COMPUTER NETWORKS

CN201

Group Assignment

Deadline: 15.08.18

Table of Contents

Introduction

According to the scenario that is given to us, it is needed to design a new network for a certain company. It is required to develop an appropriate LAN and the WAN infrastructure for this.

The scenario describes that the company's headquarters (HQ) is located in Colombo and there are two branch offices that are situated at various remote locations in the island. Therefore, the company consists of a head office (HQ) and two branches.

The Headquarters which consists of 80 nodes, is required to be divided into four functional departments namely Management, Sales, Finance and HR/Admin and each department should consist of 12 nodes.

This whole LAN and WAN infrastructure should be designed with separate IP subnets for each department and each branch. The Headquarters and the other two branches are to be connected through a WAN. The steps we took in order to design this particular network is explained vividly in this document.

Network Design

In designing the required network, a suitable topology must be selected initially. Various topologies are available when designing a certain network and we should consider advantages and disadvantages of each in selecting a suitable topology for a network.

The topologies we used and the reasons on why we chose to use them are explained below.

Basically we used Triangle topology (Partially Mesh topology) for connecting main branches to sub branches and also the main branch is again connected to the cloud as well.

Some features of Mesh topology are,

- Fully connected.
- Robust. (Deal with errors)
- Not flexible.

Advantages of Mesh topology

- Data can be transmitted from different devices simultaneously. This topology can withstand high traffic.
- Even if one of the components fails there is always an alternative present. So data transfer doesn't get affected.
- Expansion and modification in topology can be done without disrupting other nodes.

Disadvantages of Mesh topology

- There are high chances of redundancy in many of the network connections.
- Overall cost of this network is way too high as compared to other network topologies.
- Set-up and maintenance of this topology is very difficult. Even administration of the network is tough.

Moreover, we used Star topology for interconnecting the internal networks. The main reason why we used star topology was because of how easy it is to create subnets using it.

Some features of Star Topology are,

- Every node has its own dedicated connection to the hub.
- Hub acts as a repeater for data flow.
- Can be used with twisted pair, Optical Fiber or coaxial cable.

Advantages of Star Topology

- Fast performance with few nodes and low network traffic.
- Hub can be upgraded easily.
- Easy to troubleshoot.
- Easy to setup and modify.
- Only that node is affected which has failed, rest of the nodes can work smoothly.

Disadvantages of Star Topology

- Cost of installation is high.
- Expensive to use.
- If the hub fails then the whole network is stopped because all the nodes depend on the hub.
- Performance is based on the hub that is it depends on its capacity

Proposed Topology

Considering all the facts including advantages and disadvantages of above given topologies, the overall proposed topology for the entire network is the Triangle topology.

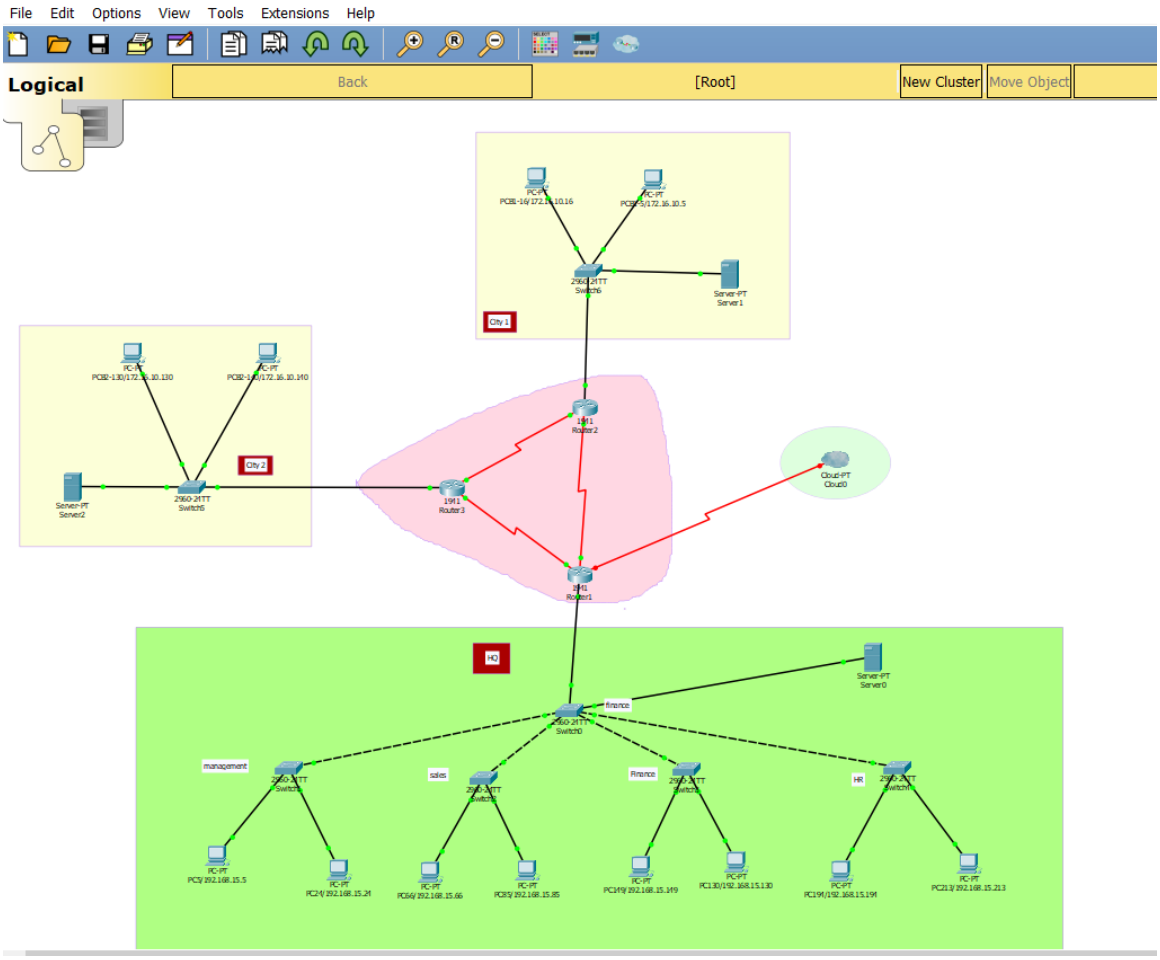
The proposed topology for the internal network is the Star topology.

All the nodes of a particular department are connected to its allocated switch. And all the switches are connected to one main switch which is connected to the router.

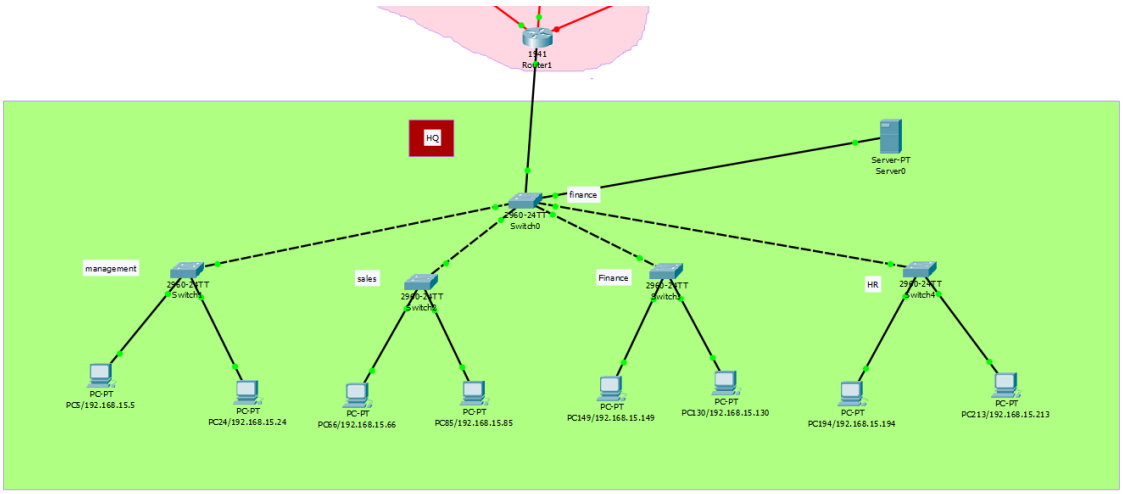
Therefore, we will be using Star Topology for this scenario with use of Switching network and Virtual Local Area Networks (VLANs).

