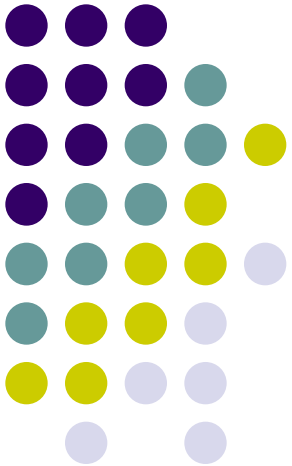
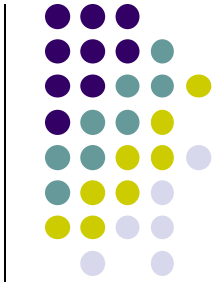
UDP – User Datagram Protocol

- Application-layer addressing
- User-data integrity

(similar to IP datagram services)



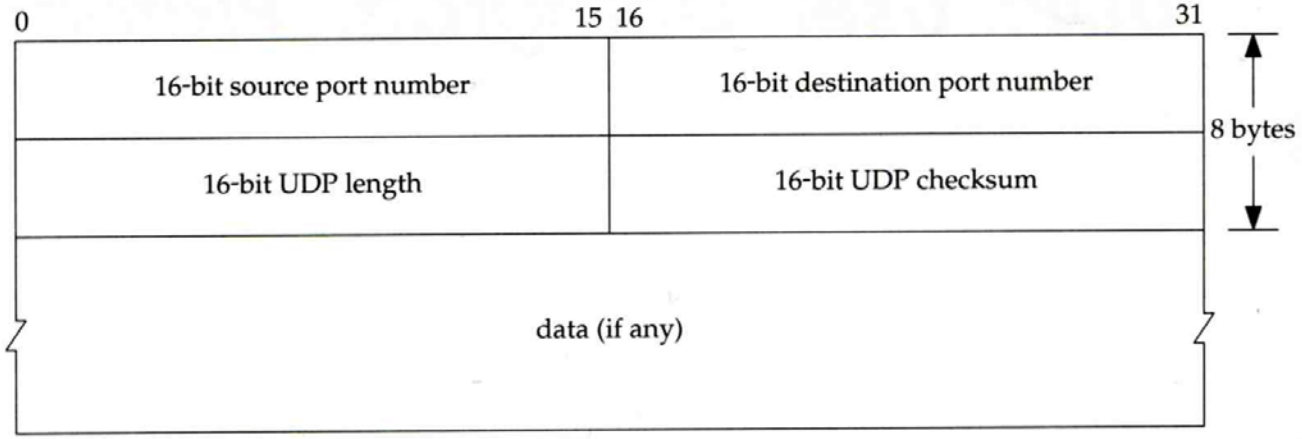
UDP Introduction



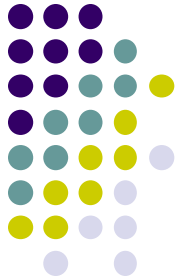
- Unreliable, connectionless, efficient, asyn. transmission protocol
  - Datagram-oriented, not stream-oriented protocol
- UDP header
  - 8 bytes, 4 fields
    - Source port and destination port
      - Identify sending and receiving process
    - Length
      - 16-bit field for UDP header and payload
    - checksum
      - 16-bit field for verifying data integrity

Data Integrity

- Checksum = Pseudo header + UDP PDU + Padding
- UDP pseudo header includes 5 fields: source IP address, dest. IP address, unused (8-bit 0's), protocol (in IP header), length (in UDP header)



Transport Layer Principles



- Transport layer is called End-to-End, Host-to-Host, or Process-to-Process protocol.
  - It has no direct relationship with the intermediate nodes when both ends of transport protocols work.
  - In IP layer, intermediate routers have interactions with the sending node.

- Reliability Issue
  - Unreliable (UDP) + unreliable (IP) = unreliable
  - Reliable (TCP) + unreliable (IP) = reliable

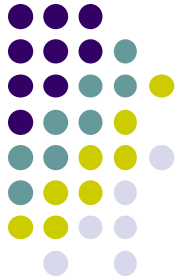
Port Number = 分機號碼

- A connection includes source IP, source port, destination IP, and destination port
  - IP + port = socket .
  - Using port number (0-65535) enables multi-tasking services on a server.

- Classification of Ports
  - Well-known ports (0 – 1023)

    eg. telnet 23, smtp 25, http 80, snmp 161, … etc
  - Registered ports (1024 – 49151)
  - Dynamic and/or private ports (49152 - 65535)

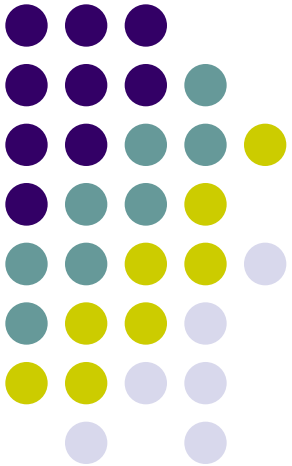
UDP Implementation



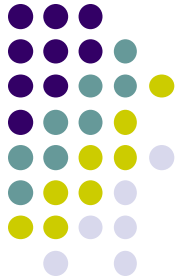
- Two approaches
  - Application-to-application pairwise interaction
  - Clients-to-server many-to-one interaction
- Protocol Port Demultiplexing
  - Using only the destination port number
    - Server implementation is easy
  - Using both source and destination port numbers
    - Multiple queues for multiple clients

TCP – Transmission Control Protocol

- Stream-oriented
- Full-duplex
- Connection-oriented
- Application-layer addressing
- In-sequence delivery
- User-data integrity
- Graceful release

