Empty, to receive more bytes from network
Occupied part of the buffer
Consumed and recycled
199
198
197
196
195
194

Sender Buffer and Sender Window



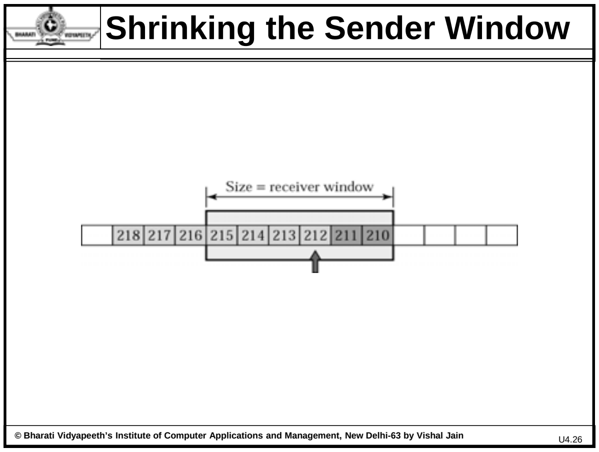
Sliding the Sender Window





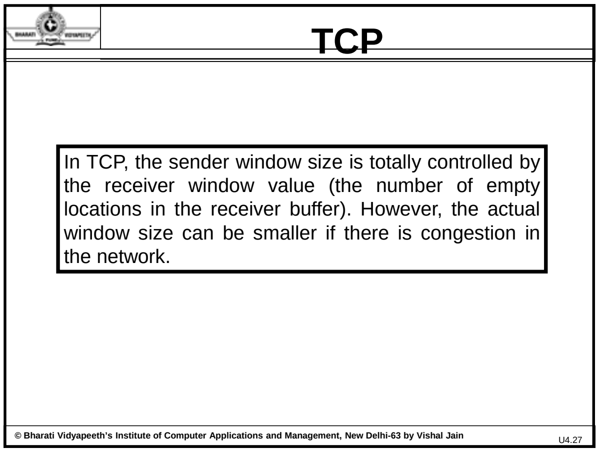
Expanding the Sender Window

© Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi-63 by Vishal Jain U4.25



Shrinking the Sender Window

© Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi-63 by Vishal Jain U4.26



TCP

In TCP, the sender window size is totally controlled by the receiver window value (the number of empty locations in the receiver buffer). However, the actual window size can be smaller if there is congestion in the network.

© Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi-63 by Vishal Jain U4.27

Silly Window Syndrome

- Syndrome created by sender
- Sol-Nagle's algorithm
- Syndrome created by receiver
- Clark Sol
- Delay ACK

© Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi-63 by Vishal Jain U4.28

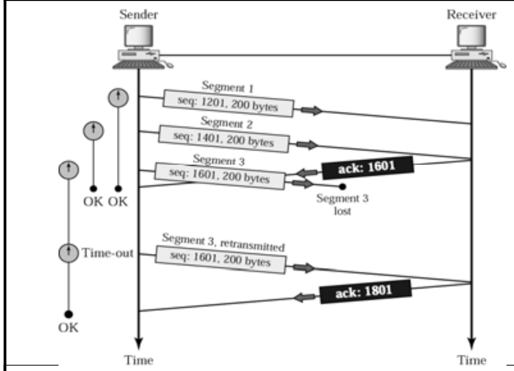
TCP

Error Control

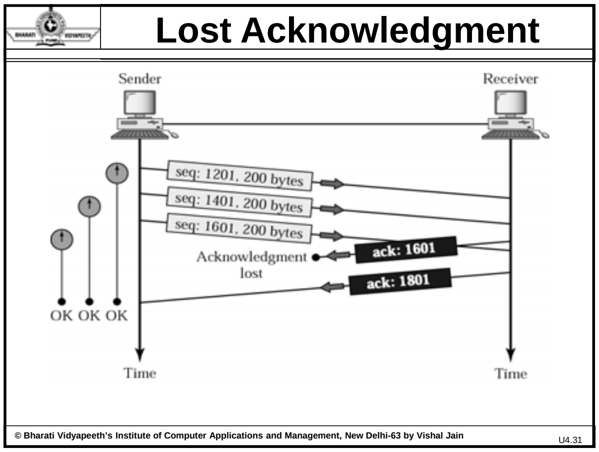
TCP provides reliability using error control, which detects corrupted, lost, out-of-order, and duplicated segments. Error control in TCP is achieved through the use of the checksum, acknowledgment, and time-out.

© Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi-63 by Vishal Jain U4.29

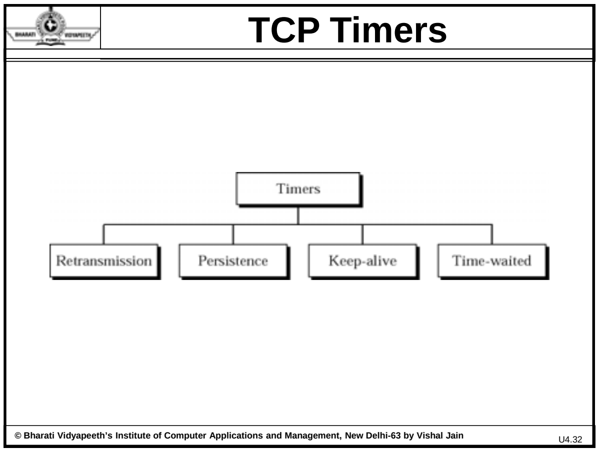
Lost Segment



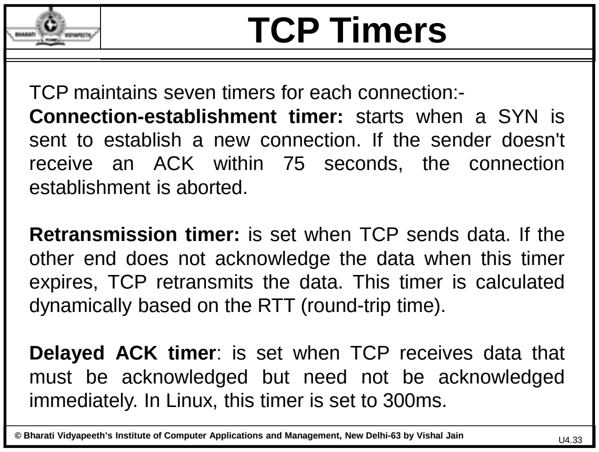
© Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi-63 by Vishal Jain U4.30



Lost Acknowledgment



TCP Timers



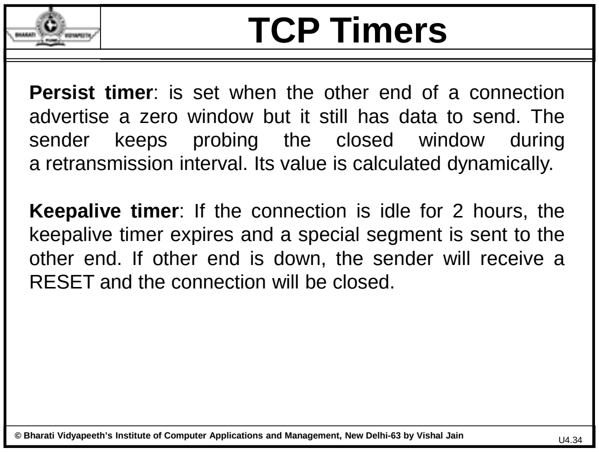
TCP Timers

TCP maintains seven timers for each connection:-

Connection-establishment timer: starts when a SYN is sent to establish a new connection. If the sender doesn't receive an ACK within 75 seconds, the connection establishment is aborted.