Diagrams of Topologies of each subnet

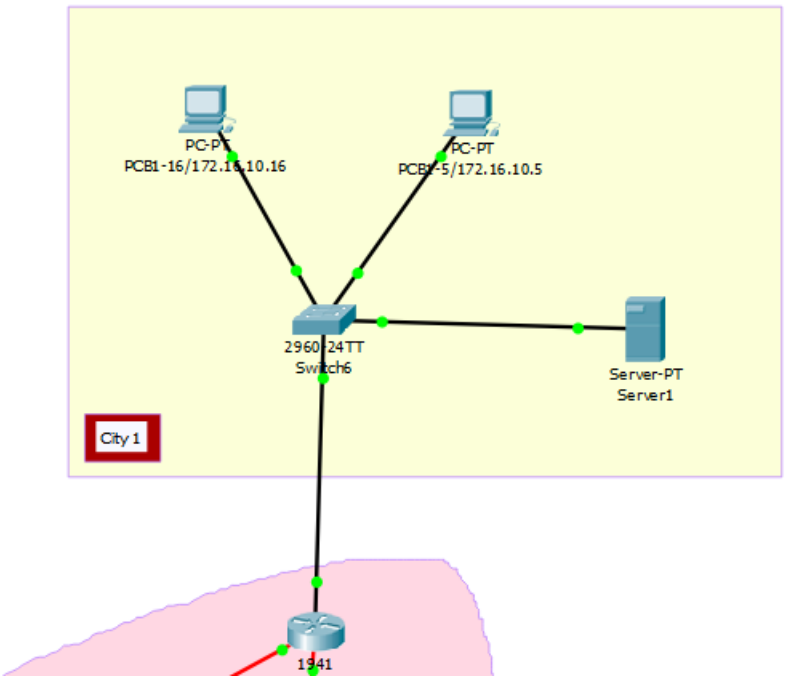
- Whole Network



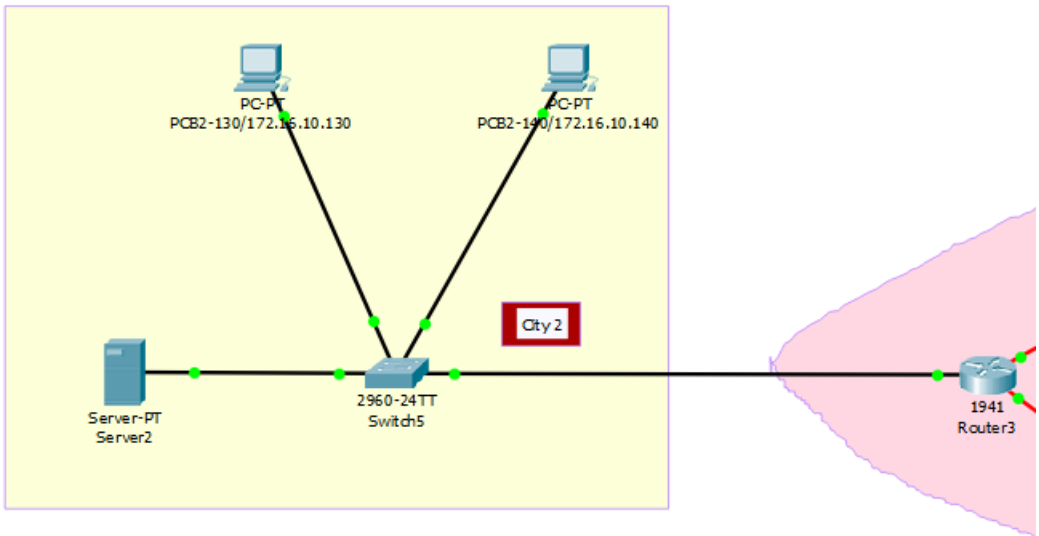
- Headquarters



- Branch 1



- Branch 2



Subnetting / IP Address allocation

What is a Subnet?

A subnet is basically the logical partition of an IP network into multiple, smaller network segments. It is typically used to divide large networks into smaller, more efficient subnetworks.

When a specific network is divided into subnets, each subnet allows its devices to communicate with each other, and routers are used to communicate between subnets. The size of a particular subnet depends on the connectivity requirements. That is, the number of hosts / nodes to be included within the subnet.

According to our scenario, the network of the company consists of 3 main parts, which are, the headquarters and its 2 other branches. The headquarters is said to have 80 nodes, which are equally divided over 4 functional departments namely, Management, Sales, Finance and HR & Admin, with 12 nodes in each branch.

Headquarters

Assumptions

- We assumed that no other departments will be added in the future, therefore, we decided to divide headquarters into 4 equal subnets, one for each department.

Decided Values

IP Address available for HQ – 192.168.15.0 /24

Therefore, we divided it into 4 equal subnets as follows.

- Subnet 1
  Network Address - 192.168.15.0 /26
  First Address - 192.168.15.1 /26
  Last Address - 192.168.15.62 /26
  Broadcast Address - 192.168.15.63 /26
  Subnet Mask - 255.255.255.192 /26
  Default-Gateway -192.168.15.1/26

- Subnet 2
  Network Address – 192.168.15.64 /26
  First Address – 192.168.15.65 /26
  Last Address - 192.168.16.126 /26
  Broadcast Address – 192.168.15.127 /26
  Subnet Mask – 255.255.255.192 /26
  Default-Gateway -192.168.15.65/26

- Subnet 3
  Network Address – 192.168.15.128 /26
  First Address – 192.168.15.129 /26
  Last Address - 192.168.16.190 /26
  Broadcast Address – 192.168.15.191 /26
  Subnet Mask - 255.255.255.192 /26
  Default-Gateway -192.168.15.129/26

- Subnet 4
  Network Address – 192.168.15.192 /26
  First Address – 192.168.15.193 /26
  Last Address - 192.168.16.254 /26
  Broadcast Address – 192. 168.15.255 /26
  Subnet Mask - 255.255.255.192
  Default-Gateway -192.168.15.193/26

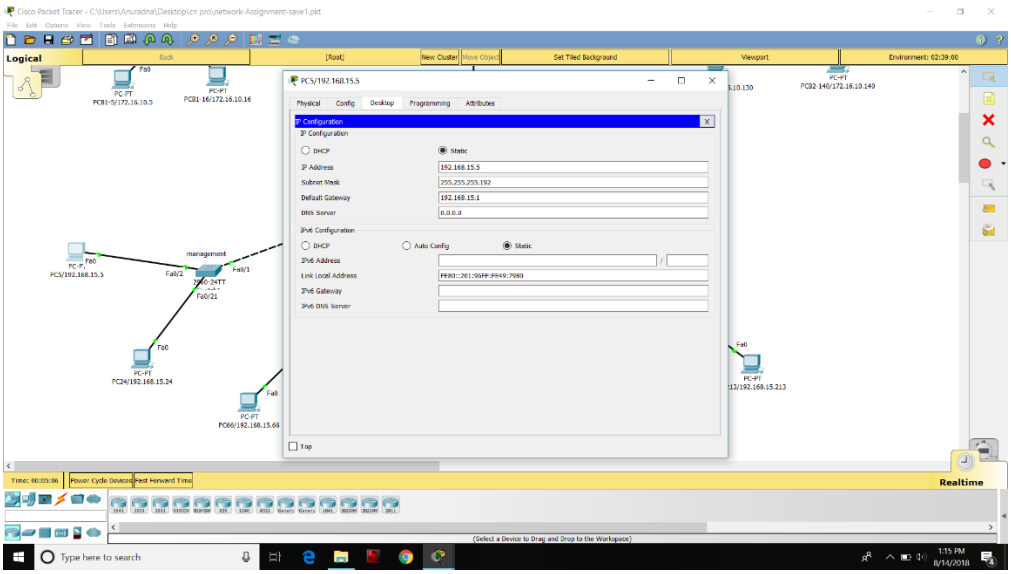
Number of hosts per each subnet = 2^6-2 = 64-2 = 62

IP Address allocation

Since the Network ID and the Broadcast ID of a particular subnet are not considered as hosts, 2 other IP addresses within the subnet's range should be selected for the nodes of each department.

