



Facultad de Ciencias
de la Computación



PROGRAMACIÓN DISTRIBUIDA

PRACTICA 1: Threeway HandShake

Profesor: GUSTAVO EMILIO MENDOZA OLGUIN

Alumno : Marco Antonio Bañuelos López

Matrícula : 201726707

18 de agosto de 2023

Otoño 2023

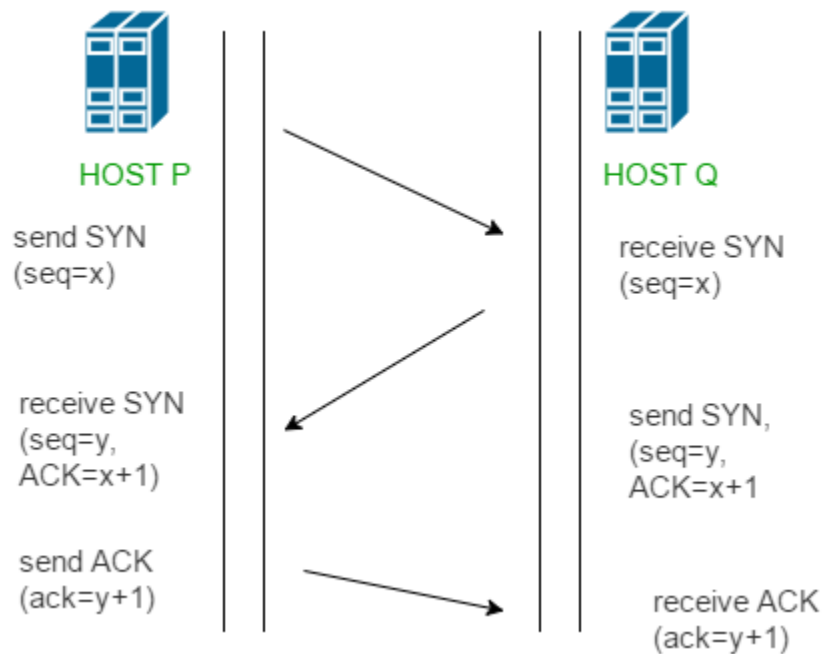
INTRODUCCION

El Protocolo de Control de Transmisión (TCP por sus siglas en inglés, Transmission Control Protocol) es un protocolo de transporte utilizado en redes de computadoras para proporcionar una comunicación confiable y orientada a la conexión entre dispositivos.

El objetivo de esta práctica es comprender el funcionamiento de uno de los pasos fundamentales que ocurren cuando ejecutamos el protocolo, específicamente al momento de establecer conexión entre dos dispositivos.

Threeway HandShake

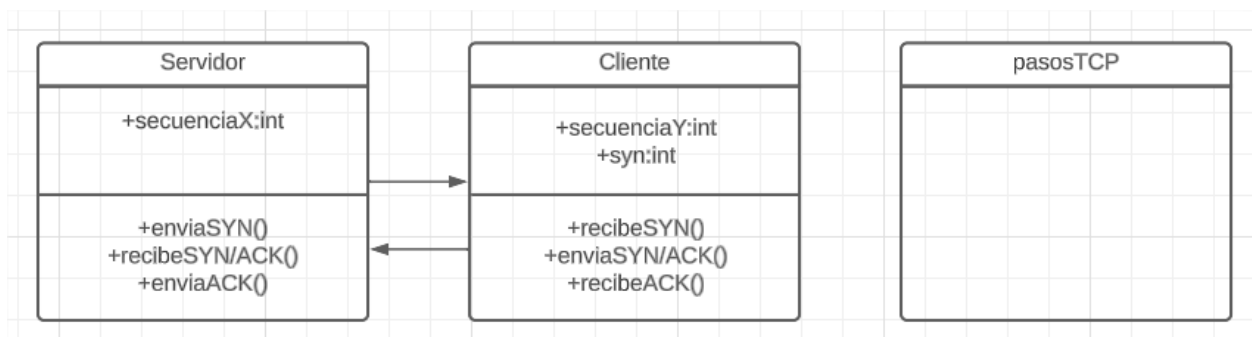
También denominado protocolo de acuerdo de tres vías consiste en un saludo y un acuerdo entre dos dispositivos antes de empezar a enviar datos. El proceso de Tres Vías de Saludo consiste en los siguientes pasos:



1. Envío de Solicitud de Inicio: El proceso comienza con el dispositivo que inicia la comunicación (HOST P) enviando un segmento SYN al dispositivo receptor (HOST Q). Este mensaje es una solicitud para establecer una conexión. Junto al segmento SYN, se añade una secuencia X de números aleatorios

2. Confirmación de Disponibilidad: En respuesta a la solicitud de inicio, el dispositivo receptor (HOST Q) responde al remitente, su respuesta consiste en incrementar la secuencia X, para luego generar una secuencia aleatoria Y, y enviar un segmento SYN/ACK al remitente, para indicar que está listo para comunicarse.
3. Confirmación de Inicio: Finalmente, el remitente envía una última confirmación en forma de segmento ACK, que ha incrementado el valor de la secuencia Y, recibiendo el receptor, para indicar que ha recibido la respuesta y está listo para comenzar la comunicación real. Esta confirmación completa el proceso de tres pasos, similar a decir "Entendido, empezaré a hablar ahora".

