

1909365

by Matthew Davies

Submission date: 29-May-2022 11:52PM (UTC+0100)

Submission ID: 181975097

File name: 15177_Matthew_Davies_1909365_169044_1130830417.pdf (9.21M)

Word count: 4218

Character count: 30479

Course: HND Electrical and Electronic Engineering

Module: SGEE5008 Networking Systems for Engineers

Portfolio: Network Design for Industry

Name: Matthew Davies

University number: 1909365

Date submitted: 29/05/2022

Table of Contents

Table of Figures.....	4
1. Element 1.....	7
1.1 Research.....	7
1.1.1 OSI 7 Layer and 4 Layer models.....	7
1.1.2 Encapsulation	9
1.1.3 Broadcasting	10
1.1.4 Port Allocation.....	11
1.2 Routing.....	12
1.2.1 Network Address Configuration.....	12
1.2.2 Static Routing.....	14
1.2.3 RIP Routing.....	16
1.2.4 Dynamic Routing.....	21
1.3 Evaluation.....	28
2. Element 2.....	29
2.1 Introduction.....	29
2.2 Design Requirements	29
2.2.1 CIDR/28 Design Requirements	29
2.2.2 Implementation of CIDR/28 RIP	32
2.2.4 CIDR/27 and CIDR/30 Networking	35
2.2.5 Implementation of CIDR/27 and CIDR/30 RIP.....	36
2.3 Design Requirements for VLAN.....	38
2.3.1 VLAN for CIDR/27 sub network.....	38
2.4 ACL	42
2.5 Evaluation.....	46
3. Element 3.....	47
3.1 Router and Switch Configuration	47
3.2 Siemens PLC Web Server Configuration.....	51
3.3 Conclusions and Evaluation.....	59

Works Cited 60

Appendix..... 61

Appendix 2.1 61

Good table of contents

Table of Figures

Figure 1 - OSI 7 layer.....	7
Figure 2 - TCP/IP 4 Layer Model.....	9
Figure 3 - Encapsulation	10
Figure 4 - Broadcasting.....	10
Figure 5 - Port Allocation.....	11
Figure 6 - Network Setup Element 1	12
Figure 7 - Element 1 IP addresses.....	13
Figure 8 - Static Routing Router 0.....	14
Figure 9 - Static Routing Router 1.....	14
Figure 10 - Static Routing Router 2.....	15
Figure 11 - Static Routing Testing.....	15
Figure 12 - RIP routing and IP config Router 0.....	16
Figure 13 - RIP routing and IP config Router 1.....	16
Figure 14 - RIP Routing and IP config Router 2	17
Figure 15 - Router 0 RIP Routing Table	17
Figure 16 - Router 1 RIP Routing Table	18
Figure 17 - Router 2 RIP Routing Table	18
Figure 18 - Router 0 RIP Config IP Route	18
Figure 19 - Router 1 RIP Config IP Route	19
Figure 20 - Router 2 RIP Config IP Route	19
Figure 21 - RIP Routing Test Table.....	20
Figure 22 - Router 0 CLI for DHCP	21
Figure 23 - PC0 pre-helper address.....	21
Figure 24 - PC1 pre-helper address.....	22
Figure 25 - PC2 pre-helper address.....	22
Figure 26 - DHCP Router 1 Helper Address.....	22
Figure 27 - DHCP Router 2 Helper Address.....	23
Figure 28 - PC1 Post-Helper Addressing	23
Figure 29 - PC 2 Post-Helper Addressing	23
Figure 30 - Router0 running configuration	24
Figure 31 - Router0 running configuration	25
Figure 32 - Router 1 running configuration.....	25
Figure 33 - Router 1 running configuration.....	26
Figure 34 - Router 2 running configuration.....	27
Figure 35 - Router 2 running configuration.....	27

Figure 36 – CIDR Subnetting Network Topology.....	29
Figure 37 - Rome RIP Configuration Element 2	32
Figure 38 - Rome Routing Table CIDR/28.....	32
Figure 39 - Paris RIP Configuration Element 2.....	32
Figure 40 - Paris Routing Table CIDR/28	33
Figure 41 - London RIP Configuration Element 2	33
Figure 42 - London Routing Table CIDR/28	33
Figure 43 - CIDR/28 Testing RIP Routing	34
Figure 44 - Rome Routing Table CIDR/27 and CIDR/30	36
Figure 45 - Paris Routing Table CIDR/27 and CIDR/30.....	36
Figure 46 - London Routing Table CIDR/27 and CIDR/30.....	37
Figure 47 - CIDR/27 and CIDR/30 Testing.....	37
Figure 48 - London Switch VLAN Config.....	38
Figure 49 - London Switch VLAN Config (1).....	39
Figure 50 - London Switch VLAN Config (2).....	40
Figure 51 - London Router VLAN Config.....	40
Figure 52 - VLAN Routing Table	41
Figure 53 - VLAN Testing.....	41
Figure 54 - Paris Router ACL Config.....	42
Figure 55 - Paris Router ACL Config (1).....	42
Figure 56 - Paris Router ACL Config (2).....	43
Figure 57 - ACL Testing PC4 to London PCs.....	44
Figure 58 - ACL Testing PC5 to London PCs.....	45
Figure 59 - ACL Testing PC5 to Rome PCs	45
Figure 60 - Element 3 Network Topology	47
Figure 61 - VLAN Configuration	47
Figure 62 - VLAN Configuration (1).....	48
Figure 63 - VLAN Configuration (2).....	48
Figure 64 - Inter VLAN Router/Switch Config	49
Figure 65 - Inter VLAN Router Config	50
Figure 66 - Ping Testing	50
Figure 67 - Ping Testing (1).....	51
Figure 68 - Web Server Activation.....	51
Figure 69 - PLC File Extensions.....	52
Figure 70 - Motor Tags.....	52
Figure 71 - Webdata Tags	52
Figure 72 - Motor PWM Main	52