Therefore,

- Management Department (Subnet 1)
  PC 1 – 192.168.15.5
  PC 2 – 192.168.15.24



A suitable IP address for each PC of each department are allocated as given above.

- Sales Department (Subnet 2)
  PC 1 – 192.168.15.66
  PC 2 – 192.168.15.85

- Finance Department (Subnet 3)
  PC 1 – 192.168.15.149
  PC 2 – 192.168.15.130

- HR & Admin Department (Subnet 4)
  PC 1- 192.168.15.194
  PC 2 – 192.168.15.213

Branches

IP Address available for branches – 172.16.10.0 /24

Since 2 branch offices are required, we divided it into 2 equal subnets as follows.

- Subnet 1
  Network Address - 172.16.10.0 /25
  First Address - 172.16.10.1 /25
  Last Address - 172.16.10.126 /25
  Broadcast Address - 172.16.10.127 /25
  Subnet Mask - 255.255.255.128 /25

- Subnet 2
  Network Address - 172.16.10.128 /25
  First Address - 172.16.10.129 /25
  Last Address - 172.16.10.254 /25
  Broadcast Address - 172.16.10.255 /25
  Subnet Mask - 255.255.255.128 /25

IP Address allocation

In the same manner as above, we allocate a particular IP address to each PC in each branch within the given IP address range (excluding network ID and broadcast ID).

- Branch 1 (Subnet 1)
  PC 1 – 172.16.10.5
  PC 2 – 172.16.10.16

- Branch 2 (Subnet 2)
  PC 1 – 172.16.10.130
  PC 2 – 172.16.10.140

Routers

IP Address assigned for routers – 10.1.1.0 /28

Since there are 3 routers in our proposed network, we divided it into 4 equal subnets in order to use 3 of them. The obtained 4 subnets are as follows.

- Router 1
  Network Address - 10.1.1.0 /30
  First Address - 10.1.1.1 /30
  Last Address - 10.1.1.2 /30
  Broadcast Address - 10.1.1.3 /30
  Subnet Mask - 255.255.255.252

- Router 2
  Network Address - 10.1.1.4 /30
  First Address - 10.1.1.5 /30
  Last Address - 10.1.1.6 /30
  Broadcast Address - 10.1.1.7 /30
  Subnet Mask - 255.255.255.252

- Router 3
  Network Address - 10.1.1.8 /30
  First Address - 10.1.1.9 /30
  Last Address - 10.1.1.10 /30
  Broadcast Address - 10.1.1.11 /30
  Subnet Mask - 255.255.255.252

- Router 4
  Network Address - 10.1.1.12 /30
  First Address - 10.1.1.13 /30
  Last Address - 10.1.1.14 /30
  Broadcast Address - 10.1.1.15 /30
  Subnet Mask - 255.255.255.252

Since we're using only 3 of the above for our 3 routers in our proposed network, one whole subnet (Router 4) is wasted.

Routing and Switching Technologies

Routing and Switching are basic functions of network communication.

Switching is basically the function of switching/ transferring data packets between devices on the same network or same LAN. Switches are used to connect multiple devices on the same network. For example, a switch can connect your computer, printer and server, creating a network of shared resources. The switch would basically serve as a controller, allowing various devices to share information and communicate with each other.

When we consider Routing, it is basically the function of routing packets between different networks or between different LANs. In simpler terms, routers are used to tie multiple networks together. The router will act as a dispatcher, choosing the best route for your information to travel in order for the user to receive it faster.

Specifically, routers and switches support,

- Sharing applications
- Speeding access to information
- Enhancing customer service
- Improving security
- Enabling remote connections

VLAN

1. Assigning VLAN id's and names

As the first step of setting up VLAN we have to assign a name and id. Here we have shown the steps of doing it.

```
Switch#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#vlan 10
Switch(config-vlan)#name hr
```

In the above example our VLAN id is 10 and VLAN name is HR(human resources).

```
Switch>en
Switch#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#vlan 20
Switch(config-vlan)#name sales
Switch(config-vlan)#ex
```

Here the id given is 20 and name is sales.

```
Switch>en
Switch#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#vlan 30
Switch(config-vlan)#name finance
Switch(config-vlan)#ex
```

Here the id is 30 and the name is finance.

```
Switch>en
Switch#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#vlan 40
Switch(config-vlan)#namemanagement
                    ^
% Invalid input detected at '^' marker.

Switch(config-vlan)#name management
```

Here the id is 40 and the name is management.

2. Assigning interfaces to VLANs

```
Switch#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#int ran fa0/2-3
Switch(config-if-range)#sw acc vlan 10
Switch(config-if-range)#ex
```

```
Switch(config)#int ran fa0/2-3
Switch(config-if-range)#sw acc vlan 20
Switch(config-if-range)#ex
```

```
Switch(config)#int ran fa0/2-3
Switch(config-if-range)#sw acc vlan 30
```

```
Switch(config)#int ran fa0/2-3
Switch(config-if-range)#sw acc vlan 40
Switch(config-if-range)#ex
```

In the above diagrams you can see that the we are informing the connected VLAN ids.

3. Assigning ip addresses to vlans

```
Switch(config)#int vlan 10
Switch(config-if)#ip add 192.168.15.20 255.255.255.192
Switch(config-if)#ex
```

Here we have assigned the IP address of 192.168.15.20 and the subnet mask of 255.255.255.192 to the VLAN 10.

Note that the IP addresses should be from the previously unassigned ip addresses of subnet.

```
Switch(config)#int vlan 20
Switch(config-if)#
%LINK-5-CHANGED: Interface Vlan20, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan20, changed
state to up

Switch(config-if)#ip add 192.168.15.90 255.255.255.192
Switch(config-if)#ex
```

To the VLAN 20 we have assignee dip address of 192.168.15.90 and the subnet mask 255.255.255.192.

```
Switch(config)#int vlan 30
Switch(config-if)#
%LINK-5-CHANGED: Interface Vlan30, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan30, changed
state to up

Switch(config-if)#ip add 192.168.15.170 255.255.255.192
Switch(config-if)#ex
```

Here we have assigned the IP address of 192.168.15.170 and the subnet mask of 255.255.255.192 to the VLAN id 30.

```
Switch(config)#int vlan 40
Switch(config-if)#
%LINK-5-CHANGED: Interface Vlan40, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan40, changed
state to up

Switch(config-if)#ip add 192.168.15.220 255.255.255.192
Switch(config-if)#ex
```

