DiffServ TOOL

Advanced Network Architectures and Wireless Systems Project

Giuliano Peraz
Valerio Tanferna
Dario Figliuzzi

a.a. 2017-2018

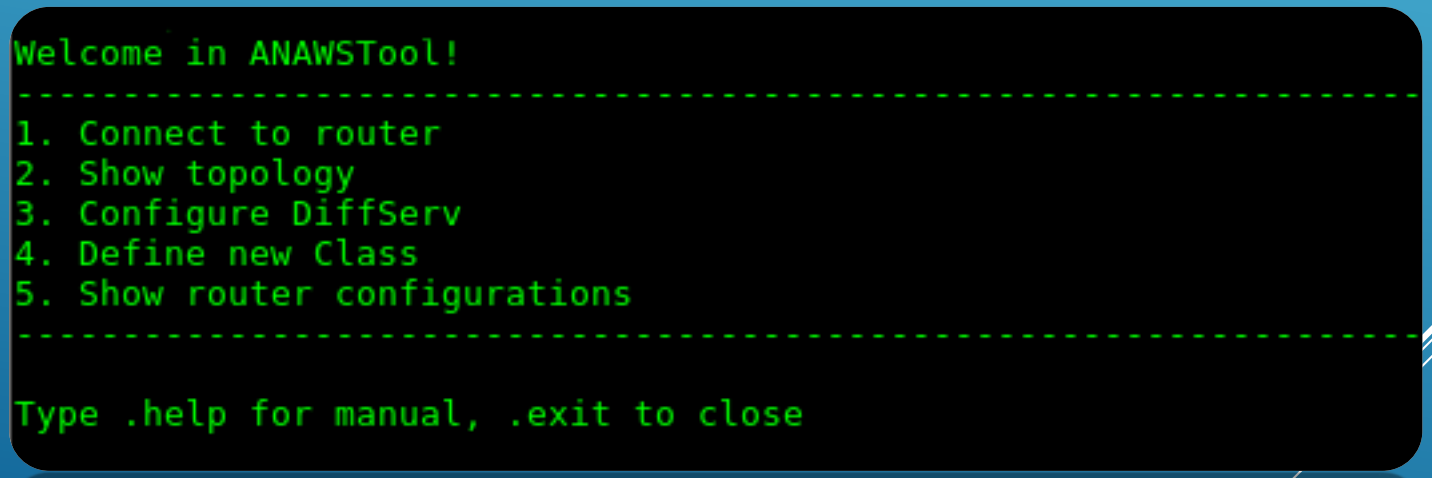
Roadmap

1. **Introduction**
2. Network Structure
3. Protocols
4. Tool Functionalities
5. Java Classes
6. Demo
7. Conclusions
8. Bibliography

1. Introduction

What is DiffServ Tool?

DiffServ Tool is a tool to allow an administrator to configure the cisco routers of a network with a certain set of diffserv traffic classes. The tool configures the routers using a remote SSH connection and exploits frrouting (*https://frrouting.org*) to retrieve the information regarding the network (e.g. the topology, etc).



Roadmap

1. Introduction
2. **Network Structure**
3. Protocols
4. Tool Functionalities
5. Java Classes
6. Demo
7. Conclusions
8. Bibliography

2. Network Structure

The network structure was designed based on the Client-Server paradigm, taking into consideration a university scenario as an example, in which each client subnet represents a Department.

The network consists of 5 subnets:

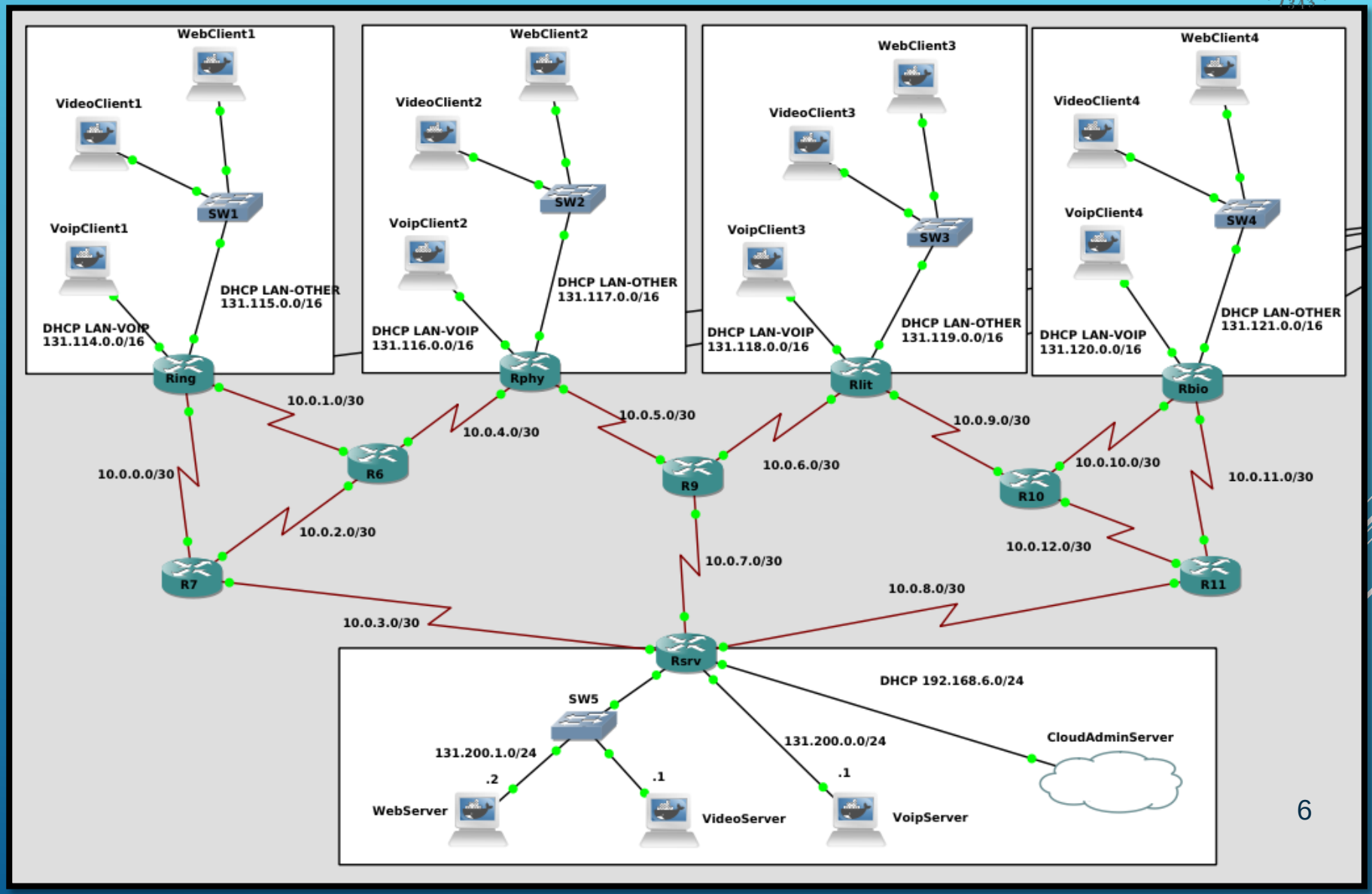
4 CLIENT SIDE each consisting of:

- Client Voip
- Client Video
- Client Web
- Switch interconnecting Client Video and Client Web.
- Edge Router
- Cloud for the admin.

1 SERVER SIDE consisting of:

- Server Voip
- Server Video
- Server Web
- Switch interconnecting Server Video and Server Web.
- Edge Router
- Cloud for the admin.

2. Network Structure

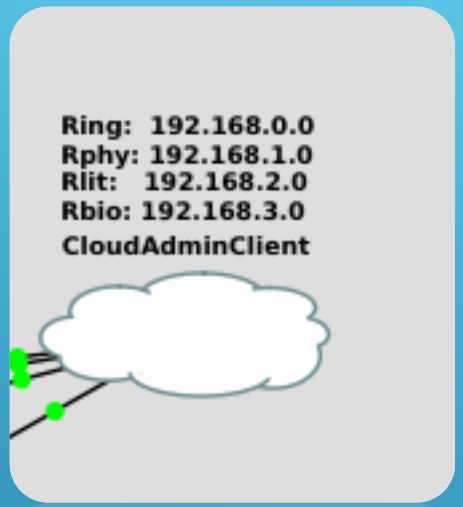


2. Network Structure

The Cloud component is used by the tool as an interface to connect to the network emulated by GNS3.

The **Cloud Client** (CloudAdminClient) is characterized by a tap interface with which you can configure the Edge Routers and the entire network.