Figure 73 - Web Server in the Main Block.....	53
Figure 74 - html index	53
Figure 75 - userfiles folder	53
Figure 76 - IOwrite file.....	53
Figure 77 - IOresult3 file.....	54
Figure 78 - IOCounter file.....	54
Figure 79 - HTML MotorSwitch	54
Figure 80 - HTML MotorSwitch (1).....	55
Figure 81 - HTML MotorSwitch (2).....	55
Figure 82 - HTML MotorSwitch (3).....	56
Figure 83 - HTML MotorSwitch (4).....	56
Figure 84 - HTML MotorSwitch (5).....	57
Figure 85 - WebServer HMI Testing.....	57
Figure 86 - WebServer HMI Testing (1).....	58
Figure 87 - Watch table for motor OFF.....	58
Figure 88 - Watch table for motor ON	58
Figure 89 - London Show Run CIDR/28 (1).....	61
Figure 90 - London Show Run CIDR/28 (2).....	61
Figure 91 - Paris Show Run CIDR/28 (1)	62
Figure 92 - Paris Show Run CIDR/28 (2)	62
Figure 93 - Rome Show Run CIDR/28 (1).....	63
Figure 94 - Rome Show Run CIDR/28 (2).....	63

Extensive list of figures

1. Element 1

1.1 Research

1.1.1 OSI 7 Layer and 4 Layer models

To ensure national and worldwide data communication, a standard must be used to maintain the compatibility of communication between devices and networks. The standard which has been developed and is now used, is called international organization of standardization, which is also known as ISO. It is called a model for open system interconnection (OSI), so it is therefore referred to as the OSI model. (studytonight, 2020) The OSI model is a 7-layer model, where these layers are:

Layer	Name of Protocol	Name of Unit exchanged
Application	Application Protocol	APDU - Application Protocol Data Unit
Presentation	Presentation Protocol	PPDU - Presentation Protocol Data Unit
Session	Session Protocol	SPDU - Session Protocol Data Unit
Transport	Transport Protocol	TPDU - Transport Protocol Data Unit
Network	Network layer host-router Protocol	Packet
Data Link	Data link layer host-router Protocol	Frame
Physical	Physical layer host-router Protocol	Bit

Good table that supports explanation

Figure 1 - OSI 7 layer

The function of each layer is explained below:

Layer 1 – Physical Layer

This layer activates, deactivates, and maintains the physical connection. It sends and receives the data, as well as defining the voltages and rates at which the data transmits and receives in the physical layer.

Layer 2 – Data Link

Error free data transmission from one node to another is the main purpose of this layer. It recognises network protocols, does error checking, and synchronises frames; and Media Access Control (MAC).

Layer 3 – Network

The network layer serves two primary purposes. One method is to divide segments into network packets and then reassemble the packets on the receiver. The alternative method is to route packets by determining the optimum path over a physical network. To send data to a destination node, the network layer employs network addresses.

Layer 4 – Transport

This layer reassembles the segments on the receiving end and turns it into data that the session layer can use. It controls the rate at which the data is sent to ensure it matches the connection speed from the receiving device. Also, if data wasn't received correctly, then it will request it again.

Layer 5 – Session

Management of opening and closing sessions of data transfer, also carrying out checkpoints during the transfers, so in the case of loss of transfer, then it can re-start from the latest checkpoint, as opposed to starting over again.

Layer 6 – Presentation

Prepares the data for the application layer by transforming the data into a form which is acceptable for the application layer.

Layer 7 – Application

The services provided by this layer are things like mail, directory, network resource etc. It transfers files to the end user and holds the applications to act on the data it has received and/or transmitted.

(imperva, 2020)

Good and accurate explanation

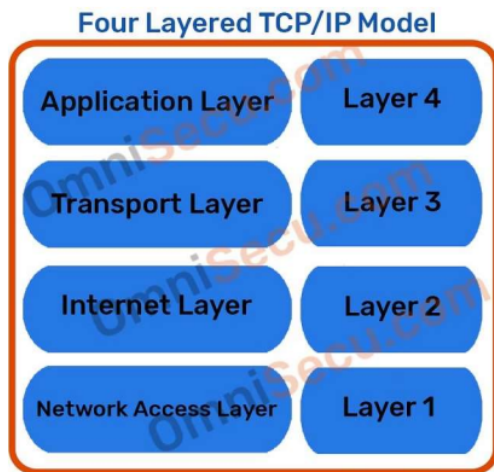


Image supports explanation

Figure 2 - TCP/IP 4 Layer Model

The above image shows the 4 different layers for the TCP/IP protocol. Below is an explanation of the role of each layer.

Layer 1 – Network Access

This describes how bits are electrically or optically indicated by hardware devices that interface directly with a network media, such as coaxial cable, optical fibre, or twisted pair copper wire, and how data is physically transmitted via the network.

Layer 3 – Transport

Main protocols used in this layer are the Transmission Control Protocol (TCP) and User Datagram Protocol (UDP). The function of this layer is to allow devices to carry on communicating with each other. It defines the connection status and the service level when transmitting and receiving data.

