

Instituto Politécnico Nacional
Escuela Superior de Cómputo

Implementación de un token-ring

Tarea 2

Alumno: Francisco Nicolas Sánchez García
Asignatura Desarrollo de Sistemas Distribuidos
Profesor: Carlos Pineda Guerrero
Grupo 4CV12

ÍNDICE

| | |
|--------------------|---|
| Introducción | 1 |
| Desarrollo | 1 |
| Conclusiones | 6 |

INTRODUCCIÓN

Un token-ring es una topología de área local en forma de anillo. En esta tarea se realiza el anillo con cuatro nodos, donde la comunicación solo va en un sentido del anillo.

Se implementan sockets seguros usando los conocimientos adquiridos en la clase de creación de certificados.

DESARROLLO

Se codificó la clase Token.java siguiendo las indicaciones de la tarea, posteriormente, se compila y se muestra su funcionamiento.

En la tarea venía las instrucciones para realizar la tarea con sockets normales, se modificaron los sockets de la comunicación de entrada y salida para hacerlos seguros. Además, también se agregó otro bloque try-catch en la línea para enviar el token entre los hosts.

Creación de certificado y capturas de pantalla:

```
C:\Users\fnico\Documents\github\desarrolloDeSistemasDistribuidos\tareas\token-ring>keytool -genkeypair -keyalg RSA -alias certificado_servidor -keystore keystore_servidor.jks -storepass 1234567
What is your first and last name?
[Unknown]: nicolas
What is the name of your organizational unit?
[Unknown]: org.001
What is the name of your organization?
[Unknown]: org.distribuidos
What is the name of your City or Locality?
[Unknown]: Mexico
What is the name of your State or Province?
[Unknown]: Mexico
What is the two-letter country code for this unit?
[Unknown]: MX
Is CN=nicolas, OU=org.001, O=org.distribuidos, L=Mexico, ST=Mexico, C=MX correct?
[no]: yes

Generating 2,048 bit RSA key pair and self-signed certificate (SHA256withRSA) with a validity of 90 days
for: CN=nicolas, OU=org.001, O=org.distribuidos, L=Mexico, ST=Mexico, C=MX

C:\Users\fnico\Documents\github\desarrolloDeSistemasDistribuidos\tareas\token-ring>keytool -exportcert -keystore keystore_servidor.jks -alias certificado_servidor -rfc -file certificado_servidor.pem
Enter keystore password:
Certificate stored in file <certificado_servidor.pem>

C:\Users\fnico\Documents\github\desarrolloDeSistemasDistribuidos\tareas\token-ring>keytool -import -alias certificado_servidor -file certificado_servidor.pem -keystore keystore_cliente.jks -storepass 123456
Owner: CN=nicolas, OU=org.001, O=org.distribuidos, L=Mexico, ST=Mexico, C=MX
Issuer: CN=nicolas, OU=org.001, O=org.distribuidos, L=Mexico, ST=Mexico, C=MX
Serial number: c391bb8e08bf2e04
Valid from: Fri Sep 10 14:49:04 CDT 2021 until: Thu Dec 09 13:49:04 CST 2021
Certificate fingerprints:
    SHA1: CD:00:7F:2F:0F:11:62:29:B3:ED:C9:8B:39:74:4C:FB:92:60:53:EE
    SHA256: 0F:EC:62:E7:46:87:F1:F9:A8:11:A4:95:60:39:13:2C:11:86:AB:02:DE:16:9B:8D:02:9E:AC:7D:4A:90:8D:43
Signature algorithm name: SHA256withRSA
Subject Public Key Algorithm: 2048-bit RSA key
```

```
C:\Windows\system32\cmd.exe
servidor.pem -keystore keystore_cliente.jks -storepass 123456
Owner: CN=nicolas, OU=org.001, O=org.distribuidos, L=Mexico, ST=Mexico, C=MX
Issuer: CN=nicolas, OU=org.001, O=org.distribuidos, L=Mexico, ST=Mexico, C=MX
Serial number: c391bb8e88bf2e84
Valid from: Fri Sep 10 14:49:04 CDT 2021 until: Thu Dec 09 13:49:04 CST 2021
Certificate fingerprints:
    SHA1: CD:00:7F:2F:0F:11:62:29:B3:ED:C9:8B:39:74:4C:FB:92:60:53:EE
    SHA256: 0F:EC:62:E7:46:87:F1:F9:A8:11:A4:95:60:39:13:2C:11:86:AB:02:DE:16:9B:8D:02:9E:AC:7D:4A:90:0D:43
Signature algorithm name: SHA256withRSA
Subject Public Key Algorithm: 2048-bit RSA key
Version: 3

Extensions:

#1: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: 86 E0 06 08 32 34 D5 1D AE D7 3A 48 4E 44 42 3D ....24....HNDB=
0010: 37 82 79 5A                                     7.yZ
]
]

Trust this certificate? [no]: yes
Certificate was added to keystore

C:\Users\fnico\Documents\github\desarrolloDeSistemasDistribuidos\tareas\token-ring>
```

