

1102216

by Sean Parry

Submission date: 29-May-2022 09:13PM (UTC+0100)

Submission ID: 181965262

File name: 3005_Sean_Parry_1102216_169044_1989175340.pdf (3.44M)

Word count: 7288

Character count: 44291

Course: HND Electrical and Electronic Engineering

Module: SGEE5008 Networking Systems for Engineers

Portfolio: Networking Systems for Engineers

Name: Sean Parry

University number: 1102216

Date Submitted: 29/05/2022

Table of Contents

Element 1	7
1. Introduction.....	7
2. Research	8
2.1 OSI 7-layer and 4-layer models with reference to the network design	8
2.2 Encapsulation	9
2.3 Broadcasting	10
2.4 Port Allocation.....	11
3. Design requirements	12
3.1 Network Address Configuration	12
3.2 Static Routing.....	12
3.3 RIP Routing.....	12
4. Implementation	13
4.1 DHCP Routing	13
5. Test and analysis.....	13
6. Evaluation.....	14
Element 2	15
1. Introduction.....	15
2. System configuration requirements	16
2.1 CIDR/28 Design Requirements.....	16
3. Implementation of CIDR/28 RIP	21
4. Testing and analysis of CIDR/28.....	21
2.2 CIDR/27 and CIDR/30 Networking	22
3.1 Implementation for CIDR/27 and CIDR/30 RIP	24
4.1 Test and Analysis for CIDR/27 and CIDR/30.....	24
2.3 Design Requirements for VLAN	24
2.3.1 VLAN London Router/Switch for CIDR/27 sub network.....	24
3.2 Implementation of VLAN	25
4.2 VLAN Testing.....	25
2.4 Access Control Lists (ACL).....	25
2.4.1 Access Control List Test	26
5. Evaluation.....	26
Element 3	27
1. Introduction.....	27
2. System configuration requirements	28
3. Implementation	34

4. Testing and analysis.....	34
5. Conclusion and recommendations.....	34
Appendix.....	35
References.....	91

[Detailed table of contents](#)

Table of figures:

Figure 1 - 7 layer OSI model.....	35
Figure 2 - The TCP/IP protocol for internetwork communications	36
Figure 3 – Comparison of the OSI 7 layer model and the TCP/IP layers.....	37
Figure 4 – Encapsulation.....	38
Figure 5 – Broadcasting	38
Figure 6 - LAN Network.....	39
Figure 7 - IP address of router 0.....	40
Figure 8 - IP address of router 1.....	40
Figure 9 - IP address of router 2.....	41
Figure 10 - Network Address of router 0.....	42
Figure 11 - Network Address of router 1.....	43
Figure 12 - Network Address of router 2.....	44
Figure 13 - Successful communications of static routing	45
Figure 14 – RIP routing address for router 0.....	46
Figure 15 - RIP routing address for router 1	46
Figure 16 - RIP routing address for router 2	46
Figure 17 - Routing table of router 0.....	47
Figure 18 - Routing table of router 1.....	47
Figure 19 - Routing table of router 2.....	47
Figure 20 - RIP IP route of router 0	48
Figure 21 - RIP IP route of router 1	49
Figure 22 - RIP IP route of router 2	50
Figure 23 - RIP routing final results	51
Figure 24 - DHCP routing of router 0.....	52
Figure 25 - PC 0 configuration.....	53
Figure 26 - PC 1 configuration.....	54
Figure 27 - PC 2 configuration.....	55
Figure 28 - PC1 Helper address	56
Figure 29 - PC2 Helper address	57
Figure 30 - DHCP Helper Address IP PC1 Successful.....	58
Figure 31 - DHCP Helper Address IP PC2 Successful.....	59
Figure 32 - DHCP Routing excluded addresses	60
Figure 33 - DHCP Routing successful with excluded addresses of PC0	61
Figure 34 - DHCP Routing successful with excluded addresses of PC1	62
Figure 35 - DHCP Routing successful with excluded addresses of PC2	63
Figure 36 - All successful router configurations	64
Figure 37 - CIDR/28 Network Topology.....	65
Figure 38 - Routing table for Paris.....	65
Figure 39 - Routing table for Rome	66
Figure 40 - Routing table for London.....	66
Figure 41 - CIDR/28 Testing Evidence.....	67
Figure 42 - CIDR/27 and CIDR/30 RIP table for London.....	68
Figure 43 - CIDR/27 and CIDR/30 RIP table for Paris.....	68
Figure 44 - CIDR/27 and CIDR/30 RIP table for Rome	69
Figure 45 - CIDR/27 and CIDR/30 Testing Evidence	69
Figure 46 - VLAN London Model	70

Figure 47 - London show VLAN	71
Figure 48 - London VLAN show interface trunk	72
Figure 49 - London VLAN Command Line Interface	73
Figure 50 - London VLAN Router Show Run	74
Figure 51 - Dynamically Updated Rome Routing Table	75
Figure 52 - VLAN Testing Ping Test PC2 to PC3	76
Figure 53 - VLAN Testing Ping Test PC3 to PC2	77
Figure 54 - Access Control List Denying London PC3.....	78
Figure 55 - Access Control List Testing	79
Figure 56 Activate the web server for the PLC.....	80
Figure 57 - Add the file extensions for the web program.....	80
Figure 58 - Create the default tags for motor control and monitor.....	80
Figure 59 - Create the tags in the web data [DB1] data block	80
Figure 60 - The motor drive PWM function is in the main part of the PLC program	81
Figure 61 - In the main function block add the web server.....	81
Figure 62 - Downloading the index .html page with http:// source for the Motor Switch.htm page..	81
Figure 63 - Set the VLAN gateway IP address (router gateway) in the PLC	82
Figure 64 - The web pages held in 'User Files' of the PLC program files	82
Figure 65 - Files stored in the 'User Files' of the TIA program files.....	82
Figure 66 - Communication test of PC connected to the switch.....	83
Figure 67 - Communication test of PC with PLC program	84
Figure 68 - Motor speed at 1%.....	85
Figure 69 - Motor speed at 40%.....	86
Figure 70 - Motor speed at 80%.....	87
Figure 71 - Motor Switch.htm program listing with comments.....	88
Figure 72 - Motor Switch.htm program listing with comments.....	88
Figure 73 - Motor Switch.htm program listing with comments.....	89
Figure 74 - Motor Switch.htm program listing with comments.....	89
Figure 75 - Motor Switch.htm program listing with comments.....	90
Figure 76 - Motor Switch.htm program listing with comments.....	90

Extensive list of figures

Table of tables:

Table 1 - London.....	17
Table 2 - Rome	18
Table 3 - Paris.....	19
London-Rome Network 4 Network address 64	
Table 4 - London-Rome	20
Rome – Paris Network 5 Network address 80	
Table 5 - Paris-Rome	20
Paris-London Network 6 Network address 96	
Table 6 - Paris-London	20
Table 7 - CIDR/27 Network Table.....	22
Table 8 - CIDR/30 Network Table.....	23
Table 9 - VLAN London Network Address.....	25
Table 10 London Router and Switch.....	25

Element 1

1. Introduction

The purpose of this particular element is to carry out research, design and implementation of a 3 LAN network system to carry out configuration and testing of dynamic addressing and routing. In order to design, implement and test the 3 LAN network, 'Cisco Packet Tracer' will provide a suitable platform to execute this. Research is necessary in order to provide an understanding of the overall system. Research into the fundamental principles related to the experimental elements of element 1 is required. The key principles being the OSI 7-layer and 4-layer models, the process of encapsulation and de-encapsulation and broadcasting and port allocation are all vital elements to this portfolio. Upon completion of the research, the practical phase will commence using the 'Cisco Packet Tracer' provides an application consisting of 3 LAN networks connected by 2 WAN networks. The structure of the overall network will consist of three routers, three network switches and three PC's. An addressing scheme was created for this particular topology. The final stage will involve testing in the form of static routing and addressing protocols. Testing will be carried out through dynamic routing and addressing protocols and then dynamic testing across all routers. Element 1 will be evaluated before proceeding to element 2 and 3.

Introduction can be more general

2. Research

2.1 OSI 7-layer and 4-layer models with reference to the network design

The Open System Interconnect (OSI) model is a conceptual framework that describes networking or telecommunications systems as seven layers, each with its own functions. The layers enables networks to visualise what is happening within their networks and can help network managers to narrow down issues as well as computer programmers, (Shaw, 2022). See figure 1 for a table displaying each layer of the 7 layer OSI model.

The TCP/IP protocol for internetwork communications is what is known as the 'internet model' or the '4-layer model'. The model shows the four functions that are necessary in order for communications. See figure 2 for a clear example of this.

The OSI 7-layer and TCP/IP 4-layer models can be compared. It can be seen that the OSI's 3rd layer, the 'Network Layer' is the same as the TCP/IP's internet layer, this describes the protocols which can address and route carious messages via the internet. The transport layer, which is known as the OSI layer 4 and TCP/IP layer is another direct comparison. This layer refers to the functions and services that enable reliability of delivery of data between the destination and source hosts. See figure 3 for an example demonstrating this.

Explanation includes significant points

2.2 Encapsulation

Data encapsulation is an object oriented programming concept that has the ability to bind a group of related properties and functions. Other members are seen as a single unit. 'Class' is the best example of data encapsulation. It is also known as data hiding that prevents the user accessing the implementation details. Encapsulation can make certain the integrity of the data contained in the object, (Net-Information, n.d.). Data encapsulation is where data is transmitted across a network to a desired device. The data that has been transmitted is wrapped with the necessary protocol information at each of the individual layers of the OSI. The layers have the ability of communicating with its receiving peer layer at the receiver. When encapsulation is being executed, data is then added and removed as it processes through each step of the OSI layers. Information can be added before or after the data. Information that is added before the data is known as a 'header' and data added after is known as the 'trailer'. See figure 4 for an example of encapsulation.

Decapsulation should be explained here also

2.3 Broadcasting

Broadcasting is the simultaneous transmission of the same message to multiple receivers. In networking, broadcasting is required when a transmitted data packet is received by all the network devices. There are some issues with broadcasting such as, security issues may occur and result in a loss of data if a network is attacked, (Techopedia, 2014). See figure 5 for an example of broadcasting. Broadcasting can be categorised into two categories, 'Limited Broadcasting' and 'Direct Broadcasting'. Limited broadcasting is used in applications where a stream of data packets is sent to all devices on the network. This is done by appending all 32 bits of the IP address set to 1, which results in a gateway of 255.255.255.255, known as a 'Limited Broadcast Address' located in the destination address of the datagram, this is reserved for the transfer of information to all its recipients from an individual sender over the network. Direct broadcasting is where a device within one network sends a packet of information to all devices on a different network.

The broadcast is explained, However the problem that it possesses should have been explained.

2.4 Port Allocation

Port numbers can range from 0 to 65535. Port numbers ranging from 0 to 1023 are reserved for common TCP/IP applications and are known as 'well known ports'. Well known ports allow client applications to identify and locate the corresponding server application processes on other hosts. For example, a client process wanting to contact a DNS process running on a server has to send the datagram to a destination port. The well-known port number for any DNS is 53, and this is where the server process should be identifying client's requests. These ports are known as 'privileged ports', however, a number of applications that formerly ran in privileged mode, such as HTTP servers, do not operate in this manner, only in the exception when binding to the port. It is important to note that it is becoming more difficult to register new applications in the space below 1023, these often use registered ports in the range of 1024 to 49151, (Goralski, 2017). *Good point identified*

3. Design requirements

3.1 Network Address Configuration

The topology of a 3 LAN network was designed and created in Cisco Packet Tracer software. The network topology was built of three routers, three network switches and three PC's. The three routers are connected via serial connections. Each router is connected to its own PC via an Ethernet switch over its LAN. See figure 6 for an example of this. The IP addresses for the network topology can be seen in figures 7, 8 and 9.

3.2 Static Routing

Static routing is executed in order to communicate packets of data between routers via serial communications. The static routing and ISO command interfaces for routers 0 to 2 are seen in figures, 10, 11 and 12.

Screen shots support this

To ensure that all routers are functional and working correctly a communication test was carried out. Packets of data were sent from PC to PC in a variety of combinations to demonstrate successful communications. The results are seen in figure 13.

Test results support explanation

3.3 RIP Routing

One efficient and effective method of routing networks is through the use of 'Routing Information Protocol', which is sometimes known as 'RIP routing'. Routers use this method for networking to exchange information regarding the network topology. RIP configured routers can send its routing table contents to its adjacent routers quickly.

The 'Command Line Interface' (CLI) prompts were entered for each router, see figures 14, 15 and 16.

From the configuration of the Command Line Interface, the RIP routing tables can be seen in figures 17, 18 and 19. The IP configurations for the RIP routing was configured and seen in figures 20, 21 and 22.

In order to confirm that all routers operations and configurations were successful and are communicating, another test for the static routing was undertaken. Packets of data were sent from PC to PC in a variety of combinations, the results can be seen in figure 23.

4. Implementation

4.1 DHCP Routing

Router 0's command line interface code for the DHCP routing was configured in figure 24. Figures 25, 26 and 27 show the device configurations.

The configuration of devices PC 1 and PC 2 were unsuccessful; as a result, they have been assigned default addresses. The router will then need helper addresses to be configured in order to communicate via the network, the purpose of these addresses is to control the serial port address of the router. Figures 28 and 29 display the DHCP helper addresses that were initiated within the packet tracer application. After initiating the DHCP helper addresses for routers 2 and 3, they were re-tested to ensure that they were working successfully. Figures 30 and 31 show that the DHCP request for PC's 1 and 2 were successful.

To secure and keep addresses for specific controller's static to ensure that they are satisfactory after the DHCP's dynamic addressing protocol, the routers were configured to exclude the first ten addresses in each LAN. Figure 32 shows code from the packet tracer application to exclude these network addresses. Figures 33, 34 and 35 show the tests proving the successful configuration of excluding addresses.

Good explanation

5. Test and analysis

To prove that the routers running configuration is correct and successful see figure 36.

There should be some comments about the results here

6. Evaluation

To conclude element 1, it is clear from the test results that it has been successful. The research has provided the foundation to the building process of this element. It provided the key principles and knowledge required to configure and construct element 1. Fundamental concepts of networking such as static routing RIP routing were utilised in this element. Comparisons can be made or the differences can be seen in the two routing methods. Static routing requires each device on its network to be manually configured, whereas, the RIP routing methods is more efficient and effective when configuring routers. It allows for remote configuration providing the internet protocol address is known. DHCP, known as Dynamic Host Communication Protocol was another fundamental tool in this element. It was a far more complex method of routing compared to that of static and RIP methods when configuring. These are some advantages to DHCP, this being its benefit of self-addressing, its IP address and simple code can be written to exclude any unwanted addresses which could cause problems. In conclusion, element 1 has provided a strong foundation for which elements 2 and 3 can be built upon.

Good evaluation

Element 2

1. Introduction

Element 2 is based upon the knowledge and understanding from the research and networking tasks carried out in element 1. However, element 2 is more aimed on classless inter-domain routing mask notation, known as CIDR. Within this element CIDR/27, CIDR/28 and CIDR/30 will be utilised in the network. Virtual Local Area Networks (VLANs) will be implemented connecting all individual networks together as an individual broadcast domain. 'Access Control List' will be utilised in order to limit the access capabilities of London's PC's to Paris's network.

Intro proves the brief is understood

2. System configuration requirements

The network for this system will consist of three different routers all connected together via a triangular topology, these routers being Rome, Paris and London. The overall network is constructed of 3 LAN's consisting of each router having its own network switch, which will connect the router to two PC's and 3 WAN's connecting the three routers together.

2.1 CIDR/28 Design Requirements

Figure 37 displays the network topology that was used within the CIDR Networking for element 2.

The CIDR/28 subnetting was constructed for the LAN's using tables 1-3.

The network diagram should have the net addresses indicated

Table 1 - London

CIDR/28 Network design

LAN Networks

CIDR/28 London Network 1 Lan address 16 *Subnet is identified*

Subnet mask	1	1	1	1	1	0	0	0	0	255.255.255.240 /28	
	0	0	0	0	0	0	0	0	0	Not used	
Net address 16	0	0	0	1	0	0	0	0	1	Lowest host address 17	192.168.6.17
Net address 16	0	0	0	1	0	0	0	1	0	Next host address 18	192.168.6.18
Default gateway	0	0	0	1	1	1	1	1	0	Fast Ethernet 0/0 address 30	192.168.6.30
	0	0	0	1	1	1	1	1	1	Broadcast 31	

Table of addresses is correct

Table 2 - Rome

CIDR/28	Rome	Network 1				Lan address 32				
Subnet mask	1	1	1	1	1	0	0	0	0	255.255.255.240 /28
	0	0	0	0	0	0	0	0	0	Not used
Net address 32	0	0	1	0	0	0	0	0	1	192.168.6.33
Net address 32	0	0	1	0	0	0	1	0	0	192.168.6.34
Default gateway	0	0	1	0	1	1	1	1	0	192.168.6.46
	0	0	1	0	1	1	1	1	1	Broadcast 47

Table of addresses is correct

Table 3 - Paris

CIDR/28	Paris	Network 3				Lan Address 48				
Subnet mask	1	1	1	1	1	0	0	0	0	255.255.255.240 /28
	0	0	0	0	0	0	0	0	0	Not used
Net address 48	0	0	1	1	1	0	0	0	1	Lowest host address 49
Net address 48	0	0	1	1	1	0	0	1	0	Next host address 50
Default gateway	0	0	1	1	1	1	1	1	0	Fast Ethernet 0/0 address 62
	0	0	1	1	1	1	1	1	1	Broadcast 63

Table for address .48 is correct

The CIDR/28 subnetting was implemented for the WAN's seen in tables 4-6.

WAN Networks

Table 4 - London-Rome

Serial link addresses are correct											
London Router	0	1	0	0	1	1	1	0	1	Serial 0/0 address 77	192.168.6.77
Rome Router	0	1	0	0	1	1	1	1	0	Serial 0/1 address 78	192.168.6.78

Rome – Paris Network 5

[illegible]

Paris-London Network 6

[illegible]

3. Implementation of CIDR/28 RIP

The routing tables for the CIDR/28 RIP routing are seen in figures 38, 39 and 40. The routing tables seen in figures 38, 39 and 40 demonstrate that each router was successfully completed the necessary configuration process for CIDR/28. This is generally seen in each routing table as each network address ends with '/28', this means that the router has correctly identified the CIDR type through the given subnet. The IP addresses are initiated using the same scheme for CIDR/28 and all networks, LAN networks seen in tables 1-3 and WAN networks seen in 4-6.

Images support explanation

4. Testing and analysis of CIDR/28

In order to confirm that each router is correctly configured and communicating, testing was carried out. Ping tests were carried out from all LAN networks. In this test, packets of data from LAN to LAN were sent. The successful status is obtained when the receiving LAN receives the data. See figure 41 for a table of successful packets delivered to each PC across the network.

Test images support explanation

2.2 CIDR/27 and CIDR/30 Networking

CIDR/27 is initiated for the LAN's. The network address used for CIDR/27 are seen in table 7.

Table 7 - CIDR/27 Network Table

Subnet is identified

CIDR/27 – Networks 1, 2 and 3										
Subnet mask	1	1	1	0	0	0	0	0	255.255.255.224	
	0	0	0	0	0	0	0	0	Not used	
Net Address 32	0	0	1	0	0	0	0	1	Lowest Host Address 33 Rome 1	192.168.6.33
Net Address 32	0	0	1	0	0	0	1	0	Next Host Address 34 Rome 2 <i>Correct for .32 address</i>	192.168.6.34
Net Address 32	0	0	1	1	1	1	1	0	Highest Router Address 62 Rome Router	192.168.6.62
Net Address 64	0	1	0	0	0	0	0	1	Lowest Host Address 65 London 1	192.168.6.65
Net Address 64	0	1	0	0	0	0	1	0	Next Host Address 66 London 2 <i>Correct for .64 address</i>	192.168.6.66
Net Address 64	0	1	0	1	1	1	1	0	Highest Router Address 94 London Router	192.168.6.94
Net Address 96	0	1	1	0	0	0	0	1	Lowest Host Address 97 Paris 1	192.168.6.97
Net Address 96	0	1	1	0	0	0	1	0	Next Host Address 98 Paris 2 <i>Correct .96 address</i>	192.168.6.98
Net Address 96	0	1	1	1	1	1	1	0	Highest Router Address 126 Paris Router	192.168.6.126

Tables are correct for the /27 scheme

CIDR/30 is initiated for the WAN's. CIDR/30 is more desired and fitting for configuring addressing between router networks that have few devices. The network addresses used for CIDR/30 are seen in table 8.