The **Cloud Server** (CloudAdminServer) works on the same principle.



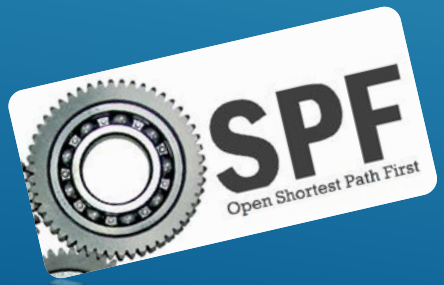
Roadmap

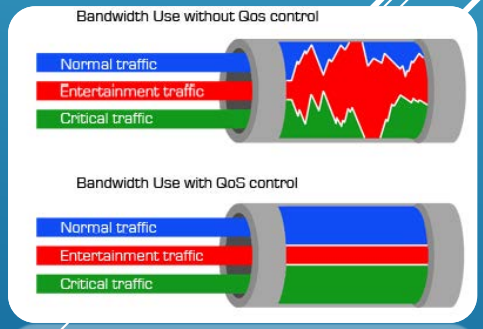
1. Introduction
2. Network Structure
3. **Protocols**
4. Tool Functionalities
5. Java Classes
6. Demo
7. Conclusions
8. Bibliography

3. Protocols

For the implementation of the project we will use the following protocols:

- **SSH**: for remote access to the routers by the administrator;
- **OSPFv2**: as a routing protocol and to obtain information about the network topology;
- **IPv4**: the initial choice was IPv6 to take advantage of all the innovations it brings (such as automatic address selection). Unfortunately FRRouting does not yet support OSPFv3 (compatible with IPv6) and the alternative would be RIPng that does not fit our needs;
- **DiffServ**: to manage and classify network traffic.





Roadmap

1. **Introduction**
2. Network Structure
3. Protocols
4. **Tool Functionalities**
5. Java Classes
6. Demo
7. Conclusions
8. Bibliography

4. Tool Functionalities

The tool is able to automate much of the work that the admin wants to perform. The admin is able to:

- access the routers remotely using informations from the network topology extracted with the help of the FRrouting daemons;
- configure the values of the Per-Hop Behavior on the border routers;
- modify the parameters for the classification of flows by setting the values, for example, of the guaranteed band;
- check that the entered commands have been correctly executed.

4. Tool Functionalities

The tool can be started from the cloud console by typing the command ./nomedeltool. The tool is interactive and will print a menu like:

1. Connect to router allows the admin to connect via ssh to a router.
2. Show topology view the network topology using the ospfd daemon.
3. Configure DiffServ allows to configure DiffServ on one or all routers of the network in an automated way; once this item is selected, the admin will be asked whether to use the standard classes (Video, Web, Voip and Excess) or to define new ones.
4. Define new class allows you to define new service classes for flows.
5. Show router configurations allows you to view the current configuration of the routers for the purpose of verification.

4. Tool Functionalities

Where necessary, once one of the items has been selected, the tool will ask the user questions for entering the parameters.

```
---   List of available routers   ---
Hostname        IP address      ABR?
R7              10.0.3.2
R6              10.0.4.2
R9              10.0.7.2
R10             10.0.12.1
R11             10.0.12.2
Ring            192.168.0.254   ABR
Rlit            192.168.2.254   ABR
Rbio            192.168.3.254   ABR
Rsrv            192.168.6.254   ABR

Choose the router IP of the list to which you want to connect to:
Enter .exit to return back
```