DIAGRAMAS



CÓDIGO

Main PasosTCP

```

package protocoloTCP;

// Definición de la clase Persona
class cliente {
    int serialX;

    generadorRandom randomGenerator; // Agregar una instancia de
    generadorRandom

    public cliente() {
        randomGenerator = new generadorRandom(); // Inicializar el generadorRandom
    }
  
```

```

        serialX = randomGenerator.generarSerial(); // Generar un número aleatorio y
asignarlo a serialX
    }

    ///
    public int getSerialX() {
        return serialX;
    }
    ///
    public void enviandoSyn() {
        // Simulate sending SYN packet
        System.out.println("Cliente: Enviando paquete SYN = " + serialX);
    }

    public void recibiendoSyn(int syn) {
        // Simulate receiving SYN-ACK packet
        System.out.println("Cliente: Recibiendo paquete SYN = "+ syn);
    }
    public void recibiendoAck(int serialY) {
        // Simulate receiving SYN-ACK packet
        System.out.println("Cliente: Recibiendo paquete ACK = "+ serialY);
    }
    public void enviandoAck() {
        // Simulate sending ACK packet
        System.out.println("Cliente: Sending ACK packet");
    }
}

```

```

class servidor {
    int serialY;
    int syn;

    generadorRandom randomGenerator; // Agregar una instancia de
    generadorRandom

    public servidor() {

        randomGenerator = new generadorRandom(); // Inicializar el
    generadorRandom

        serialY = randomGenerator.generarSerial();
    }

    public int sendAck() {
        return serialY;
    }

    public int sendSyn() {
        return syn;
    }

    public servidor(int serialX) {
        this.syn = serialX + 1;
    }

    public void recibiendoSyn(int serialX) {
        // Simulate receiving SYN packet
        System.out.println("Server: Recibiendo paquete SYN = " + serialX);
    }
}

```

```

    }

    public void enviandoSynAck() {
        // Simulate sending SYN-ACK packet
        System.out.println("Servidor: Enviando paquete SYN (sumandole + 1) = " + syn );
        System.out.println("Servidor: Enviando paquete ACK = " + serialY );
    }

    public void recibiendoAck() {
        // Simulate receiving ACK packet
        System.out.println("Server: Received ACK packet");
    }
}

public class pasosTCP {
    public static void main(String[] args) {
        cliente client = new cliente();
        servidor server = new servidor(client.getSerialX());
        int synfromServer = server.sendSyn();
        int serialYFromServer = server.sendAck();
        // Simulate three-way handshake
        System.out.println("--Cambiano el estado a ABIERTO");
        System.out.println(" Paso 1 ");
        client.enviandoSyn();
        server.recibiendoSyn(client.getSerialX());

        System.out.println(" Paso 2 ");
    }
}

```

```

server.enviandoSynAck();

System.out.println(" //////////////////////////////////");

client.recibiendoSyn(synfromServer);
client.recibiendoAck(serialYFromServer);


System.out.println(" Paso 3 ");


client.enviandoAck();
server.recibiendoAck();
}
}

```

Clase generadorRandom

```

import java.util.Random;

public class generadorRandom {
    private Random random;

    public generadorRandom() {
        random = new Random();
    }

    public int generarSerial() {
        int numero = 0; // Inicializar con un dígito aleatorio de 1 a 9

        for (int i = 0; i < 9; i++) { // Solo necesitas 9 iteraciones para un número de 10 dígitos
            numero = numero * 10 + random.nextInt(10);
        }
    }
}

```

```
    return numero;  
}  
}
```

CONCLUSIONES

El Threeway HandShake es una parte esencial en las comunicaciones TCP porque garantiza que ambas partes estén sincronizadas antes de transferir datos. Esto previene problemas como la pérdida de datos debido a dispositivos no listos, errores en el orden de transmisión y otros desafíos que pueden surgir en las redes. Asegurando además, una comunicación confiable y ordenada al establecer una conexión segura, contribuyendo así a la base de la transferencia de datos exitosa en las comunicaciones modernas.