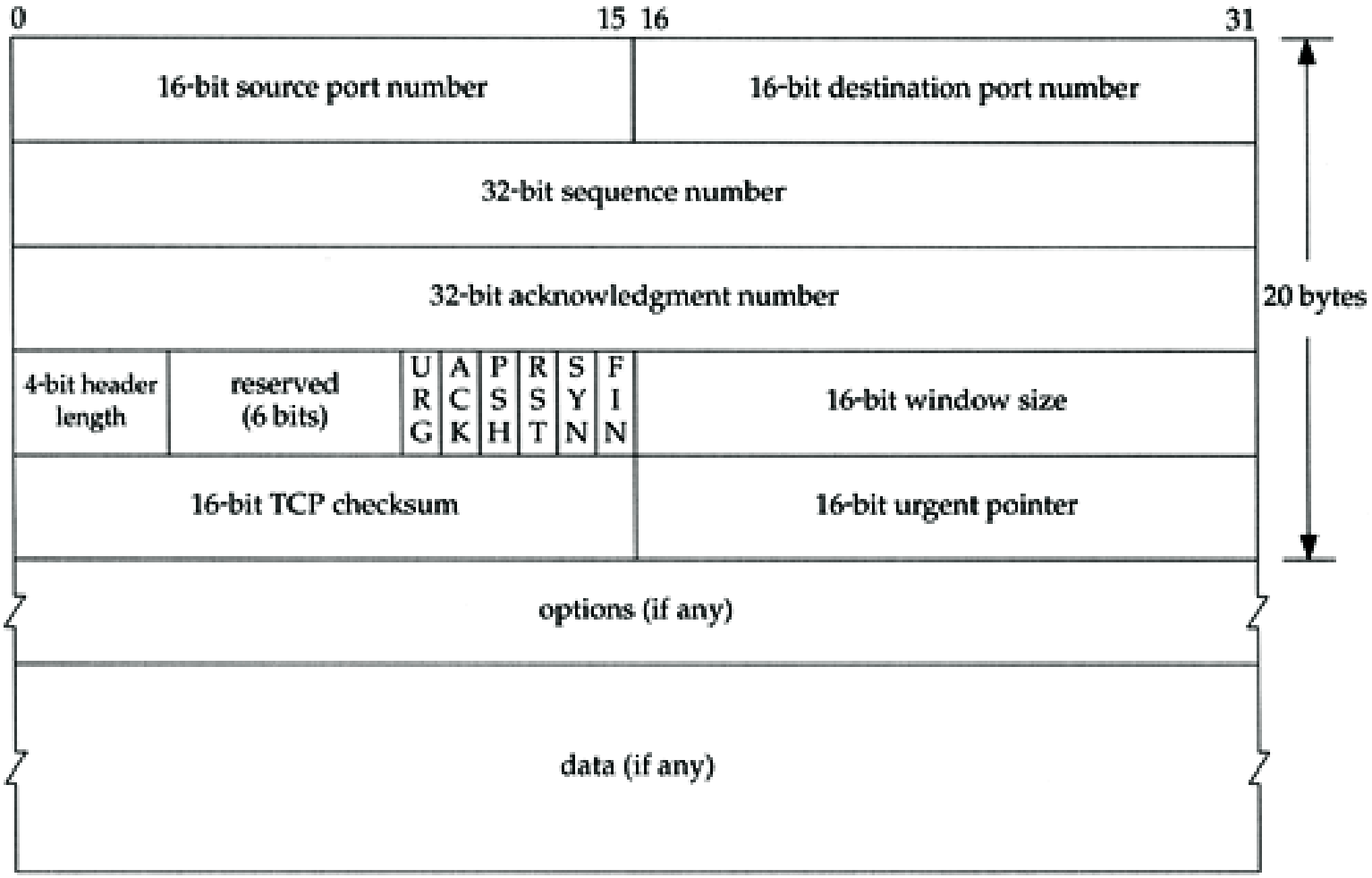


TCP

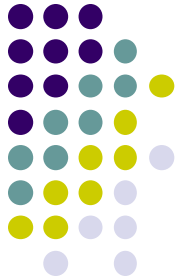


- Services
  - Connection-oriented
    - Establish TCP connection before exchanging data
  - Reliability
    - Acknowledgement when receiving data
    - Retransmission when timeout
    - Ordering
    - Discard duplicated data
    - Flow control

TCP Header (1)

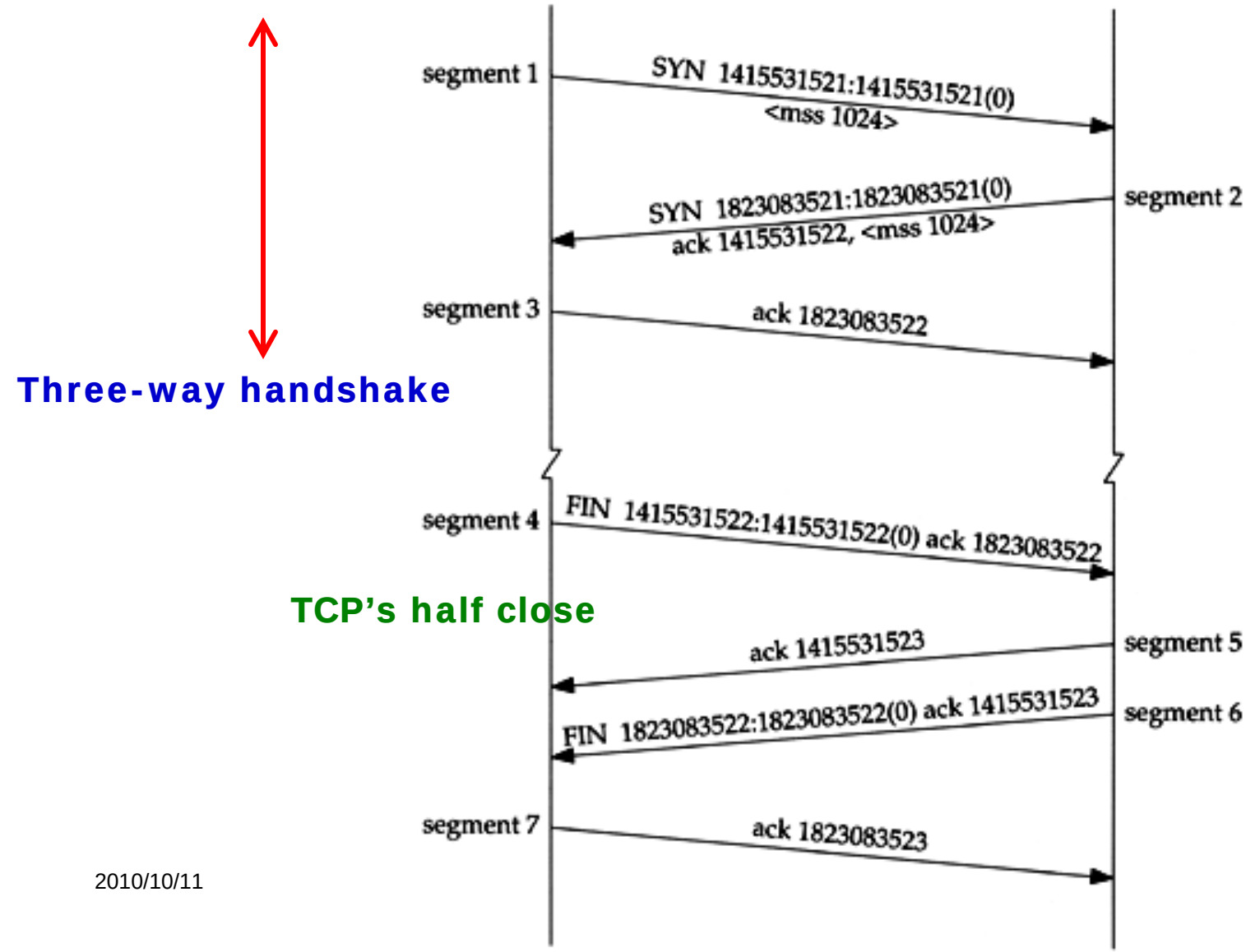

0
15 16
31
16-bit source port number
16-bit destination port number
32-bit sequence number
32-bit acknowledgment number
20 bytes
4-bit header length
reserved (6 bits)
URG
ACK
PSH
RST
SYN
FIN
16-bit window size
16-bit TCP checksum
16-bit urgent pointer
options (if any)
data (if any)

TCP Header (2)



- Flags
  - SYN
    - Establish new connection
  - ACK
    - Acknowledgement number is valid
    - Used to ack previous data that host has received
  - RST
    - Reset connection
  - FIN
    - The sender is finished sending data