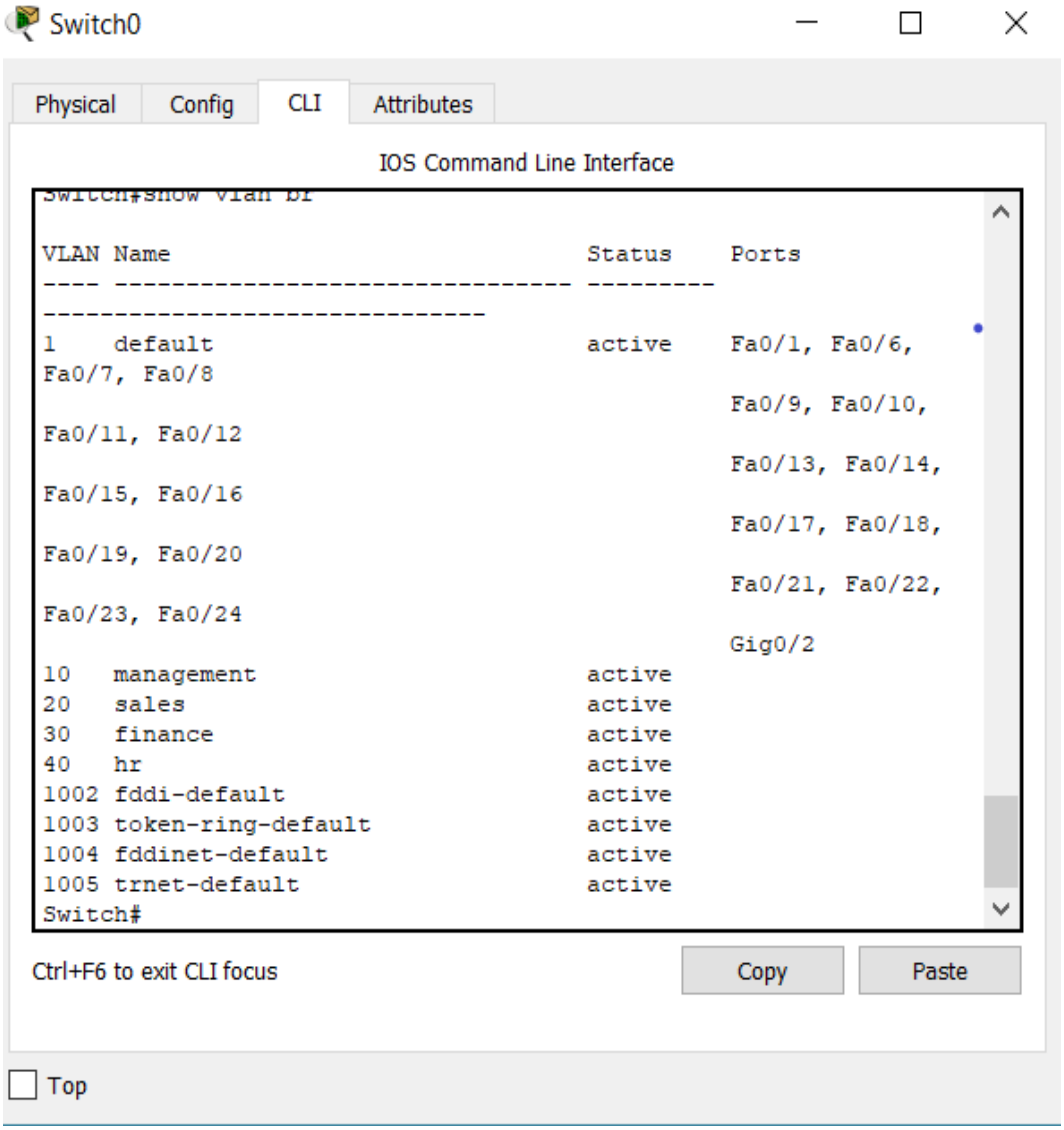
Here we have assigned the IP address of 192.168.15.220 and the subnet mask of 255.255.255.192 to the VLAN id 40.

4. Showing the created VLAN By entering the code “show vlan br” we can see all the details of the created VLAN.

Here we can see all the details about created VLANs (management, sales ,finace, hr) and also the ports they are connected to.

VLAN trunking

Virtual local area network (VLAN) trunking Protocol or VTP is a proprietary protocol from Cisco that allows networks to send network functionality through all of the switches in a domain. This technique eliminates the need for multiple configurations for VLANs throughout the system.



VTP, which is available with Cisco Catalyst products, provides efficient ways to send a VLAN through every switch. There's also the option of VLAN pruning which will avoid sending traffic through some switches. Users can make these systems pruning eligible or pruning ineligible.

One concept in VTP is that larger scale networks may need to be limited in terms of which switches will act as the VLAN servers. VTP offers various options for recovery after a crash or for efficiently serving up redundant network traffic.

In general, the idea of VLAN trunking is similar to other kinds of IT trunking. By locating resources in specific arrangements, data has to do less work to get to specific parts of a network system, or administrators need to do less work to accommodate these data transfers. The trunks between switches are part of this efficiency mechanism which allows for faster and more efficient network traffic.

```
Switch>
Switch>en
Switch#show int tr
Port        Mode         Encapsulation  Status        Native vlar
Fa0/1       on           802.1q         trunking      1

Port        Vlans allowed on trunk
Fa0/1       1-1005

Port        Vlans allowed and active in management domain
Fa0/1       1,40

Port        Vlans in spanning tree forwarding state and not
pruned
```

Inter Vlan Routing

```
interface GigabitEthernet0/0.10
 encapsulation dot1Q 10
 ip address 192.168.15.1 255.255.255.192
!
interface GigabitEthernet0/0.20
 encapsulation dot1Q 20
 ip address 192.168.15.65 255.255.255.192
!
interface GigabitEthernet0/0.30
 encapsulation dot1Q 30
 ip address 192.168.15.129 255.255.255.192
!
interface GigabitEthernet0/0.40
 encapsulation dot1Q 40
 ip address 192.168.15.193 255.255.255.192
!
interface GigabitEthernet0/1
 no ip address
 duplex auto
 speed auto
 shutdown
!
```

This defines the routing among the devices that belongs to vlans. In order to communicate between different vlans we will have to create sub interfaces of fast ethernet interface of router

OSPF as the routing protocol

Open Shortest Path First (OSPF) is a link state routing protocol (LSRP) that uses the Shortest Path First (SPF) network communication algorithm (Dijkstra's algorithm) to calculate the shortest connection path between known devices.

Routing protocols like OSPF calculate the shortest route to a destination through the network based on an algorithm. The first routing protocol that was widely implemented, the Routing Information Protocol (RIP), calculated the shortest route based on hops, that is the number of routers that an IP packet had to traverse to reach the destination host. RIP successfully implemented dynamic routing, where routing tables change if the network topology changes.

OSPF was developed so that the shortest path through a network was calculated based on the cost of the route, taking into account bandwidth, delay and load. Therefore, OSPF undertakes route cost calculation on the basis of link-cost parameters, which can be weighted by the administrator.

As a link state routing protocol, OSPF maintains link state databases, which are really network topology maps, on every router on which it is implemented. The state of a given

route in the network is the cost, and OSPF algorithm allows every router to calculate the cost of the routes to any given reachable destination. Unless the administrator has made a configuration, the link cost of a path connected to a router is determined by the bit rate (1 Gbit/s, 10 Gbit/s, etc) of the interface.

A router interface with OSPF will then advertise its link cost to neighboring routers through multicast, known as the “hello procedure”. All routers with OSPF implementation keep sending hello packets, and thus changes in the cost of their links become known to neighboring routers. The information about the cost of a link, that is the speed of a point to point connection between two routers, is then cascaded through the network because OSPF routers advertise the information they receive from one neighboring router to all other neighboring routers. This process of flooding link state information through the network is known as synchronization. Based on this information, all routers with OSPF implementation continuously update their link state databases with information about the network topology and adjust their routing tables.

