

## TÀI LIỆU TCP

### I. Giới thiệu về TCP

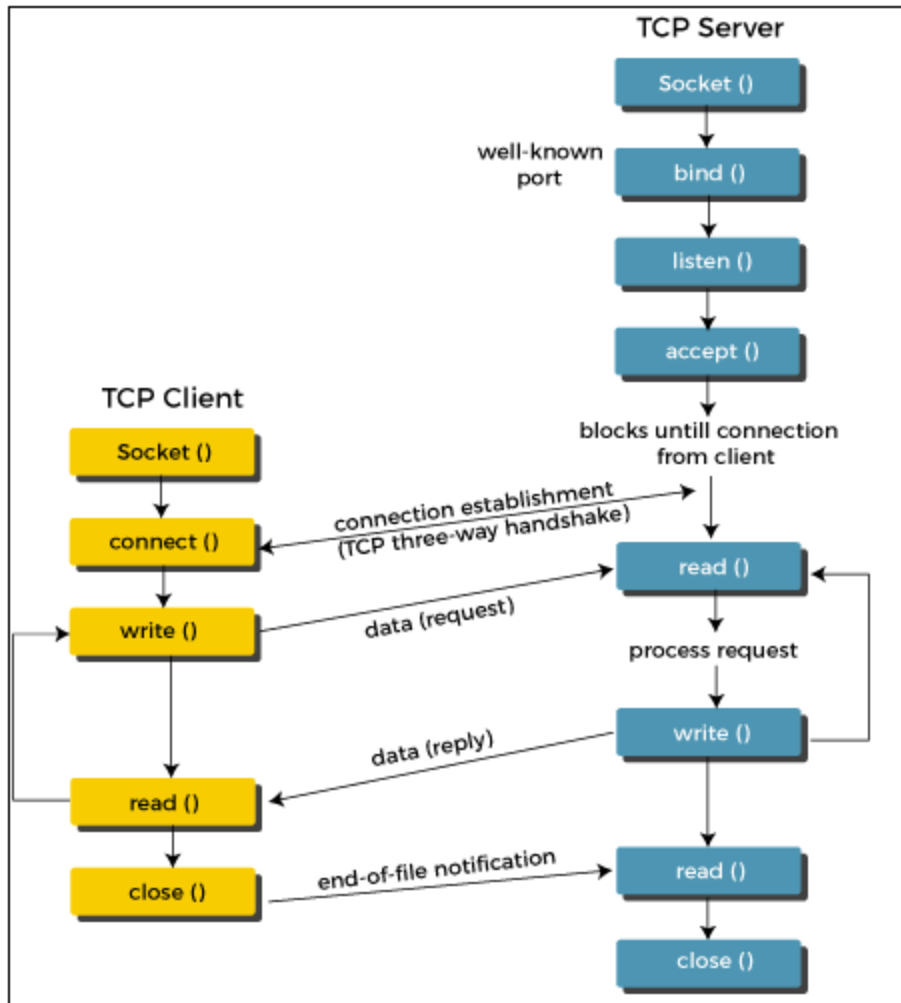
TCP (Transmission Control Protocol) là một trong những giao thức chính của bộ giao thức Internet (TCP/IP) và đóng vai trò trong việc thiết lập và duy trì một kết nối truyền thông đáng tin cậy giữa hai máy tính qua mạng.

#### **Đặc điểm chính:**

- **Connection-Oriented:** Trước khi trao đổi dữ liệu bắt đầu, TCP thiết lập một kết nối giữa bên gửi và bên nhận bằng cách sử dụng quy trình bắt tay ba bước. Điều này đảm bảo cả hai bên sẵn sàng truyền và nhận dữ liệu
- **Reliable Data Delivery:** TCP cung cấp việc giao nhận dữ liệu đáng tin cậy bằng cách sử dụng các phản hồi và số thứ tự. Khi bên gửi gửi các gói dữ liệu, bên nhận xác nhận đã nhận được. Nếu không nhận được xác nhận, bên gửi cho rằng gói dữ liệu bị mất và gửi lại.
- **Flow Control:** TCP bao gồm các cơ chế để ngăn không cho bên nhận quá nhiều dữ liệu cùng lúc. Điều này ngăn chặn tình trạng quá tải và đảm bảo việc truyền dữ liệu hiệu quả.
- **Congestion Control:** TCP được thiết kế để phát hiện tắc nghẽn mạng và điều chỉnh tốc độ truyền của nó tương ứng. Điều này giúp ngăn chặn tắc nghẽn mạng ảnh hưởng đến hiệu suất tổng thể.
- **Full Duplex Communication:** TCP hỗ trợ cho việc cả hai bên có thể gửi và nhận dữ liệu độc lập và đồng thời.
- **Segmentation and Reassembly:** TCP chia dữ liệu thành các đơn vị nhỏ hơn gọi là các phân đoạn trước khi truyền. Bên nhận sau đó lắp ráp các phân đoạn này thành dữ liệu ban đầu.
- **Header Information:** Mỗi segment TCP chứa thông tin header với các trường khác nhau, bao gồm destination port, source port, sequence number và ack, và cờ điều khiển (ví dụ: SYN, ACK, FIN).

**Socket:** một socket đại diện cho một điểm kết nối trong mạng, cho phép truyền và nhận dữ liệu giữa các máy tính hoặc quá trình khác nhau một socket kết hợp địa chỉ IP của máy tính và port để định danh một điểm giao tiếp duy nhất trong mạng. Các ứng dụng có thể sử dụng sockets để gửi và nhận dữ liệu thông qua mạng.

#### **Mô hình client-server:**



Quy trình bắt tay 3 bước:

### Client state

```

clientSocket = socket(AF_INET, SOCK_STREAM)
LISTEN
clientSocket.connect((serverName, serverPort))
  
```

SYNSENT

ESTAB

choose init seq num, x  
send TCP SYN msg

received SYNACK(x)  
indicates server is live;  
send ACK for SYNACK;  
this segment may contain  
client-to-server data

SYNbit=1, Seq=x

SYNbit=1, Seq=y  
ACKbit=1; ACKnum=x+1

ACKbit=1, ACKnum=y+1

### Server state

```

serverSocket = socket(AF_INET, SOCK_STREAM)
serverSocket.bind('', serverPort)
serverSocket.listen(1)
connectionSocket, addr = serverSocket.accept()
  
```

LISTEN

SYN RCVD

ESTAB

choose init seq num, y  
send TCP SYNACK  
msg, acking SYN

received ACK(y)  
indicates client is live

### Segment:

Trong giao thức TCP, một "segment" là một đơn vị dữ liệu cơ bản được sử dụng để truyền thông tin giữa các máy tính thông qua mạng. Mỗi segment chứa một phần của dữ liệu cần gửi, cùng với thông tin tiêu đề và các trường điều khiển liên quan đến việc giao nhận dữ liệu qua mạng.

Cấu trúc của một segmen

## TCP segment structure