Table 8 - CIDR/30 Network Table

Subnet is identified

CIDR/30 – Networks 4, 5 and 6.										
Subnet mask	1	1	1	1	1	1	0	0	255.255.255.252	
	0	0	0	0	0	0	0	0	Not used	
Net Address 4	0	0	0	0	0	1	0	1	Lowest interface Address 5 Rome serial 0/0 <i>Correct for address .4</i>	192.168.6.5
Net Address 4	0	0	0	0	0	1	1	0	Highest interface Address 6 Paris serial 0/0	192.168.6.6
Net Address 8	0	0	0	0	1	0	0	1	Lowest interface Address 9 Rome serial 0/1 <i>Correct for address .8</i>	192.168.6.9
Net Address 8	0	0	0	0	1	0	1	0	Highest interface Address 10 London serial 0/0	192.168.6.10
Net Address 12	0	0	0	0	1	1	0	1	Lowest interface Address 13 London serial 0/1 <i>Correct for address .12</i>	192.168.6.13
Net Address 12	0	0	0	0	1	1	1	0	Highest interface Address 14 Paris serial 0/1	192.168.6.14

Table is correct for /30 scheme

3.1 Implementation for CIDR/27 and CIDR/30 RIP

The RIP routing tables for CIDR/27 and CIDR/30 are seen in figures 42, 43 and 44. The CIDR class changes are automatically detected via the subnet mask as do their corresponding internet protocol addresses. *This is a significant point*

4.1 Test and Analysis for CIDR/27 and CIDR/30

In order to confirm that the CIDR/27 and CIDR/30 routers are successful and configured correctly, testing was carried out. A ping test was executed from all LAN networks. This test sends packets of data from LAN to LAN. The successful status is received when the receiving LAN receives its data. See figure 45 for a table of successful results. *Results support the explanation*

2.3 Design Requirements for VLAN

2.3.1 VLAN London Router/Switch for CIDR/27 sub network

VLAN's are designed and used for transmitting packets of information across desired domains, this transmitted data is then received by every device that is connected to the LAN. A VLAN is seen as a collection of various network nodes that have the ability to communicate with each other just like a LAN, however, they exist in one or several different LAN segments. The segments are separated from the rest of the LAN by a router or a network switch that is used for specific departments. When a department transmits a packet of information it will then be received by all other departments on the VLAN excluding those that are outside the network. This resolves problems that are encountered by LAN's such as excessive network traffic which cause collisions.

Good explanation

The network addresses for the London VLAN are seen in table 9. The packet tracer model is seen in figure 46.

Table 9 - VLAN London Network Address

VLAN London Network Addresses									
Fast Ethernet 0/0.2	0	1	0	1	1	1	1	0	Net Addresses 64 I/F Addresses 94
Fast Ethernet 0/0.2	1	0	0	1	1	1	1	0	Net Addresses 128 I/F Addresses 158 <i>Addresses are correct</i>
PC2 London	0	1	0	0	0	0	0	1	Net Addresses 64 host Addresses 65
PC3 London	1	0	0	0	0	0	0	1	Net Addresses 128 host Addresses 129

Table 10 London Router and Switch

London Router – Fast Ethernet 0/0.2 IP address 192.168.6.94 255.255.255.224
London Router – Fast Ethernet 0/0.3 IP address 192.168.6.158 255.255.255.224
London Switch – Fast Ethernet 0/24 – Linked to London Router with encapsulation VLAN 2 and VLAN 3.
London Switch – Fast Ethernet 0/2 - Set as VLAN 2 linked to PC2 London
London Switch – Fast Ethernet 0/3 – Set as VLAN 3 linked to PC3 London

Good table showing the inter VLAN routing ports

3.2 Implementation of VLAN

Figure 47 displays the evidence of successful configuration of the VLAN. The shared gateway code for the VLAN is seen in figure 48, it displays the testing of the trunk connection. By using the 'show IP integer brief command the sub-interfaces are seen in the London router CLI, this is displayed in figure 49. Figure 50 shows the London routers, 'show run configurations.' The RIP routing table for Rome is seen in figure 51, it can be seen that the VLAN sub interface gateways have been dynamically updated in the routing table.

4.2 VLAN Testing

In order to display that the VLAN was successful it is necessary to carry out a ping test. The ping test was carried out between London's PC's to ensure correct configuration. Figures 52 and 53 show that both PC's are transmitting and receiving data as expected.

2.4 Access Control Lists (ACL)

The access control list enables a router to deny all access to certain devices that are connected to the network denied. It implements a local firewall, which can be ascertained to a network. In this case, the Paris router will be used to deny access from the London's PC3 and London's PC2 is permitted access to Paris device, see figure 54. With this security measure in place, access from the Paris device to PC3 on the London VLAN should be restricted. The ACL demonstrates the security protocols that are available on VLAN.

Good explanations

2.4.1 Access Control List Test

Figure 55 displays the access control list test. It demonstrates that London's PC3's (192.168.6.129) access to Paris' network has been restricted, this is seen in the part of the program that displays 'Request timed out', it means that Paris' did not receive the data. However, London's PC2 (192.168.6.65) was not restricted and gained full access to Paris' network seen by the replay that it obtained for this network, again in figure 55.

test results support the explanation

5. Evaluation

To conclude element 2, it can be seen from the test results that have been derived that it was a success. CIDR along with VLAN methodologies were implemented and tested and thus producing a fully functioning and communicating network. CIDR/28 subnetting was successfully implemented for and Local and Wide area networks. RIP routing tables for CIDR/28 were created and configured into the cisco packet tracer software. Routing tables were obtained providing the successful configuration as each router had identified the CIDR type through the subnet mask. Testing was a crucial element in proving that the configuration was successful. Ping tests proved useful and was carried out from all LAN networks. Packets of data were transmitted from LAN to LAN. The successful status was obtained when the receiving LAN received its data. This network was altered accordingly for the CIDR/27 and CIDR/30 within the packet tracer application to enable the use of a VLAN. It was then configured on the London network to reduce traffic across the broadcast domain and as a result reduced the potential collisions. Again, testing was performed in order to prove the functionality of the VLAN through the use of a simple ping test where packets of data were transmitted and received from PC to PC. The implementation of the Access Control List to the Paris network was also successful. The purpose behind this experiment was to prevent the communication of data between London's PC3 and Paris' network. Thus, filtering London's access to the network by programming the router and network switch allowed only certain network addresses to receive data, however, it was designed in a specific manner that other devices on the London network could fully communicate with Paris. The ping function proved successful of the intended design as London's PC3 could not communicate with Paris' network, whereas PC2 had full communication abilities with no data loss. All testing methods undertaken for element 2 proved successful. It proved that all implementation, design and testing stages of the CIDR/28, CIDR/27 and CIDR/30 stages were successful. This now laid a solid foundation to build element 3.

Good evaluation

Element 3

1. Introduction

The purpose of element 3 is to improve the network performance of the network using logical grouping of end devices. This will be done using a VLAN implemented on the London switch. Element 3 will implement and configure part of the network practically using identical layer 1, 2, and 3 hardware, and test the connectivity and security of the devices on the network. It will include 2 routers and 2 switches with a device on each switch. Testing is necessary. The network will be tested using HTTP protocol by using open source software that can access PLC data across the network. Successful implementation of the VLAN on the London switch is necessary in element 3. The network will be tested using the HTTP protocol with a PLC web server and a PC client station. A web page program will then send and receive data between the PLC and PC client station.

Relevant intro

2. System configuration requirements

In order to adhere to the requirements of element 3 it is necessary to begin by creating a new project in Siemens TIA and configure the web server. Activation of the web server for the PLC is seen in figure 56. The file extensions for the web program were then added and seen in figure 57. Here the file extensions are .html, htm and .js. Other file extensions can be .json. HTML, Hypertext Mark-up Language. HTM, Allowable short version of HTML. Both extensions are permitted. .js – java script (jQuery, ajax). .json – an extension library of java script. The .js generally is not accepted on PLC model lower than 1500. Therefore, the .js program can be embedded in the HTML or HTM using .json. Graphic extensions can be used also such as .png and .gif can be embedded in the ajax scripts. The LED's are .png images. The next step is to create the default tags for motor control and monitor, this can be seen in figure 58. Then create the tags in the web data [DB1] data block, this is seen in figure 59. The motor drive PWM function is in the main part of the PLC program seen in figure 60. In the main function block is added the web server as seen in figure 61. The web server was then configured and is ready for web pages to be downloaded. The next stage as to download the index .html page with the http:// source for the Motor Switch.htm page as seen in figure 62. Once the page was ready generation of the web data blocks were generated to download the web pages on the PLC server. The VLAN gateway was set with an IP address (router gateway) in the PLC as seen in figure 63. The web pages were all held in the same location in the 'User Files' of the PLC program files as seen in figure 64. Good point

A short embedded program using HTML and tow images can visually represent PLC conditions. This is the way that Industry 4.0 is being developed especially with PLC's. The next development stage is to implement the Java script libraries such as jQuery and json which enable real time data monitoring without page refresh. This will have a smoother graphic and increased speed of data refresh. Applications such as motor speed control and variable data values can be implemented as web pages for remote control monitoring. The pages here are in the 'User Files' and include: Motor switch.htm which is the main file downloaded to the PLC server. The other files are:

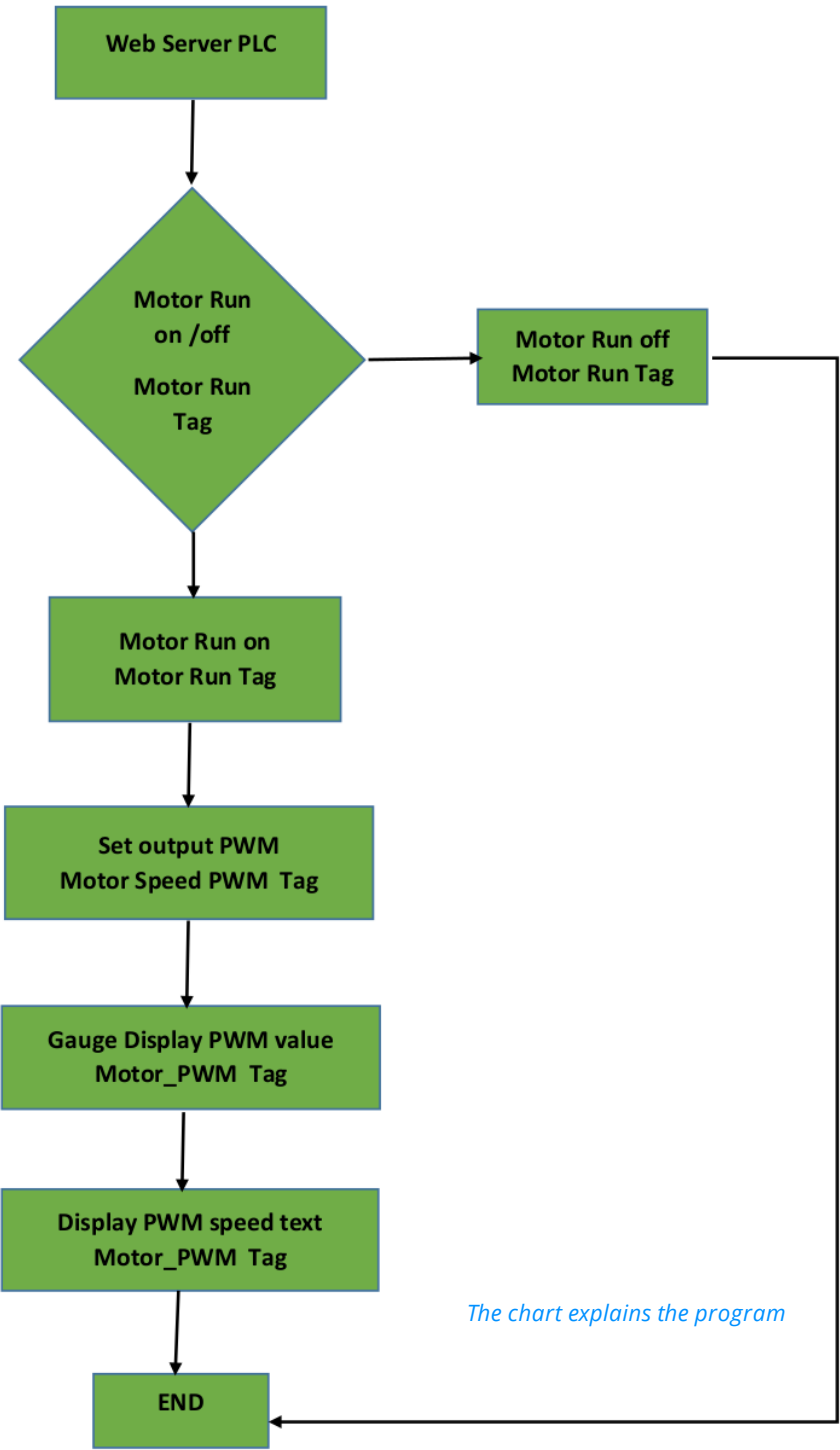
The configuration of the Cisco hardware should be explained

Js	Contain the jQuery libraries downloaded from the open source webs site.
IOCounter.json	Contains the java script ID's for the PLC tag values.
IOresult3.htm	Contains the PWM tag ':= "webdata".Motor_PWM:' which is read in the ajax function in the Motor Switch.htm main file.
IOwrite.htm	Contains the PWM drive tag 'Motor Speed PWM' from the PLC.
Motor Switch	Contains the main program where all the other tags and library functions are called.
LED 0	This is just an image of a red rectangle to represent an LED.
LED 1	This is just an image of a green rectangle to represent an LED.

These are the files used in the web server program. These files are stored in the 'User Files' of the TIA program files seen in figure 65.

There is no explanation for the VLAN config for the Cisco hardware. There is too much focus on the PLC.

Program flow chart



The chart explains the program

Program explanation:

IOwrite.htm:

```
<!--AWP_In_Variable Name = "Motor Speed PWM"-->:= "Motor Speed PWM";
```

This is the variable that holds the set value of the PWM sent to the motor. This is called from the 'Motor Switch.htm' file using its ID name.

IOresult3.htm:

This is the variable that stores the PWM value for the motor and it is read from the 'Motor Switch.htm' file using its ID name.

IOCounter.json:

```
"Motor Speed PWM" -      ":= "Motor Speed PWM":",
```

```
"Motor PWM" -           ":= "web data". Motor PWM:",
```

```
"Motor RUN" -           ":= "Motor RUN":."
```

This assigns the ID names to the PLC tags which are then called in the main 'Motor Switch.htm' file.

Motor Switch.htm:

This is the main file which is downloaded to the PLC and contains all the ajax function calls. Everything that is displayed in the web page is contained in the <body></body> tags in this program.

Tag data is identified

The dynamic Ajax functions are:

This functions reads the motor on/off status binary value and the value of the PWM driving the motor.

```
<script type="text/javascript">

$(document).ready(function()
{

$.ajaxSetup({ cache: false });
setInterval(function()
{
$.getJSON("IOCounter.json", function(result)
{
$('#Motor_PWM').text(result["Motor_PWM"]);
$('#Motor RUN').text(result["Motor RUN"]);

if(result["Motor RUN"]==0){

$('#img9').html('');
}
else
{

$('#img9').html('');
}

});
},
1000);

//END
```

This function reads the value of the PWM for the gauge.

```
$.ajaxSetup({ cache: false });
setInterval(function() {
$.get("IOresult3.htm", function(result){

    g1.refresh(parseInt(result));

});

},1000);
```

This function writes the PWM value in the text box and sends using a button press to the PLC tag ID.

```
$("#button").click(function(){  
    url="IOwrite.htm";  
    name="Motor Speed PWM";  
    val=$('#input[id=setvar]').val();  
    sdata=escape(name)+'='+val;  
    $.post(url,sdata,function(result){});  
    document.getElementById('setvar').value="";  
});  
});
```

Functions are identified

3. Implementation

Figure 66 and 67 shows a check of the communication using the ping test feature.

4. Testing and analysis

Figure 68 shows the testing of the developed web control system which has been created to control the speed of the motor using the Java Script program. This program represents the future of industry 4.0. Figure 68 shows the motor speed at 1%. Figure 69 shows the motor speed at 40%. Figure 70 shows motor speed at 80%. Figures 71-76 show 'Motor Switch.htm' program listings with comments.

Test results support this

5. Conclusion and recommendations

Element 3 has proven to be successful. It has also proven how simple it is to create a PLC monitoring the program that can be accessed from everywhere in the world with a use of a web page. This demonstrates the capabilities and the potential of 4.0 and how attractive it will become to industries worldwide. Several machines are able to communicate and work together over large distances and users can monitor and change the program that is run by simply logging into the PLC by using the IP address of the PLC. This is ideal if a system needs to be accessed and a fault quickly diagnosed by technicians from anywhere. The setup that was created was very simple and basic in terms of features and interconnected systems. It is possible to design a more complicated setup, which could monitor multiple interconnected systems and production lines. With the implementation of security protocols and passwords. This provides technicians with the ability of accessing the program from anywhere to run fault diagnosis.

Good informative evaluation

Appendix

Figure 1 - 7 layer OSI model

7	Application Layer	Human-computer interaction layer, where applications can access the network services
6	Presentation Layer	Ensures that data is in a usable format and is where data encryption occurs
5	Session Layer	Maintains connections and is responsible for controlling ports and sessions
4	Transport Layer	Transmits data using transmission protocols including TCP and UDP
3	Network Layer	Decides which physical path the data will take
2	Data Link Layer	Defines the format of data on the network
1	Physical Layer	Transmits raw bit stream over the physical medium

Appropriate image that supports the explanation

Figure 2 - The TCP/IP protocol for internetwork communications

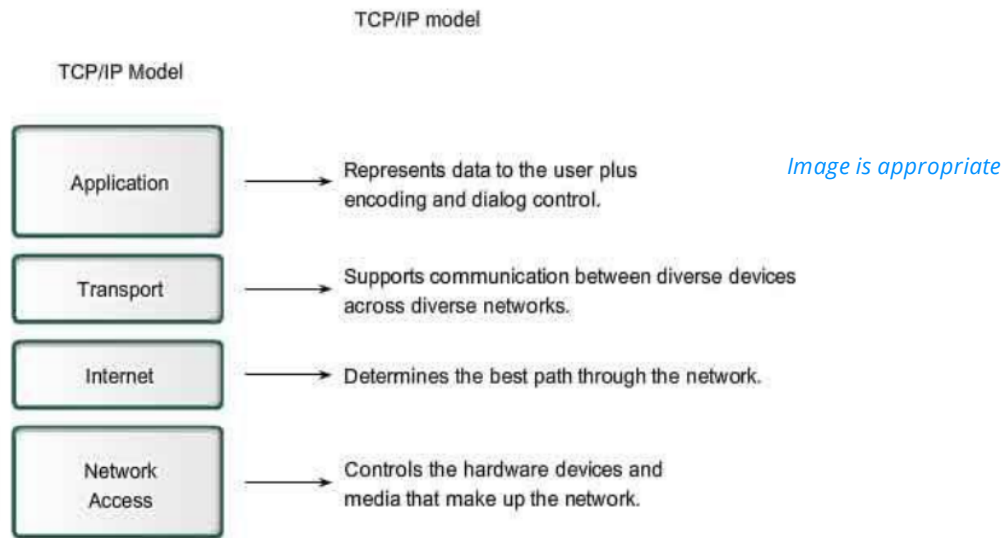
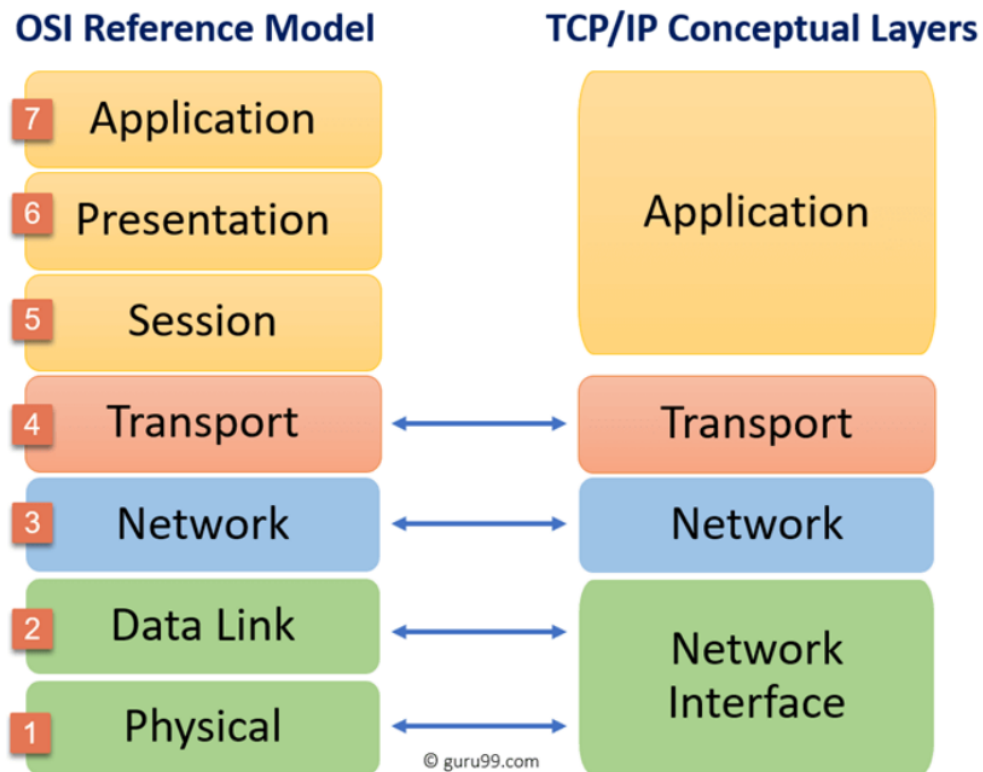