Retransmission timer: is set when TCP sends data. If the other end does not acknowledge the data when this timer expires, TCP retransmits the data. This timer is calculated dynamically based on the RTT (round-trip time).

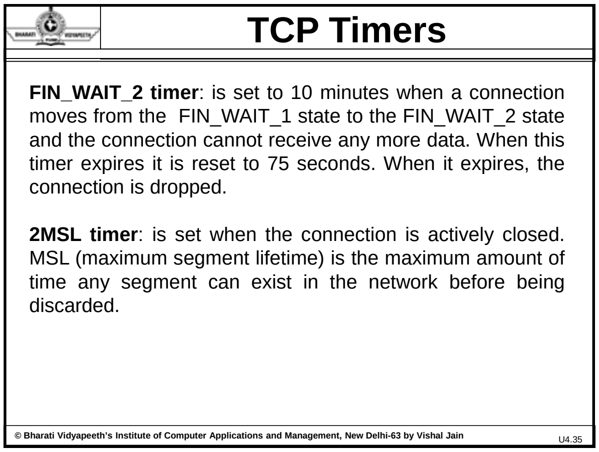
Delayed ACK timer: is set when TCP receives data that must be acknowledged but need not be acknowledged immediately. In Linux, this timer is set to 300ms.



TCP Timers

Persist timer: is set when the other end of a connection advertise a zero window but it still has data to send. The sender keeps probing the closed window during a retransmission interval. Its value is calculated dynamically.

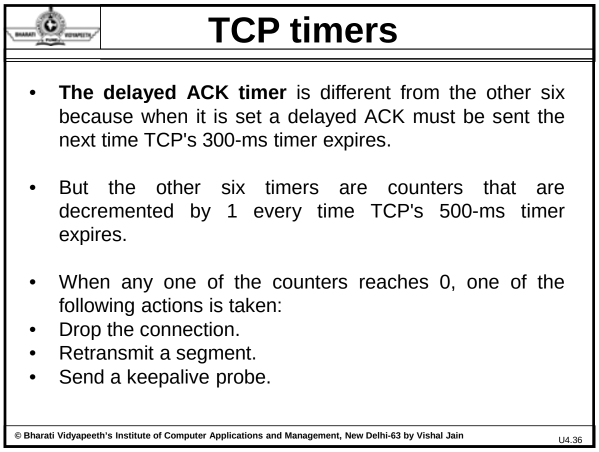
Keepalive timer: If the connection is idle for 2 hours, the keepalive timer expires and a special segment is sent to the other end. If other end is down, the sender will receive a RESET and the connection will be closed.



TCP Timers

FIN_WAIT_2 timer: is set to 10 minutes when a connection moves from the FIN_WAIT_1 state to the FIN_WAIT_2 state and the connection cannot receive any more data. When this timer expires it is reset to 75 seconds. When it expires, the connection is dropped.

2MSL timer: is set when the connection is actively closed. MSL (maximum segment lifetime) is the maximum amount of time any segment can exist in the network before being discarded.

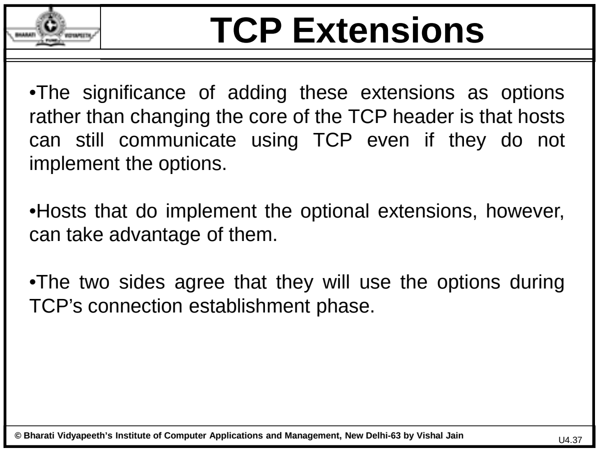


TCP timers

- **The delayed ACK timer** is different from the other six because when it is set a delayed ACK must be sent the next time TCP's 300-ms timer expires.

- But the other six timers are counters that are decremented by 1 every time TCP's 500-ms timer expires.

- When any one of the counters reaches 0, one of the following actions is taken:
- Drop the connection.
- Retransmit a segment.
- Send a keepalive probe.

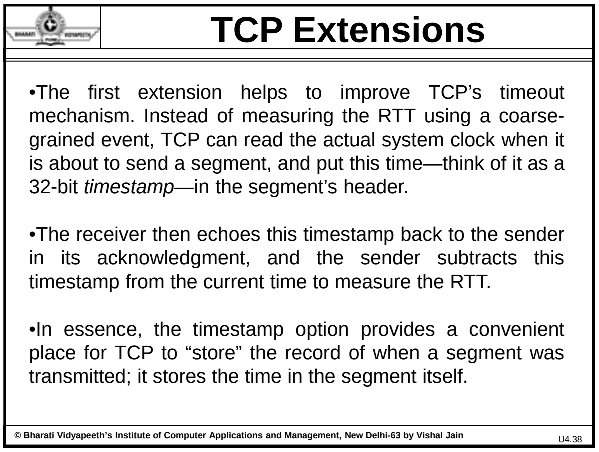


TCP Extensions

- The significance of adding these extensions as options rather than changing the core of the TCP header is that hosts can still communicate using TCP even if they do not implement the options.

- Hosts that do implement the optional extensions, however, can take advantage of them.

- The two sides agree that they will use the options during TCP's connection establishment phase.