Código fuente de Token.java

```
import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.IOException;
import java.net.ServerSocket;
import java.net.Socket;

import javax.net.ServerSocketFactory;
import javax.net.ssl.SSLServerSocketFactory;
import javax.net.ssl.SSLSocketFactory;

class Token {
    static DataInputStream entrada;
    static DataOutputStream salida;
    static boolean inicio = true;
    static String ip;
    static int nodo;
    static long token;

    static class Worker extends Thread {
        public void run() {
            // algoritmo 1
            try {
                System.setProperty("javax.net.ssl.keyStore",
"keystore_servidor.jks");
                System.setProperty("javax.net.ssl.keyStorePassword",
"1234567");

                ServerSocketFactory socketFactory =
(SSLServerSocketFactory) SSLServerSocketFactory.getDefault();
                ServerSocket servidor;
```

```

        servidor = socketFactory.createServerSocket(20000 +
nodo);

        Socket conexion;
        conexion = servidor.accept();
        entrada = new DataInputStream(conexion.getInputStream());

        } catch (Exception e) {
            System.out.println("Error:" + e.getMessage());
        }
    }

}

public static void main(String[] args) throws Exception {
    if (args.length != 2) {
        System.err.println(
            "Se debe pasar como parametros el numero del nodo y
la IP del siguiente nodo en el anillo");
        System.exit(1);
    }

    nodo = Integer.valueOf(args[0]);
    ip = args[1];

    // Algoritmo 2
    Worker w;
    w = new Worker();
    w.start();

    System.setProperty("javax.net.ssl.trustStore",
"keystore_cliente.jks");
    System.setProperty("javax.net.ssl.trustStorePassword", "123456");
    SSLSocketFactory cliente = (SSLSocketFactory)
SSLSocketFactory.getDefault();
    Socket conexion = null;
    while (true) {
        try {
            conexion = cliente.createSocket(ip, 20000 + (nodo + 1) %
4);

            // conexion = new Socket(ip, 20000 + (nodo + 1) % 4);
            break;
        } catch (Exception e) {
            Thread.sleep(500);
        }
    }
    salida = new DataOutputStream(conexion.getOutputStream());
    w.join();
    while (true) {
        if (nodo == 0) {
            if (inicio) {
                inicio = false;
                token = 1;
            } else {
                token = entrada.readLong();
                token++;
                System.out.println("Nodo: " + nodo + ", token = " +
token);
            }
        }
    }
}

```

```

    }
} else {
    try {
        token = entrada.readLong();
    } catch (Exception e) {
        System.out.println("Conexion terminada:" +
e.getMessage());
        break;
    }
    token++;
    System.out.println("Nodo: " + nodo + ", token = " +
token);
}

if (nodo == 0 && token >= 1000) {
    break;
}
salida.writeLong(token);
}
Thread.sleep(1000);
}
}
}