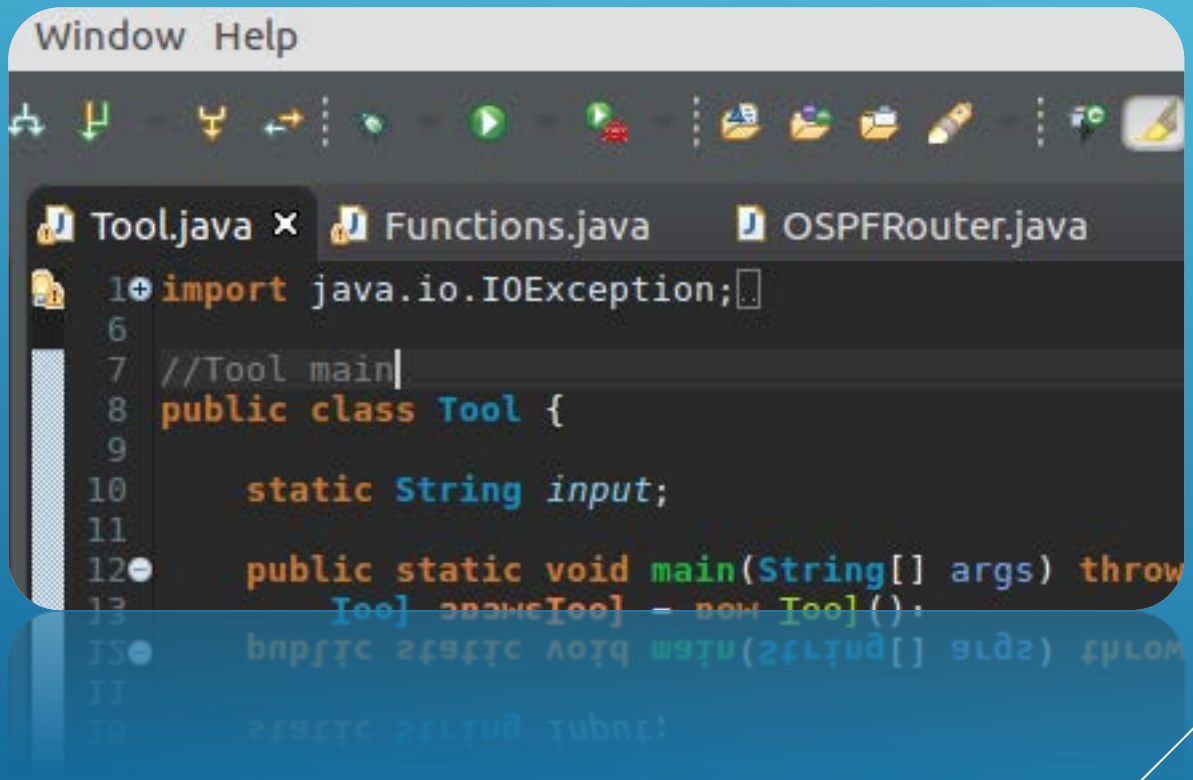
Roadmap

1. Introduction
2. Network Structure
3. Protocols
4. Tool Functionalities
5. **Java Classes**
6. Demo
7. Conclusions
8. Bibliography

5. Java Classes

The whole project has been implemented using Eclipse IDE.

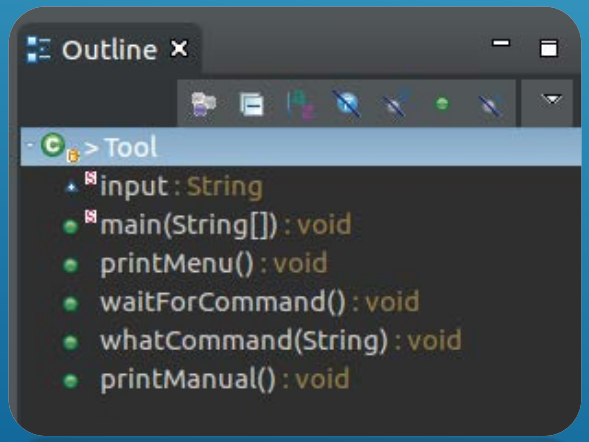
It is composed of three java classes.



5. Java Classes

Tool.java

This class implements the main menu of the tool that goes to call the related functions in *Functions.java.*