Figure 3 – Comparison of the OSI 7 layer model and the TCP/IP layers



Good comparison image where the higher level layers are combined

Figure 4 – Encapsulation

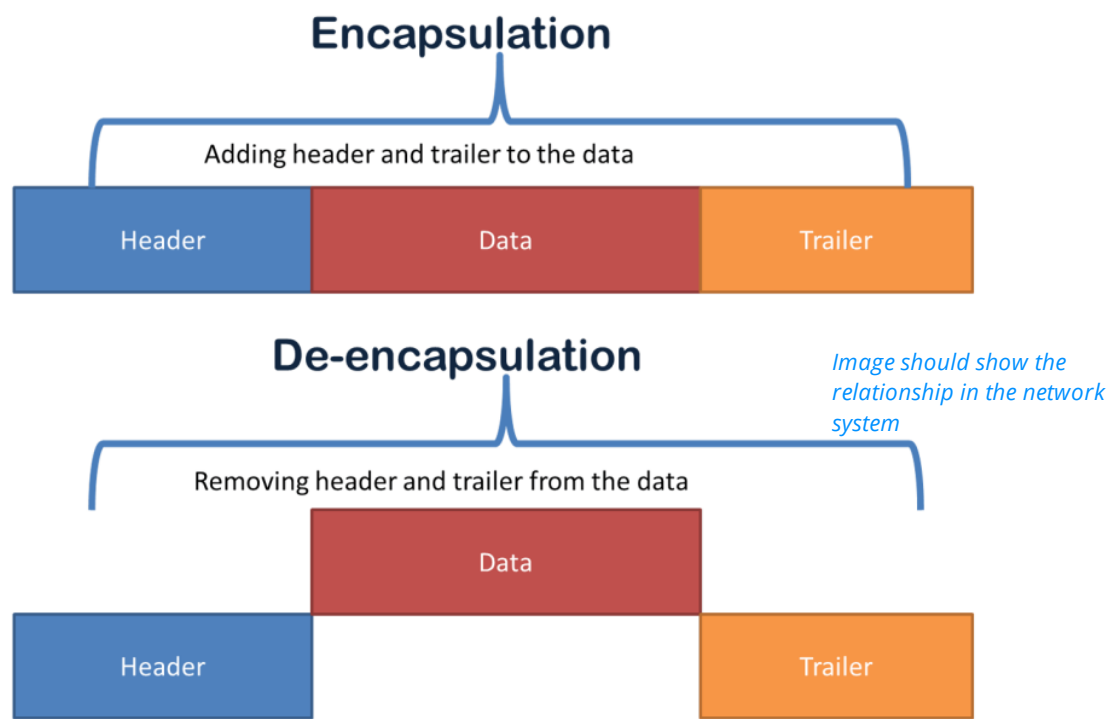


Figure 5 – Broadcasting

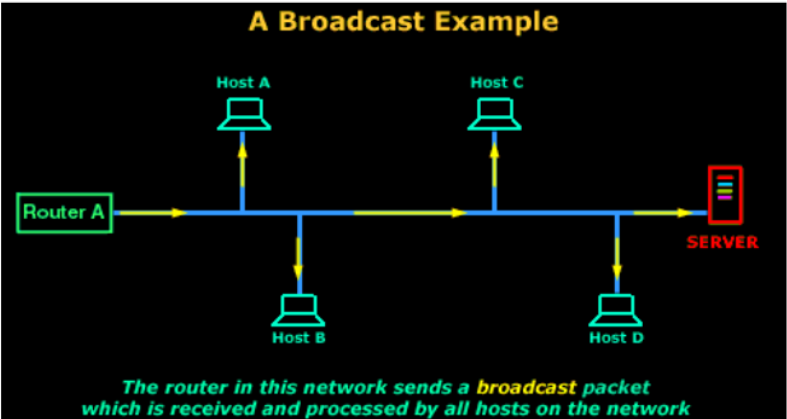


Figure 6 - LAN Network

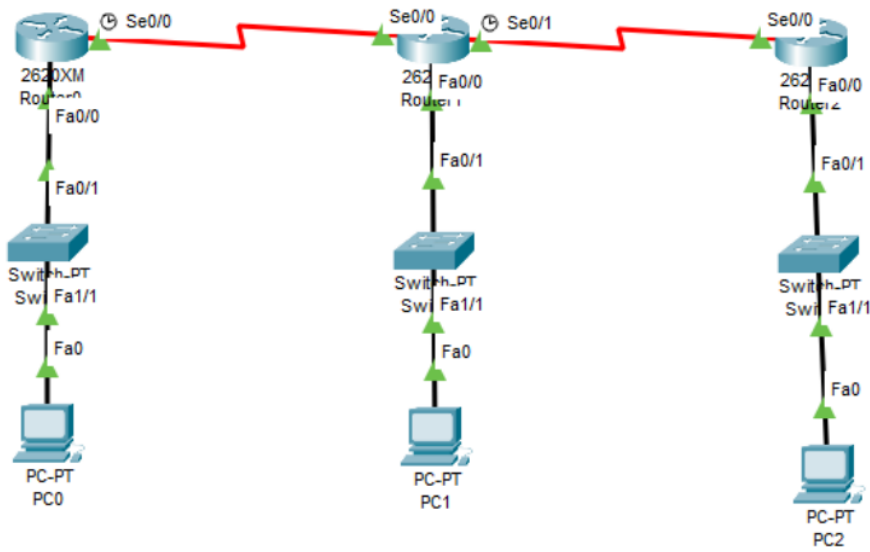


Figure 7 - IP address of router 0

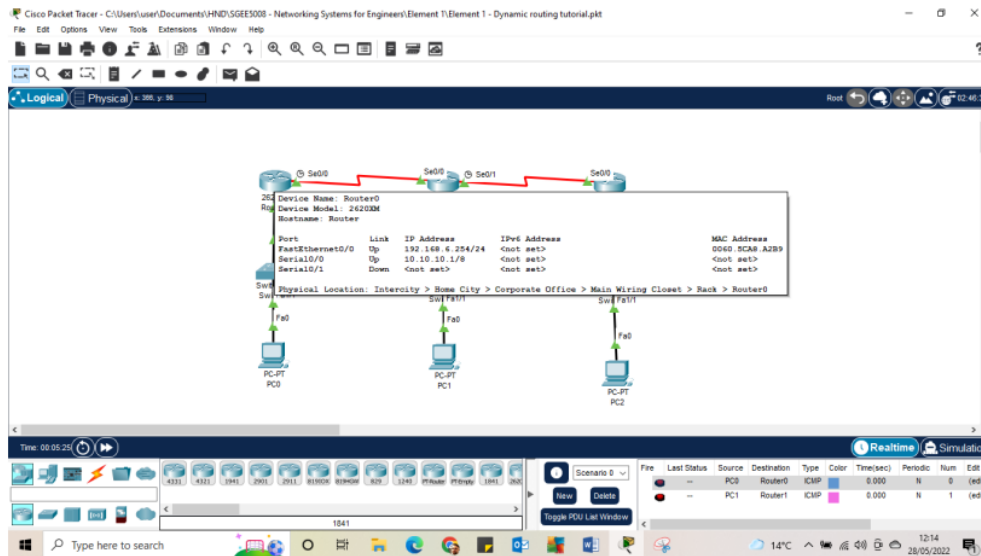


Figure 8 - IP address of router 1

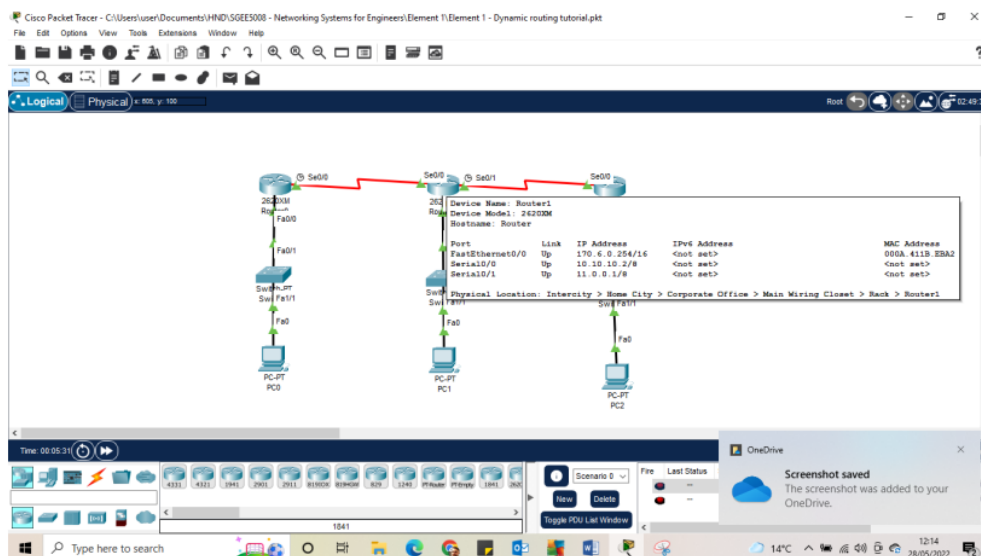
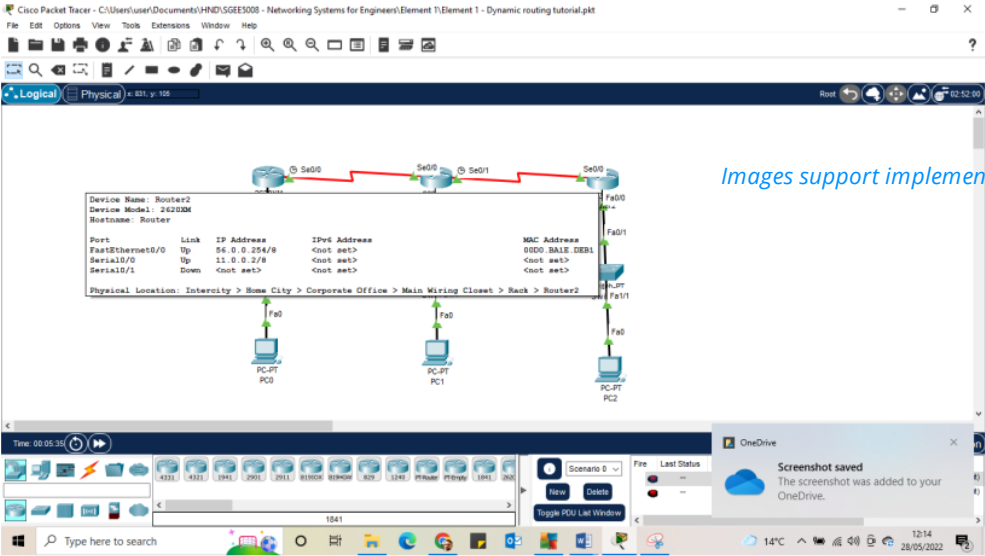


Figure 9 - IP address of router 2



Images support implementation

Figure 10 - Network Address of router 0

Network Address
56.0.0.0/8 via 10.10.10.2
170.6.0.0/16 via 10.10.10.2

The screenshot shows the Router0 CLI interface with the following content:

```

Router0
Physical Config CLI Attributes
IOS Command Line Interface
Router(config)#
Router(config)#
Router(config)#
Router(config)#show IP route
^
% Invalid input detected at '^' marker.

Router(config)#^Z
Router#
%SYS-5-CONFIG_I: Configured from console by console

Router#conf t
^
% Invalid input detected at '^' marker.

Router#enable
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#do show IP route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
        D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
        i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
        * - candidate default, U - per-user static route, o - ODR
        P - periodic downloaded static route

Gateway of last resort is not set

C    10.0.0.0/8 is directly connected, Serial0/0
S    56.0.0.0/8 [1/0] via 10.10.10.2
S    170.6.0.0/16 [1/0] via 10.10.10.2
C    192.168.6.0/24 is directly connected, FastEthernet0/0

Router(config)#
  
```

At the bottom of the window, there is a status bar with "Ctrl+F6 to exit CLI focus" and buttons for "Copy" and "Paste". A "Top" button is also visible at the bottom left.

Figure 11 - Network Address of router 1

Network Address
56.0.0.0/8 via 11.0.0.2
192.168.6.0/24 via 10.10.10.1

Router1

Physical Config **CLI** Attributes

IOS Command Line Interface

```

%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/1, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0, changed state to up

Router>enable
Router#
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#
Router(config)#conf t
%Invalid hex value
Router(config)#enable
% Incomplete command.
Router(config)#
Router(config)#conf t
%Invalid hex value
Router(config)#do show IP route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
        D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
        i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
        * - candidate default, U - per-user static route, o - ODR
        P - periodic downloaded static route

Gateway of last resort is not set

C    10.0.0.0/8 is directly connected, Serial0/0
C    11.0.0.0/8 is directly connected, Serial0/1
S    56.0.0.0/8 [1/0] via 11.0.0.2
C    170.6.0.0/16 is directly connected, FastEthernet0/0
S    192.168.6.0/24 [1/0] via 10.10.10.1

Router(config)#

```

Ctrl+F6 to exit CLI focus

Copy Paste

☐ Top

Figure 12 - Network Address of router 2

Network Address
170.6.0.0/16 via 11.0.0.1
192.168.6.0/24 via 11.0.0.1

Screen shots support main report

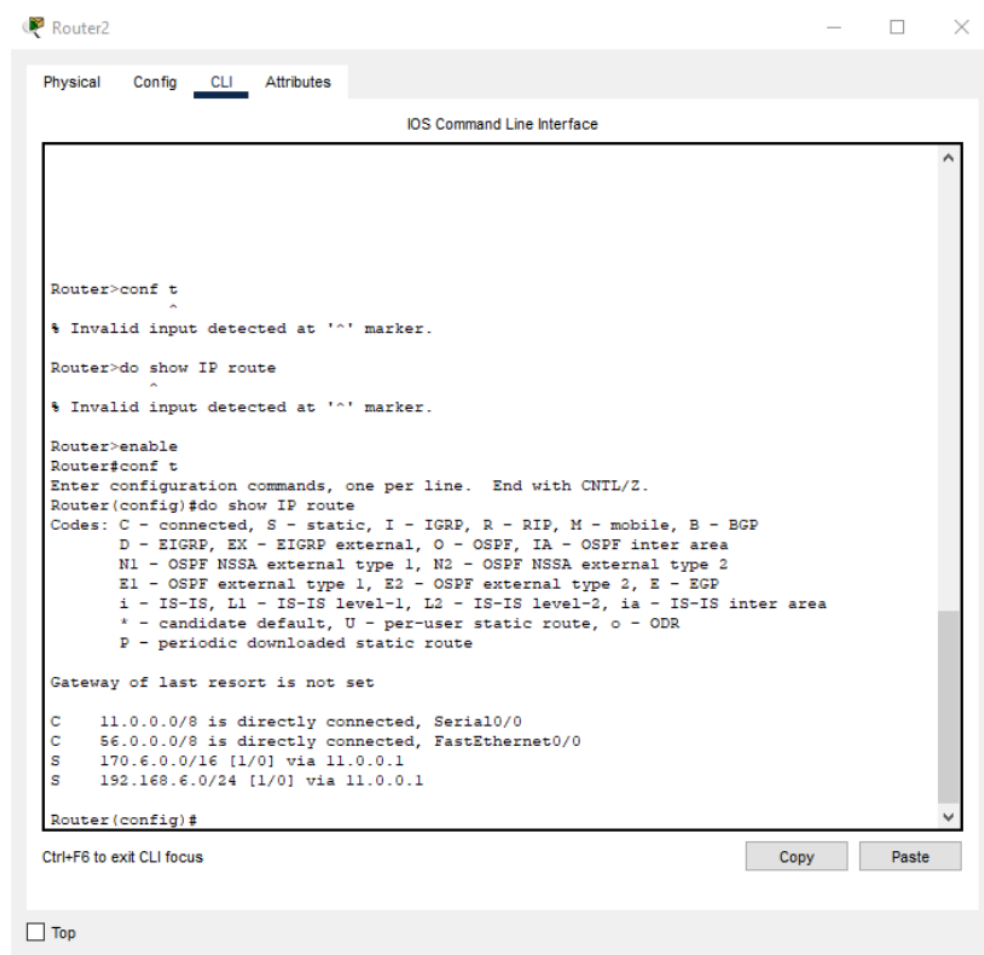


Figure 13 - Successful communications of static routing

PDU List Window

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	D
	Successful	PC0	PC1	ICMP		0.000	N	0	(edit)	
	Successful	PC1	PC0	ICMP		0.000	N	1	(edit)	
	Successful	PC1	PC2	ICMP		0.000	N	2	(edit)	
	Successful	PC2	PC1	ICMP		0.000	N	3	(edit)	
	Successful	PC0	PC2	ICMP		0.000	N	4	(edit)	
	Successful	PC2	PC0	ICMP		0.000	N	5	(edit)	

Test results support explanation in main report

Figure 14 – RIP routing address for router 0

```
%SYS-5-CONFIG_I: Configured from console by console
conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#router rip
Router(config-router)#ver 2
Router(config-router)#net 192.168.6.1
Router(config-router)#net 10.0.0.0
Router(config-router)#net 56.0.0.0
Router(config-router)#^Z
Router#
%SYS-5-CONFIG_I: Configured from console by console
```

RIP implementation

Figure 15 - RIP routing address for router 1

```
%SYS-5-CONFIG_I: Configured from console by console
conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#router rip
Router(config-router)#ver 2
Router(config-router)#net 56.0.0.0
Router(config-router)#net 10.0.0.0
Router(config-router)#net 192.168.6.1
Router(config-router)#net 11.0.0.0
Router(config-router)#net 170.6.0.0
Router(config-router)#^Z
Router#
%SYS-5-CONFIG_I: Configured from console by console
```

Figure 16 - RIP routing address for router 2

```
%SYS-5-CONFIG_I: Configured from console by console
conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#router rip
Router(config-router)#ver 2
Router(config-router)#net 170.6.0.0
Router(config-router)#net 11.0.0.0
Router(config-router)#net 56.0.0.0
Router(config-router)#^Z
Router#
%SYS-5-CONFIG_I: Configured from console by console
```

Figure 17 - Routing table of router 0

Routing Table for Router0				
Type	Network	Port	Next Hop IP	Metric
C	10.0.0.0/8	Serial0/0	---	0/0
R	11.0.0.0/8	Serial0/0	10.10.10.2	120/1
R	56.0.0.0/8	Serial0/0	10.10.10.2	120/2
R	170.6.0.0/16	Serial0/0	10.10.10.2	120/1
C	192.168.6.0/24	FastEthernet0/0	---	0/0

Figure 18 - Routing table of router 1

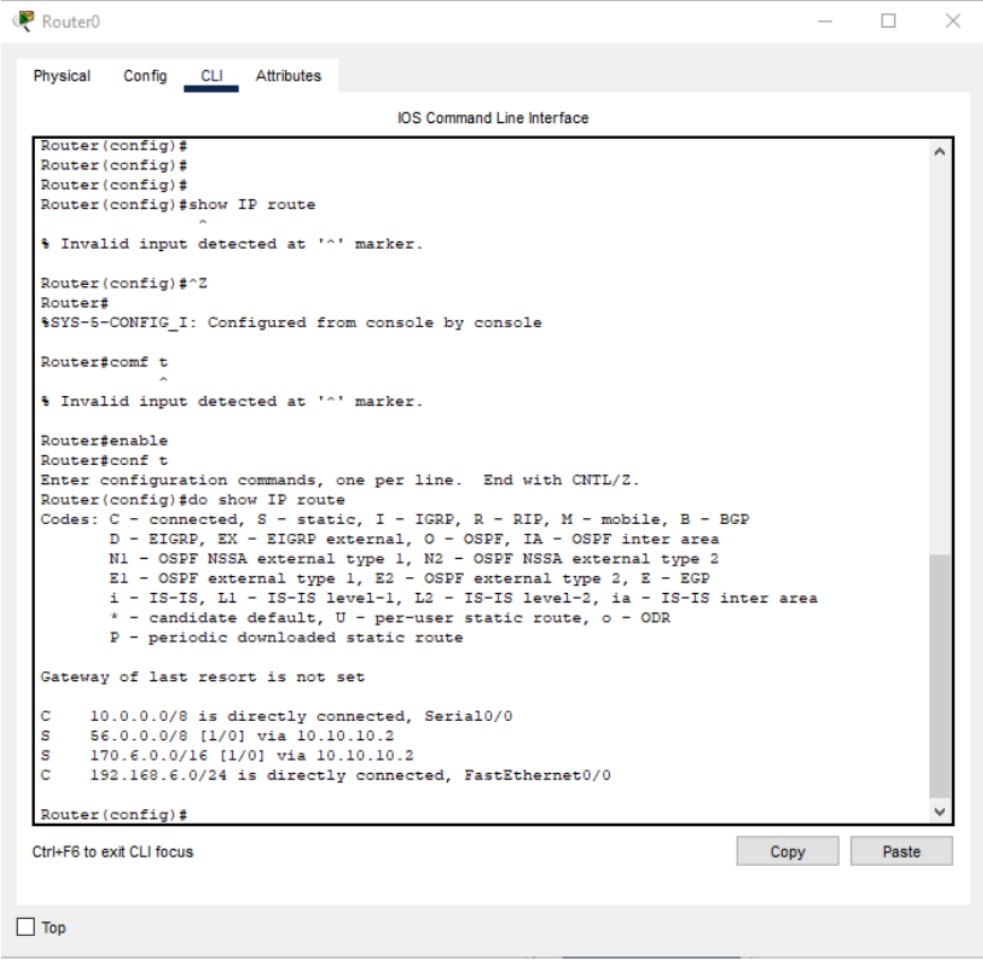
Routing Table for Router1				
Type	Network	Port	Next Hop IP	Metric
C	10.0.0.0/8	Serial0/0	---	0/0
C	11.0.0.0/8	Serial0/1	---	0/0
R	56.0.0.0/8	Serial0/1	11.0.0.2	120/1
C	170.6.0.0/16	FastEthernet0/0	---	0/0
R	192.168.6.0/24	Serial0/0	10.10.10.1	120/1