```

Compilación del programa:

```

C:\Users\fnico\Documents\github\desarrolloDeSistemasDistribuidos\tareas\token-ring>dir
El volumen de la unidad C no tiene etiqueta.
El número de serie del volumen es: 0E29-1C7C

Directorio de C:\Users\fnico\Documents\github\desarrolloDeSistemasDistribuidos\tareas\token-ring
10/09/2021 02:49 p. m. <DIR> .
10/09/2021 02:49 p. m. <DIR> ..
10/09/2021 02:49 p. m. 1,294 certificado_servidor.pem
10/09/2021 02:49 p. m. 1,226 keystore_cliente.jks
10/09/2021 02:49 p. m. 2,651 keystore_servidor.jks
08/09/2021 09:22 p. m. 3,389 Token.java
4 archivos 8,560 bytes
2 dirs 310,129,172,480 bytes libres

C:\Users\fnico\Documents\github\desarrolloDeSistemasDistribuidos\tareas\token-ring>javac Token.java

C:\Users\fnico\Documents\github\desarrolloDeSistemasDistribuidos\tareas\token-ring>dir
El volumen de la unidad C no tiene etiqueta.
El número de serie del volumen es: 0E29-1C7C

Directorio de C:\Users\fnico\Documents\github\desarrolloDeSistemasDistribuidos\tareas\token-ring
10/09/2021 02:50 p. m. <DIR> .
10/09/2021 02:50 p. m. <DIR> ..
10/09/2021 02:49 p. m. 1,294 certificado_servidor.pem
10/09/2021 02:49 p. m. 1,226 keystore_cliente.jks
10/09/2021 02:49 p. m. 2,651 keystore_servidor.jks
10/09/2021 02:50 p. m. 1,744 Token$Worker.class
10/09/2021 02:50 p. m. 2,779 Token.class
08/09/2021 09:22 p. m. 3,389 Token.java
6 archivos 13,083 bytes
2 dirs 310,129,029,120 bytes libres

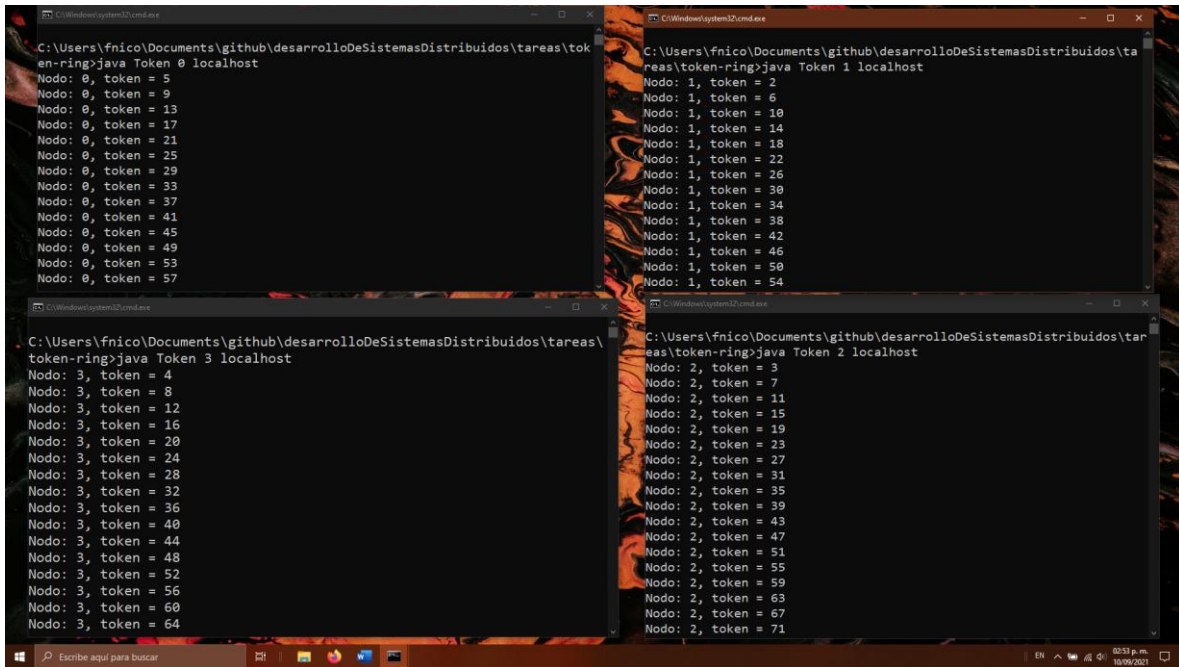
C:\Users\fnico\Documents\github\desarrolloDeSistemasDistribuidos\tareas\token-ring>

```

Para la ejecución, se corre el programa compilado en cuatro ventanas, se pasa como parámetro el número del nodo y la dirección IP.