Layer 4 – Application

This can be described as the combined high level layer

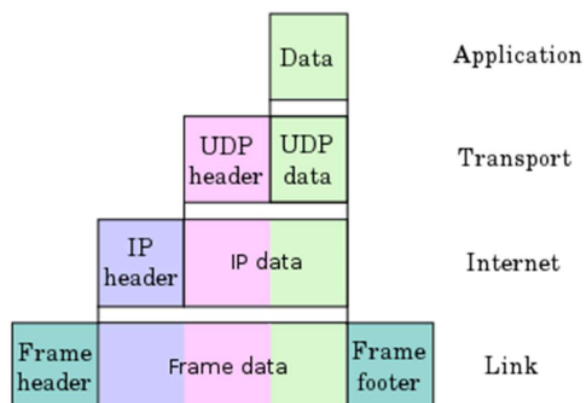
This layer defines how host programs interface with the network to transmit and receive the required data. It uses a variety of high-level protocols, such as DNS, HTTP, DHCP, Telnet, SMTP etc. (OmniSecu, 2020)

1.1.2 Encapsulation

This is a fundamental of Object-Oriented Programming (OOP), it bundles data and the methods working on the data in a defined unit/class. The idea behind it is to provide read or

This explanation is not accurate

write access to allow specific information to be hidden, if so desired, and to control the access.
(Janssen, 2022)

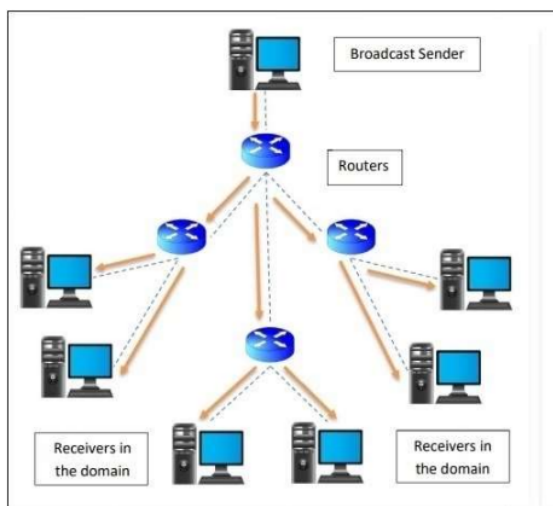


The image is correct but the explanation does not support this

Figure 3 - Encapsulation

1.1.3 Broadcasting

This is used for sending information to a high number of devices that sit on the same LAN, or potentially across multiple networks. It is an 'all-to-all' model where every device on the network will receive the same information. It can be explained clearly in the image below.



Good supporting image

Figure 4 - Broadcasting

It ensures faster delivery of information as all devices are getting it, so it minimises the processing overheads. However, it can't accommodate a very high number of devices, and it

Explanation is correct, however the domain can be any size but is undesirable. Smaller domains are more reliable and faster

won't allow personalisation on messaging, as all devices will receive the same message. (Moumita, 2019)

1.1.4 Port Allocation

The port numbers can vary from 0 to 65535, however only port numbers 0 to 1023 are classified as well-known ports and are kept for specific services. The server process's contact port is specified by the following list of well-known port numbers.

Some well-known ports are HTTP – port 80, FTP – port 20 and DNS – port 53.

The port is an endpoint to the communication, in computer networking terms. However, in software terms, the port number is a logical construct that identifies itself with a specific type of network service. The ports are identified for each protocol and the addresses are a combination of 16-bit unsigned number, which refer to as the port number. TCP and UDP are the protocols that most commonly use the port numbers. (bytesofgigabytes, 2020)

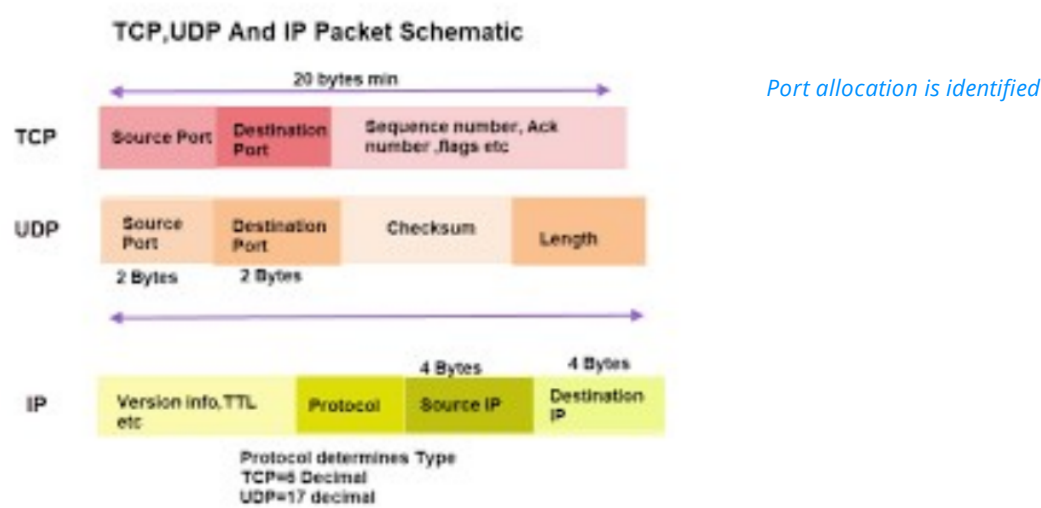


Figure 5 - Port Allocation

1.2 Routing

1.2.1 Network Address Configuration

The network was set up with 3 routers, 3 network switches and 3 PC's. Each router can be seen to be connected via serial links and the PCs are connected to the routers via the network switches using the Local Area Network (LAN).

The IP addresses for the network can be seen in the figure below, showing the 3 different network classes (A, B and C).