Figure 19 - Routing table of router 2

Routing Table for Router2				
Type	Network	Port	Next Hop IP	Metric
R	10.0.0.0/8	Serial0/0	11.0.0.1	120/1
C	11.0.0.0/8	Serial0/0	---	0/0
C	56.0.0.0/8	FastEthernet0/0	---	0/0
R	170.6.0.0/16	Serial0/0	11.0.0.1	120/1
R	192.168.6.0/24	Serial0/0	11.0.0.1	120/2

RIP tables support report

Figure 20 - RIP IP route of router 0



The screenshot shows a window titled "Router0" with tabs for "Physical", "Config", "CLI", and "Attributes". The "CLI" tab is active, displaying the "IOS Command Line Interface". The terminal output shows the following sequence of commands and responses:

```
Router(config)#
Router(config)#
Router(config)#
Router(config)#show IP route
^
% Invalid input detected at '^' marker.

Router(config)#^Z
Router#
%SYS-5-CONFIG_I: Configured from console by console

Router#conf t
^
% Invalid input detected at '^' marker.

Router#enable
Router#conf t
Enter configuration commands, one per line. End with CNTRL/Z.
Router(config)#do show IP route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
        D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
        i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
        * - candidate default, U - per-user static route, o - ODR
        P - periodic downloaded static route

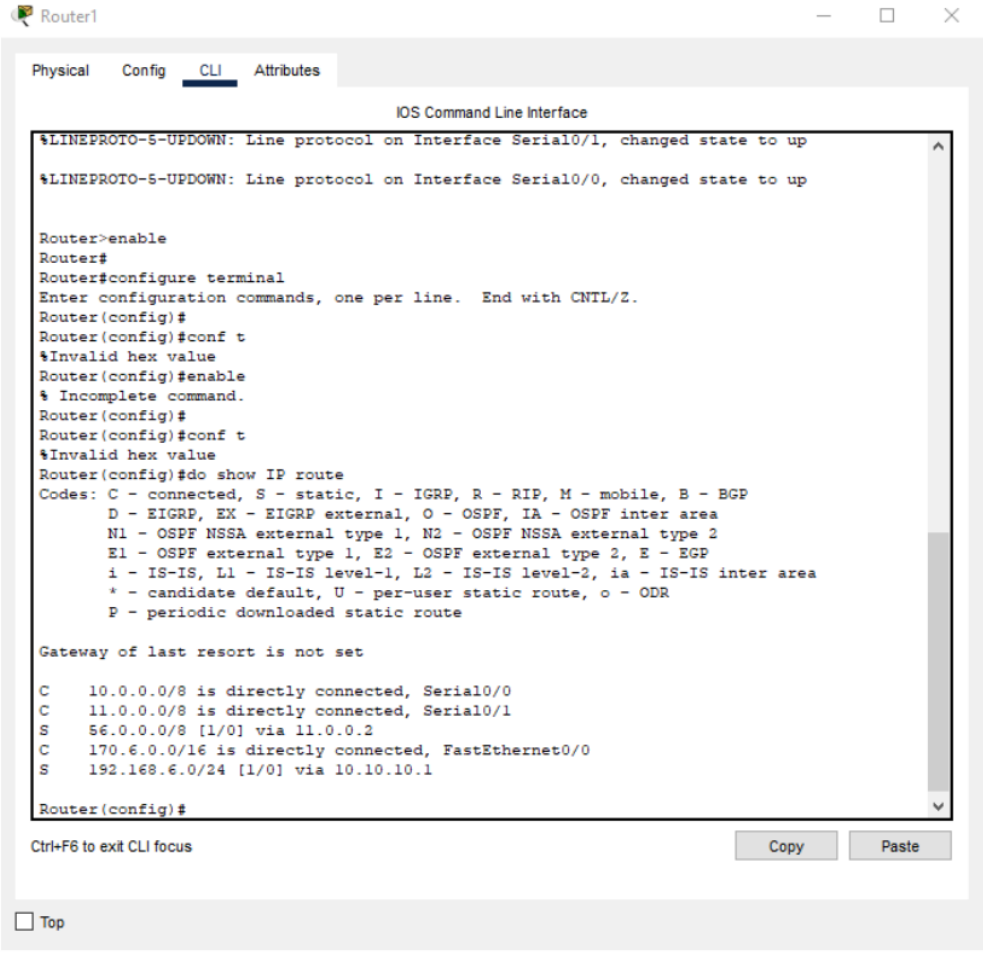
Gateway of last resort is not set

C    10.0.0.0/8 is directly connected, Serial0/0
S    56.0.0.0/8 [1/0] via 10.10.10.2
S    170.6.0.0/16 [1/0] via 10.10.10.2
C    192.168.6.0/24 is directly connected, FastEthernet0/0

Router(config)#
```

Below the terminal window, there is a "Ctrl+F6 to exit CLI focus" message and "Copy" and "Paste" buttons. At the bottom left, there is a "Top" button with a checkbox.

Figure 21 - RIP IP route of router 1



The screenshot shows a Cisco Router CLI window titled "Router1". The window has tabs for "Physical", "Config", "CLI", and "Attributes", with "CLI" being the active tab. The main area displays the "IOS Command Line Interface". The output shows the router's configuration and the results of the "show ip route" command.

```
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/1, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0, changed state to up

Router>enable
Router#
Router#configure terminal
Enter configuration commands, one per line. End with CNIL/Z.
Router(config)#
Router(config)#conf t
%Invalid hex value
Router(config)#enable
% Incomplete command.
Router(config)#
Router(config)#conf t
%Invalid hex value
Router(config)#do show IP route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
        D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
        i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
        * - candidate default, U - per-user static route, o - ODR
        P - periodic downloaded static route

Gateway of last resort is not set

C    10.0.0.0/8 is directly connected, Serial0/0
C    11.0.0.0/8 is directly connected, Serial0/1
S    56.0.0.0/8 [1/0] via 11.0.0.2
C    170.6.0.0/16 is directly connected, FastEthernet0/0
S    192.168.6.0/24 [1/0] via 10.10.10.1

Router(config)#
```

At the bottom of the CLI window, there is a "Ctrl+F6 to exit CLI focus" message and two buttons: "Copy" and "Paste". Below the CLI window, there is a "Top" button.

Figure 22 - RIP IP route of router 2

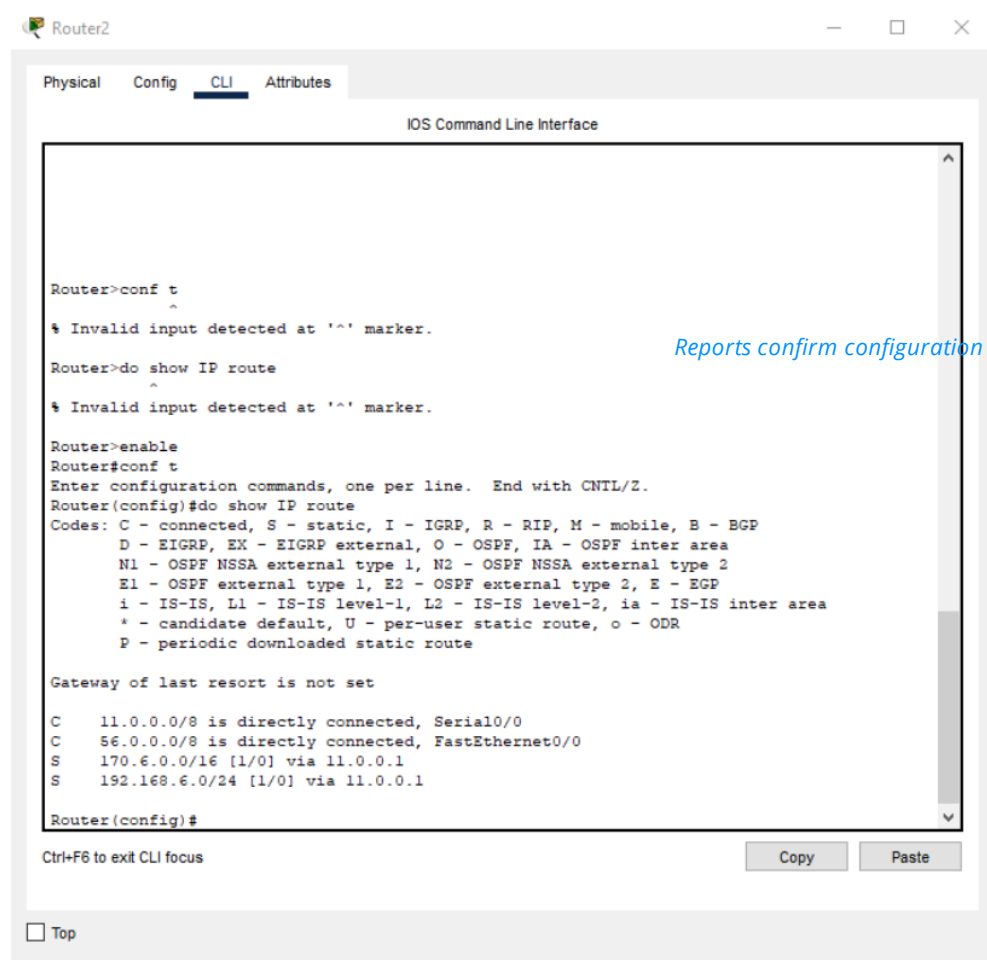


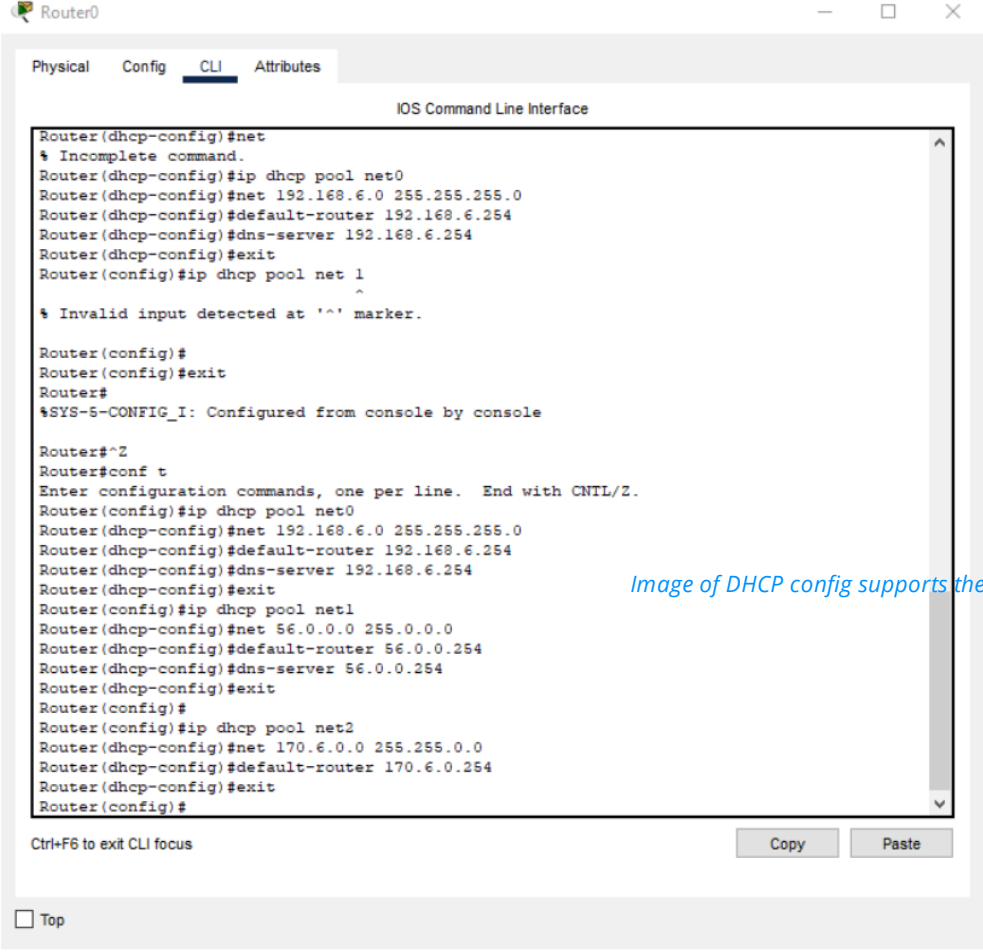
Figure 23 - RIP routing final results

PDU List Window

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	D
	Successful	PC0	PC1	ICMP		0.000	N	0	(edit)	
	Successful	PC1	PC0	ICMP		0.000	N	1	(edit)	
	Successful	PC1	PC2	ICMP		0.000	N	2	(edit)	
	Successful	PC2	PC1	ICMP		0.000	N	3	(edit)	
	Successful	PC0	PC2	ICMP		0.000	N	4	(edit)	
	Successful	PC2	PC0	ICMP		0.000	N	5	(edit)	

Test results show protocol is working

Figure 24 - DHCP routing of router 0



The screenshot shows the CLI interface of Router0. The tabs at the top are Physical, Config, CLI (selected), and Attributes. The main window displays the following commands and output:

```
Router(dhcp-config)#net
% Incomplete command.
Router(dhcp-config)#ip dhcp pool net0
Router(dhcp-config)#net 192.168.6.0 255.255.255.0
Router(dhcp-config)#default-router 192.168.6.254
Router(dhcp-config)#dns-server 192.168.6.254
Router(dhcp-config)#exit
Router(config)#ip dhcp pool net 1
Router(config)#^
% Invalid input detected at '^' marker.

Router(config)#
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console

Router#^Z
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#ip dhcp pool net0
Router(dhcp-config)#net 192.168.6.0 255.255.255.0
Router(dhcp-config)#default-router 192.168.6.254
Router(dhcp-config)#dns-server 192.168.6.254
Router(dhcp-config)#exit
Router(config)#ip dhcp pool net1
Router(dhcp-config)#net 56.0.0.0 255.0.0.0
Router(dhcp-config)#default-router 56.0.0.254
Router(dhcp-config)#dns-server 56.0.0.254
Router(dhcp-config)#exit
Router(config)#
Router(config)#ip dhcp pool net2
Router(dhcp-config)#net 170.6.0.0 255.255.0.0
Router(dhcp-config)#default-router 170.6.0.254
Router(dhcp-config)#exit
Router(config)#
```

At the bottom of the CLI window, there is a status bar with the text "Ctrl+F6 to exit CLI focus" and two buttons: "Copy" and "Paste". Below the CLI window, there is a checkbox labeled "Top".

Image of DHCP config supports the report

Figure 25 - PC 0 configuration

PC0

Physical

Config

Desktop

Programming

Attributes

IP Configuration

X

Interface

FastEthernet0

IP Configuration

☒ DHCP

☐ Static

DHCP request successful.

IPv4 Address

192.168.6.1

Subnet Mask

255.255.255.0

Default Gateway

192.168.6.254

DNS Server

192.168.6.254

IPv6 Configuration

☐ Automatic

☒ Static

IPv6 Address

/

Link Local Address

FE80::205:5EFF:FE8D:6A35

Default Gateway

DNS Server

802.1X

☐ Use 802.1X Security

Authentication

MD5

Username

Password

☐ Top

Figure 26 - PC 1 configuration

PC1

Physical

Config

Desktop

Programming

Attributes

IP Configuration

X

Interface

FastEthernet0

IP Configuration

☒ DHCP

☐ Static

DHCP failed. APIPA is being used.

IPv4 Address

169.254.114.165

Subnet Mask

255.255.0.0

Default Gateway

0.0.0.0

DNS Server

0.0.0.0

IPv6 Configuration

☐ Automatic

☒ Static

IPv6 Address

/

Link Local Address

FE80::201:43FF:FE77:72A5

Default Gateway

DNS Server

802.1X

☐ Use 802.1X Security

Authentication

MDS

Username

Password

☐ Top

Figure 27 - PC 2 configuration

PC2

Physical

Config

Desktop

Programming

Attributes

IP Configuration

X

Interface

FastEthernet0

IP Configuration

☒ DHCP

☐ Static

DHCP failed. APIPA is being used.

IPv4 Address

169.254.90.123

Subnet Mask

255.255.0.0

Default Gateway

0.0.0.0

DNS Server

0.0.0.0

IPv6 Configuration

☐ Automatic

☒ Static

IPv6 Address

/

Link Local Address

FE80::260:3EFF:FE40:5A7B

Default Gateway

DNS Server

802.1X

☐ Use 802.1X Security

Authentication

MDS

Username

Password

Top

Images support test results

Figure 28 - PC1 Helper address

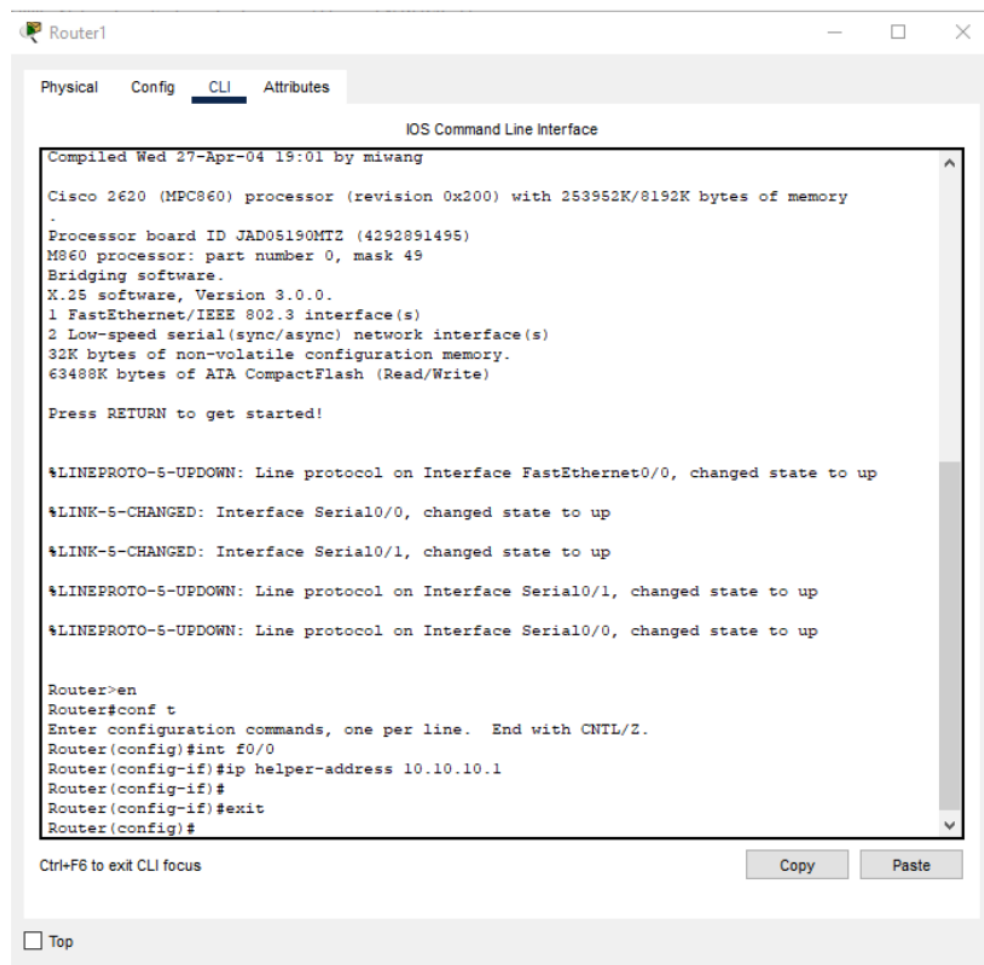


Figure 29 - PC2 Helper address

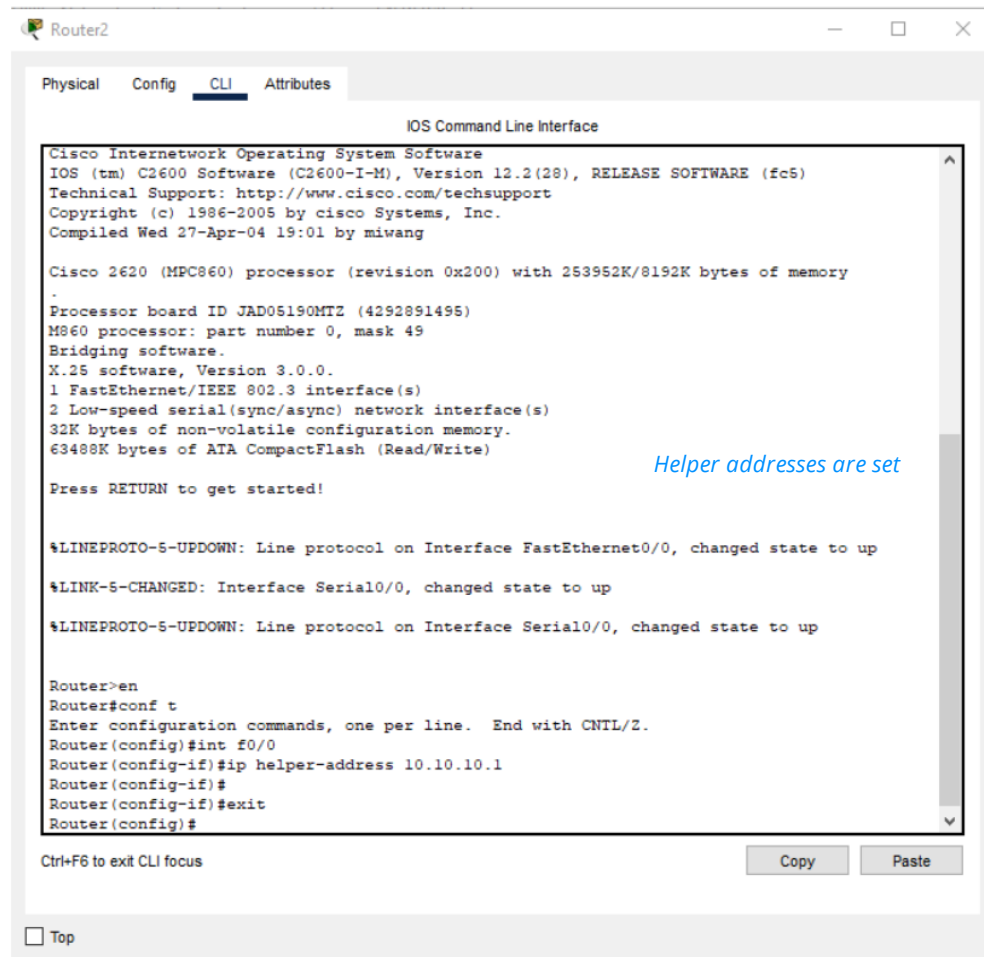


Figure 30 - DHCP Helper Address IP PC1 Successful

PC1

Physical

Config

Desktop

Programming

Attributes

IP Configuration

X

Interface

FastEthernet0

IP Configuration

DHCP

Static

DHCP request successful.