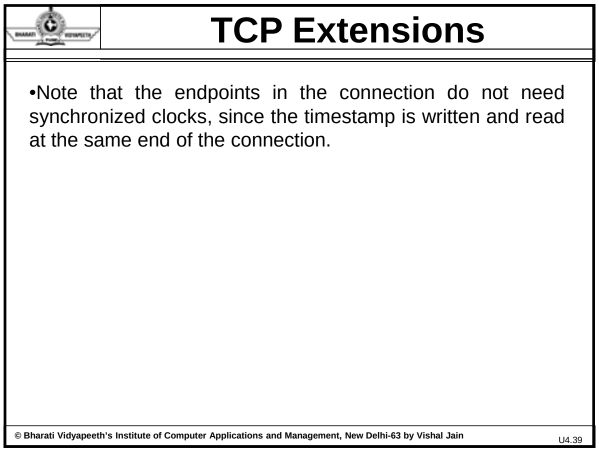


TCP Extensions

- The first extension helps to improve TCP's timeout mechanism. Instead of measuring the RTT using a coarse-grained event, TCP can read the actual system clock when it is about to send a segment, and put this time—think of it as a 32-bit *timestamp*—in the segment's header.

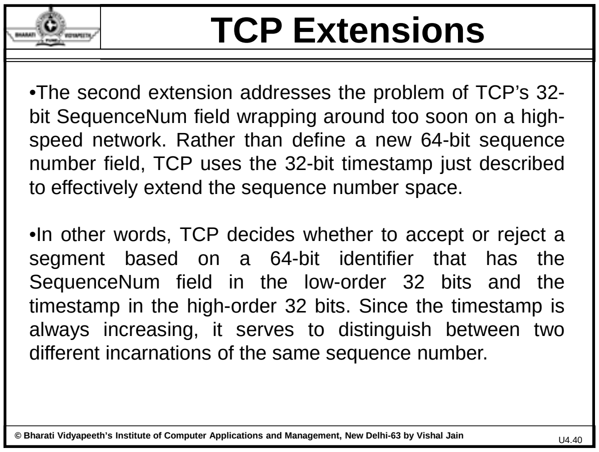
- The receiver then echoes this timestamp back to the sender in its acknowledgment, and the sender subtracts this timestamp from the current time to measure the RTT.

- In essence, the timestamp option provides a convenient place for TCP to "store" the record of when a segment was transmitted; it stores the time in the segment itself.



TCP Extensions

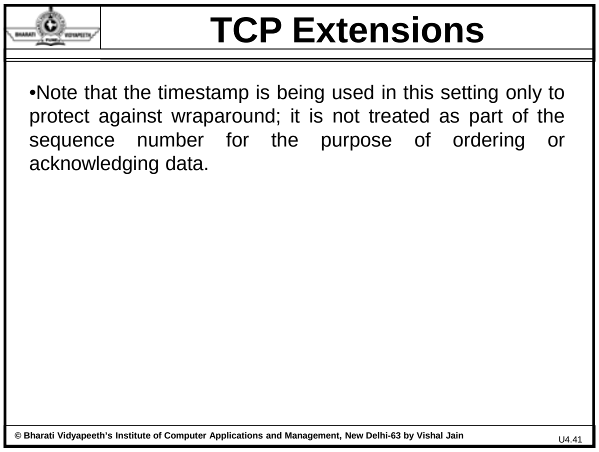
- Note that the endpoints in the connection do not need synchronized clocks, since the timestamp is written and read at the same end of the connection.



TCP Extensions

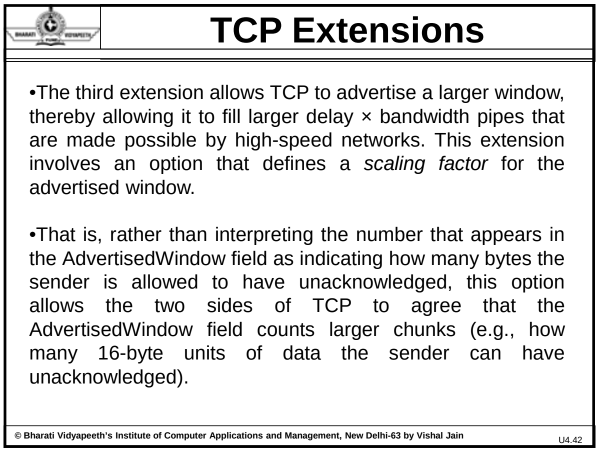
•The second extension addresses the problem of TCP’s 32-bit SequenceNum field wrapping around too soon on a high-speed network. Rather than define a new 64-bit sequence number field, TCP uses the 32-bit timestamp just described to effectively extend the sequence number space.

•In other words, TCP decides whether to accept or reject a segment based on a 64-bit identifier that has the SequenceNum field in the low-order 32 bits and the timestamp in the high-order 32 bits. Since the timestamp is always increasing, it serves to distinguish between two different incarnations of the same sequence number.



TCP Extensions

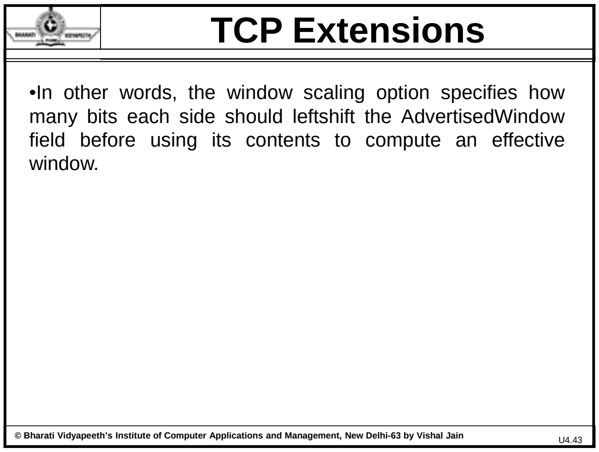
•Note that the timestamp is being used in this setting only to protect against wraparound; it is not treated as part of the sequence number for the purpose of ordering or acknowledging data.



TCP Extensions

•The third extension allows TCP to advertise a larger window, thereby allowing it to fill larger delay × bandwidth pipes that are made possible by high-speed networks. This extension involves an option that defines a *scaling factor* for the advertised window.