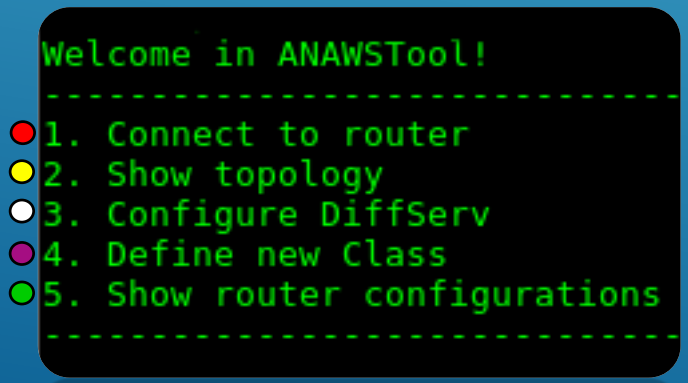


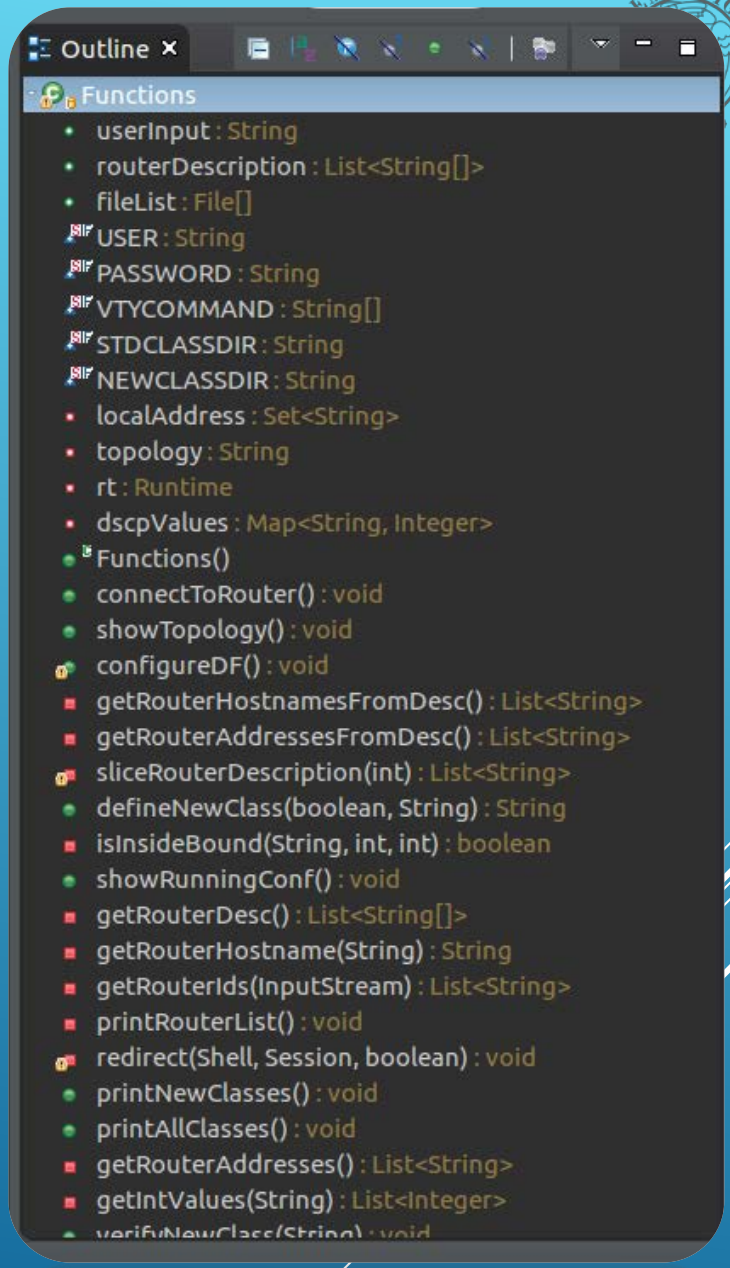

```java
//Tool main
public class Tool {

    static String input;

    public static void main(String[] args) throws IOException {
        Tool anawsTool = new Tool();
        System.out.println("Welcome in ANAWSTool!");
        while(true) {
            anawsTool.printMenu();
        }
    }

    public void printMenu() throws IOException {
        System.out.println("-----------------------------------------------
                + "1. Connect to router\n"
                + "2. Show topology\n"
                + "3. Configure DiffServ\n"
                + "4. Define new Class\n"
                + "5. Show router configurations\n"
                + "-----------------------------------------------
        System.out.println("Type .help for manual, .exit to close\n");
        waitForCommand();
    }

    public void waitForCommand() throws IOException {
        try {
            input = System.console().readLine();
            whatCommand(input);
        }catch (NumberFormatException e) {
        }
    }
```

5. Java Classes

Functions.java

It implements the functionality of the tool and contains the attributes to save the information of the network, such as routers that are there, whether or not the Border Routers and more, so as to avoid continuous connections to the routers of the network.