IPv4 Address

170.6.0.1

Subnet Mask

255.255.0.0

Default Gateway

170.6.0.254

DNS Server

170.6.0.254

IPv6 Configuration

Automatic

Static

IPv6 Address

/

Link Local Address

FE80::201:43FF:FE77:72A5

Default Gateway

DNS Server

802.1X

Use 802.1X Security

Authentication

MDS

Username

Password

Top

Figure 31 - DHCP Helper Address IP PC2 Successful

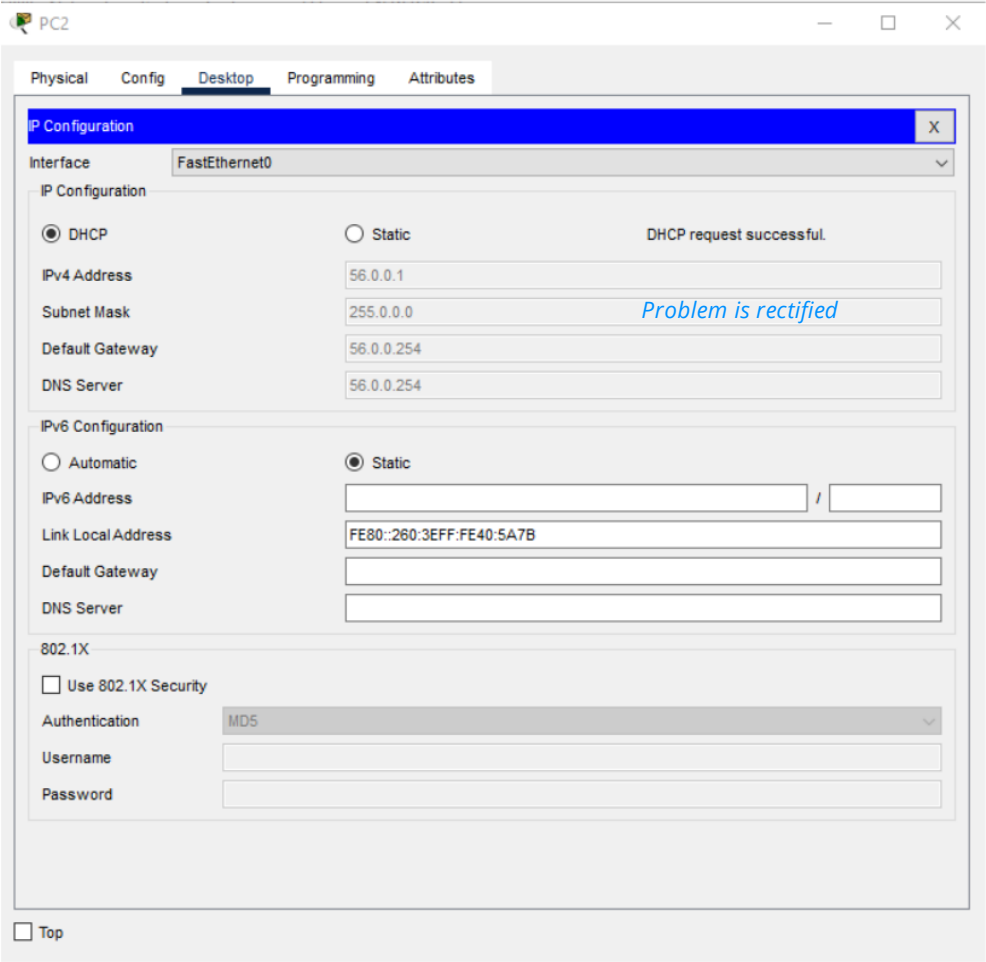


Figure 32 - DHCP Routing excluded addresses

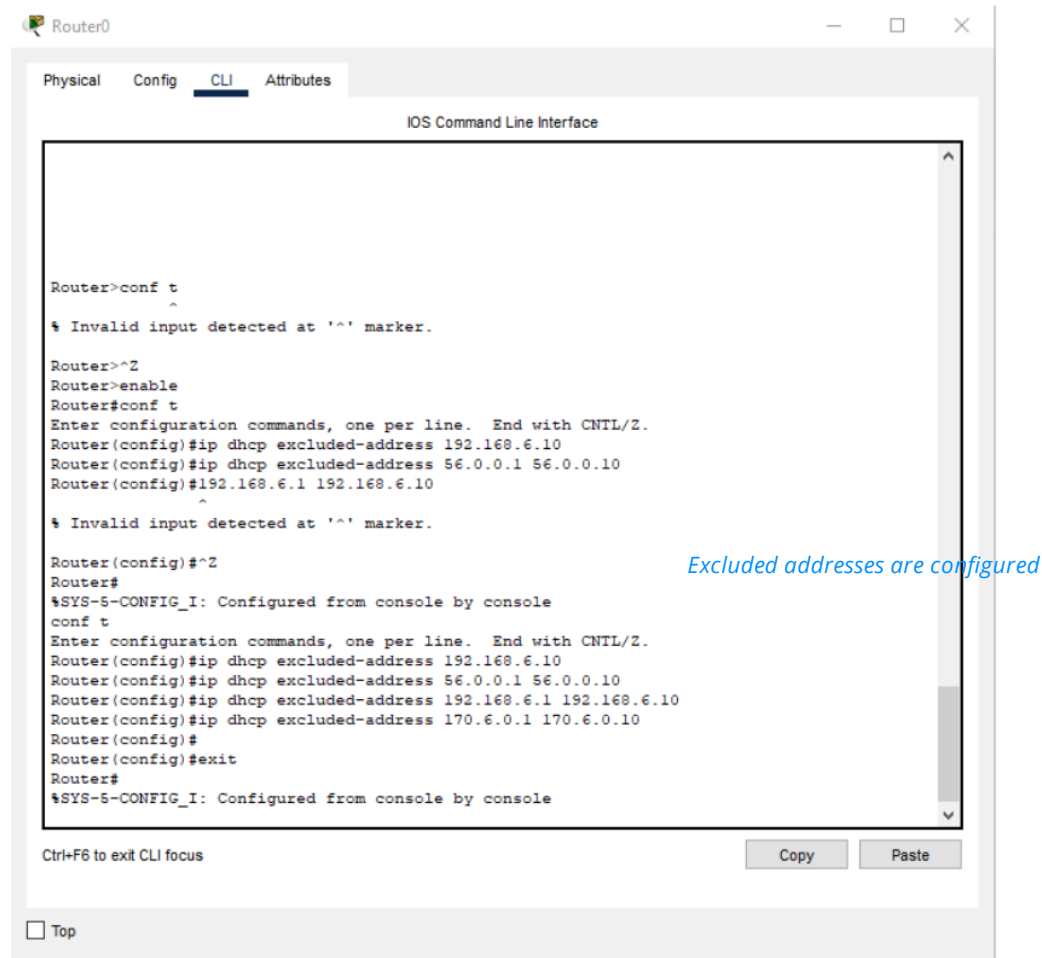


Figure 33 - DHCP Routing successful with excluded addresses of PC0

The screenshot shows the configuration window for PC0, specifically the 'Desktop' tab. The 'IP Configuration' section is active, showing the 'FastEthernet0' interface. The 'DHCP' option is selected, and the 'Static' option is also visible. The 'DHCP request successful.' message is displayed. The IPv4 Address is 192.168.6.11, Subnet Mask is 255.255.255.0, Default Gateway is 192.168.6.254, and DNS Server is 192.168.6.254. The IPv6 Configuration section shows the 'Static' option selected, with the Link Local Address set to FE80::205:5EFF:FE8D:6A35. The 802.1X section shows the 'Use 802.1X Security' checkbox unchecked, and the Authentication dropdown set to MD5. The Username and Password fields are empty. A 'Top' button is located at the bottom left of the window.

PC0

Physical Config **Desktop** Programming Attributes

IP Configuration X

Interface FastEthernet0

IP Configuration

☒ DHCP ☐ Static DHCP request successful.

IPv4 Address 192.168.6.11

Subnet Mask 255.255.255.0

Default Gateway 192.168.6.254

DNS Server 192.168.6.254

IPv6 Configuration

☐ Automatic ☒ Static

IPv6 Address /

Link Local Address FE80::205:5EFF:FE8D:6A35

Default Gateway

DNS Server

802.1X

☐ Use 802.1X Security

Authentication MD5

Username

Password

☐ Top

Figure 34 - DHCP Routing successful with excluded addresses of PC1

PC1

Physical Config Desktop Programming Attributes

IP Configuration

Interface FastEthernet0

IP Configuration

☒ DHCP ☐ Static DHCP request successful.

IPv4 Address 170.6.0.11

Subnet Mask 255.255.0.0

Default Gateway 170.6.0.254

DNS Server 170.6.0.254

IPv6 Configuration

☐ Automatic ☒ Static

IPv6 Address /

Link Local Address FE80::201:43FF:FE77:72A5

Default Gateway

DNS Server

802.1X

☐ Use 802.1X Security

Authentication MD5

Username

Password

☐ Top

Figure 35 - DHCP Routing successful with excluded addresses of PC2

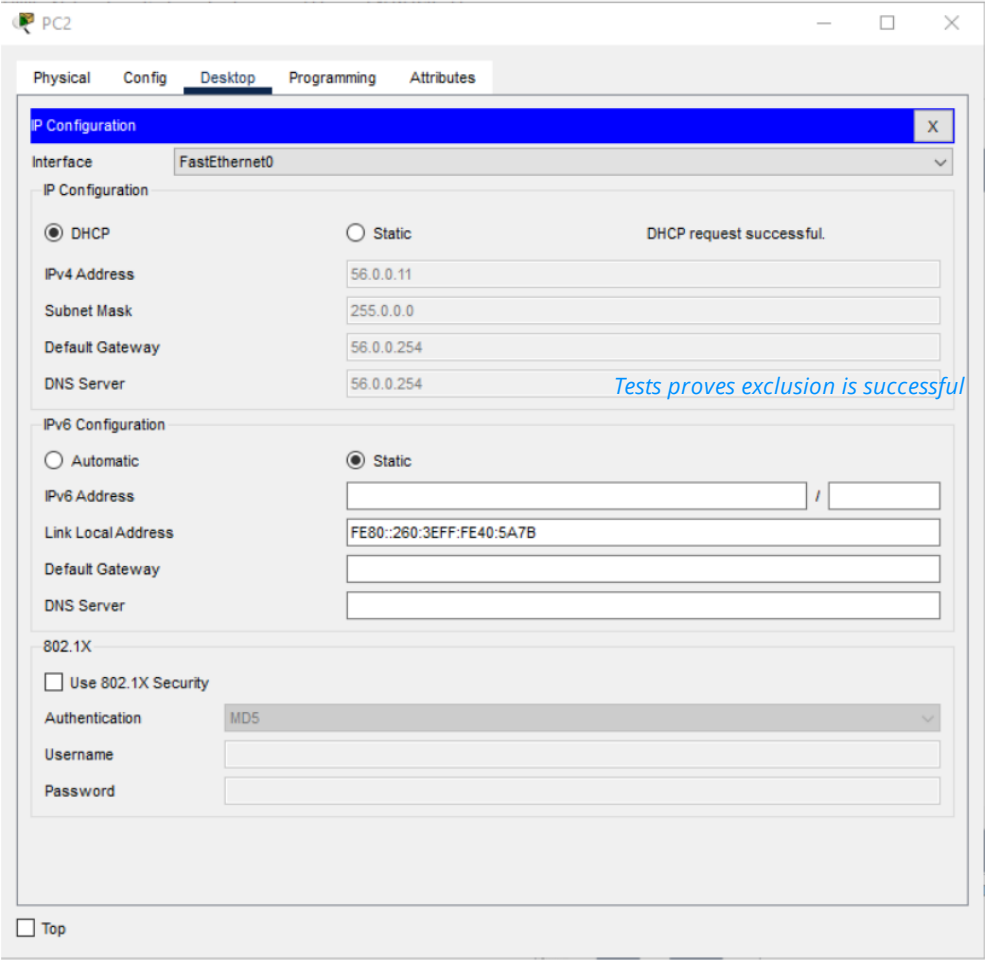
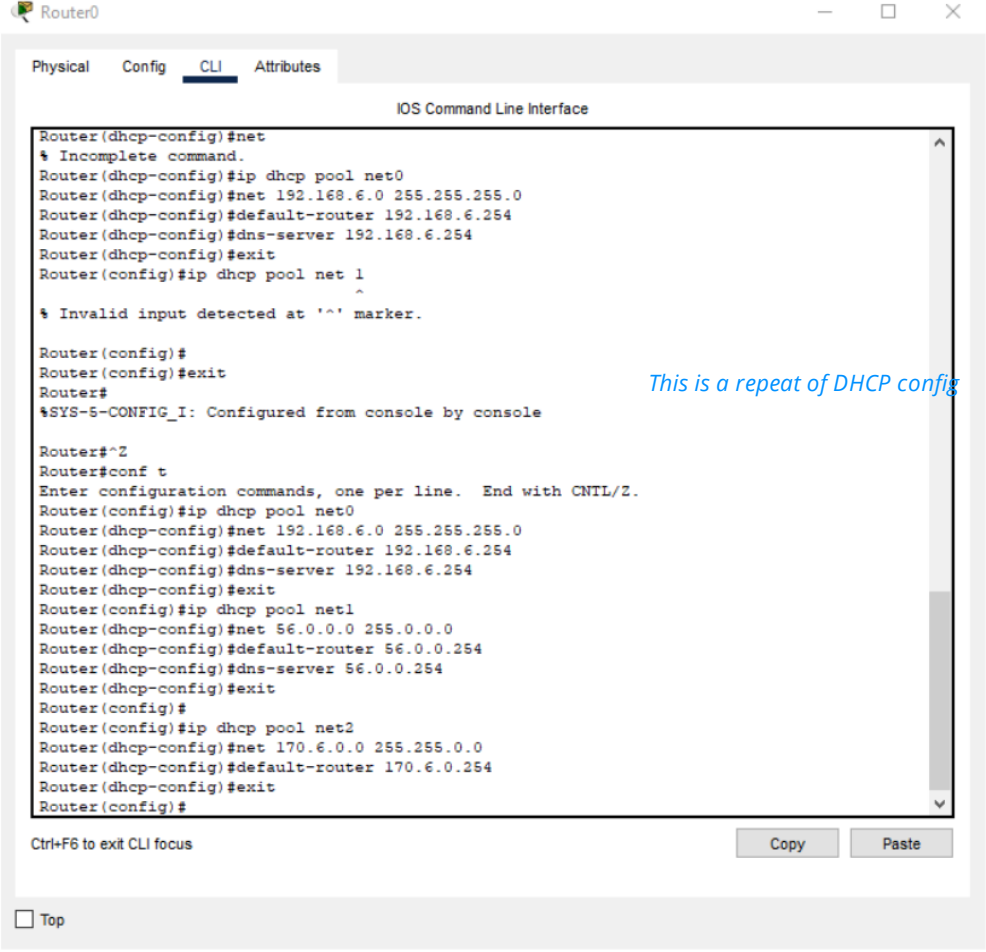


Figure 36 - All successful router configurations



```
Router0
Physical Config CLI Attributes
IOS Command Line Interface

Router(dhcp-config)#net
% Incomplete command.
Router(dhcp-config)#ip dhcp pool net0
Router(dhcp-config)#net 192.168.6.0 255.255.255.0
Router(dhcp-config)#default-router 192.168.6.254
Router(dhcp-config)#dns-server 192.168.6.254
Router(dhcp-config)#exit
Router(config)#ip dhcp pool net 1
Router(config)#^
% Invalid input detected at '^' marker.

Router(config)#
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console

Router#^Z
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#ip dhcp pool net0
Router(dhcp-config)#net 192.168.6.0 255.255.255.0
Router(dhcp-config)#default-router 192.168.6.254
Router(dhcp-config)#dns-server 192.168.6.254
Router(dhcp-config)#exit
Router(config)#ip dhcp pool net1
Router(dhcp-config)#net 56.0.0.0 255.0.0.0
Router(dhcp-config)#default-router 56.0.0.254
Router(dhcp-config)#dns-server 56.0.0.254
Router(dhcp-config)#exit
Router(config)#
Router(config)#ip dhcp pool net2
Router(dhcp-config)#net 170.6.0.0 255.255.0.0
Router(dhcp-config)#default-router 170.6.0.254
Router(dhcp-config)#exit
Router(config)#
```

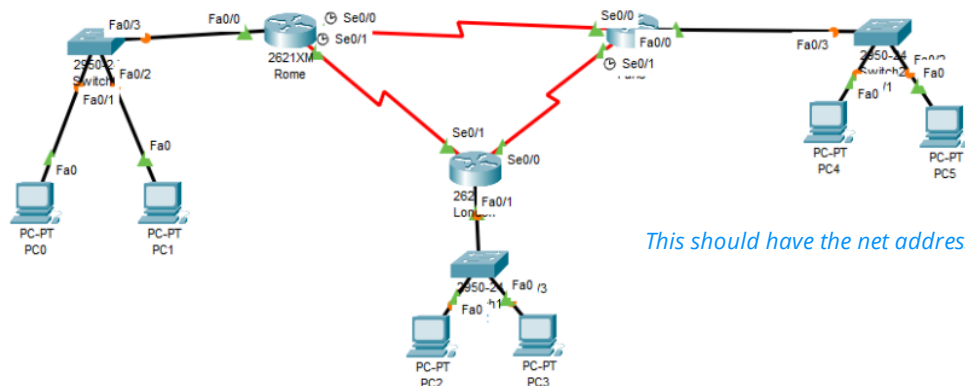
This is a repeat of DHCP config

Ctrl+F6 to exit CLI focus

Copy Paste

☐ Top

Figure 37 - CIDR/28 Network Topology



This should have the net addresses indicated

Figure 38 - Routing table for Paris

Routing Table for Paris

Type	Network	Port	Next Hop IP	Metric
R	192.168.6.16/28	Serial0/1	192.168.6.110	120/1
R	192.168.6.32/28	Serial0/0	192.168.6.94	120/1
C	192.168.6.48/28	FastEthernet0/0	---	0/0
R	192.168.6.64/28	Serial0/0	192.168.6.94	120/1
C	192.168.6.80/28	Serial0/0	---	0/0
C	192.168.6.96/28	Serial0/1	---	0/0

Figure 39 - Routing table for Rome

Routing Table for Rome				
Type	Network	Port	Next Hop IP	Metric
R	192.168.6.16/28	Serial0/1	192.168.6.77	120/1
C	192.168.6.32/28	FastEthernet0/0	---	0/0
R	192.168.6.48/28	Serial0/0	192.168.6.93	120/1
C	192.168.6.64/28	Serial0/1	---	0/0
C	192.168.6.80/28	Serial0/0	---	0/0
R	192.168.6.96/28	Serial0/1	192.168.6.77	120/1
R	192.168.6.96/28	Serial0/0	192.168.6.93	120/1

Figure 40 - Routing table for London

Routing Table for London				
Type	Network	Port	Next Hop IP	Metric
C	192.168.6.16/28	FastEthernet0/0	---	0/0
R	192.168.6.32/28	Serial0/1	192.168.6.78	120/1
R	192.168.6.48/28	Serial0/0	192.168.6.109	120/1
C	192.168.6.64/28	Serial0/1	---	0/0
R	192.168.6.80/28	Serial0/1	192.168.6.78	120/1
R	192.168.6.80/28	Serial0/0	192.168.6.109	120/1
C	192.168.6.96/28	Serial0/0	---	0/0

Router tables are correct

Figure 41 - CIDR/28 Testing Evidence

PDU List Window

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
	Successful	PC1	PC2	ICMP		0.000	N	0	(edit)	(delete)
	Successful	PC2	PC1	ICMP		0.000	N	1	(edit)	(delete)
	Successful	PC3	PC4	ICMP		0.000	N	2	(edit)	(delete)
	Successful	PC4	PC3	ICMP		0.000	N	3	(edit)	(delete)
	Successful	PC4	PC1	ICMP		0.000	N	4	(edit)	(delete)
	Successful	PC1	PC4	ICMP		0.000	N	5	(edit)	(delete)