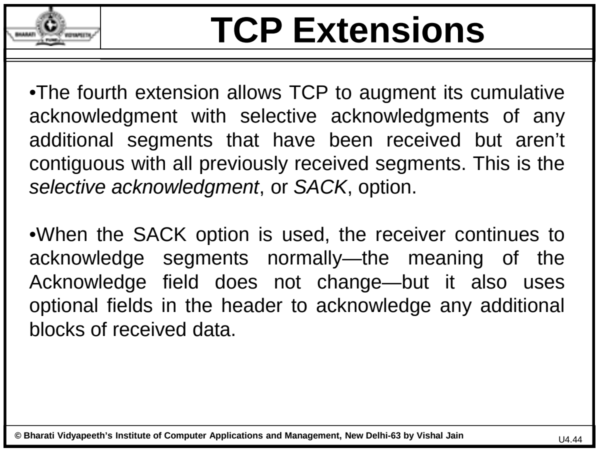
•That is, rather than interpreting the number that appears in the AdvertisedWindow field as indicating how many bytes the sender is allowed to have unacknowledged, this option allows the two sides of TCP to agree that the AdvertisedWindow field counts larger chunks (e.g., how many 16-byte units of data the sender can have unacknowledged).



TCP Extensions

•In other words, the window scaling option specifies how many bits each side should leftshift the AdvertisedWindow field before using its contents to compute an effective window.

U4.43

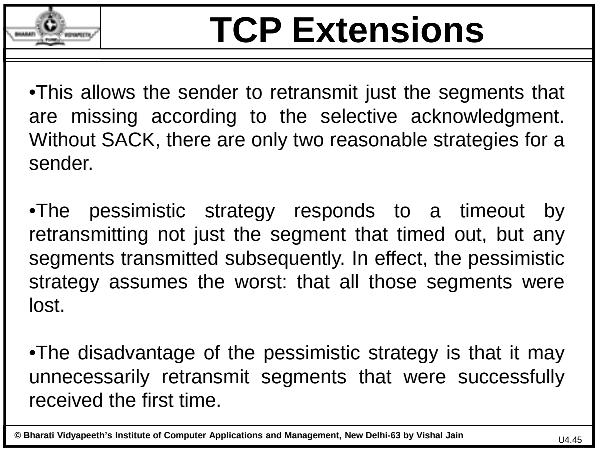


TCP Extensions

•The fourth extension allows TCP to augment its cumulative acknowledgment with selective acknowledgments of any additional segments that have been received but aren’t contiguous with all previously received segments. This is the *selective acknowledgment*, or *SACK*, option.

•When the SACK option is used, the receiver continues to acknowledge segments normally—the meaning of the Acknowledge field does not change—but it also uses optional fields in the header to acknowledge any additional blocks of received data.

U4.44



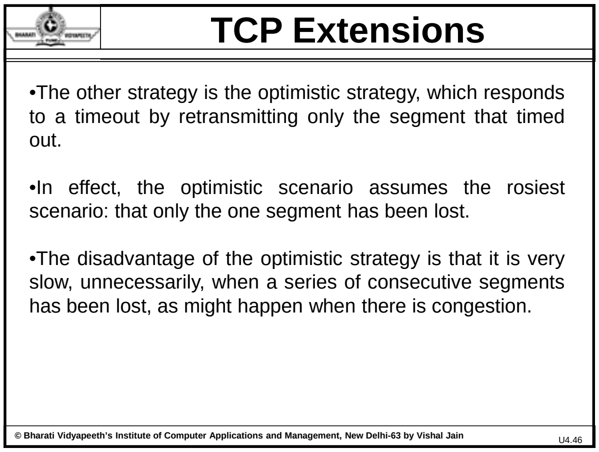
TCP Extensions

•This allows the sender to retransmit just the segments that are missing according to the selective acknowledgment. Without SACK, there are only two reasonable strategies for a sender.

•The pessimistic strategy responds to a timeout by retransmitting not just the segment that timed out, but any segments transmitted subsequently. In effect, the pessimistic strategy assumes the worst: that all those segments were lost.

•The disadvantage of the pessimistic strategy is that it may unnecessarily retransmit segments that were successfully received the first time.

U4.45

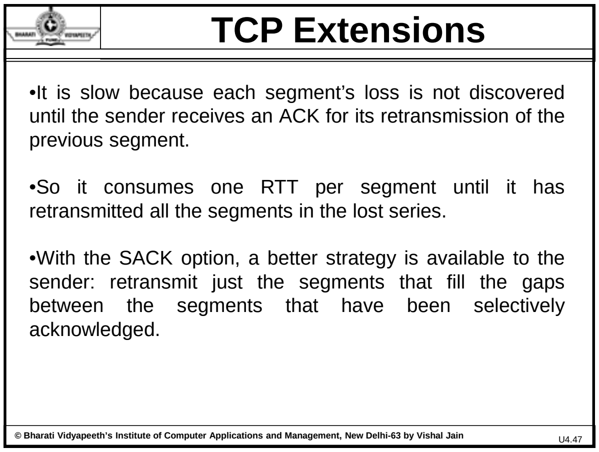


TCP Extensions

•The other strategy is the optimistic strategy, which responds to a timeout by retransmitting only the segment that timed out.

•In effect, the optimistic scenario assumes the rosiest scenario: that only the one segment has been lost.

•The disadvantage of the optimistic strategy is that it is very slow, unnecessarily, when a series of consecutive segments has been lost, as might happen when there is congestion.

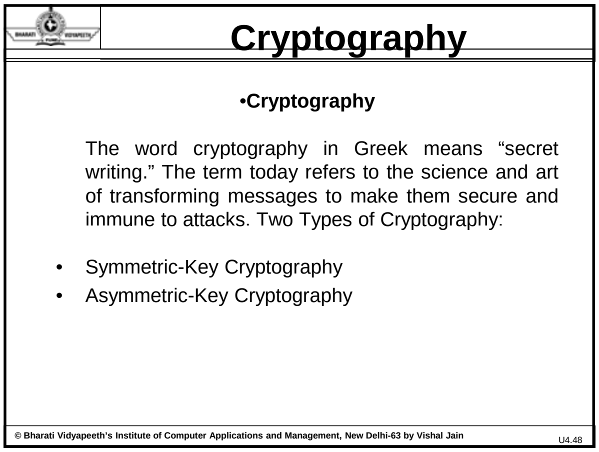


TCP Extensions

•It is slow because each segment's loss is not discovered until the sender receives an ACK for its retransmission of the previous segment.

•So it consumes one RTT per segment until it has retransmitted all the segments in the lost series.

•With the SACK option, a better strategy is available to the sender: retransmit just the segments that fill the gaps between the segments that have been selectively acknowledged.