TCP connection establishment and termination

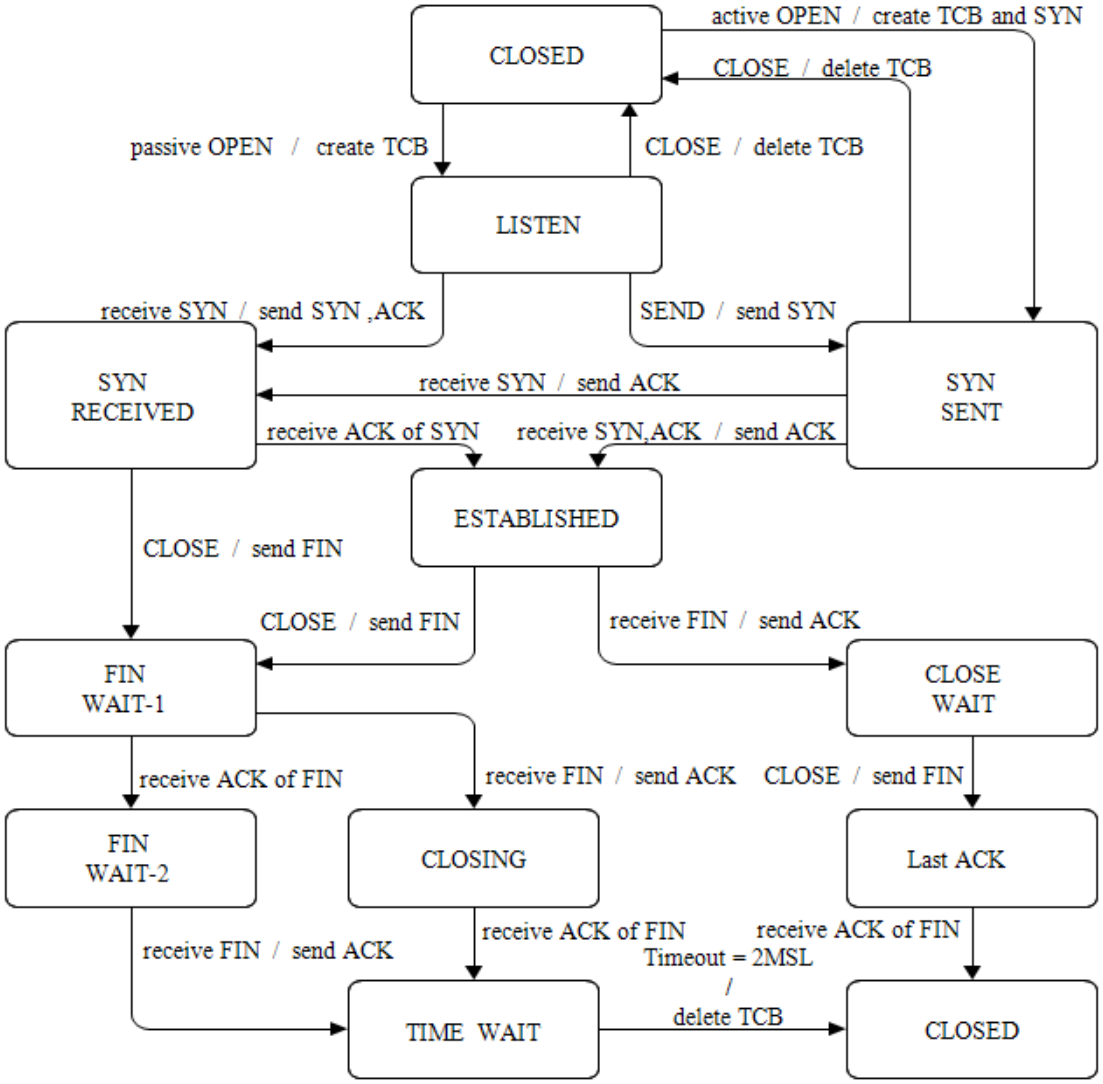


TCP state diagram (1/4)

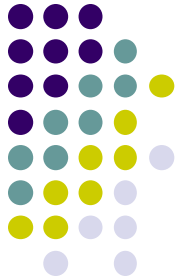
- Building a connection (3-way handshaking)
- Data transfer
- Graceful disconnection

(補充資料如後)

TCP state diagram (2/4)

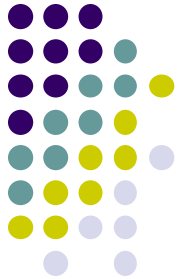


TCP state diagram (3/4)



- 狀態(state)，通常是使用『長方形』來表示，用來表示該主機在當時的狀況為何
- 轉移(transition)，通常是使用一條『有向線』來表示，用來連接兩個狀態，表示從一個狀態轉換成另一個狀態。轉移又可分為兩種
- 一種為『無觸發式轉移』(trigger-less transition)，也就是當狀態一結束後，自動轉換到下一個狀態
- 另一種為『觸發式轉移』(trigger transition)，也就是當一個『事件』(event)發生時，會觸發一個『動作』(action)的執行，該『動作』執行結束後，便會將狀態轉換到下一個狀態

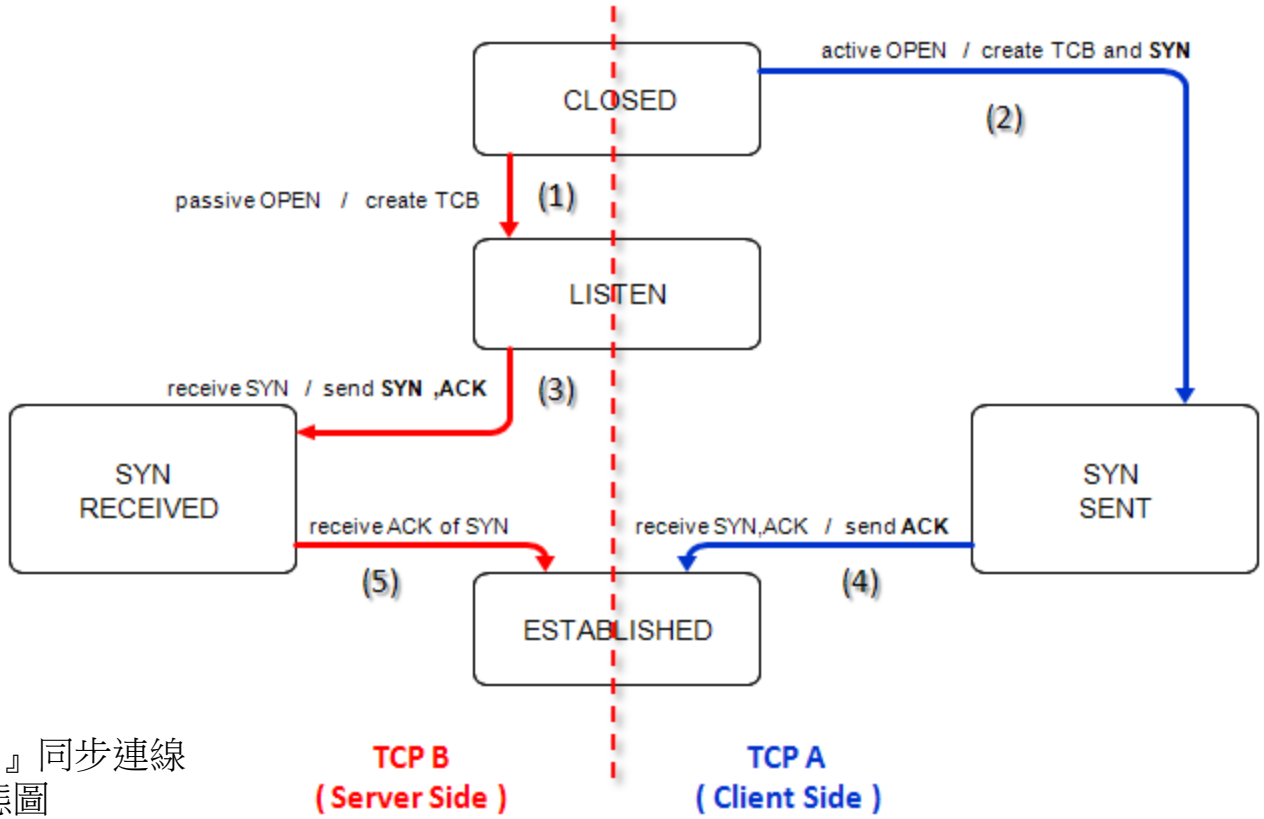
TCP state diagram (4/4)