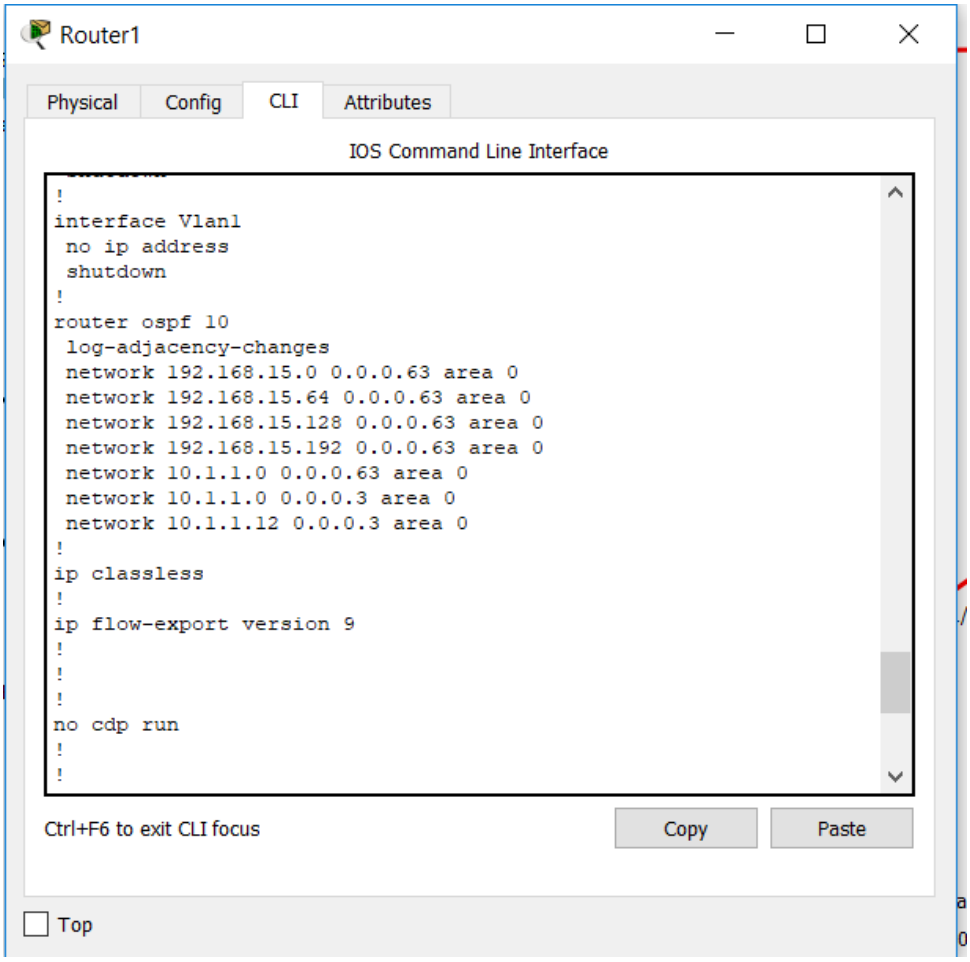
An OSPF network can be structured, or subdivided, into routing areas to simplify administration and optimize traffic and resource utilization.

Areas are identified by 32-bit numbers, expressed either simply in decimal, or often in the same dot-decimal notation used for IPv4 addresses. By convention, area 0 (zero), or 0.0.0.0, represents the core or backbone area of an OSPF network. While the identifications of other areas may be chosen at will, administrators often select the IP address of a main router in an area as the area identifier.

Each additional area must have a connection to the OSPF backbone area. Such connections are maintained by an interconnecting router, known as an area border router (ABR). An ABR maintains separate link-state databases for each area it serves and maintains summarized routes for all areas in the network.

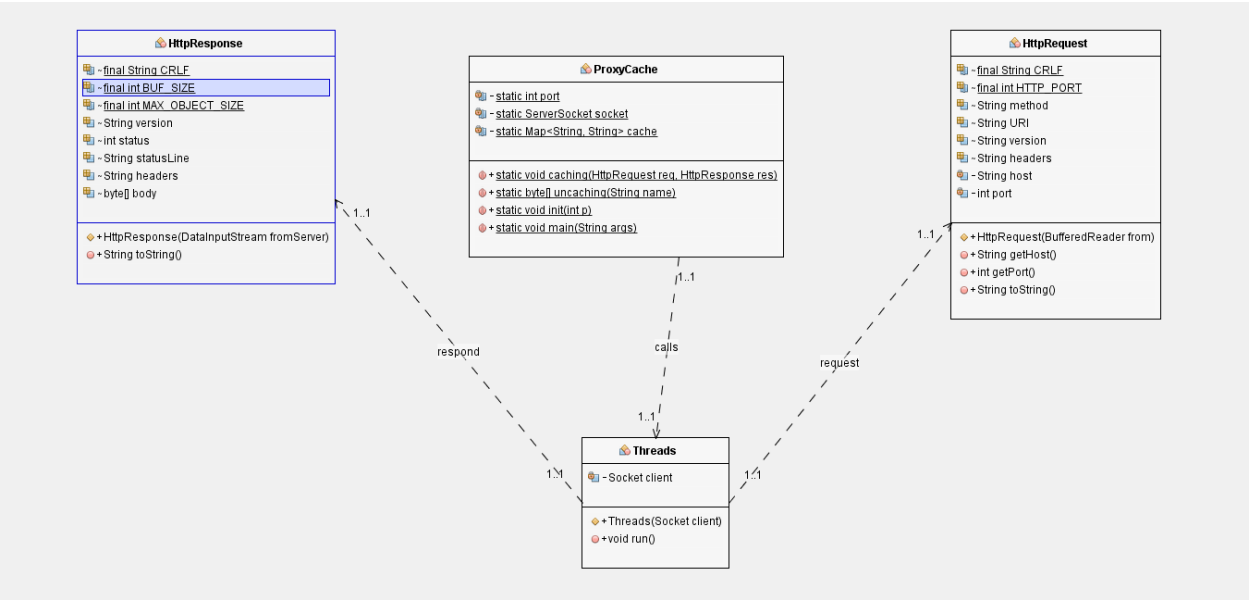
OSPF detects changes in the topology, such as link failures, and converges on a new loop-free routing structure within seconds.

OSPF has become a popular dynamic routing protocol. Other commonly used dynamic routing protocols are the RIP and the Border Gateway
Protocol (BGP). Today routers support a least one interior gateway protocol to advertise their routing tables within a local area network. Frequently implemented interior gateway protocols besides OSPF are RIP, IS-IS, and the proprietary Interior Gateway Routing Protocol (IGRP) by Cisco.


Router1
Physical
Config
CLI
Attributes
IOS Command Line Interface
!
interface Vlan1
 no ip address
 shutdown
!
router ospf 10
 log-adjacency-changes
 network 192.168.15.0 0.0.0.63 area 0
 network 192.168.15.64 0.0.0.63 area 0
 network 192.168.15.128 0.0.0.63 area 0
 network 192.168.15.192 0.0.0.63 area 0
 network 10.1.1.0 0.0.0.63 area 0
 network 10.1.1.0 0.0.0.3 area 0
 network 10.1.1.12 0.0.0.3 area 0
!
ip classless
!
ip flow-export version 9
!
!
!
no cdp run
!
!
Ctrl+F6 to exit CLI focus
Copy
Paste
Top

Proxy

Class Diagram



Java Code Implementation

change.log | README.md | Threads.java | HttpRequest.java | HttpResponse.java

```java
import java.io.*;
import java.net.*;
import java.util.*;

public class HttpRequest {
    /** Help variables */
    final static String CRLF = "\r\n";
    final static int HTTP_PORT = 80;
    /** Store the request parameters */
    String method;
    String URI;
    String version;
    String headers = "";
    /** Server and port */
    private String host;
    private int port;

    /** Create HttpRequest by reading it from the client socket */
    public HttpRequest(BufferedReader from) {
    String firstLine = "";
    try {
        firstLine = from.readLine();
    } catch (IOException e) {
        System.out.println("Error reading request line: " + e);
    }

    String[] tmp = firstLine.split(" ");

    /* Formato GET = GET http://www.google.pt HTTP/1.0 */
    method = tmp[0]; /* method GET */
    URI =  tmp[1]; /* URI */
    version =  tmp[2]; /* HTTP version */

    System.out.println("URI is: " + URI);
```

Start Page | Proxy.java | HttpRequest.java | HttpResponse.java | ProxyCache.java | Proxy 2018.08.14 20-41-26.cdg