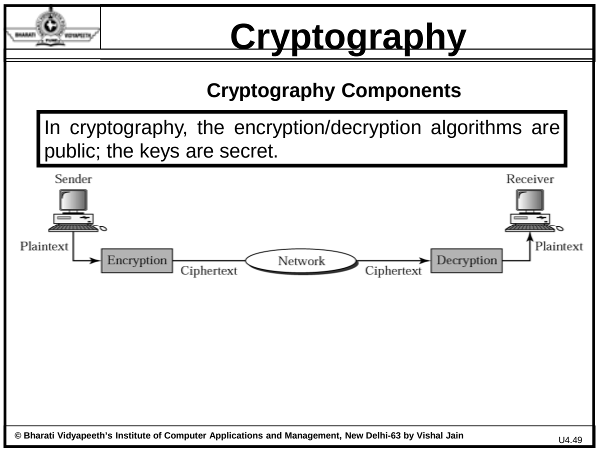


Cryptography

•Cryptography

The word cryptography in Greek means "secret writing." The term today refers to the science and art of transforming messages to make them secure and immune to attacks. Two Types of Cryptography:

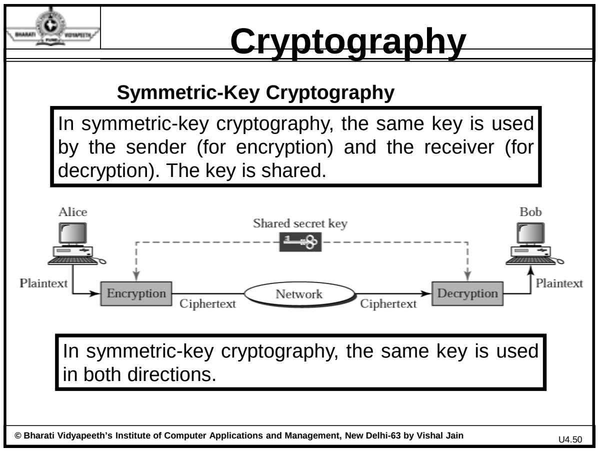
- Symmetric-Key Cryptography
- Asymmetric-Key Cryptography



Cryptography

Cryptography Components

In cryptography, the encryption/decryption algorithms are public; the keys are secret.

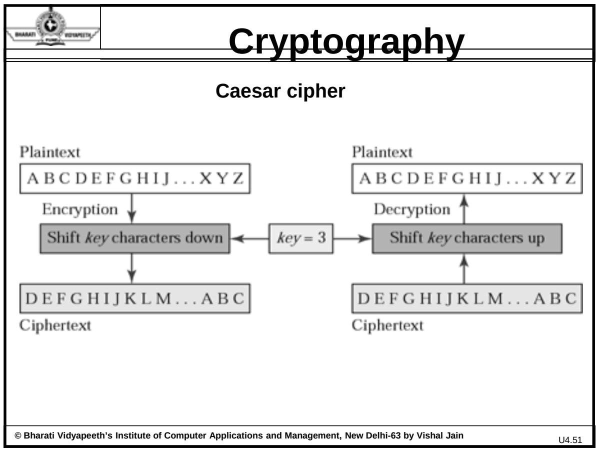


Cryptography

Symmetric-Key Cryptography

In symmetric-key cryptography, the same key is used by the sender (for encryption) and the receiver (for decryption). The key is shared.

In symmetric-key cryptography, the same key is used in both directions.



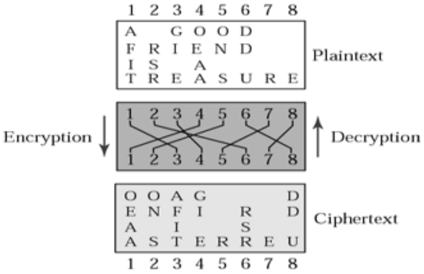
Cryptography

Caesar cipher

Plaintext
A B C D E F G H I J . . . X Y Z
Encryption
Shift key characters down
key = 3
D E F G H I J K L M . . . A B C
Ciphertext
Plaintext
A B C D E F G H I J . . . X Y Z
Decryption
Shift key characters up
D E F G H I J K L M . . . A B C
Ciphertext

Cryptography

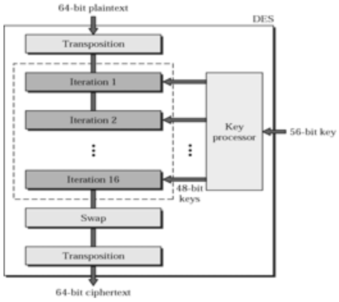
Transpositional cipher



U4.52

Cryptography

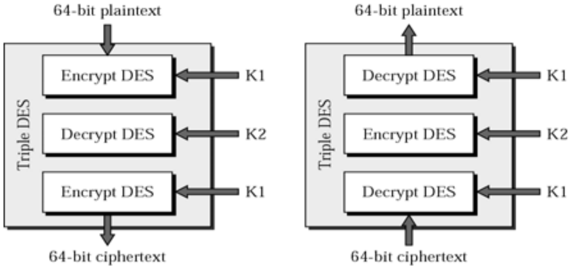
DES



U4.53

Cryptography

Triple DES



a. Encryption triple DES

b. Decryption triple DES

U4.54

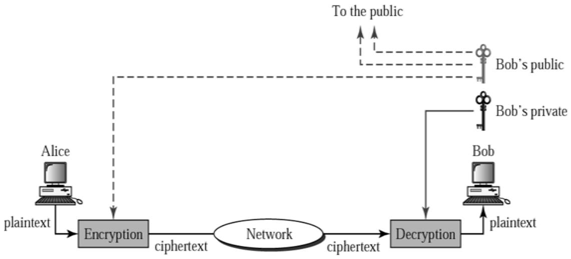
Cryptography

The DES cipher uses the same concept as the Caesar cipher, but the encryption/ decryption algorithm is much more complex.

© Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi-63 by Vishal Jain U4.55

Cryptography

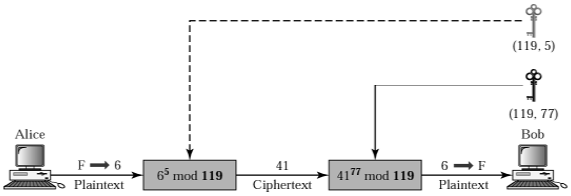
Public Key Cryptography



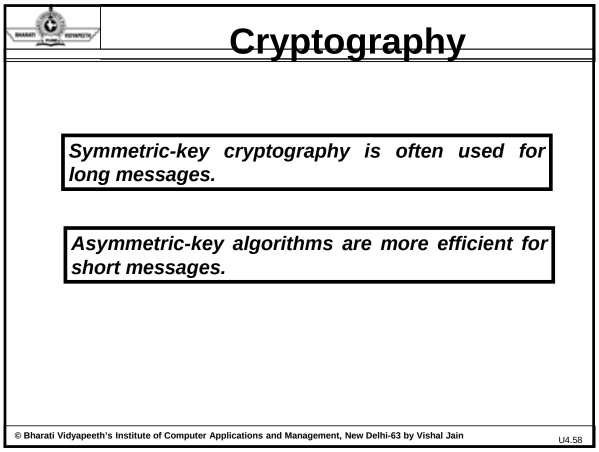
© Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi-63 by Vishal Jain U4.56