TCP連線狀態圖中，共有五種不同的『狀態』(state)，分別說明如下：

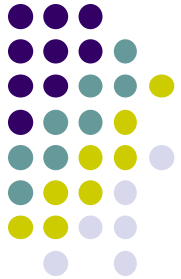
- 『CLOSED』(已關閉連線)，表示該主機的連線呈現關閉中
- 『LISTEN』(聆聽中)，表示該主機的『服務』已準備好，等待接受外界的連線請求
- 『SYN RECEIVED』(已接受到同步訊息SYN)，表示已接收到對方的SYN訊息
- 『SYN SENT』(已傳送出同步訊息SYN)，表示已送出SYN訊息
- 『ESTABLISHED』(已建立連線)，表示已完成雙方連線的建立

Establish Connection (1/6)



基本之『三向交握』同步連線
(a) 基本的連線狀態圖

Establish Connection (2/6)

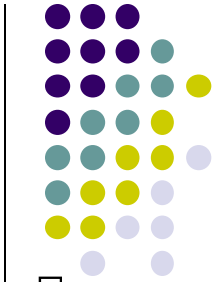


(0)雙方皆處於『CLOSED』狀態

(1)[Server端]

當Server端發生『passive OPEN』的事件後，便會進行『create TCB』的動作，並進入『LISTEN』的狀態。因為在Server端啟動『服務』(service)，並使用TCP協定時，是屬於被動的角色，所以在Server端對TCP的開啟，稱之為『passive OPEN』，隨即便會在系統中，建立一個『傳輸控制區塊』(Transmission Control Block,簡稱TCB)，用來儲存Server端有關TCP的所有資訊。也由於Server端必須先進入『LISTEN』狀態才能接受Client端的連線請求

Establish Connection (3/6)



(2)[Client端]

當Client發生『active OPEN』事件後，即進行『create TCB and SYN』的動作，並進入『SYN SENT』的狀態。因為在Client端使用TCP協定時，是屬於主動的角色，所以在Client端對TCP的開啟，稱之為『active OPEN』，隨即便會在系統中，建立一個『傳輸控制區塊』(Transmission Control Block, 簡稱TCB)，用來儲存Client端有關TCP的所有資訊，同時也送出SYN同步訊息給Server端，也就是進行『三向交握』的第一個動作

Establish Connection (4/6)

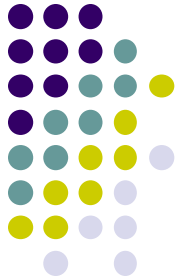
(3)[Server端]

當Server發生『receive SYN』事件後，即進行『send SYN, ACK』的動作。換言之，當Server端收到Client端傳送過來同步訊息SYN時，便會傳送SYN與ACK給Client端，也就是進行『三向交握』的第二個動作

(4)[Client端]

當Client發生『receive SYN, ACK』事件後，即進行『send ACK』的動作。換言之，當Client端收到Server端回應的確認訊息ACK與同步訊息SYN時，便會回應一個確認訊息ACK給Server端，也就是進行『三向交握』的第三個動作，並進入『ESTABLISHED』狀態

Establish Connection (5/6)



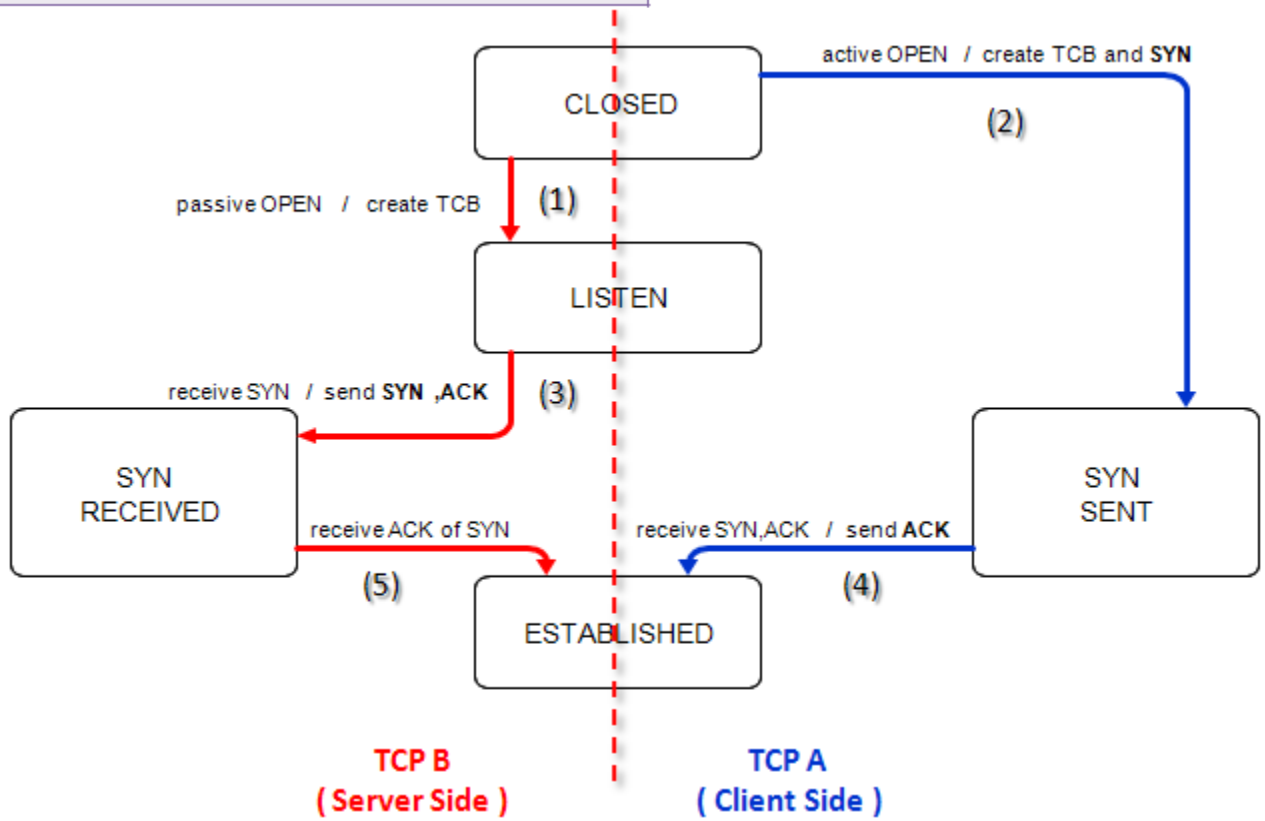
(5)[Server端]

當Server發生『receive ACK of SYN』事件後，不做任何動作，直接進入『ESTABLISHED』狀態。換言之，就是在收到Client端回應的確認訊息ACK後，不執行任何動作，直接進入『ESTABLISHED』狀態

Establish Connection (6/6)

序號	TCP B (Server)		Action		TCP A (Client)
1	LISTEN				CLOSED
2	SYN-RECEIVED	←	<SEQ=100><CTL=SYN>	←	SYN-SENT
3	SYN-RECEIVED	→	<SEQ=300><ACK=101><CTL=SYN,ACK>	→	ESTABLISHED
4	ESTABLISHED	←	<SEQ=101><ACK=301><CTL=ACK>	←	ESTABLISHED
5	ESTABLISHED	←	<SEQ=101><ACK=301><CTL=ACK><DATA>	←	ESTABLISHED