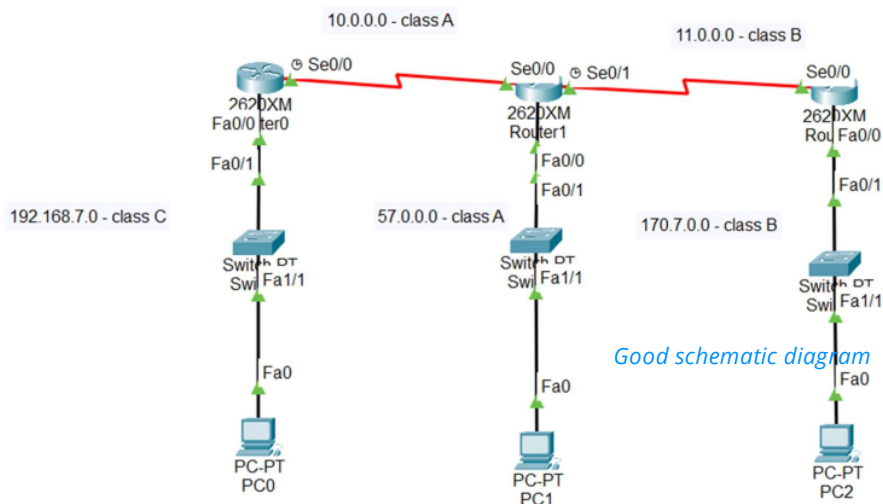


Figure 6 - Network Setup Element 1

The IP addresses for the network can be seen below, showing the network addresses, gateway addresses and the device addresses. All 3 classes of IP addressing have been used, as can be seen.

Class C:

192.168.7.0 – Network Address

192.168.7.254 – Gateway Address

192.168.7.1 – Initial Device Address

Serial Addresses:

10.0.0.0 – Network Address

10.0.0.1 – Serial0/0 Router 0

11.0.0.1 – Serial 0/1 Router 1

Class B:

170.7.0.0 – Network Address

170.7.254 – Gateway Address

170.7.0.1 – Initial Device Address

Serial Addresses:

11.0.0.0 – Network Address

10.0.0.2 – Serial0/0 Router 1

11.0.0.2 – Serial 0/0 Router 2

Class A:

57.0.0.0 – Network Address

57.0.0.254 – Gateway Address

57.0.0.1 – Initial Device Address

Address scheme is correct and can be used in the implementation

Figure 7 - Element 1 IP addresses

1.2.2 Static Routing

The images below show the static routing for all 3 routers.