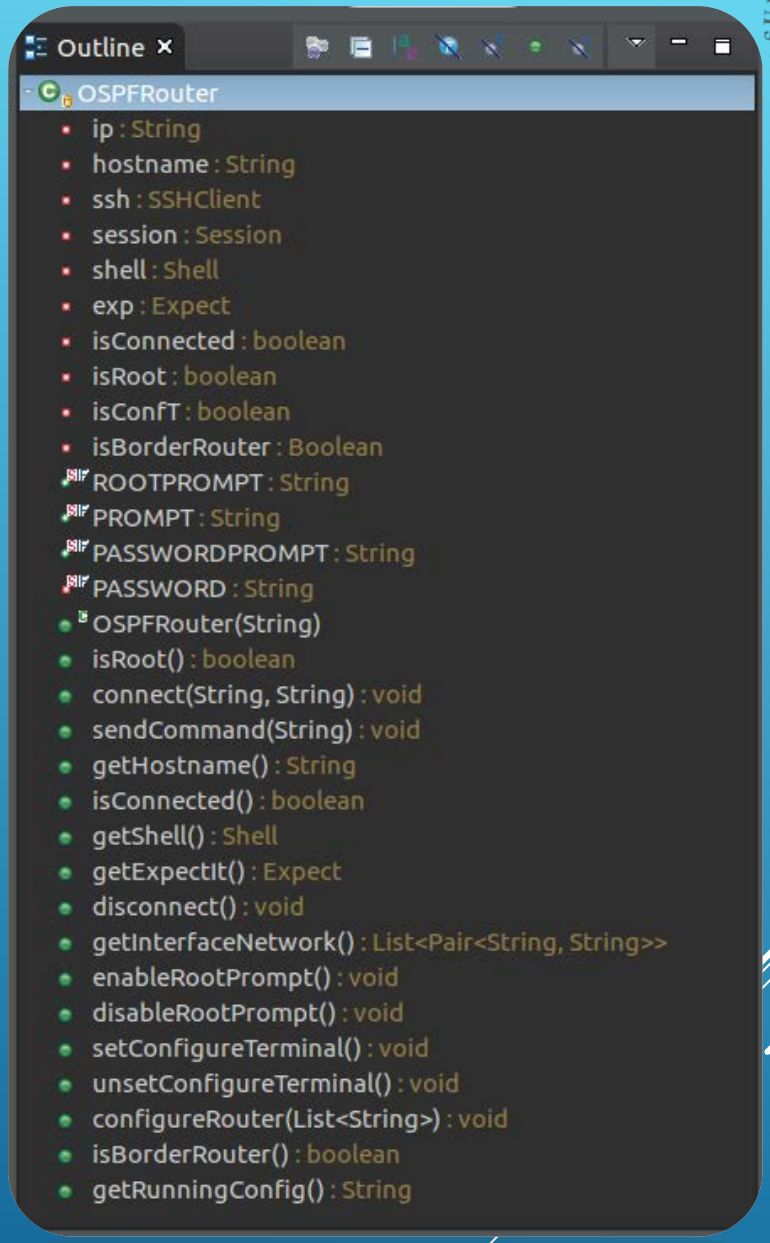




5. Java Classes

OSPFRouter.java

It models a router and takes care to connect/disconnect itself, to request information from the routers of the network, to execute the commands on the routers and manages the information that the methods of *Functions* class requires, such as the hostname, the name of the interfaces and the IP addresses of these, running configuration and others.



Roadmap

1. Introduction
2. Network Structure
3. Protocols
4. Tool Functionalities
5. Java Classes
6. **Demo**
7. Conclusions
8. Bibliography

6. Demo

Roadmap

1. Introduction
2. Network Structure
3. Protocols
4. Tool Functionalities
5. Java Classes
6. Demo
7. **Conclusions**
8. Bibliography

7. Conclusions

The tool is a good demonstration of how to automate the process of configuring Cisco routers within a network. Since it is not a commercial product, it has basic structural defects:

- login credentials
- verification of configuration commands
- user interface
- connection optimization

Roadmap

1. Introduction
2. Network Structure
3. Protocols
4. Tool Functionalities
5. Java Classes
6. Demo
7. Conclusions
8. **Bibliography**

8. Bibliography

- **SSHJ**, Jeroen van Erp, https://github.com/hierynomus/sshj
- **ExpectIT**, Alexey Gavrilov, https://github.com/Alexey1Gavrilov/ExpectI
- **Cisco IOS Configuration Fundamentals Command Reference**, Cisco Systems, Inc., https://www.cisco.com/c/en/us/td/docs/ios/fundamentals/command/reference/cf_book.html
- **OSPFD**, FRRouting Project, https://frrouting.org/user-guide/ospfd.html