基本之『三向交握』同步連線
(b) 連線之基本動作表列



Close Connection (1/7)

- TCP的『關閉連線』方式，依據IETF的標準文件rfc793中所描述的情形(可參考IETF的官方資料(http://www.rfc-editor.org/rfc/rfc793.txt)，可分為以下二種不同狀況
  - Normal Close Sequence
  - Simultaneous Close Sequence
- 以下僅針對第一個最基本的『關閉連線』方式(Normal Close Sequence)進行說明

Close Connection (2/7)

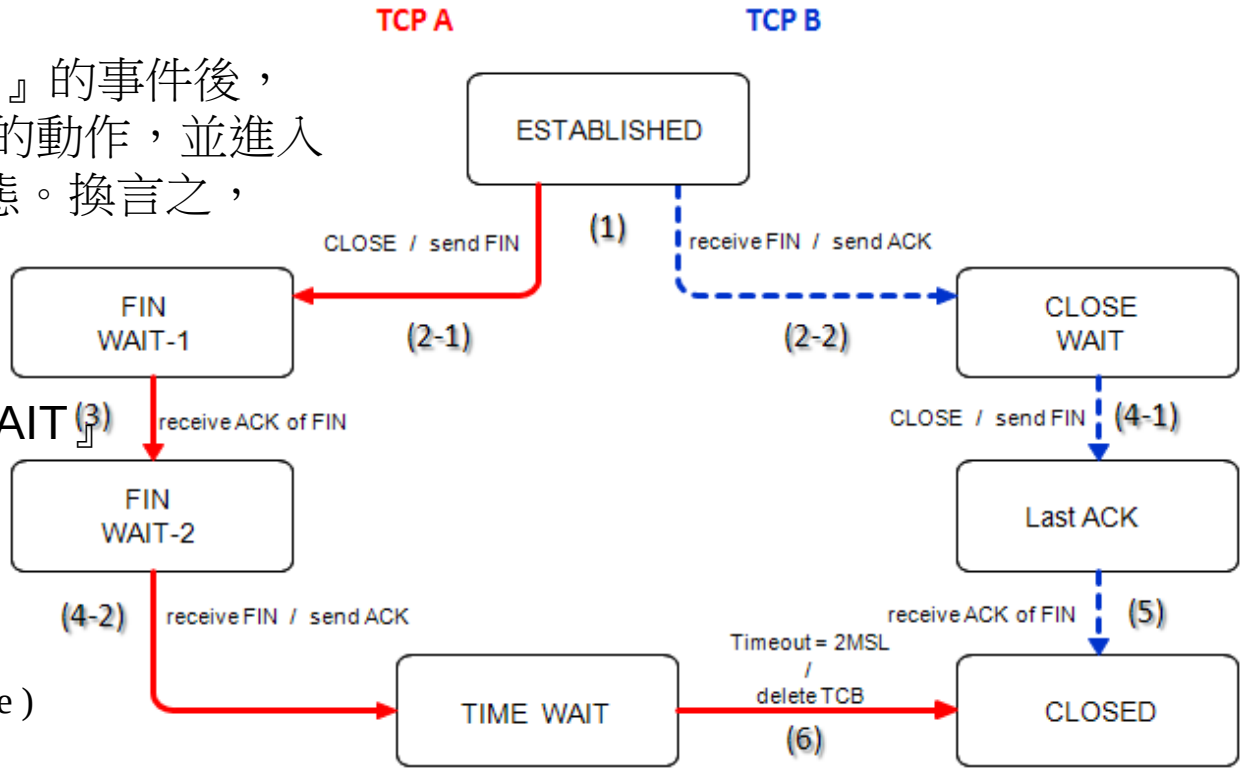
1) 雙方皆已處於『ESTABLISHED』狀態。
2) 分為以下兩部份

(2-1) [TCP A]

當A端發生『CLOSE』的事件後，便會進行『send FIN』的動作，並進入『FIN WAIT-1』的狀態。換言之，就是在A端的程式要準備關閉連線時，會主動送出完成的訊息FIN給對方，並進入『FIN WAIT-1』的狀態。

(2-2) [TCP B]

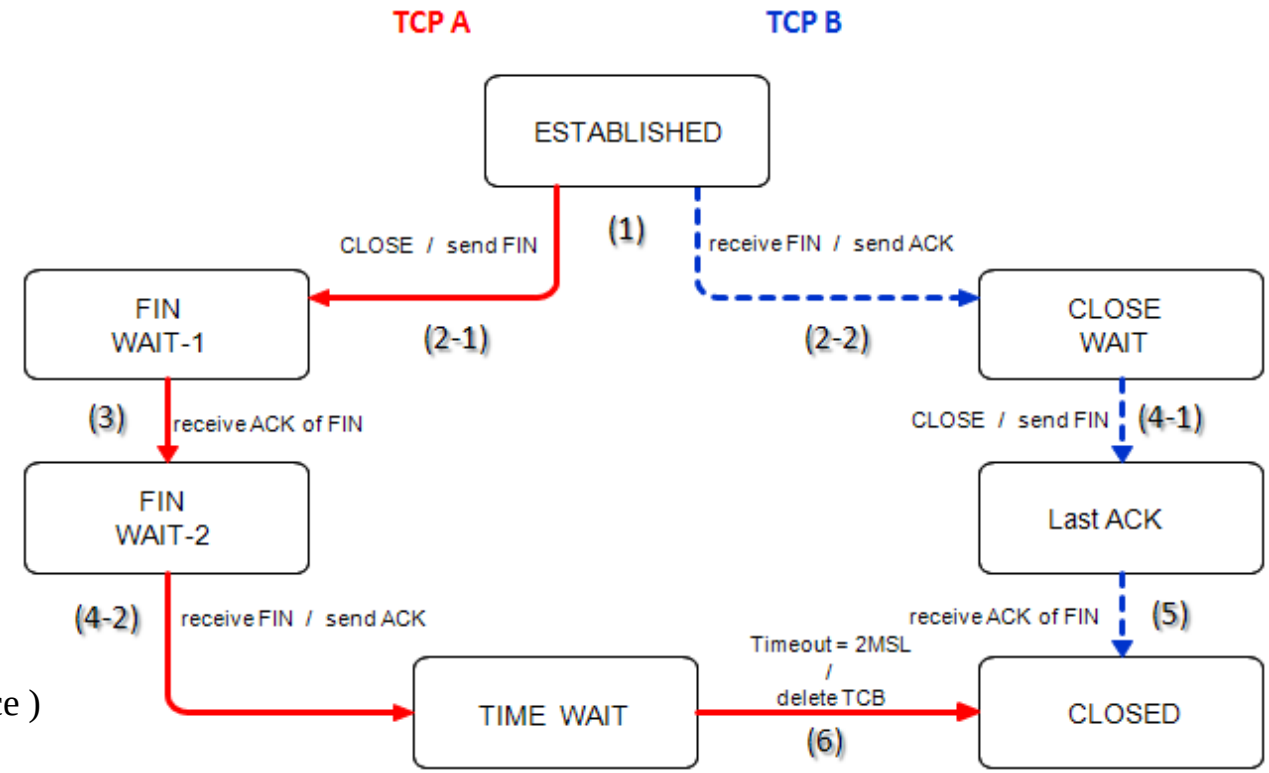
當B端發生『receive FIN』的事件後，便會進行『send ACK』的動作，並進入『CLOSE WAIT』的狀態。換言之，就是B端接受到A端送來的完成訊息FIN時，會送出確認訊息ACK給對方，並進入『CLOSE WAIT』的狀態