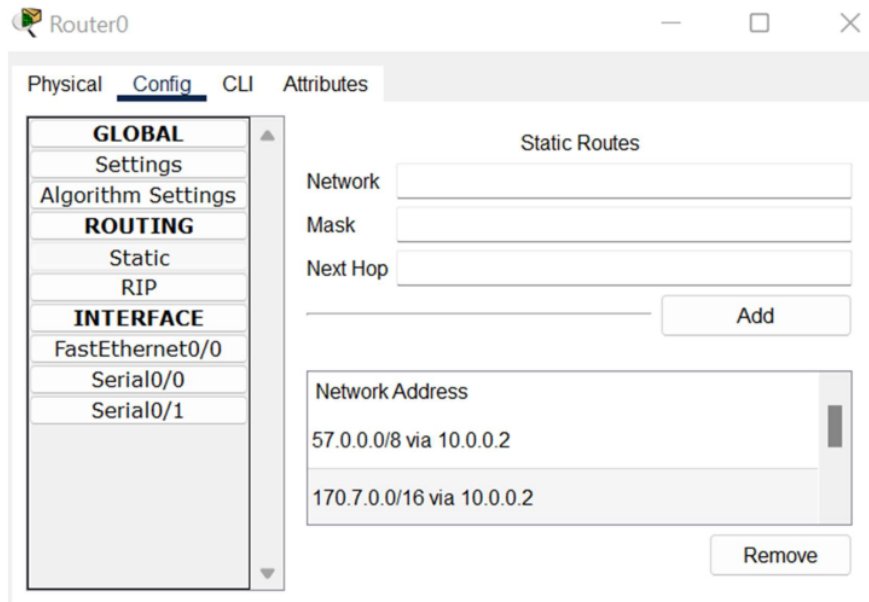


Figure 8 - Static Routing Router 0

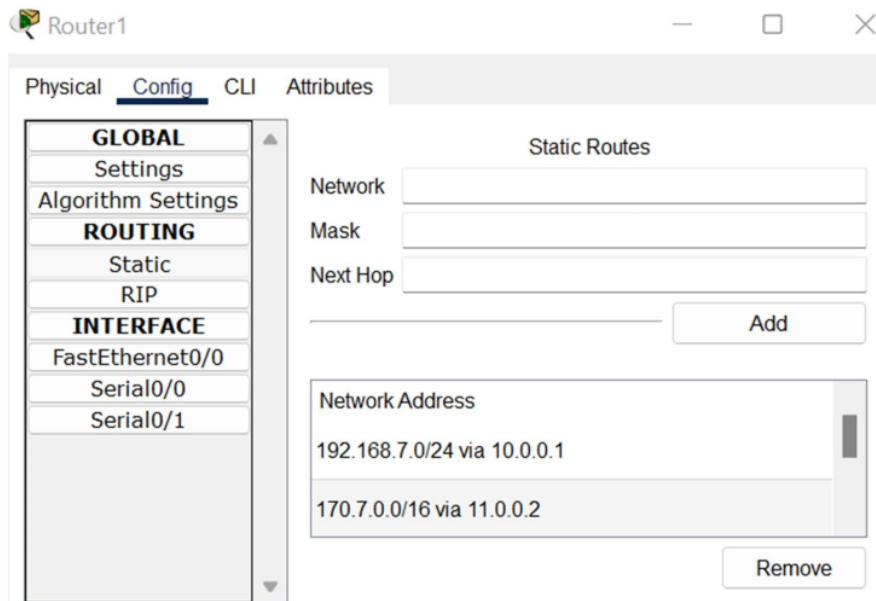


Figure 9 - Static Routing Router 1

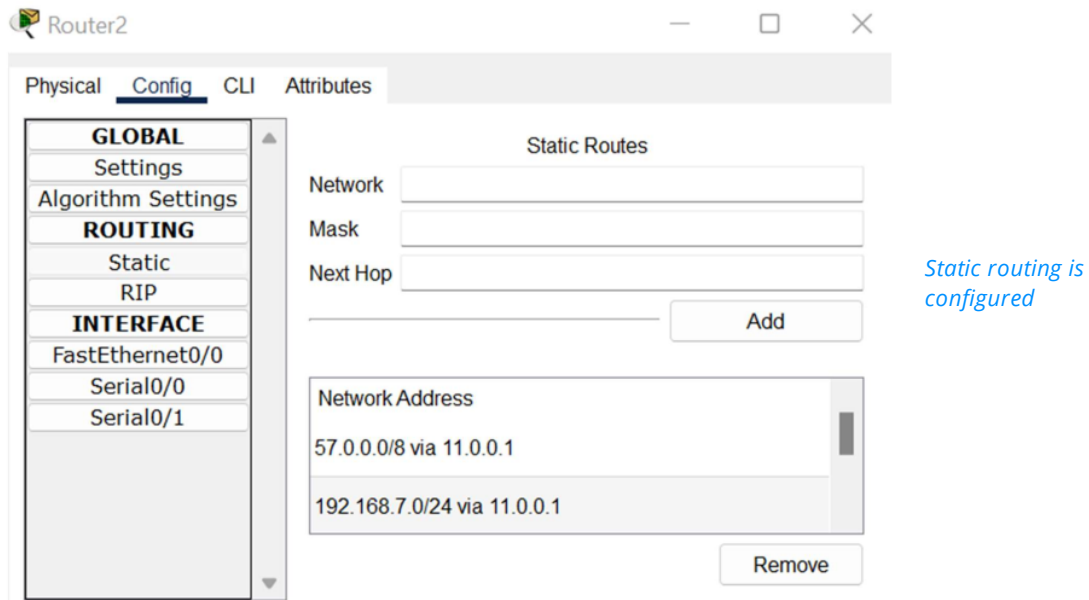


Figure 10 - Static Routing Router 2

Testing was carried out for the static routing by sending messages from each PC to each other. This tested that the routing was set up correctly in the routers to allow communication between the PCs. This can be seen below.

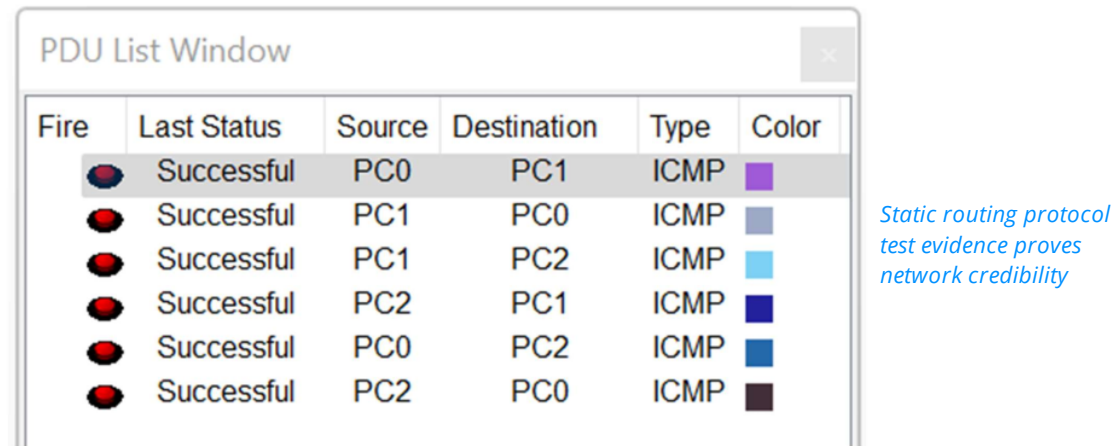
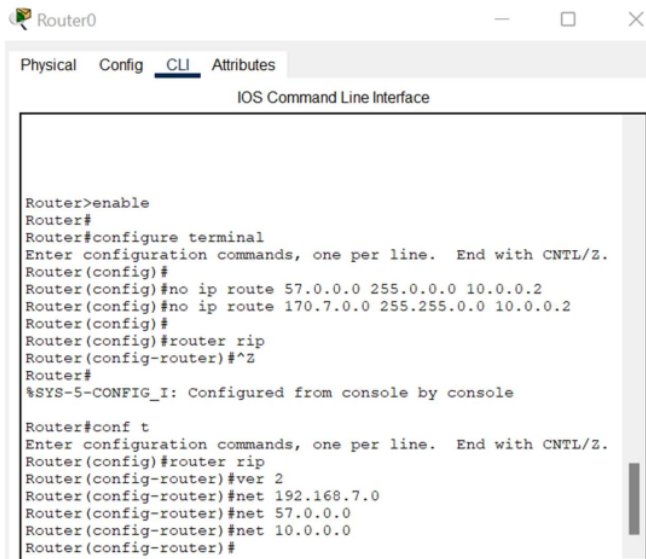


Figure 11 - Static Routing Testing

1.2.3 RIP Routing

The images below show the IP configuration and routing for each router.