Test results support the main report

Figure 42 - CIDR/27 and CIDR/30 RIP table for London

Type	Network	Port	Next Hop IP	Metric
R	192.168.6.4/30	Serial0/0	192.168.6.14	120/1
R	192.168.6.4/30	Serial0/1	192.168.6.9	120/1
C	192.168.6.8/30	Serial0/1	---	0/0
C	192.168.6.12/30	Serial0/0	---	0/0
R	192.168.6.32/27	Serial0/1	192.168.6.9	120/1
C	192.168.6.64/27	FastEthernet0/0	---	0/0
R	192.168.6.96/27	Serial0/0	192.168.6.14	120/1

Figure 43 - CIDR/27 and CIDR/30 RIP table for Paris

Type	Network	Port	Next Hop IP	Metric
C	192.168.6.4/30	Serial0/0	---	0/0
R	192.168.6.8/30	Serial0/0	192.168.6.5	120/1
R	192.168.6.8/30	Serial0/1	192.168.6.13	120/1
C	192.168.6.12/30	Serial0/1	---	0/0
R	192.168.6.32/27	Serial0/0	192.168.6.5	120/1
R	192.168.6.64/27	Serial0/1	192.168.6.13	120/1
C	192.168.6.96/27	FastEthernet0/0	---	0/0

Figure 44 - CIDR/27 and CIDR/30 RIP table for Rome

Routing Table for Rome					
Type	Network	Port	Next Hop IP	Metric	
C	192.168.6.4/30	Serial0/0	---	0/0	
C	192.168.6.8/30	Serial0/1	---	0/0	
R	192.168.6.12/30	Serial0/0	192.168.6.6	120/1	
R	192.168.6.12/30	Serial0/1	192.168.6.10	120/1	
C	192.168.6.32/27	FastEthernet0/0	---	0/0	
R	192.168.6.64/27	Serial0/1	192.168.6.10	120/1	
R	192.168.6.96/27	Serial0/0	192.168.6.6	120/1	

Routing tables auto update with new CIDR scheme

Figure 45 - CIDR/27 and CIDR/30 Testing Evidence

PDU List Window										
Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
	Successful	PC0	PC1	ICMP		0.000	N	0	(edit)	(delete)
	Successful	PC1	PC0	ICMP		0.000	N	1	(edit)	(delete)
	Successful	PC2	PC3	ICMP		0.000	N	2	(edit)	(delete)
	Successful	PC3	PC2	ICMP		0.000	N	3	(edit)	(delete)
	Successful	PC4	PC5	ICMP		0.000	N	4	(edit)	(delete)
	Successful	PC5	PC4	ICMP		0.000	N	5	(edit)	(delete)
	Successful	PC0	PC2	ICMP		0.000	N	6	(edit)	(delete)
	Successful	PC1	PC3	ICMP		0.000	N	7	(edit)	(delete)
	Successful	PC3	PC4	ICMP		0.000	N	8	(edit)	(delete)
	Successful	PC5	PC2	ICMP		0.000	N	9	(edit)	(delete)
	Successful	PC0	PC4	ICMP		0.000	N	10	(edit)	(delete)
	Successful	PC5	PC1	ICMP		0.000	N	11	(edit)	(delete)
	Successful	PC1	PC3	ICMP		0.000	N	12	(edit)	(delete)
	Successful	PC0	PC4	ICMP		0.000	N	13	(edit)	(delete)
	Successful	PC3	PC2	ICMP		0.000	N	14	(edit)	(delete)

Test evidence supports the report

Figure 46 - VLAN London Model

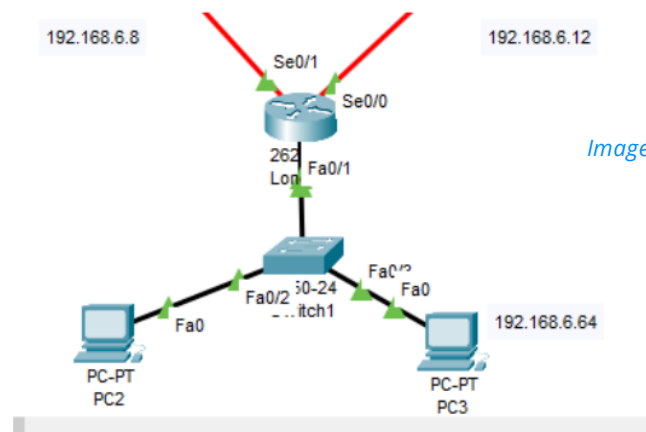


Figure 47 - London show VLAN

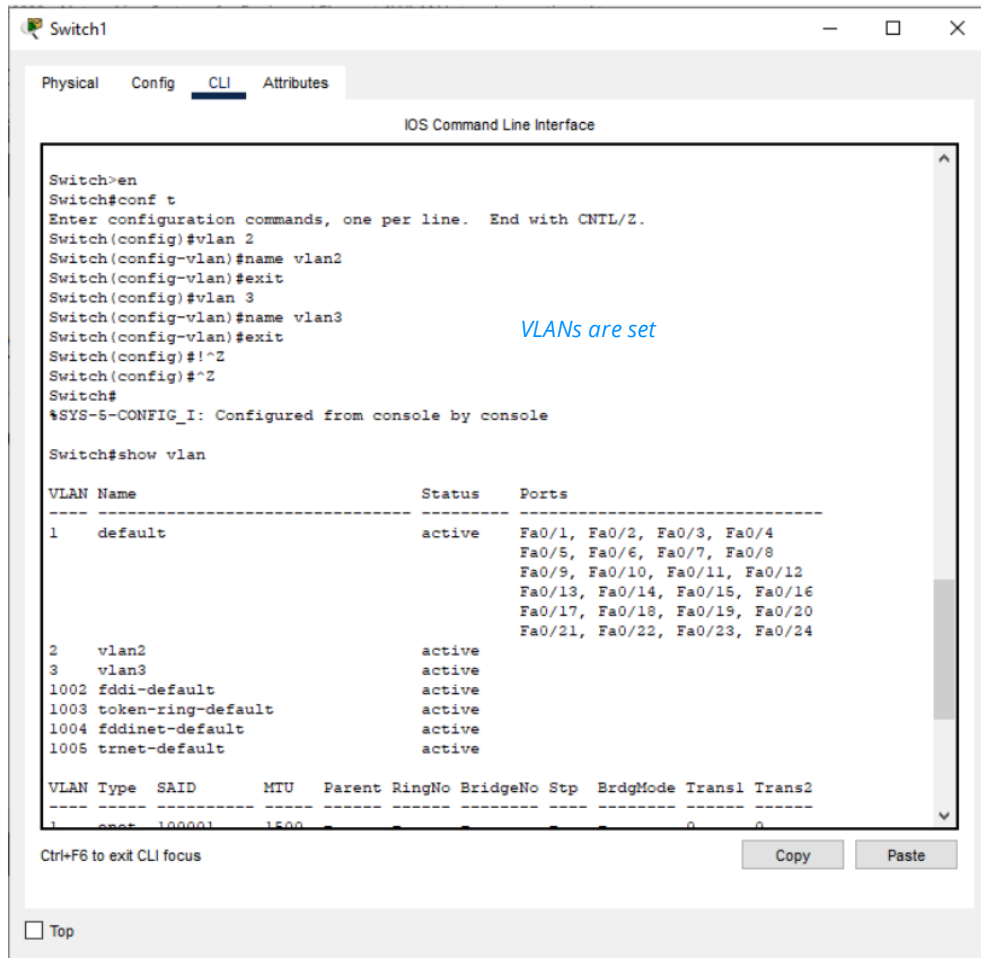


Figure 48 - London VLAN show interface trunk

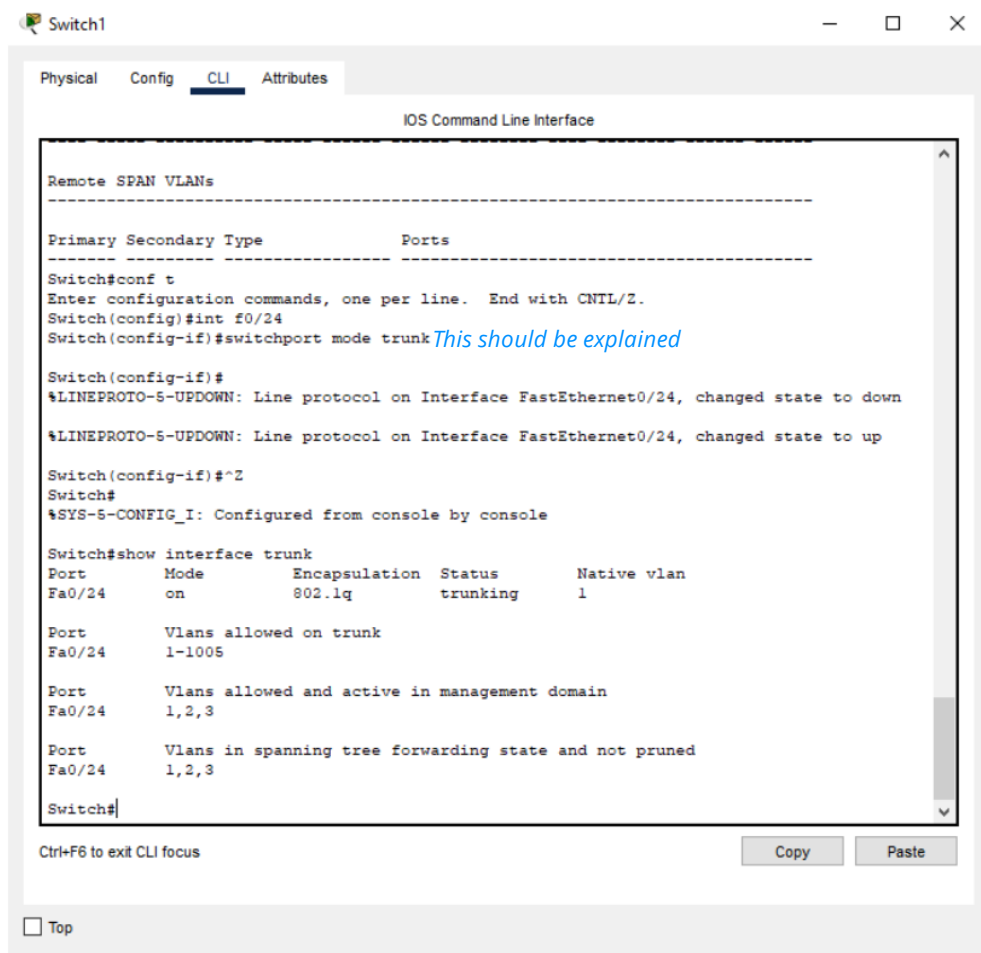


Figure 49 - London VLAN Command Line Interface

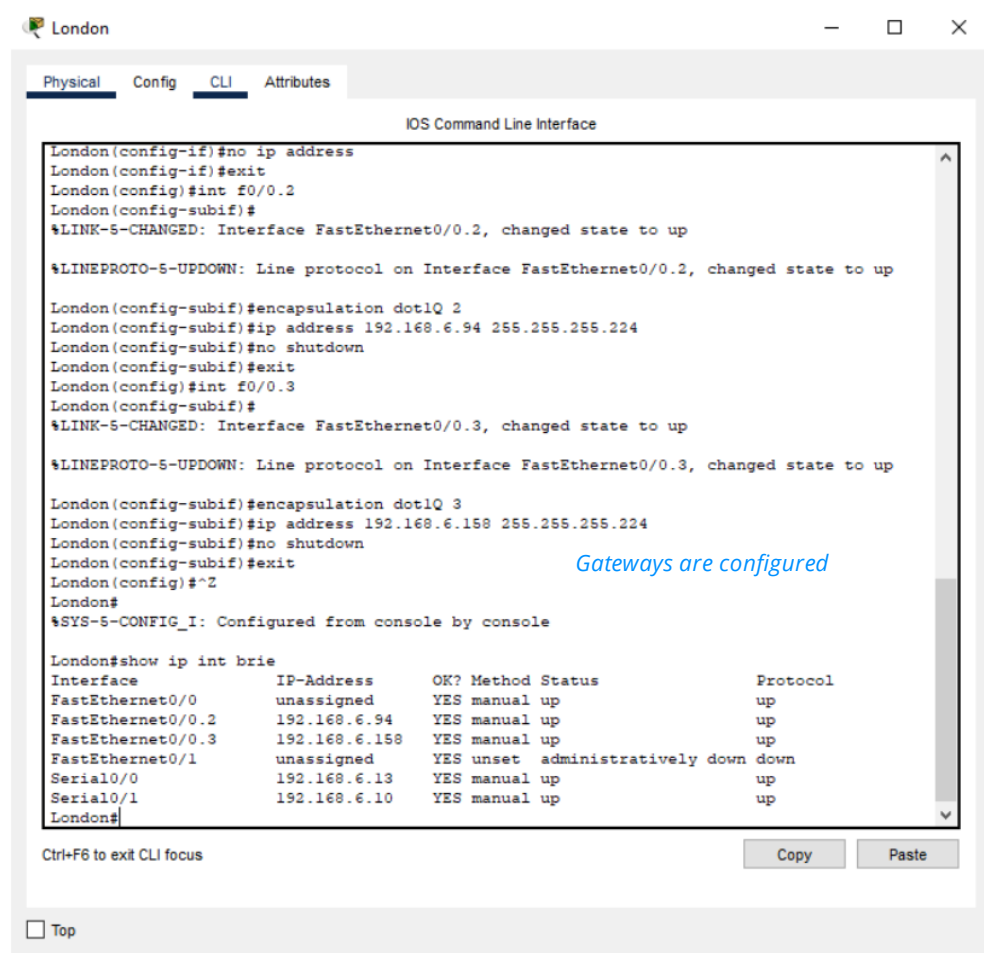


Figure 50 -London VLAN Router Show Run

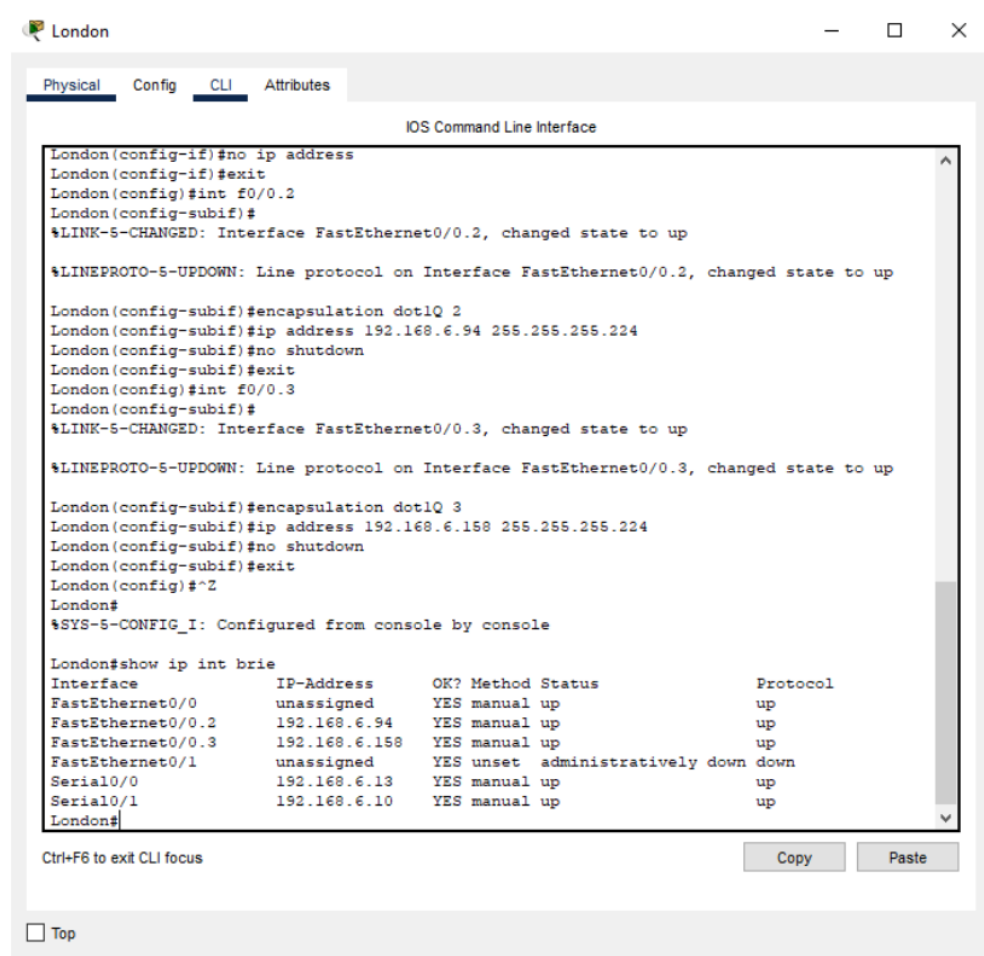
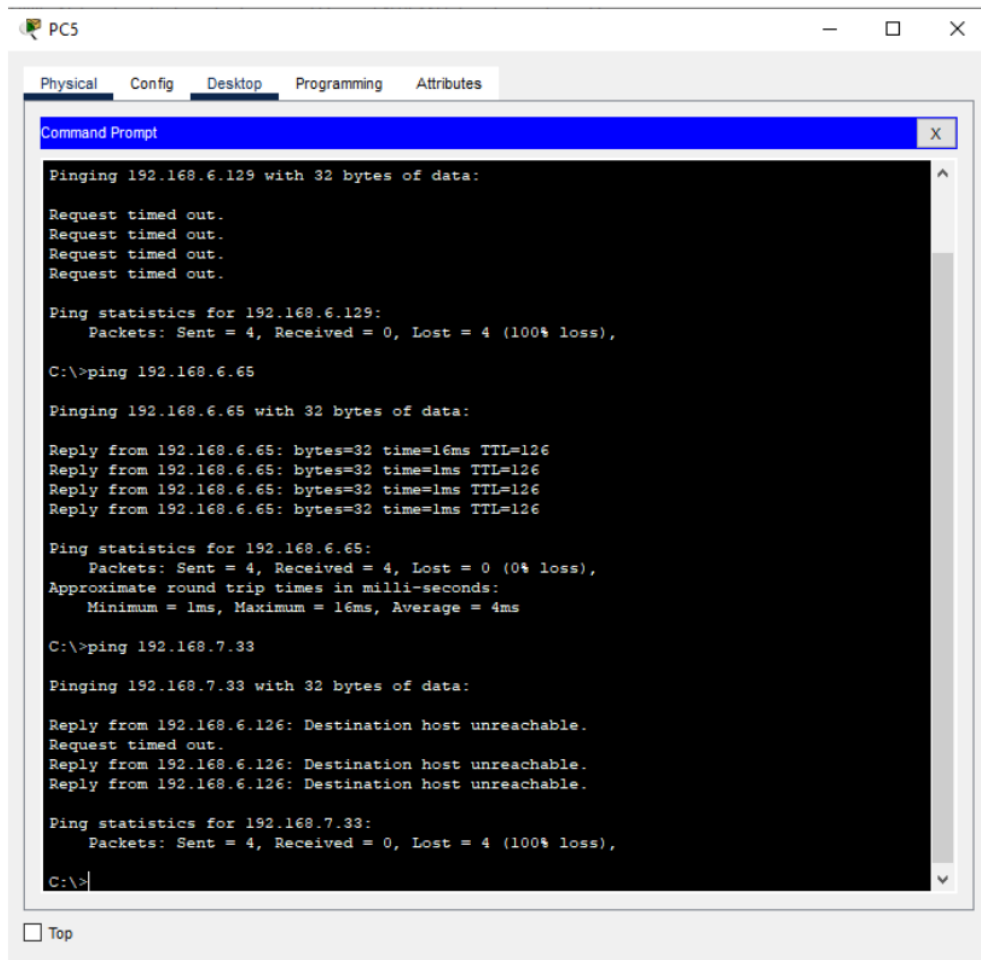


Figure 51 - Dynamically Updated Rome Routing Table

Routing Table for Rome					
Type	Network	Port	Next Hop IP	Metric	
C	192.168.6.4/30	Serial0/0	---	0/0	
C	192.168.6.8/30	Serial0/1	---	0/0	
R	192.168.6.12/30	Serial0/1	192.168.6.10	120/1	
R	192.168.6.12/30	Serial0/0	192.168.6.6	120/1	
C	192.168.6.32/27	FastEthernet0/0	---	0/0	
R	192.168.6.64/27	Serial0/1	192.168.6.10	120/1	
R	192.168.6.96/27	Serial0/0	192.168.6.6	120/1	
R	192.168.6.12...	Serial0/1	192.168.6.10	120/1	