正常關閉連線 (Normal Close Sequence)
(a) 正常『關閉連線』之狀態圖

Close Connection (3/7)

3) [TCP A]
當A端發生『receive ACK of FIN』事件後，便直接進入『FIN WAIT-2』的狀態。換言之，就是收到B端回應的確認訊息ACK之後，不執行任何動作，便直接進入『FIN WAIT-2』的狀態



正常關閉連線 (Normal Close Sequence)
(a) 正常『關閉連線』之狀態圖

Close Connection (4/7)

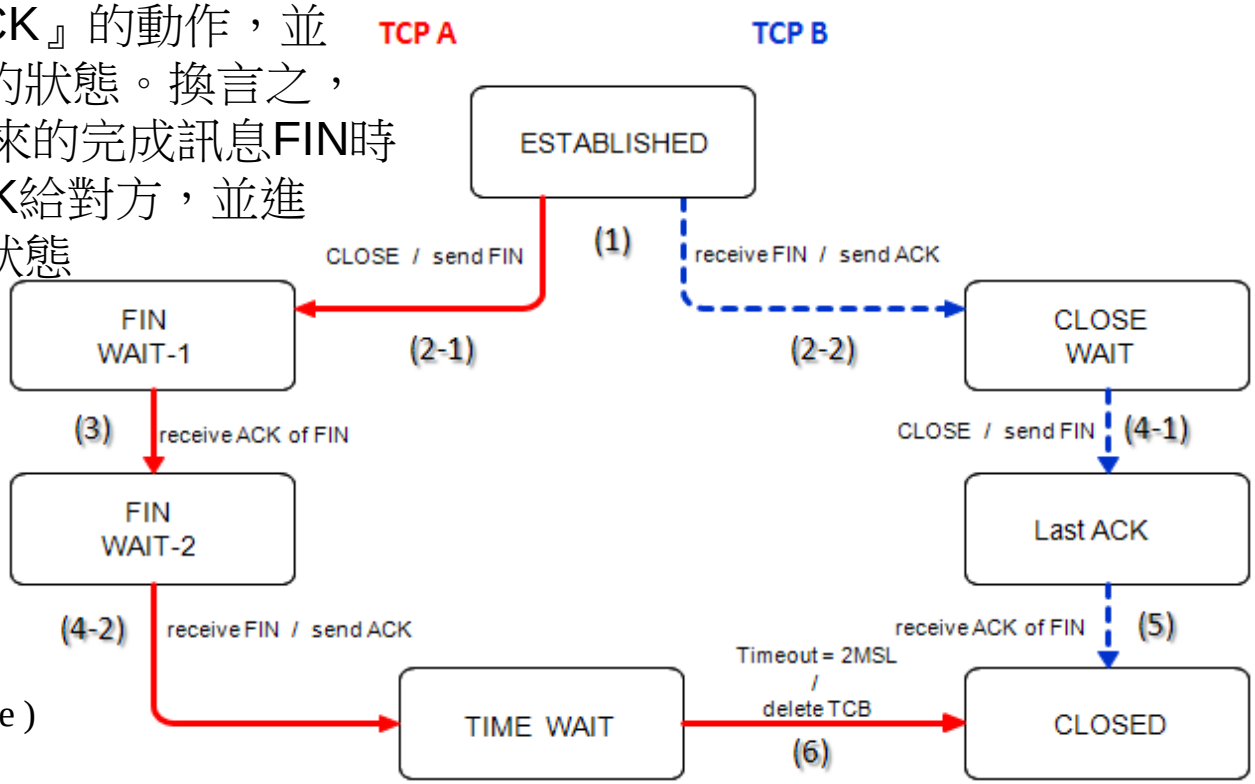
4) 分為以下兩部份

(4-1) [TCP B]

當B端發生『CLOSE』的事件後，便會進行『send FIN』的動作，並進入『LAST ACK』的狀態。換言之，就是在B端的程式要準備關閉連線時，會主動送出完成的訊息FIN給對方，並進入『LAST ACK』的狀態。

(4-2) [TCP A]

當A端發生『receive FIN』的事件後，便會進行『send ACK』的動作，並進入『TIME WAIT』的狀態。換言之，就是A端接受到B端送來的完成訊息FIN時，會送出確認訊息ACK給對方，並進入『TIME WAIT』的狀態

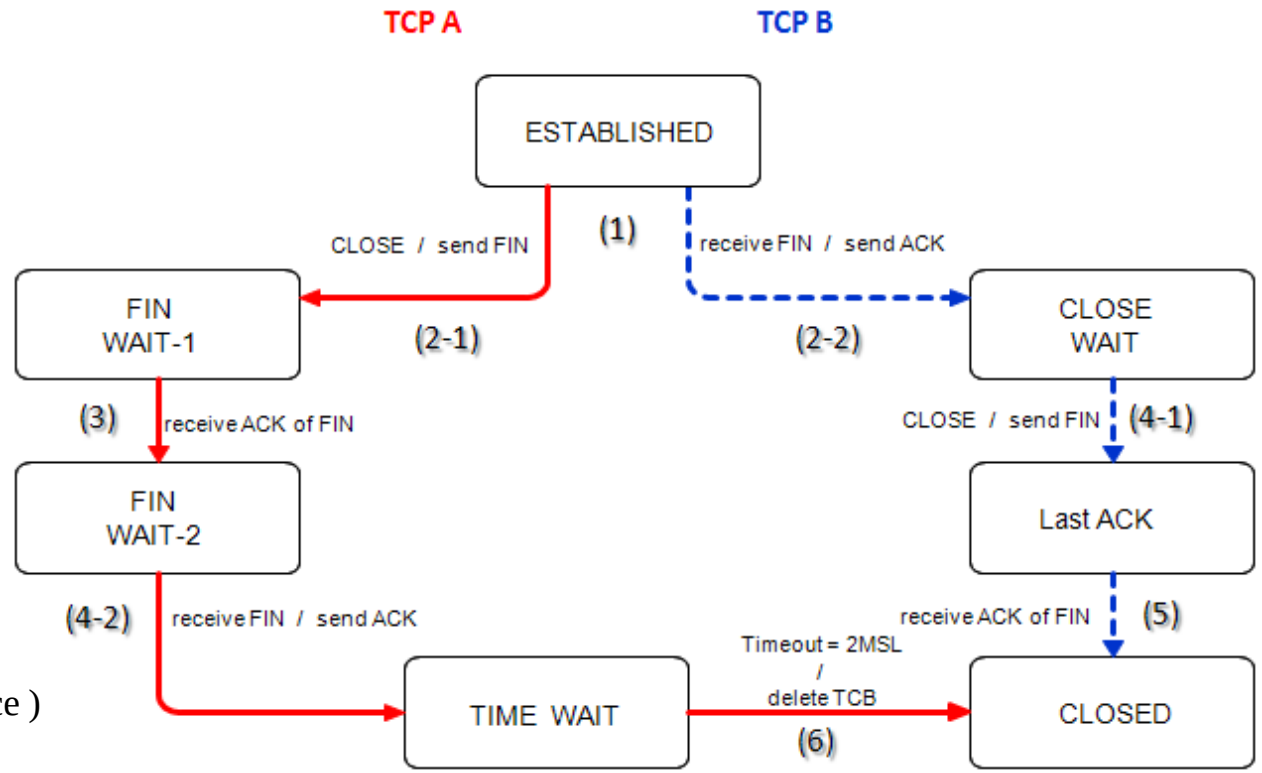


正常關閉連線 (Normal Close Sequence)