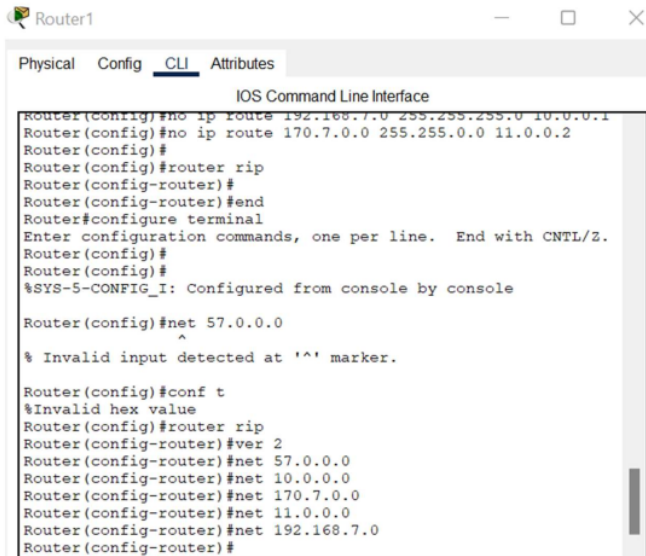


The screenshot shows the CLI of Router0. The tabs at the top are Physical, Config, CLI (selected), and Attributes. The title bar says 'Router0'. The main window is titled 'IOS Command Line Interface'. The command history is as follows:

```
Router>enable
Router#
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#
Router(config)#no ip route 57.0.0.0 255.0.0.0 10.0.0.2
Router(config)#no ip route 170.7.0.0 255.255.0.0 10.0.0.2
Router(config)#
Router(config)#router rip
Router(config-router)#^Z
Router#
%SYS-5-CONFIG_I: Configured from console by console

Router#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 192.168.7.0
Router(config-router)#net 57.0.0.0
Router(config-router)#net 10.0.0.0
Router(config-router)#
```

Figure 12 - RIP routing and IP config Router 0



The screenshot shows the CLI of Router1. The tabs at the top are Physical, Config, CLI (selected), and Attributes. The title bar says 'Router1'. The main window is titled 'IOS Command Line Interface'. The command history is as follows:

```
Router(config)#no ip route 192.168.7.0 255.255.255.0 10.0.0.1
Router(config)#no ip route 170.7.0.0 255.255.0.0 11.0.0.2
Router(config)#
Router(config)#router rip
Router(config-router)#
Router(config-router)#end
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#
Router(config)#
%SYS-5-CONFIG_I: Configured from console by console

Router(config)#net 57.0.0.0
^
% Invalid input detected at '^' marker.

Router(config)#conf t
%Invalid hex value
Router(config)#router rip
Router(config-router)#ver 2
Router(config-router)#net 57.0.0.0
Router(config-router)#net 10.0.0.0
Router(config-router)#net 170.7.0.0
Router(config-router)#net 11.0.0.0
Router(config-router)#net 192.168.7.0
Router(config-router)#
```

Figure 13 - RIP routing and IP config Router 1

```

Router2
Physical Config CLI Attributes
IOS Command Line Interface

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

Router>router rip
^
% Invalid input detected at '^' marker.

Router>en
Router#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 170.7.0.0
Router(config-router)#net 11.0.0.0
Router(config-router)#net 57.0.0.0
Router(config-router)#

```

Dynamic routing is configured for the addresses in the schematic

Figure 14 - RIP Routing and IP config Router 2

The routing tables for each router can be seen below. The 'R' type shows the RIP routing configuration, and the IP addresses used for their next hops.

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	10.0.0.2 120/1	
R	57.0.0.0/8	Serial0/0	10.0.0.2 120/1	
R	170.7.0.0/16	Serial0/0	10.0.0.2 120/2	
C	192.168.7.0/24	FastEthernet0/0	---	0/0

Figure 15 - Router 0 RIP Routing Table

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
C	57.0.0.0/8	FastEthernet0/0	---	0/0
R	170.7.0.0/16	Serial0/1	11.0.0.2	120/1
R	192.168.7.0/24	Serial0/0	10.0.0.1	120/1

Figure 16 - Router 1 RIP Routing Table

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
R	57.0.0.0/8	Serial0/0	11.0.0.1	120/1
C	170.7.0.0/16	FastEthernet0/0	---	0/0
R	192.168.7.0/24	Serial0/0	11.0.0.1	120/2

Reports show the connected addresses and routed

Figure 17 - Router 2 RIP Routing Table

The following images show the RIP configuration for each router, which shows their IP route.

```
Router>en
Router#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
R    11.0.0.0/8 [120/1] via 10.0.0.2, 00:00:08, Serial0/0
R    57.0.0.0/8 [120/1] via 10.0.0.2, 00:00:08, Serial0/0
R    170.7.0.0/16 [120/2] via 10.0.0.2, 00:00:08, Serial0/0
C    192.168.7.0/24 is directly connected, FastEthernet0/0
```

Figure 18 - Router 0 RIP Config IP Route

```

Router>en
Router#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
C    57.0.0.0/8 is directly connected, FastEthernet0/0
R    170.7.0.0/16 [120/1] via 11.0.0.2, 00:00:04, Serial0/1
R    192.168.7.0/24 [120/1] via 10.0.0.1, 00:00:03, Serial0/0

```

Figure 19 - Router 1 RIP Config IP Route

```

Router>en
Router#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

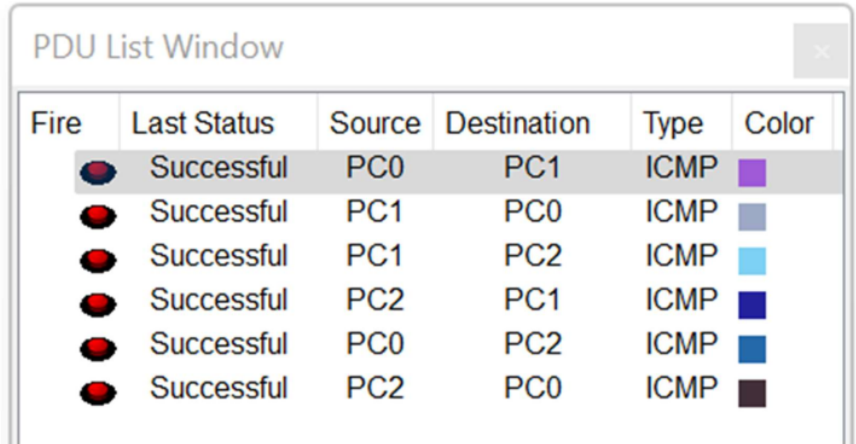
R    10.0.0.0/8 [120/1] via 11.0.0.1, 00:00:18, Serial0/0
C    11.0.0.0/8 is directly connected, Serial0/0
R    57.0.0.0/8 [120/1] via 11.0.0.1, 00:00:18, Serial0/0
C    170.7.0.0/16 is directly connected, FastEthernet0/0
R    192.168.7.0/24 [120/2] via 11.0.0.1, 00:00:18, Serial0/0