Cryptography

RSA



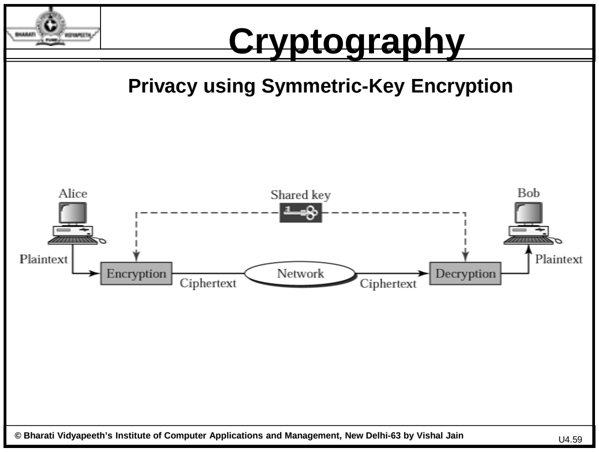
© Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi-63 by Vishal Jain U4.57



Cryptography

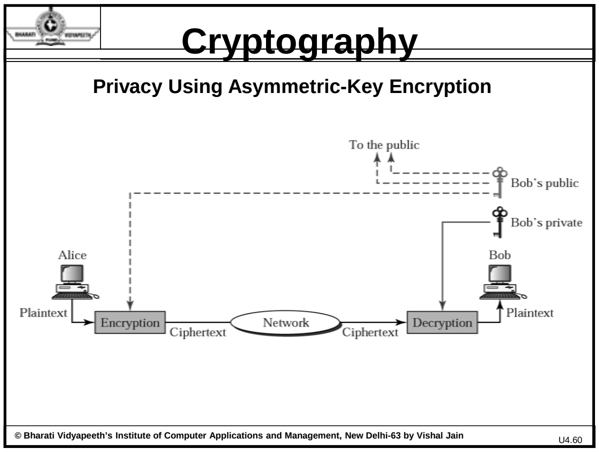
Symmetric-key cryptography is often used for long messages.

Asymmetric-key algorithms are more efficient for short messages.



Cryptography

Privacy using Symmetric-Key Encryption



Cryptography

Privacy Using Asymmetric-Key Encryption

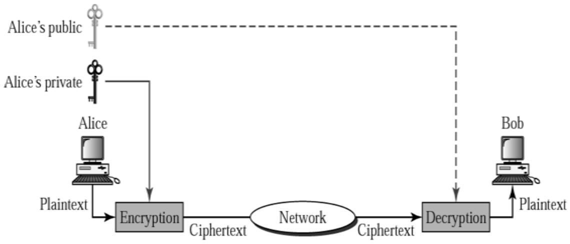
Cryptography

Digital Signature

Digital signature can provide authentication, integrity, and nonrepudiation for a message.

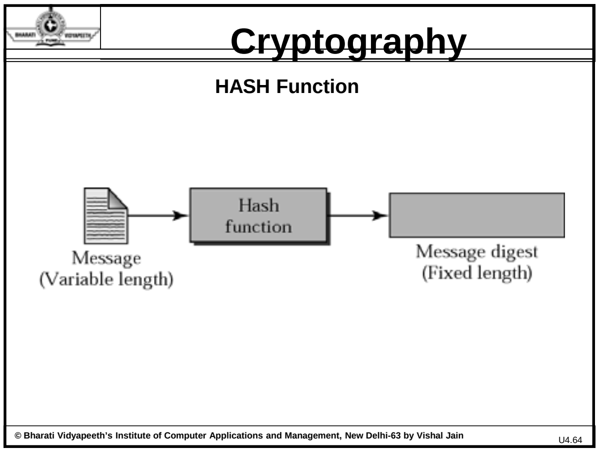
Cryptography

Signing the Whole Document



Cryptography

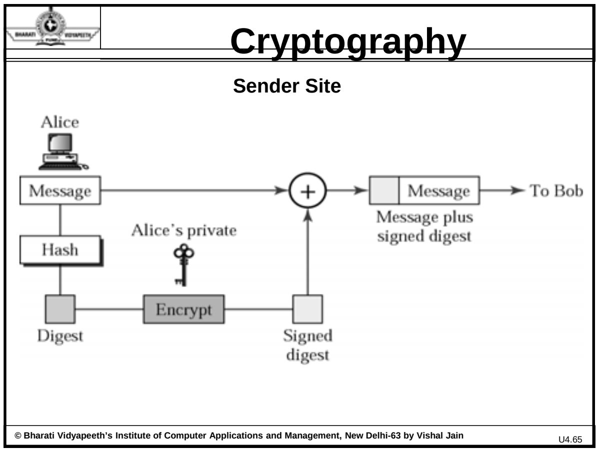
Digital signature does not provide privacy. If there is a need for privacy, another layer of encryption/decryption must be applied.



Cryptography

HASH Function

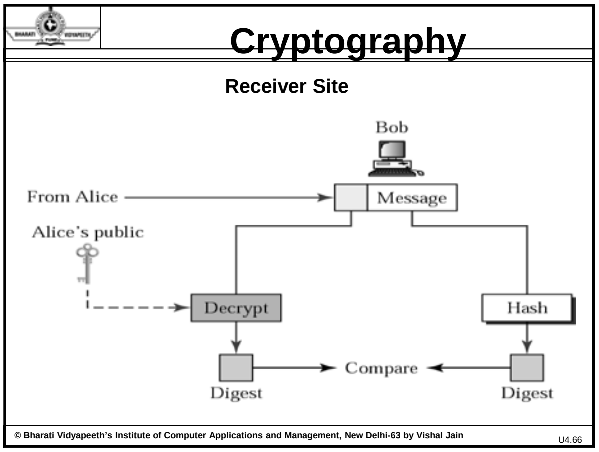
U4.64



Cryptography

Sender Site

U4.65



Cryptography

Receiver Site

U4.66

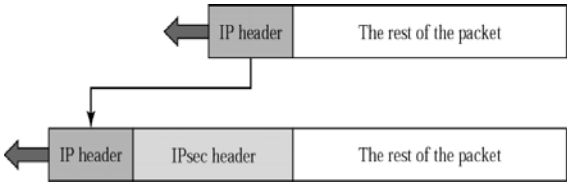
IP Sec

Security in the Internet

- In this section we discuss a security method for each of the top 3 layers of the Internet model.
- At the IP level we discuss a protocol called IPSec;
- at the transport layer we discuss a protocol that "glues" a new layer to the transport layer; at the application layer we discuss a security method called PGP.