Figure 52 - VLAN Testing Ping Test PC2 to PC3



*Test across
VLANs are
successful*

Figure 53 - VLAN Testing Ping Test PC3 to PC2

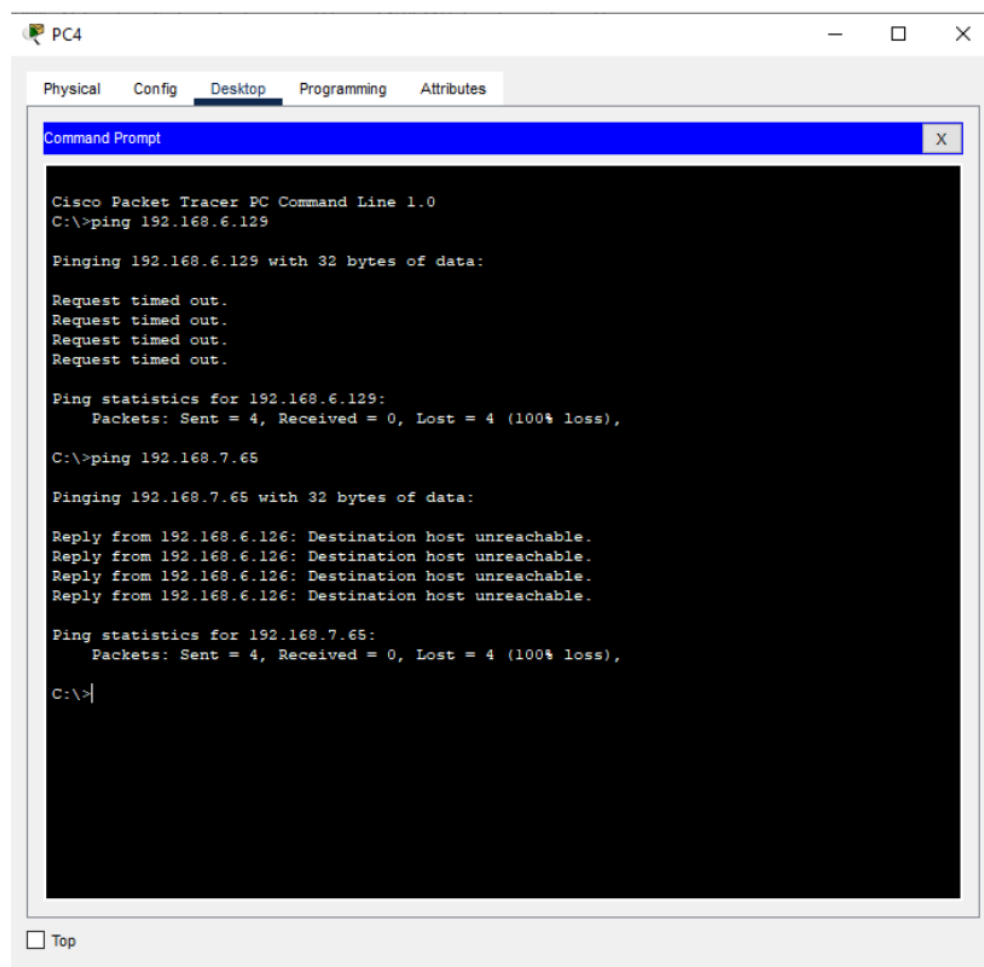


Figure 54 - Access Control List Denying London PC3

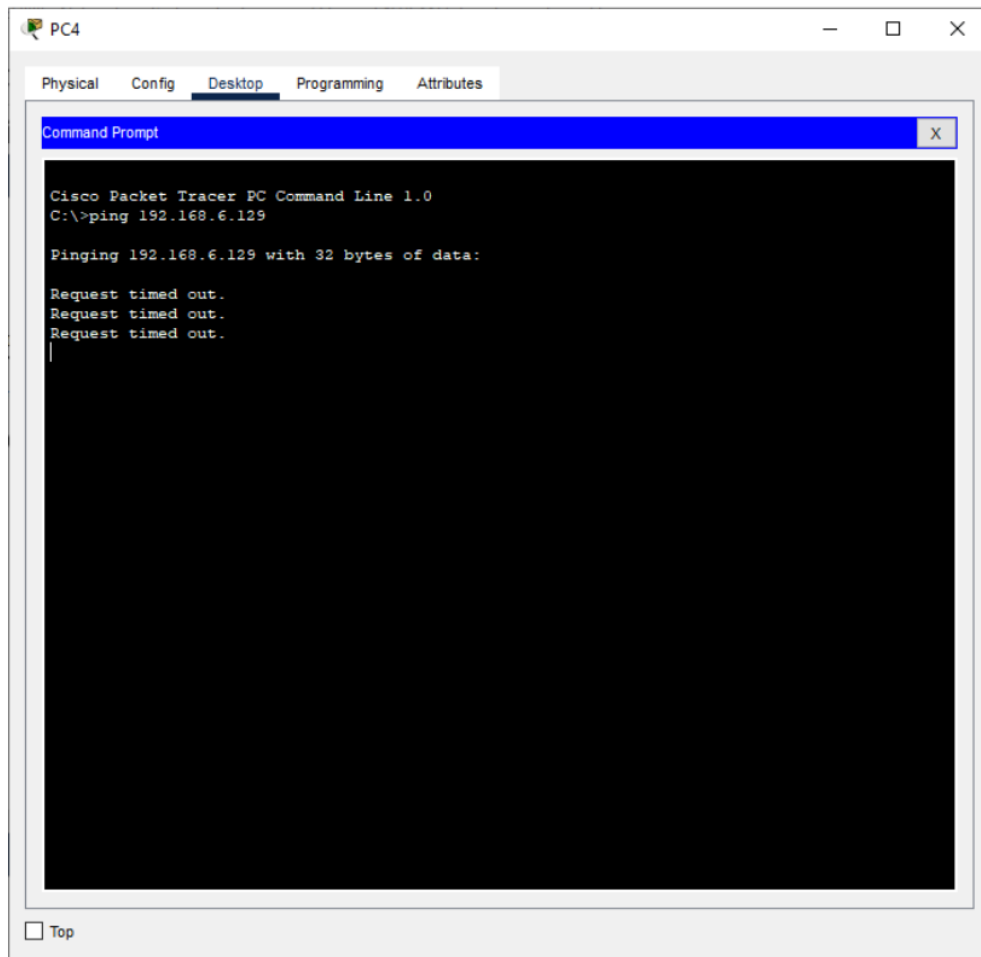


Figure 55 - Access Control List Testing

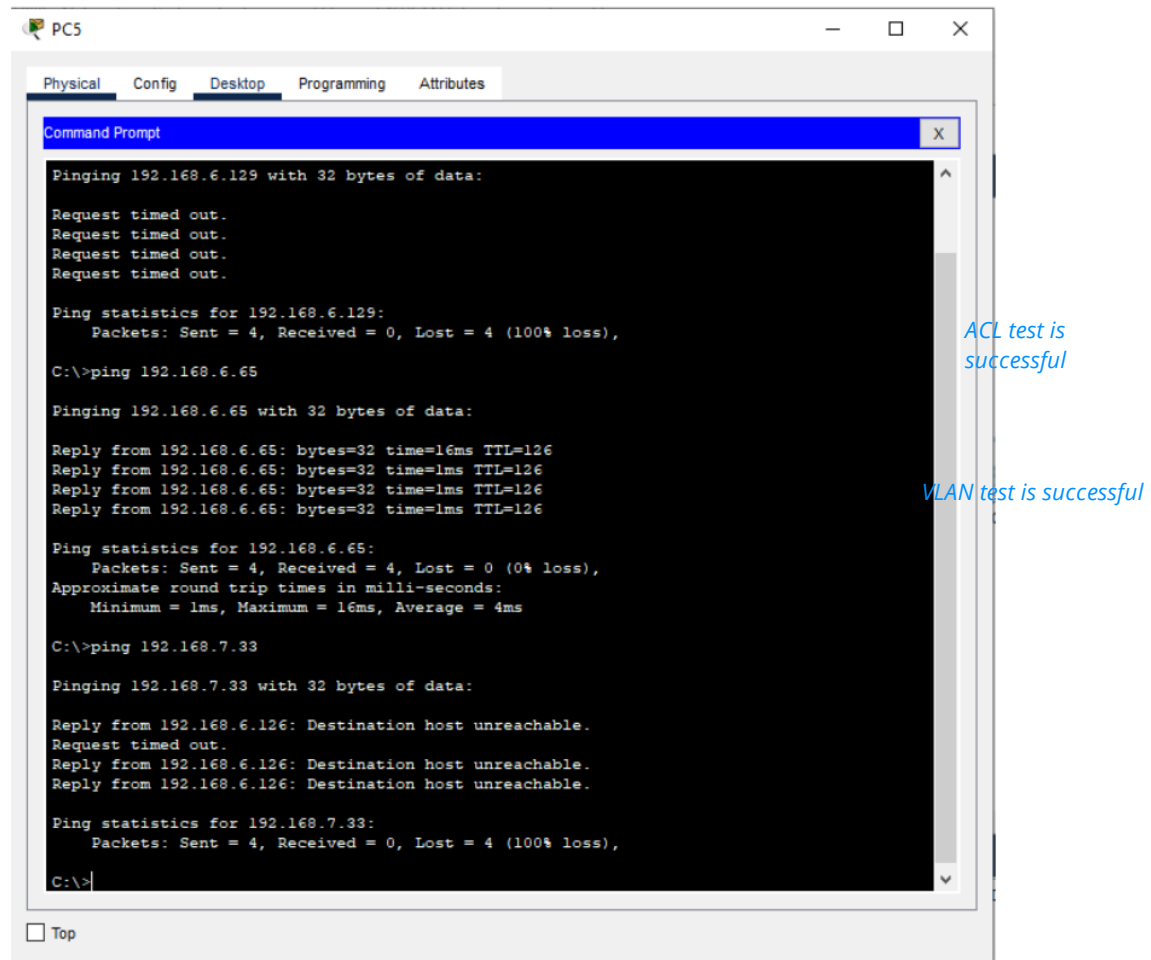


Figure 56 Activate the web server for the PLC

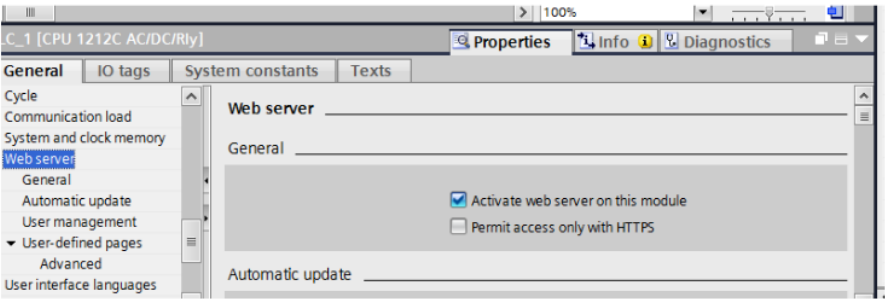


Figure 57 - Add the file extensions for the web program

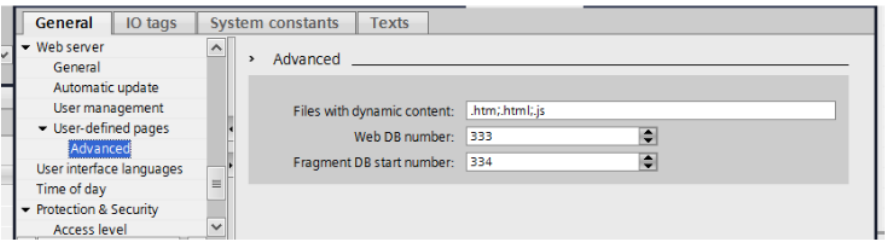


Figure 58 - Create the default tags for motor control and monitor

Default tag table

	Name	Data type	Address
	Motor Speed PWM	Word	%QW1000
	Motor RUN	Bool	%I0.4
	Motor PWM output	Bool	%Q1.1

PLC web server is activated

Figure 59 - Create the tags in the web data [DB1] data block

11		Motor_PWM	Int	0
3		web_ok	Int	

Figure 60 - The motor drive PWM function is in the main part of the PLC program

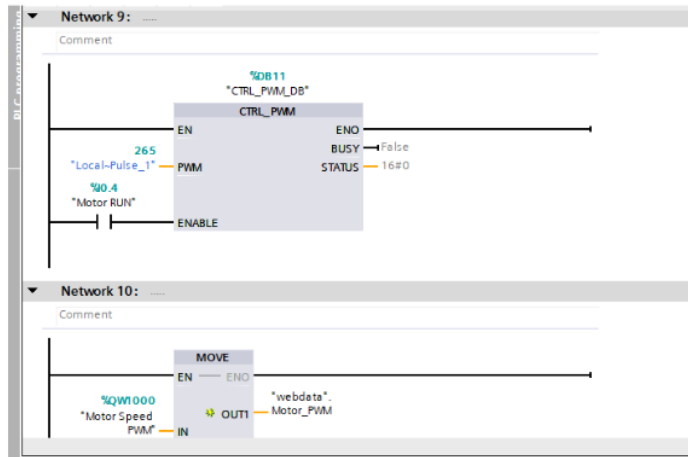


Figure 61 - In the main function block add the web server

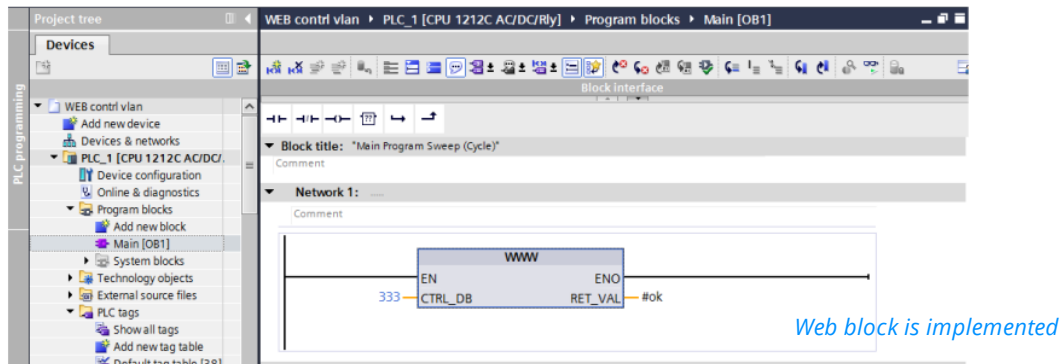


Figure 62 - Downloading the index .html page with http:// source for the Motor Switch.htm page

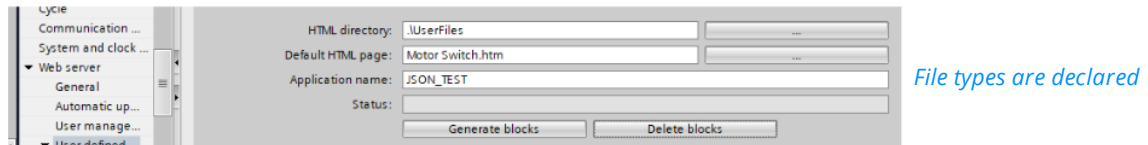
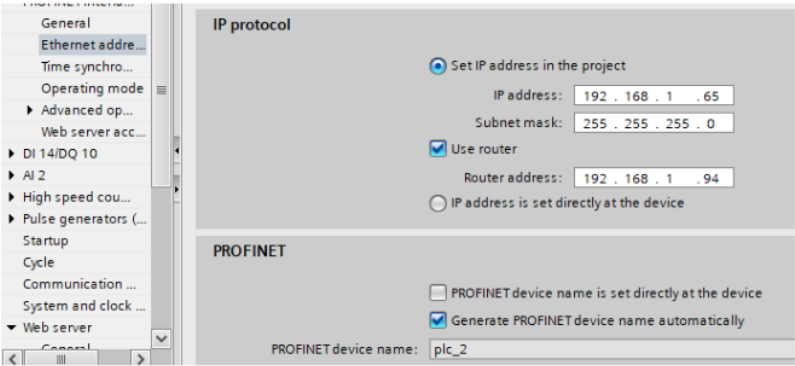


Figure 63 - Set the VLAN gateway IP address (router gateway) in the PLC



Address and gateway are configured

Figure 64 - The web pages held in 'User Files' of the PLC program files

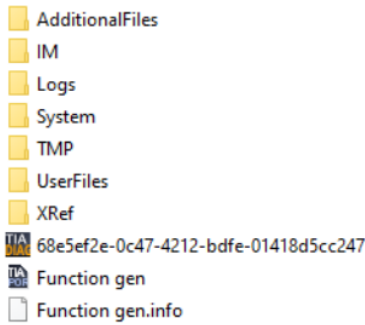


Figure 65 - Files stored in the 'User Files' of the TIA program files

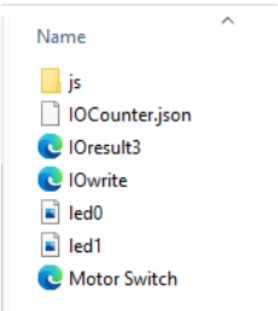
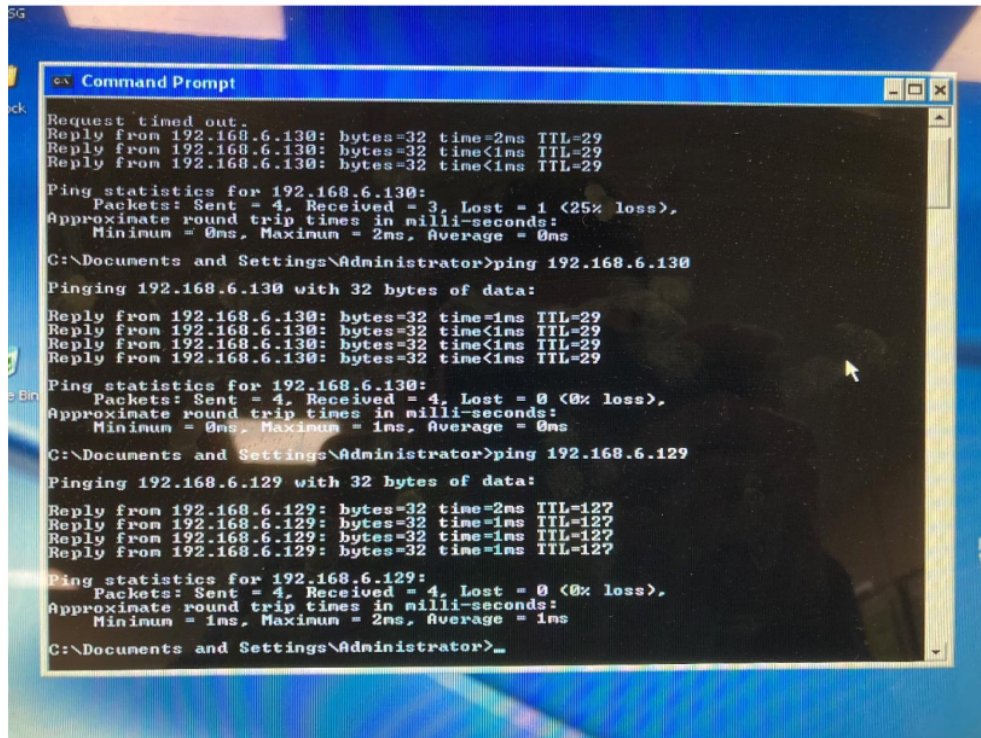


Figure 66 - Communication test of PC connected to the switch



```
Request timed out.
Reply from 192.168.6.130: bytes=32 time=2ms TTL=29
Reply from 192.168.6.130: bytes=32 time<1ms TTL=29
Reply from 192.168.6.130: bytes=32 time<1ms TTL=29
Ping statistics for 192.168.6.130:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 2ms, Average = 0ms
C:\Documents and Settings\Administrator>ping 192.168.6.130
Pinging 192.168.6.130 with 32 bytes of data:
Reply from 192.168.6.130: bytes=32 time=1ms TTL=29
Reply from 192.168.6.130: bytes=32 time<1ms TTL=29
Reply from 192.168.6.130: bytes=32 time<1ms TTL=29
Reply from 192.168.6.130: bytes=32 time<1ms TTL=29
Ping statistics for 192.168.6.130:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms
C:\Documents and Settings\Administrator>ping 192.168.6.129
Pinging 192.168.6.129 with 32 bytes of data:
Reply from 192.168.6.129: bytes=32 time=2ms TTL=127
Reply from 192.168.6.129: bytes=32 time=1ms TTL=127
Reply from 192.168.6.129: bytes=32 time=1ms TTL=127
Reply from 192.168.6.129: bytes=32 time=1ms TTL=127
Ping statistics for 192.168.6.129:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 2ms, Average = 1ms
C:\Documents and Settings\Administrator>
```