A continuación, se muestra la ejecución de las cuatro ventanas.

Inicio de ejecución del programa:



```
C:\Users\fnico\Documents\github\desarrolloDeSistemasDistribuidos\tareas\token-ring>java Token 0 localhost
Nodo: 0, token = 5
Nodo: 0, token = 9
Nodo: 0, token = 13
Nodo: 0, token = 17
Nodo: 0, token = 21
Nodo: 0, token = 25
Nodo: 0, token = 29
Nodo: 0, token = 33
Nodo: 0, token = 37
Nodo: 0, token = 41
Nodo: 0, token = 45
Nodo: 0, token = 49
Nodo: 0, token = 53
Nodo: 0, token = 57

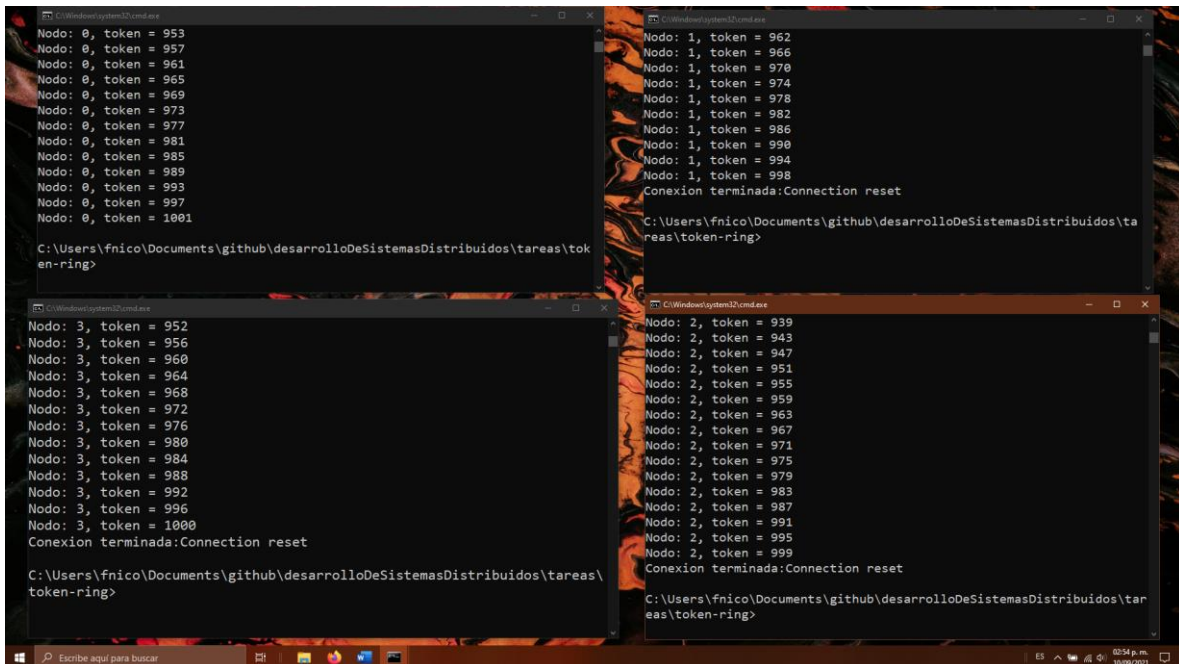
C:\Users\fnico\Documents\github\desarrolloDeSistemasDistribuidos\tareas\token-ring>java Token 1 localhost
Nodo: 1, token = 2
Nodo: 1, token = 6
Nodo: 1, token = 10
Nodo: 1, token = 14
Nodo: 1, token = 18
Nodo: 1, token = 22
Nodo: 1, token = 26
Nodo: 1, token = 30
Nodo: 1, token = 34
Nodo: 1, token = 38
Nodo: 1, token = 42
Nodo: 1, token = 46
Nodo: 1, token = 50
Nodo: 1, token = 54

C:\Users\fnico\Documents\github\desarrolloDeSistemasDistribuidos\tareas\token-ring>java Token 2 localhost
Nodo: 2, token = 3
Nodo: 2, token = 7
Nodo: 2, token = 11
Nodo: 2, token = 15
Nodo: 2, token = 19
Nodo: 2, token = 23
Nodo: 2, token = 27
Nodo: 2, token = 31
Nodo: 2, token = 35
Nodo: 2, token = 39
Nodo: 2, token = 43
Nodo: 2, token = 47
Nodo: 2, token = 51
Nodo: 2, token = 55
Nodo: 2, token = 59
Nodo: 2, token = 63
Nodo: 2, token = 67
Nodo: 2, token = 71

C:\Users\fnico\Documents\github\desarrolloDeSistemasDistribuidos\tareas\token-ring>java Token 3 localhost
Nodo: 3, token = 4
Nodo: 3, token = 8
Nodo: 3, token = 12
Nodo: 3, token = 16
Nodo: 3, token = 20
Nodo: 3, token = 24
Nodo: 3, token = 28
Nodo: 3, token = 32
Nodo: 3, token = 36
Nodo: 3, token = 40
Nodo: 3, token = 44
Nodo: 3, token = 48
Nodo: 3, token = 52
Nodo: 3, token = 56
Nodo: 3, token = 60
Nodo: 3, token = 64
```