Source | History

```java
/*
 * To change this license header, choose License Headers in Project Properties
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package proxy;

/**
 *
 * @author ASUS
 */
public class Proxy {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
    }

}
```

```java
    try {
        String line = from.readLine();
        while (line.length() != 0) {
        headers += line + CRLF;
        /* We need to find host header to know which server to
         * contact in case the request URI is not complete. */
        if (line.startsWith("Host:")) {
            tmp = line.split(" ");
            if (tmp[1].indexOf(':') > 0) {
            String[] tmp2 = tmp[1].split(":");
            host = tmp2[0];
            port = Integer.parseInt(tmp2[1]);
            } else {
            host = tmp[1];
            port = HTTP_PORT;
            }
        }
        line = from.readLine();
        }
    } catch (IOException e) {
        System.out.println("Error reading from socket: " + e);
        return;
    }
    System.out.println("Host to contact is: " + host + " at port " + port);
    }

    /** Return host for which this request is intended */
    public String getHost() {
    return host;
    }

    /** Return port for server */
    public int getPort() {
    return port;
    }
```

```java
    /**
     * Convert request into a string for easy re-sending.
     */
    public String toString() {
    String req = "";

    req = method + " " + URI + " " + version + CRLF;
    req += headers;
    /* This proxy does not support persistent connections */
    req += "Connection: close" + CRLF;
    req += CRLF;

    return req;
    }
}
```

change.log | README.md | Threads.java | HttpRequest.java | HttpResponse.java

```
13
14  public class HttpResponse {
15      final static String CRLF = "\r\n";
16      /** How big is the buffer used for reading the object */
17      final static int BUF_SIZE = 8192;
18      /** Maximum size of objects that this proxy can handle. For the
19       * moment set to 100 KB. You can adjust this as needed. */
20      final static int MAX_OBJECT_SIZE = 100000;
21      /** Reply status and headers */
22      String version;
23      int status;
24      String statusLine = "";
25      String headers = "";
26      /* Body of reply */
27      byte[] body = new byte[MAX_OBJECT_SIZE];
28
29      /** Read response from server. */
30      public HttpResponse(DataInputStream fromServer) {
31      /* Length of the object */
32      int length = -1;
33      boolean gotStatusLine = false;
34
35      /* First read status line and response headers */
36      try {
37          String line =fromServer.readLine(); /* Ler inputstream do servidor */
38          while (line.length() != 0) {
39          if (!gotStatusLine) {
40              statusLine = line;
41              gotStatusLine = true;
42          } else {
43              headers += line + CRLF;
44          }
45
46          /* Get length of content as indicated by
47           * Content-Length header. Unfortunately this is not
48           * present in every response. Some servers return the
49           * header "Content-Length", others return
50           * "Content-length". You need to check for both
51           * here. */
52          if (line.startsWith("Content-Length:") ||
53              line.startsWith("Content-length:")) {
54              String[] tmp = line.split(" ");
55              length = Integer.parseInt(tmp[1]);
56          }
57          line = fromServer.readLine();
58          }
59      } catch (IOException e) {
60          System.out.println("Error reading headers from server: " + e);
61      }
62
63      try {
64          int bytesRead = 0;
65          byte buf[] = new byte[BUF_SIZE];
66          boolean loop = false;
67
68          /* If we didn't get Content-Length header, just loop until
69           * the connection is closed. */
70          if (length == -1) {
71          loop = true;
72          }
73
74          /* Read the body in chunks of BUF_SIZE and copy the chunk
75           * into body. Usually replies come back in smaller chunks
76           * than BUF_SIZE. The while-loop ends when either we have
77           * read Content-Length bytes or when the connection is
78           * closed (when there is no Connection-Length in the
79           * response. */
```

```
        while (bytesRead < length || loop) {
        /* Read it in as binary data */
        int res = fromServer.read(buf, 0, BUFF_SIZE); /* Read binary date until BUFF_SIZE to buf (array in bytes) */
        if (res == -1) {
            break;
        }
        /* Copy the bytes into body. Make sure we don't exceed
         * the maximum object size. */
        for (int i = 0;
             i < res && (i + bytesRead) < MAX_OBJECT_SIZE;
             i++) {
            body[bytesRead + i] = buf[i]; /* copy bytes read to buf to body */
        }
        bytesRead += res;
        }
    } catch (IOException e) {
        System.out.println("Error reading response body: " + e);
        return;
    }


    }

    /**
     * Convert response into a string for easy re-sending. Only
     * converts the response headers, body is not converted to a
     * string.
     */
    public String toString() {
    String res = "";

    res = statusLine + CRLF;
    res += headers;
    res += CRLF;

    return res;
    }
}
```

change.log | README.md | Threads.java | HttpRequest.java | HttpResponse.java

```
import java.net.*;
import java.io.*;
import java.util.*;
import java.lang.*;

public class Threads implements Runnable{

    private final Socket client;

    public Threads(Socket client) {
        this.client = client;
    }


    public void run() {
    Socket server = null;
    HttpRequest request = null;
    HttpResponse response = null;

    /* Process request. If there are any exceptions, then simply
     * return and end this request. This unfortunately means the
     * client will hang for a while, until it timeouts. */

    /* Read request */
    try {
        BufferedReader fromClient = new BufferedReader(new InputStreamReader(client.getInputStream()));
        request = new HttpRequest(fromClient);
    } catch (IOException e) {
        System.out.println("Error reading request from client: " + e);
        return;
    }
    /* Send request to server */
    try {
        /* Open socket and write request to socket */
        server = new Socket(request.getHost(), request.getPort()); /* Create socket */
        DataOutputStream toServer = new DataOutputStream(server.getOutputStream()); /* Create outputstream for server on socket */
        toServer.writeBytes(request.toString()); /* Create outputstream for the serviceScribe request to the outputstreamidor on the socket */
    } catch (UnknownHostException e) {
        System.out.println("Unknown host: " + request.getHost());
        System.out.println(e);
        return;
    } catch (IOException e) {
        return;
    }
```

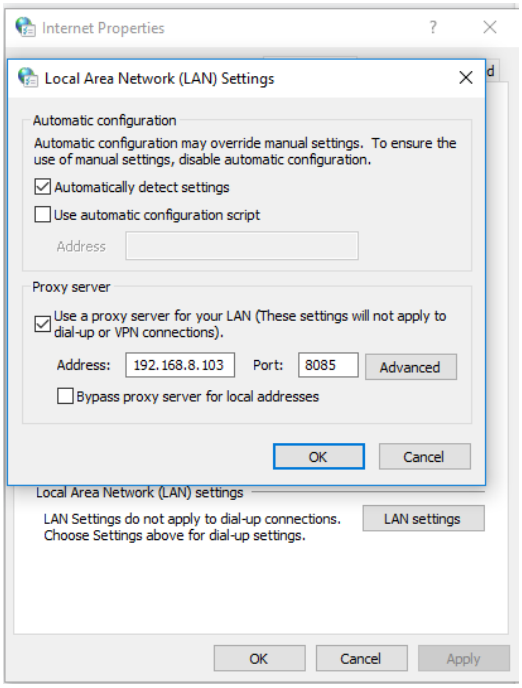
```
45      /* Read response and forward it to client */
46      try {
47          byte[] cache = ProxyCache.uncaching(request.URI);
48          if(cache.length==0) {
49              DataInputStream fromServer = new DataInputStream(server.getInputStream()); /* Create server inputstream */
50              response = new HttpResponse(fromServer); /* Create object with server response */
51              DataOutputStream toClient = new DataOutputStream(client.getOutputStream());
52
53
54
55              toClient.writeBytes(response.toString()); /* Write headers */
56              toClient.write(response.body); /* Write body */
57              /* Write response to client. First headers, then body */
58
59              ProxyCache.caching(request, response); /* Save to cache */
60
61              client.close();
62              server.close();
63              /* Insert object into the cache */
64              /* Fill in (optional exercise only) */
65          }
66          else{
67              DataOutputStream toClient = new DataOutputStream(client.getOutputStream());
68              toClient.write(cache);
69              client.close();
70              server.close();
71          }
72
73      } catch (IOException e) {
74          System.out.println("Error writing response to client: " + e);
75      }
76      }
77  }
78
```

Creating Proxy server

Step 1- Obtain the ip address of the interface which is currently connected to the internet using “ipconfif” command in cmd.