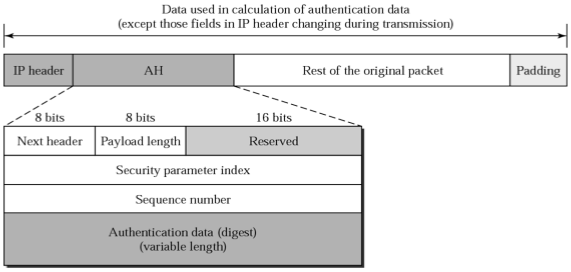
IP Sec

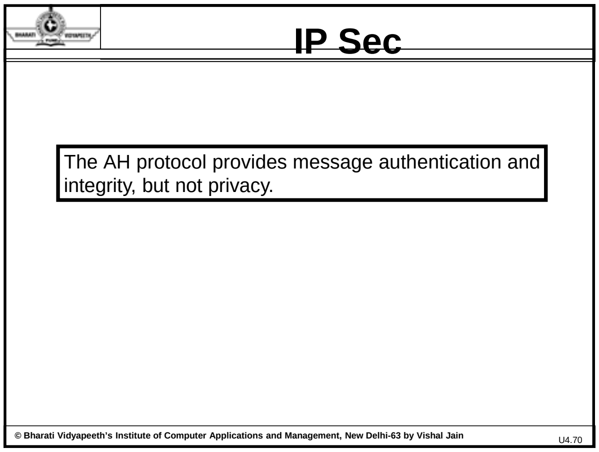
Transport Mode



IP Sec

AH


Data used in calculation of authentication data
(except those fields in IP header changing during transmission)
IP header
AH
Rest of the original packet
Padding
8 bits
8 bits
16 bits
Next header
Payload length
Reserved
Security parameter index
Sequence number
Authentication data (digest)
(variable length)

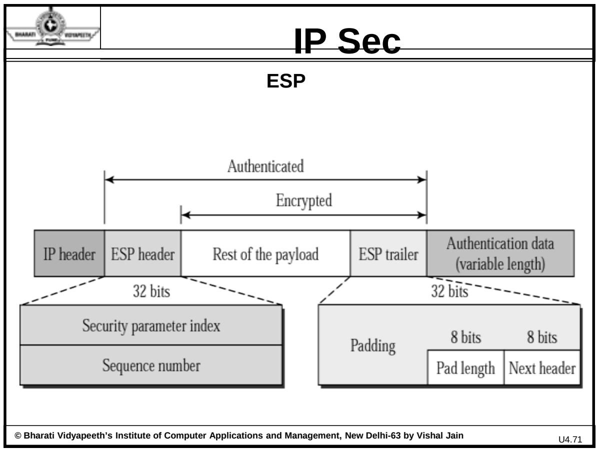


IP Sec

The AH protocol provides message authentication and integrity, but not privacy.

© Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi-63 by Vishal Jain

U4.70

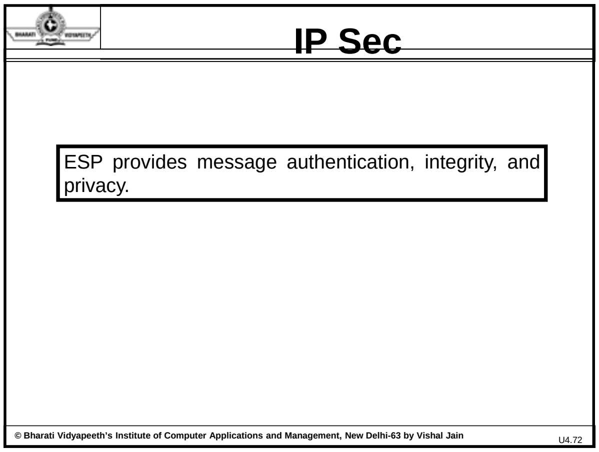


IP Sec

ESP

© Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi-63 by Vishal Jain

U4.71



IP Sec

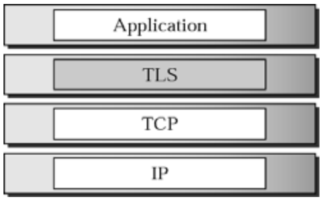
ESP provides message authentication, integrity, and privacy.

© Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi-63 by Vishal Jain

U4.72

IP Sec

Position of TLS



Application

TLS

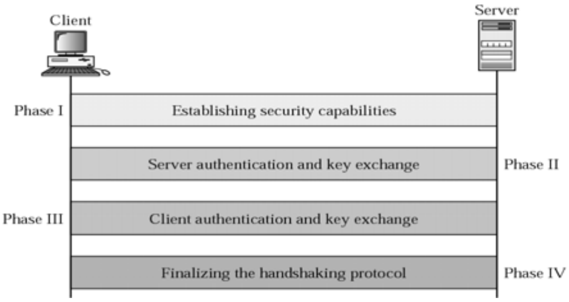
TCP

IP

© Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi-63 by Vishal Jain U4.73

Cryptography

Handshake protocol



Phase I Establishing security capabilities

Server authentication and key exchange Phase II

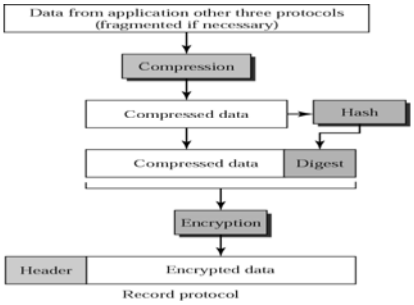
Phase III Client authentication and key exchange

Finalizing the handshaking protocol Phase IV

© Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi-63 by Vishal Jain U4.74

Cryptography

Record protocol



Data from application other three protocols (fragmented if necessary)