Figure 67 - Communication test of PC with PLC program

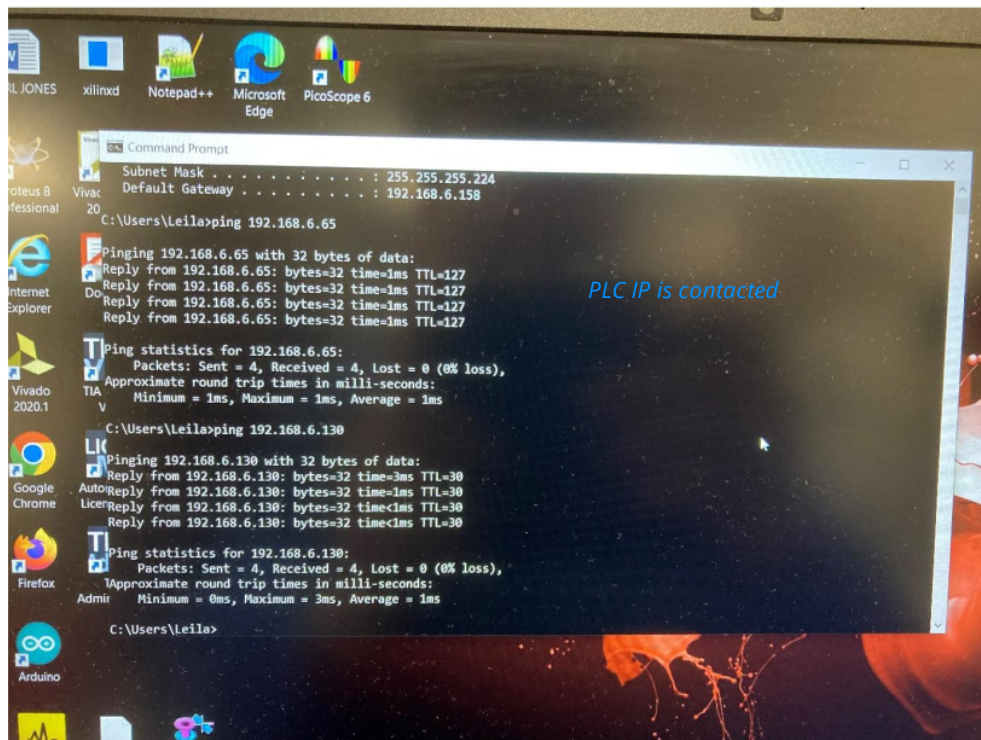


Figure 68 - Motor speed at 1%

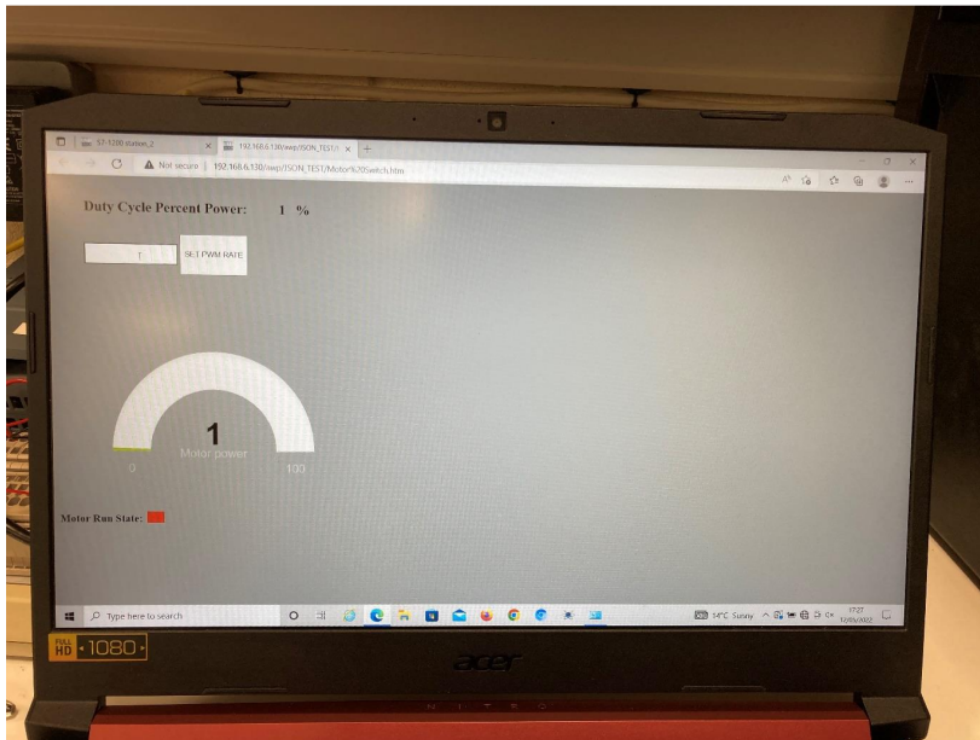


Figure 69 - Motor speed at 40%

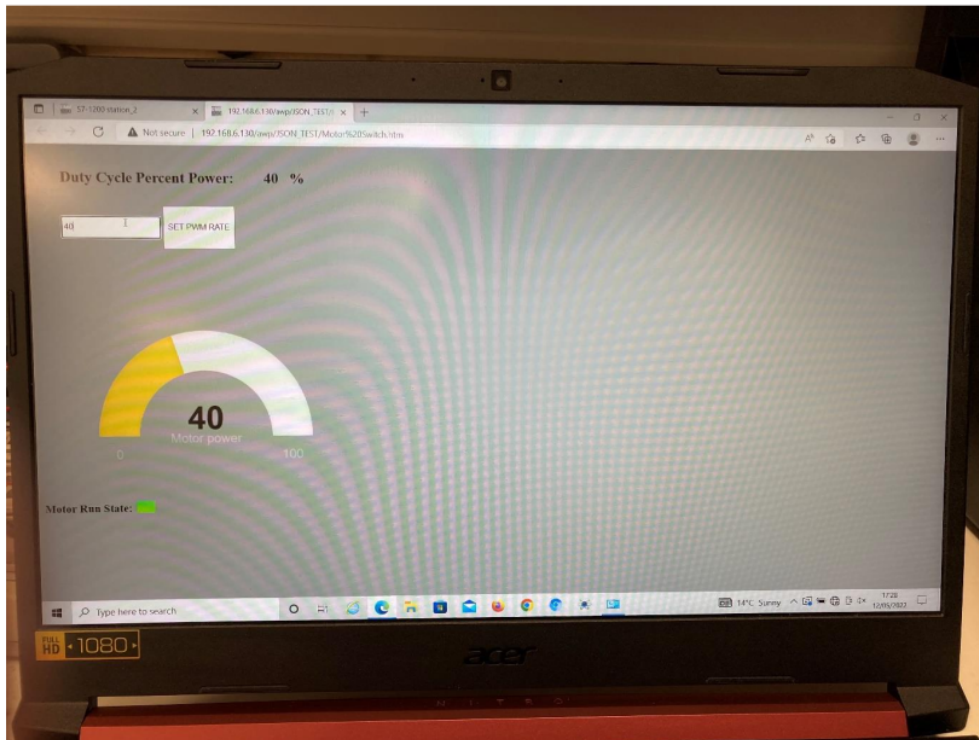
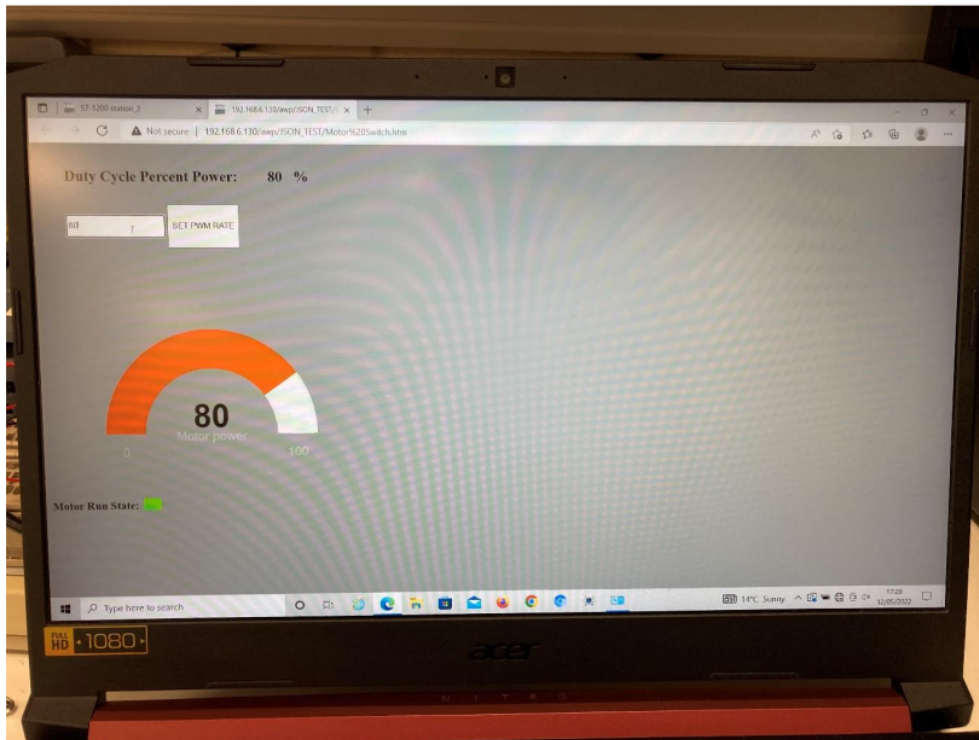


Figure 70 - Motor speed at 80%



Tests support evidence

Figure 71 - Motor Switch.htm program listing with comments

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4  <meta charset="utf-8">
5  <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
6  <meta name="viewport" content="user-scalable=yes, width=device-width, initial-scale=1.0, maximum-scale=1.0">
7
8
9  <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
10 <!--All the above code is standard and is given in all HTML program pages. This does not effect the program function
11
12 <!-- Aligns the guage graphic. gl is the dynamic variable for the guage. This is contained in a <style> tag-->
13 <style>
14     body {
15         text-align: center;
16
17     }
18
19     #gl {
20         width:500px; height:350px;
21         display: inline-block;
22         margin: 1em;
23         title: "Result 1";
24     }
25
26     p {
27         display: block;
28         width: 450px;
29         margin: 2em auto;
30         text-align: left;
31     }
32 </style>

```

Figure 72 - Motor Switch.htm program listing with comments

```

33
34
35 </head> <!--Nothing in the <head> </head> tags is displayed in the web page. Only the values between the <body> and </body> tags is displayed
36
37 <!--Everything inside the <body> tag is displayed on screen-->
38 <body bgcolor="#999999">
39
40 <!--table for a text heading and then read the duty cycle % value-->
41 <table style="height 100px" width="450px" font size="16" face="calibri" cellpadding="10">
42 <tr>
43     <th><h2>Duty Cycle Percent Power:</h2></th>
44     <td><h2><label id="Motor_PWM">0</label></h2></td>
45     <td><h2>%</h2></td>
46 </tr>
47
48 <!--this is the button for setting the duty cycle %-->
49 <!--This button modifies the value in the background using JavaScript without a postback-->
50 <tr>
51     <td><input id="setvar" type="text" style="width: 150px; height: 30px; background-color:white;color:black;"/>
52     <button style="height: 70px" width="70px"> SET PWM RATE </button></td>
53 </tr>
54 </table>
55
56 <!--declare .js sources which are given libraries-->
57 <script src="js/jquery-2.0.2.min.js"></script>
58 <script src="js/raphael.2.1.0.min.js"></script>
59 <script src="js/justgage.1.0.1.min.js"></script>
60

```

Figure 73 - Motor Switch.htm program listing with comments

```

60
61
62 <!--Give an id to the g1 dynamic guage variable contained inside a <div> tag -->
63 <div style="float: center">
64   <table>
65     <tr>
66       <td><div id="g1"></div></td>
67     </tr>
68   </table>
69 </div>
70
71 <!--Give id to the variables for reading the digital value for the MOTOR RUN state LED.
72 The img variable will hold the red or green LED images-->
73 <div>
74   <table>
75     <tr>
76       <th><h3>Motor Run State:</h3></th>
77       <td><label id="Motor RUN"></label></td>
78       <td><label id="img9"></label></td>
79     </tr>
80   </table>
81 </div>
82
83 </body>
84
85 <!--End of the <body> tag. Everything from here on are function call which will send data values
86 to all the ID tag names in the <body> tag-->
87

```

Figure 74 - Motor Switch.htm program listing with comments

```

87
88
89 <!-- This is the JSON function call for reading the Motor_PWM vcalue from the tag in the IOCounter.json file.
90 Script to refresh every 1000ms -->
91
92 <script type="text/javascript">
93   //All ajax and JSON is held in this document.
94   $(document).ready(function()
95   {
96
97     //ajax JSON to read the PLC Tag values
98     $.ajaxSetup({ cache: false });
99     setInterval(function()
100     {
101       $.getJSON("IOCounter.json", function(result)
102       {
103         $('#Motor_PWM').text(result["Motor_PWM"]);
104         $('#Motor RUN').text(result["Motor RUN"]);
105
106         //condition image RED / GREEN for 'Motor RUN' tag. Images are held in the User file in the PLC program directory
107         if(result["Motor RUN"]==0){
108
109           //change image to red if condition is off
110           $('#img9').html('');
111         }
112         else
113         {
114           //chnage image to green if condition is on
115           $('#img9').html('');
116         }
117       }
118     }
119   )
120 }

```

Figure 75 - Motor Switch.htm program listing with comments

```

117
118
119
120 -});
121 -},
122 //Refresh rate in ms.
123 1000); //this timing value is critical for the write to tag function
124 //If this is too small the write misses the first pole and takes too long to update.
125 //If this is too large the inputs will take too long to update. 1000ms is about right.
126
127 //END OF THE JSON FUNCTION CALL
128
129
130
131
132 <!--Initialise the gauge dynamic variables. Only g1, g2 and g3 used. g1 and g3 are dynamic variables form PLC Tags-->
133 var g1;
134
135 var g1 = new JustGage({
136     id: "g1",
137     value: getRandomInt(0, 100),
138     min: 0,
139     max: 100,
140     title: "Duty Cycle %",
141     label: "Motor power"
142 });
143
144

```

Figure 76 - Motor Switch.htm program listing with comments

```

144
145 //ajax call function for g1 dynamic variable which is linked to Motor_PWM tag in file IOresult3 in PLC directory/////
146 //call the ajax function. This can easily be converted to JSON if necessary
147 $.ajaxSetup({ cache: false });
148
149 setInterval(function() {
150     $.get("IOresult3.htm", function(result){
151         g1.refresh(parseInt(result));
152     });
153 }, 1000); //END OF THIS AJAX FUNCTION CALL
154
155
156
157
158
159
160 //modify the counter value from button. This is the part that writes to the PLC Tag variable i.e.Motor Speed PWM
161 //This is the function that writes the PWM values and sends using the button
162 $("#button").click(function(){
163     url="IOwrite.htm";
164     name="Motor Speed PWM";
165     val=$('#input[id=setvar]').val();
166     sdata=escape(name)+'='+val;
167     $.post(url,sdata,function(result){});
168     document.getElementById('setvar').value='';
169 });
170
171
172
173
174
175 </script>
176
177 </html>

```

References

Goralski, W., 2017. Well-Known Ports. In: *The Illustrated Network*. s.l.:s.n.

Net-Information, n.d. *Encapsulation*. [Online]

Available at: <http://net-informations.com/faq/oops/encapsulation.htm>

[Accessed 7 May 2022].

Shaw, K., 2022. *The OSI model explained and how to easily remember its 7 layers*. [Online]

Available at: [https://www.networkworld.com/article/3239677/the-osi-model-explained-and-how-to-easily-remember-its-7-](https://www.networkworld.com/article/3239677/the-osi-model-explained-and-how-to-easily-remember-its-7-layers.html#:~:text=The%207%20layers%20of%20the,Presentation%3B%20Layer%207%E2%80%9494A)

[layers.html#:~:text=The%207%20layers%20of%20the,Presentation%3B%20Layer%207%E2%80%9494A](https://www.networkworld.com/article/3239677/the-osi-model-explained-and-how-to-easily-remember-its-7-layers.html#:~:text=The%207%20layers%20of%20the,Presentation%3B%20Layer%207%E2%80%9494A)

[pplication.](https://www.networkworld.com/article/3239677/the-osi-model-explained-and-how-to-easily-remember-its-7-layers.html#:~:text=The%207%20layers%20of%20the,Presentation%3B%20Layer%207%E2%80%9494A)

[Accessed 7 May 2022].

Techopedia, 2014. *Broadcasting*. [Online]

Available at:

<https://www.techopedia.com/definition/6271/broadcasting#:~:text=Broadcasting%20is%20the%20simultaneous%20transmission,network%20is%20attacked%20by%20intruders.>

[Accessed 7 May 2022].

Limited number of sources

FINAL GRADE

GENERAL COMMENTS

Instructor

64/100

PAGE 1

PAGE 2

PAGE 3

Text Comment. Detailed table of contents

PAGE 4

PAGE 5

Text Comment. Extensive list of figures

PAGE 6

PAGE 7

Text Comment. Introduction can be more general

PAGE 8

Text Comment. Explanation includes significant points

PAGE 9

Text Comment. Decapsulation should be explained here also

PAGE 10

Text Comment. The broadcast is explained, However the problem that it possesses should have been explained.

PAGE 11

Text Comment. Good point identified

PAGE 12

Text Comment. Screen shots support this

Text Comment. Test results support explanation

PAGE 13

Text Comment. Good explanation

Text Comment. There should be some comments about the results here

PAGE 14

Text Comment. Good evaluation

PAGE 15

Text Comment. Intro proves the brief is understood

PAGE 16

Text Comment. The network diagram should have the net addresses indicated

PAGE 17

Text Comment. Subnet is identified

Text Comment. Table of addresses is correct

PAGE 18

Text Comment. Table of addresses is correct

PAGE 19

Text Comment. Table for address .48 is correct

Text Comment. Serial link addresses are correct

Text Comment. Table for address .80 is correct

Text Comment. Table for address .96 is correct

Text Comment. Images support explanation

Text Comment. Test images support explanation

Text Comment. Subnet is identified

Text Comment. Correct for .32 address

Text Comment. Correct for .64 address

Text Comment. Correct .96 address

Text Comment. Tables are correct for the /27 scheme

Text Comment. Subnet is identified

Text Comment. Correct for address .4

Text Comment. Correct for address .8

Text Comment. Correct for address .12

Text Comment. Table is correct for /30 scheme

Text Comment. This is a significant point

Text Comment. Results support the explanation

Text Comment. Good explanation

PAGE 25

Text Comment. Addresses are correct

Text Comment. Good table showing the inter VLAN routing ports

Text Comment. Good explanations

PAGE 26

Text Comment. test results support the explanation

Text Comment. Good evaluation

PAGE 27

Text Comment. Relevant intro

PAGE 28

Text Comment. Good point

Text Comment. The configuration of the Cisco hardware should be explained

PAGE 29

Text Comment. There is no explanation for the VLAN config for the Cisco hardware. There is too much focus on the PLC.

PAGE 30

Text Comment. The chart explains the program

PAGE 31

Text Comment. Tag data is identified

PAGE 32

PAGE 33

Text Comment. Functions are identified

PAGE 34

Text Comment. Test results support this

Text Comment. Good informative evaluation

PAGE 35

Text Comment. Appropriate image that supports the explanation

PAGE 36

Text Comment. Image is appropriate

PAGE 37

Text Comment. Good comparison image where the higher level layers are combined

PAGE 38

Text Comment. Image should show the relationship in the network system

PAGE 39

PAGE 40

PAGE 41

Text Comment. Images support implementation

PAGE 42

PAGE 43

PAGE 44

Text Comment. Screen shots support main report

PAGE 45

Text Comment. Test results support explanation in main report

Text Comment. RIP implementation

Text Comment. RIP tables support report

Text Comment. Reports confirm configuration

Text Comment. Test results show protocol is working

Text Comment. Image of DHCP config supports the report

Text Comment. Images support test results

Text Comment. Helper addresses are set

Text Comment. Problem is rectified

Text Comment. Excluded addresses are configured

PAGE 61

PAGE 62

PAGE 63

Text Comment. Tests proves exclusion is successful

PAGE 64

Text Comment. This is a repeat of DHCP config

PAGE 65

Text Comment. This should have the net addresses indicated

PAGE 66

Text Comment. Router tables are correct

PAGE 67

Text Comment. Test results support the main report

PAGE 68

PAGE 69

Text Comment. Routing tables auto update with new CIDR scheme

Text Comment. Test evidence supports the report

PAGE 70

Text Comment. Image does not include the VLANs

PAGE 71

Text Comment. VLANs are set

PAGE 72

Text Comment. This should be explained

Text Comment. Gateways are configured

Text Comment. Test across VLANs are successful

Text Comment. ACL test is successful

Text Comment. VLAN test is successful

Text Comment. PLC web server is activated

Text Comment. Web block is implemented

Text Comment. File types are declared

Text Comment. Address and gateway are configured

Text Comment. PLC IP is contacted

Text Comment. Tests support evidence

Text Comment. Limited number of sources

GRADING FORM: CSG-RUBIC

SEAN PARRY

64

RESEARCH-INFORMED LITERATURE



Research has parts missing, however relevant ref sources are included. 5/10 Sources are limited

5

KNOWLEDGE AND UNDERSTANDING



Element 1 is missing the address scheme table. Repeated evidence in the report. The implementation is successful but the evidence here is missing. 6/10 Element 2 diagram should indicate net addresses. Address tables are correct. VLAN diagram does include VLANs and Trunk explanation is missing. 6.5/10. Element 3 practical

19

VLAN config for Cisco hardware is not explained. The function calls have been explained briefly. The VLAN diagram does not indicate VLANs. 6.5/10

ANALYSIS



E1 test evidence is included and explained. The tests are successful and in a good order. Good evaluation. 3.5/5 E2 - The test results prove the operation of the network for all CIDR arrangements. The VLAN is developed from the final CIDR design and tests prove the configuration. ACL testing proved successful with evidence of the test outcomes. Limited evaluation. 7.5/10 E3 - Practical VLAN test evidence is provided and indicate the correct functioning of the network. The PLC functions are tested but with no evidence directly from the PLC for browser matching. A reasonable evaluation. 7/10

18

PRACTICAL APPLICATION AND DEP.



E1 - Implementation has been carried out, however address schema table evidence is missing. Testing procedure tested every device in every direction and proved successful. 2.5/5 E2 - Implementations for CIDR are successful. Network was further developed for VLAN configuration and ACL functions implemented. Some of the explanations could be more informative. 3.5/5 E3 - The practical network has been successfully implemented and tested. The PLC router test evidence to match browser is missing. VLAN diagram plan does not include VLANs and missing explanation. Trunking

12

method is not explained. The function calls have not been explained. 6/10

SKILLS FOR PROFESSIONAL PRAC.



The report follows a sequence. However diagrams in the appendix should be separated by elements. Difficult to follow from main report. 10/15

10