Step 2- Then go to internet properties and select LAN settings, in there enable the option for the proxy server by applying the tick, Type the ipv4 address of the currently connected interface.

Step 3-select a port Number




```
Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . : local
   Link-local IPv6 Address . . . . . : fe80::f074:fe2f:b3a8:a535%16
   IPv4 Address. . . . . . . . . . . : 192.168.8.103
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.8.1
```

CACHE

Caching the request and responses to a text file.

Step 1-Compile the file.

Step 2-Run the application with port number previously defined in the LAN settings.

```
PS C:\Users\SHASHIN\Desktop\ProxyServer-master> javac -deprecation *.java
HttpResponse.java:37: warning: [deprecation] readLine() in DataInputStream has been deprecated
            String line =fromServer.readLine(); /* Ler inputstream do servidor */
                                    ^
HttpResponse.java:57: warning: [deprecation] readLine() in DataInputStream has been deprecated
                line = fromServer.readLine();
                                  ^
2 warnings
PS C:\Users\SHASHIN\Desktop\ProxyServer-master> java ProxyCache 8085
```

Step 3- Then browse any website, Then the caching starts

```
URI is: http://nsbm.lk/
Host to contact is: nsbm.lk at port 80
Caching: No hit on http://nsbm.lk/
URI is: http://nsbm.lk/css/bootstrap.min.css
Host to contact is: nsbm.lk at port 80
Caching: No hit on http://nsbm.lk/css/bootstrap.min.css
URI is: fonts.googleapis.com:443
URI is: http://nsbm.lk/css/style.css
URI is: http://www.bw2017.lk/new/stats.php?page=2017
URI is: http://nsbm.lk/css/flexslider.css
URI is: http://nsbm.lk/css/responsive.css
Host to contact is: fonts.googleapis.com at port 443
URI is: http://nsbm.lk/css/nsbm.css
URI is: http://nsbm.lk/css/simple-sidebar.css
Host to contact is: nsbm.lk at port 80
Host to contact is: nsbm.lk at port 80
Host to contact is: nsbm.lk at port 80
Host to contact is: www.bw2017.lk at port 80
Host to contact is: nsbm.lk at port 80
Host to contact is: nsbm.lk at port 80
URI is: http://nsbm.lk/css/owl.carousel.css
Host to contact is: nsbm.lk at port 80
Caching: No hit on http://nsbm.lk/css/nsbm.css
Caching: No hit on http://nsbm.lk/css/responsive.css
Caching: No hit on http://nsbm.lk/css/style.css
Caching: No hit on http://nsbm.lk/css/flexslider.css
Caching: No hit on http://nsbm.lk/css/owl.carousel.css
Caching: No hit on http://nsbm.lk/css/simple-sidebar.css
Caching: No hit on fonts.googleapis.com:443
URI is: http://nsbm.lk/css/owl.theme.css
Host to contact is: nsbm.lk at port 80
Caching: No hit on http://nsbm.lk/css/owl.theme.css
Exception in thread "Thread-19" java.lang.NullPointerException
        at HttpResponse.<init>(HttpResponse.java:38)
        at Threads.run(Threads.java:50)
        at java.lang.Thread.run(Thread.java:748)
URI is: clients4.google.com:443
Host to contact is: clients4.google.com at port 443
URI is: http://nsbm.lk/css/bootstrap.vertical-tabs.css
URI is: http://nsbm.lk/font-awesome-4.6.3/css/font-awesome.css
Host to contact is: nsbm.lk at port 80
Host to contact is: nsbm.lk at port 80
Caching: No hit on clients4.google.com:443
Caching: No hit on http://nsbm.lk/css/bootstrap.vertical-tabs.css
Caching: No hit on http://nsbm.lk/font-awesome-4.6.3/css/font-awesome.css
URI is: http://www.nsbm.lk/images/nsbm-green-uni-logo.png
Host to contact is: www.nsbm.lk at port 80
URI is: http://www.nsbm.lk/images/facility-icon.png
Host to contact is: www.nsbm.lk at port 80
Exception in thread "Thread-23" java.lang.NullPointerException
        at HttpResponse.<init>(HttpResponse.java:38)
        at Threads.run(Threads.java:50)
        at java.lang.Thread.run(Thread.java:748)
Caching: No hit on http://www.nsbm.lk/images/nsbm-green-uni-logo.png
Caching: No hit on http://www.nsbm.lk/images/facility-icon.png
Caching from: http://nsbm.lk/css/flexslider.css para C:\Users\SHASHIN\Desktop\ProxyServer-master\cachedSites.txt
Caching from: http://nsbm.lk/css/nsbm.css para C:\Users\SHASHIN\Desktop\ProxyServer-master\cachedSites.txt
Caching from: http://nsbm.lk/css/bootstrap.min.css para C:\Users\SHASHIN\Desktop\ProxyServer-master\cachedSites.txt
Caching from: http://nsbm.lk/css/style.css para C:\Users\SHASHIN\Desktop\ProxyServer-master\cachedSites.txt
Caching from: http://nsbm.lk/css/responsive.css para C:\Users\SHASHIN\Desktop\ProxyServer-master\cachedSites.txt
Caching from: http://nsbm.lk/font-awesome-4.6.3/css/font-awesome.css para C:\Users\SHASHIN\Desktop\ProxyServer-master\cachedSites.txt
Caching from: http://nsbm.lk/css/owl.theme.css para C:\Users\SHASHIN\Desktop\ProxyServer-master\cachedSites.txt
Caching from: http://nsbm.lk/ para C:\Users\SHASHIN\Desktop\ProxyServer-master\cachedSites.txt
Caching from: http://nsbm.lk/css/owl.carousel.css para C:\Users\SHASHIN\Desktop\ProxyServer-master\cachedSites.txt
Caching from: http://nsbm.lk/css/bootstrap.vertical-tabs.css para C:\Users\SHASHIN\Desktop\ProxyServer-master\cachedSites.txt
Caching from: http://nsbm.lk/css/simple-sidebar.css para C:\Users\SHASHIN\Desktop\ProxyServer-master\cachedSites.txt
```

```
URI is: http://nsbm.lk/js/jquery.js
Host to contact is: nsbm.lk at port 80
URI is: http://nsbm.lk/js/bootstrap.js
Host to contact is: nsbm.lk at port 80
Caching from: http://nsbm.lk/images/nsbm-green-uni-logo.png para C:\Users\SHASHIN\Desktop\ProxyServer-master\cachedSites.txt
Caching from: http://nsbm.lk/images/facility-icon.png para C:\Users\SHASHIN\Desktop\ProxyServer-master\cachedSites.txt
Caching: No hit on http://nsbm.lk/js/bootstrap.js
Caching: No hit on http://nsbm.lk/js/jquery.js
URI is: http://nsbm.lk/js/cust.js
Host to contact is: nsbm.lk at port 80
URI is: http://nsbm.lk/js/jquery.flexslider.js
Host to contact is: nsbm.lk at port 80
Caching: No hit on http://nsbm.lk/js/cust.js
Caching: No hit on http://nsbm.lk/js/jquery.flexslider.js
Caching from: http://nsbm.lk/js/bootstrap.js para C:\Users\SHASHIN\Desktop\ProxyServer-master\cachedSites.txt
Caching from: http://nsbm.lk/js/jquery.js para C:\Users\SHASHIN\Desktop\ProxyServer-master\cachedSites.txt
URI is: http://nsbm.lk/js/owl.carousel.min.js
Host to contact is: nsbm.lk at port 80
Caching from: http://nsbm.lk/js/cust.js para C:\Users\SHASHIN\Desktop\ProxyServer-master\cachedSites.txt
Caching: No hit on http://nsbm.lk/js/owl.carousel.min.js
Caching from: http://nsbm.lk/js/jquery.flexslider.js para C:\Users\SHASHIN\Desktop\ProxyServer-master\cachedSites.txt
URI is: http://nsbm.lk/js/marquee.js
Host to contact is: nsbm.lk at port 80
Caching: No hit on http://nsbm.lk/js/marquee.js
URI is: http://www.nsbm.lk/images/facbook.png
Host to contact is: www.nsbm.lk at port 80
Caching: No hit on http://www.nsbm.lk/images/facbook.png
Caching from: http://nsbm.lk/js/owl.carousel.min.js para C:\Users\SHASHIN\Desktop\ProxyServer-master\cachedSites.txt
URI is: http://www.nsbm.lk/images/twitter.png
Host to contact is: www.nsbm.lk at port 80
URI is: http://nsbm.lk/images/facbook.png
Host to contact is: nsbm.lk at port 80
Caching from: http://nsbm.lk/js/marquee.js para C:\Users\SHASHIN\Desktop\ProxyServer-master\cachedSites.txt
Caching: No hit on http://www.nsbm.lk/images/twitter.png
Caching: No hit on http://nsbm.lk/images/facbook.png
URI is: http://nsbm.lk/images/twitter.png
Host to contact is: nsbm.lk at port 80
Caching from: http://www.nsbm.lk/images/facbook.png para C:\Users\SHASHIN\Desktop\ProxyServer-master\cachedSites.txt
Caching: No hit on http://nsbm.lk/images/twitter.png
Caching from: http://www.nsbm.lk/images/twitter.png para C:\Users\SHASHIN\Desktop\ProxyServer-master\cachedSites.txt
URI is: http://www.nsbm.lk/images/youtube.png
Host to contact is: www.nsbm.lk at port 80
Caching from: http://nsbm.lk/images/facbook.png para C:\Users\SHASHIN\Desktop\ProxyServer-master\cachedSites.txt
Caching: No hit on http://www.nsbm.lk/images/youtube.png
Caching from: http://nsbm.lk/images/twitter.png para C:\Users\SHASHIN\Desktop\ProxyServer-master\cachedSites.txt
Caching from: http://www.nsbm.lk/images/youtube.png para C:\Users\SHASHIN\Desktop\ProxyServer-master\cachedSites.txt
URI is: http://nsbm.lk/images/slide/edit.jpg
Host to contact is: nsbm.lk at port 80
URI is: http://nsbm.lk/images/youtube.png
Host to contact is: nsbm.lk at port 80
Caching: No hit on http://nsbm.lk/images/slide/edit.jpg
Caching: No hit on http://nsbm.lk/images/youtube.png
Caching from: http://nsbm.lk/images/youtube.png para C:\Users\SHASHIN\Desktop\ProxyServer-master\cachedSites.txt
URI is: http://nsbm.lk/images/slide/page%20(1).jpg
Host to contact is: nsbm.lk at port 80
Caching: No hit on http://nsbm.lk/images/slide/page%20(1).jpg
Unknown host: www.bw2017.lk
java.net.UnknownHostException: www.bw2017.lk
URI is: www.cisco.com:443
Host to contact is: www.cisco.com at port 80
Caching from: http://nsbm.lk/images/slide/edit.jpg para C:\Users\SHASHIN\Desktop\ProxyServer-master\cachedSites.txt
URI is: http://nsbm.lk/images/slide/page.jpg
Host to contact is: nsbm.lk at port 80
Caching: No hit on www.cisco.com:443
URI is: beacons.gcp.gvt2.com:443
Host to contact is: beacons.gcp.gvt2.com at port 443
```