Compression

Compressed data

Hash

Compressed data Digest

Encryption

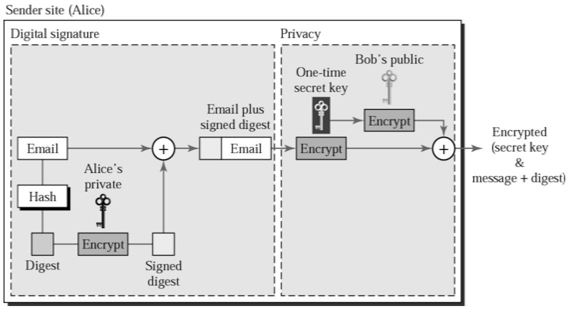
Header Encrypted data

Record protocol

© Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi-63 by Vishal Jain U4.75

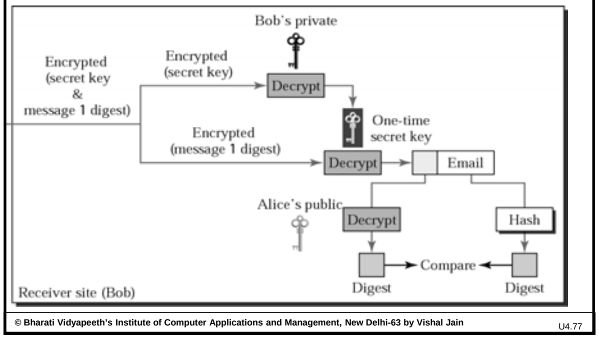
Cryptography

PGP at Sender Site



Cryptography

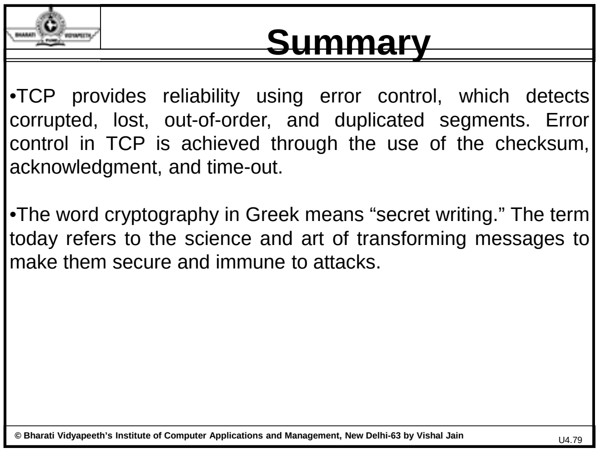
PGP at Receiver Site



Summary

•Flow control regulates the amount of data a source can send before receiving an acknowledgment from the destination. TCP defines a window that is imposed on the buffer of data delivered from the application program.

•In TCP, the sender window size is totally controlled by the receiver window value (the number of empty locations in the receiver buffer). However, the actual window size can be smaller if there is congestion in the network.

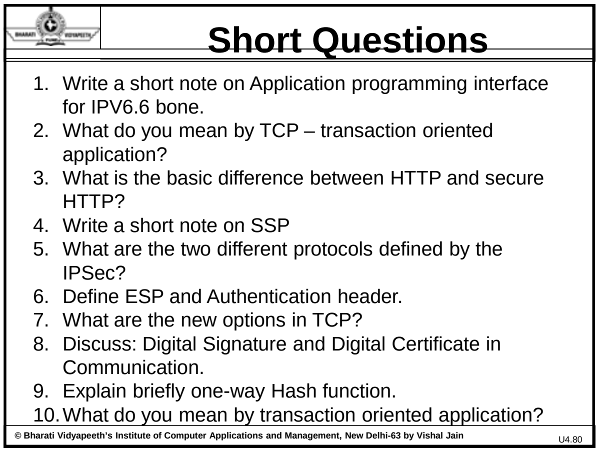


Summary

•TCP provides reliability using error control, which detects corrupted, lost, out-of-order, and duplicated segments. Error control in TCP is achieved through the use of the checksum, acknowledgment, and time-out.

•The word cryptography in Greek means “secret writing.” The term today refers to the science and art of transforming messages to make them secure and immune to attacks.

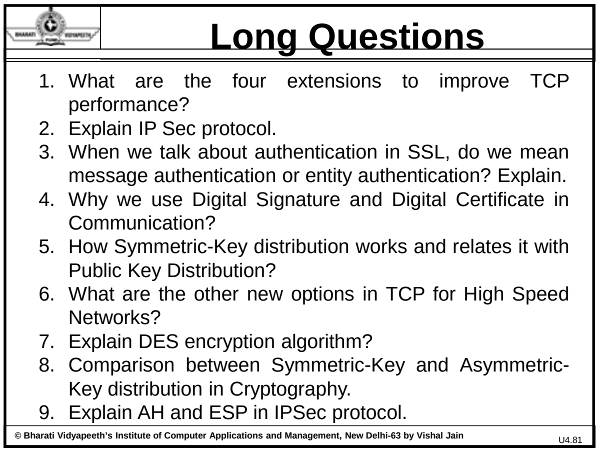
© Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi-63 by Vishal Jain U4.79



Short Questions

1. Write a short note on Application programming interface for IPV6.6 bone.
2. What do you mean by TCP – transaction oriented application?
3. What is the basic difference between HTTP and secure HTTP?
4. Write a short note on SSP
5. What are the two different protocols defined by the IPSec?
6. Define ESP and Authentication header.
7. What are the new options in TCP?
8. Discuss: Digital Signature and Digital Certificate in Communication.
9. Explain briefly one-way Hash function.
10. What do you mean by transaction oriented application?

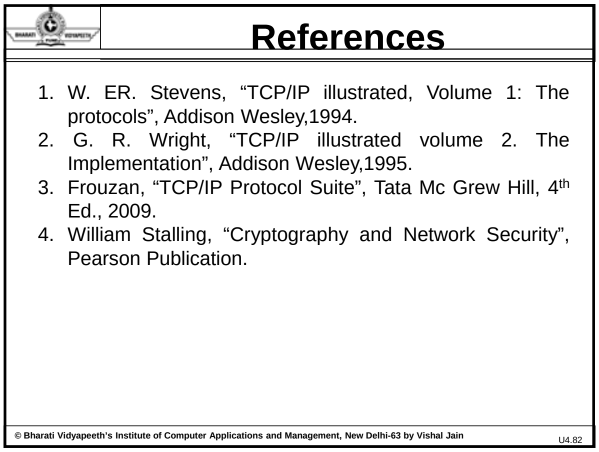
© Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi-63 by Vishal Jain U4.80



Long Questions

1. What are the four extensions to improve TCP performance?
2. Explain IP Sec protocol.
3. When we talk about authentication in SSL, do we mean message authentication or entity authentication? Explain.
4. Why we use Digital Signature and Digital Certificate in Communication?
5. How Symmetric-Key distribution works and relates it with Public Key Distribution?
6. What are the other new options in TCP for High Speed Networks?
7. Explain DES encryption algorithm?
8. Comparison between Symmetric-Key and Asymmetric-Key distribution in Cryptography.
9. Explain AH and ESP in IPSec protocol.

© Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi-63 by Vishal Jain U4.81



References

1. W. ER. Stevens, “TCP/IP illustrated, Volume 1: The protocols”, Addison Wesley,1994.
2. G. R. Wright, “TCP/IP illustrated volume 2. The Implementation”, Addison Wesley,1995.
3. Frouzan, “TCP/IP Protocol Suite”, Tata Mc Grew Hill, 4th Ed., 2009.
4. William Stalling, “Cryptography and Network Security”, Pearson Publication.