(a) 正常『關閉連線』之狀態圖

Close Connection (5/7)

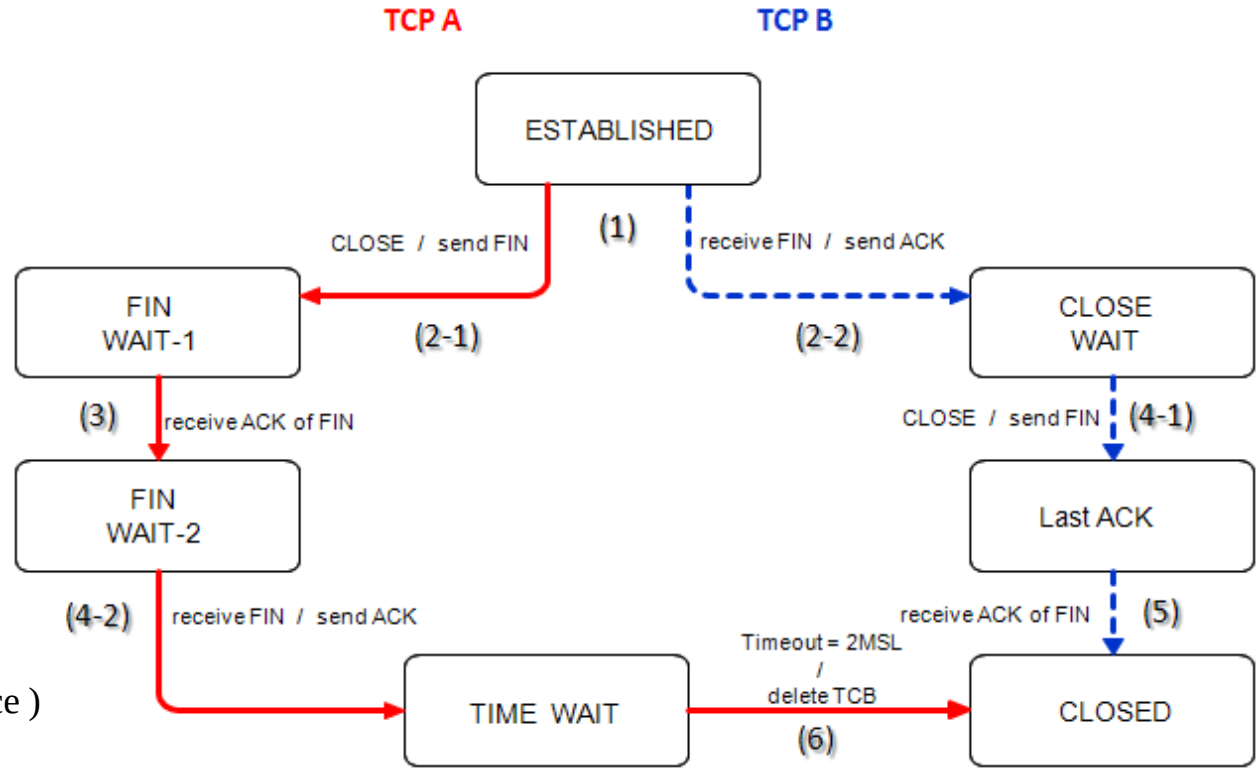
5) [TCP B]
當B端發生『receive ACK of FIN』事件後，便直接進入『CLOSED』的狀態。換言之，就是收到A端回應的確認訊息ACK之後，不執行任何動作，便直接進入『CLOSED』的狀態



正常關閉連線 (Normal Close Sequence)
(a) 正常『關閉連線』之狀態圖

Close Connection (6/7)

6) [TCP A]
當A端發生[Timeout=2MSL]事件後，便會執行[delete TCB]的動作，並進入[CLOSED]的狀態。換言之，就是A端等待2MSL(Max Segment Lifetime, 簡稱MSL)的時間過後，便將系統在『建立連線』階段所建立的TCB刪除，並進入[CLOSED]的狀態。MSL是指封包在網路傳輸中，能存活的最長時間，所以此處等待2倍的MSL，即可確保舊的連線封包，不會殘留在網中，而干擾到新的連線封包

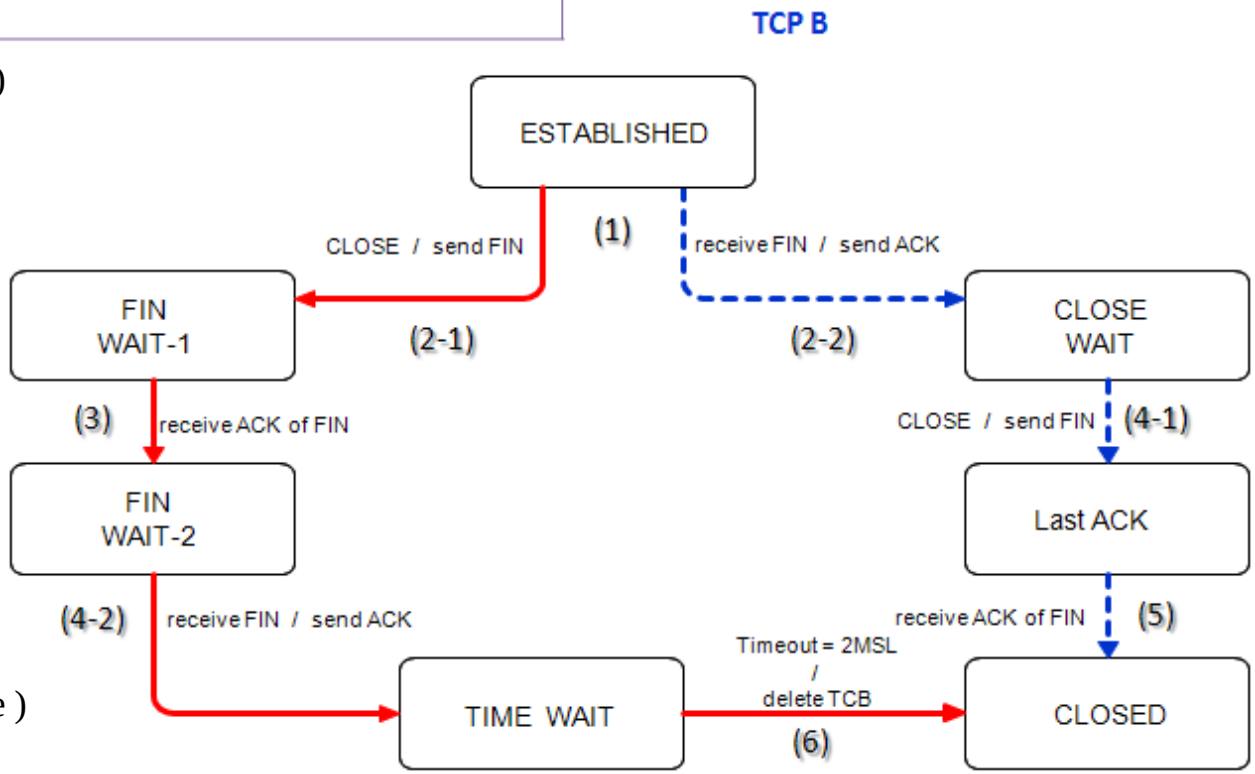


正常關閉連線 (Normal Close Sequence)
(a) 正常『關閉連線』之狀態圖

Close Connection (7/7)