Como podemos ver, la primera vez que se imprime el valor del token es el número 5 en el nodo 0 ya que no se incluye ninguna instrucción para imprimir el token cuando se inicializa y la bandera de inicio pasa a ser falsa.

Finalización del programa:



```
Nodo: 0, token = 953
Nodo: 0, token = 957
Nodo: 0, token = 961
Nodo: 0, token = 965
Nodo: 0, token = 969
Nodo: 0, token = 973
Nodo: 0, token = 977
Nodo: 0, token = 981
Nodo: 0, token = 985
Nodo: 0, token = 989
Nodo: 0, token = 993
Nodo: 0, token = 997
Nodo: 0, token = 1001

C:\Users\fnico\Documents\github\desarrolloDeSistemasDistribuidos\tareas\token-ring>

Nodo: 1, token = 962
Nodo: 1, token = 966
Nodo: 1, token = 970
Nodo: 1, token = 974
Nodo: 1, token = 978
Nodo: 1, token = 982
Nodo: 1, token = 986
Nodo: 1, token = 990
Nodo: 1, token = 994
Nodo: 1, token = 998
Conexion terminada:Connection reset

C:\Users\fnico\Documents\github\desarrolloDeSistemasDistribuidos\tareas\token-ring>

Nodo: 2, token = 939
Nodo: 2, token = 943
Nodo: 2, token = 947
Nodo: 2, token = 951
Nodo: 2, token = 955
Nodo: 2, token = 959
Nodo: 2, token = 963
Nodo: 2, token = 967
Nodo: 2, token = 971
Nodo: 2, token = 975
Nodo: 2, token = 979
Nodo: 2, token = 983
Nodo: 2, token = 987
Nodo: 2, token = 991
Nodo: 2, token = 995
Nodo: 2, token = 999
Conexion terminada:Connection reset

C:\Users\fnico\Documents\github\desarrolloDeSistemasDistribuidos\tareas\token-ring>

Nodo: 3, token = 952
Nodo: 3, token = 956
Nodo: 3, token = 960
Nodo: 3, token = 964
Nodo: 3, token = 968
Nodo: 3, token = 972
Nodo: 3, token = 976
Nodo: 3, token = 980
Nodo: 3, token = 984
Nodo: 3, token = 988
Nodo: 3, token = 992
Nodo: 3, token = 996
Nodo: 3, token = 1000
Conexion terminada:Connection reset

C:\Users\fnico\Documents\github\desarrolloDeSistemasDistribuidos\tareas\token-ring>
```

CONCLUSIONES

Si bien el uso de redes de token-ring fue muy utilizada, principalmente por IBM en los 80's, el uso de este tipo de redes fue sustituidos por ethernet gracias al menor costo que este suponía de implementar.

Además, una de las dificultades de este tipo de redes que encontré mientras realizaba la ejecución es la gran dependencia de nodos con sus vecinos y la inestabilidad que pueden tener en caso de que alguno de los nodos caiga o se cierren los flujos de entrada, salida o la misma conexión entre los nodos.