```

*Router reports confirm
configuration*

Figure 20 - Router 2 RIP Config IP Route

Testing was carried out to check that the RIP routing was configured correctly, allowing communication between the PCs on the different routers. The table below shows that it was working correctly, by proving the ping commands between the PCs were successfully communicating.



Fire	Last Status	Source	Destination	Type	Color
	Successful	PC0	PC1	ICMP	
	Successful	PC1	PC0	ICMP	
	Successful	PC1	PC2	ICMP	
	Successful	PC2	PC1	ICMP	
	Successful	PC0	PC2	ICMP	
	Successful	PC2	PC0	ICMP	

Figure 21 - RIP Routing Test Table

RIP routing test evidence supports network functionality

1.2.4 Dynamic Routing

The 3 routers were then configured for dynamic routing using DHCP. Router 0 was configured to be the control router, and routers 1 & 2 were configured without helper addresses for dynamic addressing with DHCP.

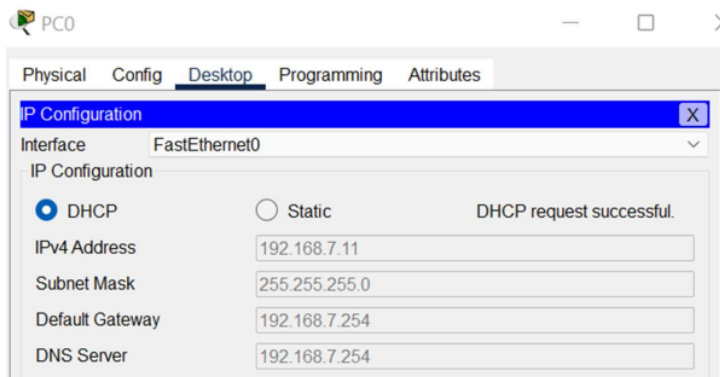
The Command Line Interface (CLI) for Router 0 can be seen below.

```
Router(config)#ip dhcp pool net1
Router(dhcp-config)#net 192.168.7.0 255.255.255.0
Router(dhcp-config)#default-router 192.168.7.254
Router(dhcp-config)#dns-server 192.168.7.254
Router(dhcp-config)#exit
Router(config)#
Router(config)#ip dhcp pool net2
Router(dhcp-config)#net 100.0.0.0 255.0.0.0
Router(dhcp-config)#net 57.0.0.0 255.0.0.0
Router(dhcp-config)#default-router 57.0.0.254
Router(dhcp-config)#dns-server 57.0.0.254
Router(dhcp-config)#exit
Router(config)#
Router(config)#ip dhcp pool net3
Router(dhcp-config)#net 170.7.0.0 255.255.0.0
Router(dhcp-config)#default-router 170.7.0.254
Router(dhcp-config)#dns-server 170.7.0.254
Router(dhcp-config)#exit
Router(config)#
```

DHCP is configured

Figure 22 - Router 0 CLI for DHCP

Before the helper addresses were configured to routers 1 & 2, the DHCP IP address configuration should be successful for PC0, and it should have failed for PC1 and PC2. This is proven in the images below.



The address here should be .1 not .11

Figure 23 - PC0 pre-helper address

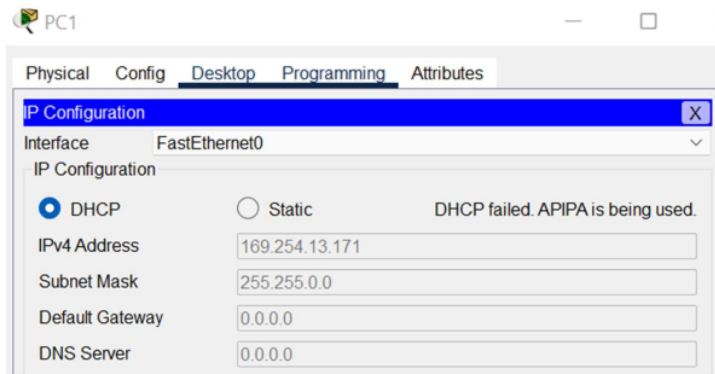
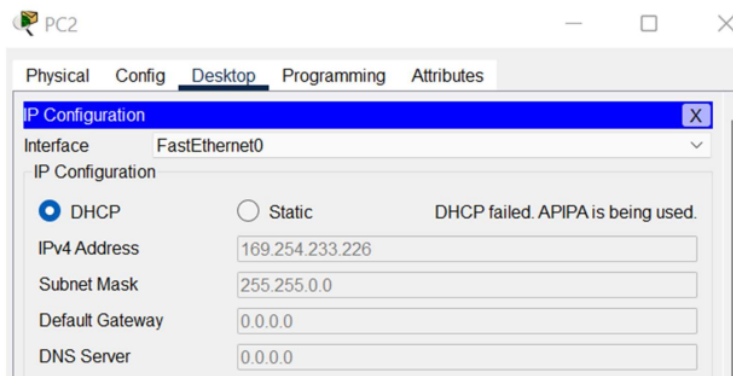


Figure 24 - PC1 pre-helper address



Test prior to helper addresses configurations indicate the failure

Figure 25 - PC2 pre-helper address

A default address was set for PC1 and PC2 as the router needs a helper address to communicate across the network.