Cached text file

```
aced 0005 7372 0013 6a61 7661 2e75 7469
6c2e 4861 7368 7461 626c 6513 bb0f 2521
4ae4 b803 0002 4600 0a6c 6f61 6446 6163
746f 7249 0009 7468 7265 7368 6f6c 6478
703f 4000 0000 0000 4777 0800 0000 5f00
0000 3c74 0021 6874 7470 3a2f 2f6e 7362
6d2e 6c6b 2f69 6d61 6765 732f 796f 7574
7562 652e 706e 6774 003b 433a 5c55 7365
7273 5c53 4841 5348 494e 5c44 6573 6b74
6f70 5c50 726f 7879 5365 7276 6572 2d6d
6173 7465 725c 6361 6368 6564 5369 7465
732e 7478 7474 001c 6874 7470 3a2f 2f6e
7362 6d2e 6c6b 2f63 7373 2f73 7479 6c65
2e63 7373 7400 3b43 3a5c 5573 6572 735c
5348 4153 4849 4e5c 4465 736b 746f 705c
5072 6f78 7953 6572 7665 722d 6d61 7374
6572 5c63 6163 6865 6453 6974 6573 2e74
7874 7400 2568 7474 703a 2f2f 6e73 626d
2e6c 6b2f 696d 6167 6573 2f62 7573 696e
6573 2d68 6d65 2e6a 7067 7400 3b43 3a5c
5573 6572 735c 5348 4153 4849 4e5c 4465
736b 746f 705c 5072 6f78 7953 6572 7665
722d 6d61 7374 6572 5c63 6163 6865 6453
6974 6573 2e74 7874 7400 2768 7474 703a
2f2f 6e73 626d 2e6c 6b2f 696d 6167 6573
2f70 6c79 6d6f 7574 685f 6c6f 676f 2e6a
7067 7400 3b43 3a5c 5573 6572 735c 5348
4153 4849 4e5c 4465 736b 746f 705c 5072
6f78 7953 6572 7665 722d 6d61 7374 6572
5c63 6163 6865 6453 6974 6573 2e74 7874
7400 2d68 7474 703a 2f2f 6e73 626d 2e6c
6b2f 696d 6167 6573 2f6e 7362 6d2d 6772
6565 6e2d 756e 692d 6c6f 676f 2e70 6e67
7400 3b43 3a5c 5573 6572 735c 5348 4153
4849 4e5c 4465 736b 746f 705c 5072 6f78
7953 6572 7665 722d 6d61 7374 6572 5c63
6163 6865 6453 6974 6573 2e74 7874 7400
2d68 7474 703a 2f2f 6e73 626d 2e6c 6b2f
696d 6167 6573 2f65 6d62 6c65 6d5f 6f66
5f73 7269 5f6c 616e 6b61 2e6a 7067 7400
3b43 3a5c 5573 6572 735c 5348 4153 4849
4e5c 4465 736b 746f 705c 5072 6f78 7953
6572 7665 722d 6d61 7374 6572 5c63 6163
6865 6453 6974 6573 2e74 7874 7400 3668
7474 703a 2f2f 6e73 626d 2e6c 6b2f 666f
6e74 2d61 7765 736f 6d65 2d34 2e36 2e33
2f63 7373 2f66 6f6e 742d 6177 6573 6f6d
652e 6373 7374 003b 433a 5c55 7365 7273
5c53 4841 5348 494e 5c44 6573 6b74 6f70
5c50 726f 7879 5365 7276 6572 2d6d 6173
7465 725c 6361 6368 6564 5369 7465 732e
7478 7474 0028 6874 7470 3a2f 2f6e 7362
6d2e 6c6b 2f69 6d61 6765 732f 7465 6368
6e6f 6c6f 6779 2d68 6d65 2e6a 7067 7400
```

Group Members

- BSC-UCD-CSC-16.2-005 O. Nethmi S. Fonseka(10009222)
- BSC-UCD-CSC-16.2-001 S. R. Dharmage (10009223)
- BSC-PLY-COM-16.2-337 V. A. R. Senanayake
- BSC-PLY-COM-16.2-313 Wijewardhane Shashin P(10009118)
- BSC-PLY-COM-16.2-157 Dhananja Y.S. Lekamge(10008972)