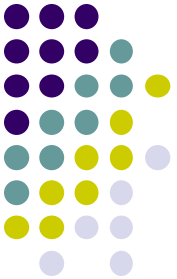
序號	TCP A	Action	TCP B
1	ESTABLISHED		ESTABLISHED
2	(Close) FIN-WAIT-1	→ <SEQ=100><ACK=300><CTL=FIN,ACK> →	CLOSE-WAIT
3	FIN-WAIT-2	← <SEQ=300><ACK=101><CTL=ACK> ←	CLOSE-WAIT
4	TIME-WAIT	← <SEQ=300><ACK=101><CTL=FIN,ACK> ←	(Close) LAST-ACK
5	TIME-WAIT	→ <SEQ=101><ACK=301><CTL=ACK> →	CLOSED
6	(2 MSL) CLOSED		

正常關閉連線 (Normal Close Sequence)
(b) 關閉連線之基本動作表列



正常關閉連線 (Normal Close Sequence)
(a) 正常『關閉連線』之狀態圖

TCP三向交握的弱點



TCP的SYN氾濫攻擊法(TCP SYN Flooding Attack)，主要是針對『三向交握』(Three-Way Handshake) 的弱點進行攻擊，所謂的『三向交握』具有以下三步驟:

1)Client端藉由送出一個SYN的同步訊息給伺服端，並要求連線

2)伺服端在收到Client端的同步訊息，並被要求連線時，便會在佇列(Queue)中建立一個『TCP連線項目』(TCP Connection Entry)，直到完成連線後才移除，同時也送出另一個同步訊息SYN，以及回應一個ACK給Client送來的SYN同步訊息，也就是SYN+ACK

3)Client針對伺服端送來的同步訊息，再回應一個確認ACK，此時連線才正式完成

但是在第三步驟Client的回應ACK訊息，倘若遲遲無法回應給伺服端，伺服端將會一直『等待』(wait)，直到『逾時』(timeout)後，才會將佇列(Queue)中的『TCP連線項目』(TCP Connection Entry)移除

因為每一個伺服器的佇列(Queue)能容納『TCP連線項目』(TCP Connection Entry)的個數有線，一旦佇列存滿後，後來的連線將會被丟棄(discard)

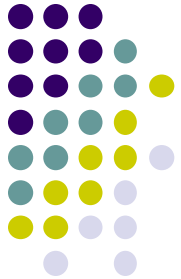
TCP SYN Flooding Attack

首先，先偽造自己的IP Address為其他主機或是不存在的主機IP Address，不會使用自己真正的IP Address來要求與伺服器連線，稱為『偽造IP』(IP Spoofing)

當攻擊者在進行『三向交握』的第一步驟，將同步訊息SYN送達伺服器，此時伺服器要回應『三向交握』的第二步驟SYN+ACK時，所回應的目的IP Address已不是真正的攻擊者，而是被偽裝的IP Address，此時受攻擊者伺服器將會一直等待(wait)『三向交握』的第三步驟ACK回應，直到逾時(timeout)，形成『半開啟連線』(half-open connection)

當攻擊者連續發出大量的『半開啟連線』(half-open connection)，將會迅速地充爆伺服器的佇列(Queue)，造成伺服器的網路服務無法正常地工作，此種攻擊也是『阻斷服務』(Daniel of Service, 簡稱DoS) 的一種

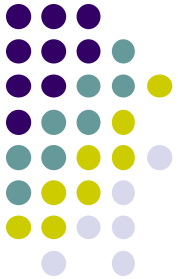
Addressing in various layers



- IP-layer --- type field of Interface header
- TCP/UDP-layer --- Protocol field of IP header
- Application-layer --- port field of TCP/UDP header

  In general, an application entity uses both port number and IP address to uniquely identify each other

User-data integrity of TCP/UDP



- Checksum over pseudo-header & entire TCP/UDP segment
- Pseudo-header (12 Bytes) includes

  (1) source IP address (2) destination IP address

  (3) zero-valued octet (4) 8-bit protocol field

  (5) 2-byte of UDP/TCP length
- Checksum is done byte by byte, then take its 1's complement

  Note: the checksum field is first initialized by zeroes

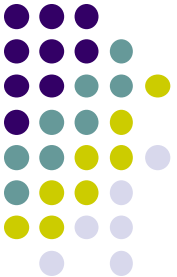
Performance Issues over Transport layer (1/3)

- Performance requirements for application: response time and throughput

- Performance metrics for data networks and the Internet: delay (transmission, propagation, processing, and queueing) and throughput (in general, transmission delay dominates the propagation delay in the case of slow data links)

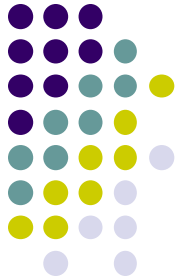
- The delay depends on data rate, distance, velocity of propagation signal, and the size of the packet. Define a= propagation delay / transmission delay, that is propagation delay x data rate (throughput), which is transmission channel bit-length / PDU bit-length

Performance Issues over Transport layer (2/3)



- a = R*D/L is a single critical system parameter
- Flow control, error control, and congestion control are three issues to be considered, especially when a is large.
- Several mechanisms are proposed in data link layer for flow control and error control.
- The object of congestion control is to maintain the number of packets within the network below the level at which performance falls off dramatically.

Performance Issues over Transport layer (3/3)

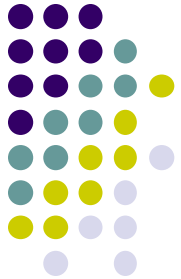


- Backpressure – can be selectively applied to logical connections so that the flow is restricted or halted on some connections, this restriction propagates back along to the source.
- Choke Packet – is a control packet generated at a congested node and transmitted back to a source node to restrict traffic flow. (such as ICMP source quench)
- Implicit Congestion Signaling – a source can detect increased delay and packet discarded
- Explicit Congestion Signaling – the network alerts end systems to growing congestion within the network and the end systems take steps to reduce the offered load to the network.

Interconnecting LANs (1/3)

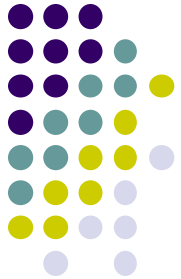
- A segment is a section of a network separated by bridges, switches, and routers
- Bridge is a layer-2 device, uses MAC address to make decisions regarding forwarding data packets
- Bridge forwards broadcasts and is susceptible to broadcast storms.
- Hub provides a technology for sharing access to network with all computers connected to its ports in the LAN but lacks the capability to isolate the data traffic and to provide a direct data connection.

Interconnecting LANs (2/3)



- Layer-2 switch provides direct data connections, minimizing data collisions, and maximizing bandwidth use.
- Switch has multiple ports (similar to hub) and switch in data port to port (similar to bridge), sometime it is called multiport bridge.
- Layer-3 or layer-4 switches are becoming available.
- Router is a layer-3 device, uses IP address to making routing decision regarding forwarding data packets.

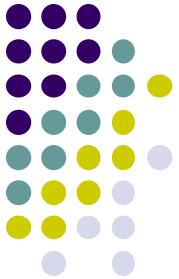
Interconnecting LANs (3/3)



- A router must have a minimum of three ports.

- Router is configured to know how to route data packets entering or exiting the LAN (while layer-2 switch knows how to forward data to hosts physically connected to its port)

- Gateway describes the networking device that enables hosts in a LAN to connect to networks (and hosts) outside the LAN.



Troubleshooting TCP/IP

- Testing Basic Connectivity
  - Ping
- Testing Network Access
  - Ifconfig, netstat, arp
- Tracing Route (checking routing)
  - Traceroute, Ripquery, netstat
- Checking Name Space
  - Nslookup, dig
- Analyzing Protocol Problems
  - Etherfind (Wireshark), Tcpdump