The helper address was then added to the routers, as seen below.

```
Router>en
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#int f0/0
Router(config-if)#ip helper-address 10.0.0.1
Router(config-if)#
Router(config-if)#exit
Router(config)#
```

Figure 26 - DHCP Router 1 Helper Address

```

Router>
Router>en
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#int f0/0
Router(config-if)#ip helper-address 10.0.0.1
Router(config-if)#
Router(config-if)#exit
Router(config)#

```

Helper addresses are configured

Figure 27 - DHCP Router 2 Helper Address

Once these helper addresses were added, PC 1 & 2 were now able to have dynamic addressing.

☒ DHCP	☐ Static	DHCP request successful.
IPv4 Address	57.0.0.11	
Subnet Mask	255.0.0.0	
Default Gateway	57.0.0.254	
DNS Server	57.0.0.254	

Figure 28 - PC1 Post-Helper Addressing

PC2

Physical Config **Desktop** Programming Attributes

IP Configuration [X]

Interface FastEthernet0

IP Configuration

☒ DHCP	☐ Static	DHCP request successful.
IPv4 Address	170.7.0.11	
Subnet Mask	255.255.0.0	
Default Gateway	170.7.0.254	
DNS Server	170.7.0.254	

The test proves the helper addresses solve the problem.

Figure 29 - PC 2 Post-Helper Addressing

The exclusion configuration is missing

The following images show the running configuration of each router after the helper addresses and the IP DHCP excluded addresses were added.

```
Router#show run
Building configuration...

Current configuration : 1166 bytes
!
version 12.2
no service timestamps log datetime msec
no service timestamps debug datetime msec
no service password-encryption
!
hostname Router
!
!
!
!
ip dhcp excluded-address 192.168.7.0
ip dhcp excluded-address 192.168.7.1 192.168.7.10
ip dhcp excluded-address 170.7.0.1 170.7.0.10
ip dhcp excluded-address 192.168.7.10
ip dhcp excluded-address 57.0.0.1 57.0.0.10
!
ip dhcp pool net1
 network 192.168.7.0 255.255.255.0
 default-router 192.168.7.254
 dns-server 192.168.7.254
ip dhcp pool net2
 network 57.0.0.0 255.0.0.0
 default-router 57.0.0.254
 dns-server 57.0.0.254
ip dhcp pool net3
 network 170.7.0.0 255.255.0.0
 default-router 170.7.0.254
 dns-server 170.7.0.254
```

Exclusion addresses are set here but this is missing in the config section. The DHCP should be tested with and without exclusion addresses

Figure 30 - Router0 running configuration

```

interface FastEthernet0/0
ip address 192.168.7.254 255.255.255.0
duplex auto
speed auto
!
interface Serial0/0
ip address 10.0.0.1 255.0.0.0
clock rate 64000
!
interface Serial0/1
no ip address
clock rate 2000000
shutdown
!
router rip
version 2
network 10.0.0.0
network 57.0.0.0
network 192.168.7.0
!
ip classless
!
ip flow-export version 9
.

line con 0
!
line aux 0
!
line vty 0 4
login
!
!
!
end

```

Figure 31 - Router0 running configuration

```

Router#show run
Building configuration...

Current configuration : 689 bytes
!
version 12.2
no service timestamps log datetime msec
no service timestamps debug datetime msec
no service password-encryption
!
hostname Router
!
!
!
!
!
!
!
no ip cef
no ipv6 cef

```

Figure 32 - Router 1 running configuration

```

interface FastEthernet0/0
 ip address 57.0.0.254 255.0.0.0
 ip helper-address 10.0.0.1
 duplex auto
 speed auto
!
interface Serial0/0
 ip address 10.0.0.2 255.0.0.0
!
interface Serial0/1
 ip address 11.0.0.1 255.0.0.0
 clock rate 64000
!
router rip
 version 2
 network 10.0.0.0
 network 11.0.0.0
 network 57.0.0.0
 network 170.7.0.0
 network 192.168.7.0
!
ip classless
!
ip flow-export version 9

line con 0
!
line aux 0
!
line vty 0 4
 login
!
!
!
end

```

Figure 33 - Router 1 running configuration

```

Router#show run
Building configuration...

Current configuration : 884 bytes
!
version 12.2
no service timestamps log datetime msec
no service timestamps debug datetime msec
no service password-encryption
!
hostname Router
!
!
!
!
!
ip dhcp pool net1
  network 192.168.7.0 255.255.255.0
  default-router 192.168.7.254
  dns-server 192.168.7.254
ip dhcp pool net2
  network 57.0.0.0 255.0.0.0
  default-router 57.0.0.254
  dns-server 57.0.0.254
!
!
!
no ip cef
no ipv6 cef

```

Figure 34 - Router 2 running configuration

```

interface FastEthernet0/0
  ip address 170.7.0.254 255.255.0.0
  ip helper-address 10.10.10.1
  ip helper-address 10.0.0.1
  duplex auto
  speed auto
!
interface Serial0/0
  ip address 11.0.0.2 255.0.0.0
!
interface Serial0/1
  no ip address
  clock rate 2000000
  shutdown
!
router rip
  version 2
  network 11.0.0.0
  network 57.0.0.0
  network 170.7.0.0
!
ip classless
!
ip flow-export version 9

line con 0
!
line aux 0
!
line vty 0 4
  login
!
!
!
end

```

Router report show the configuration for each router

Figure 35 - Router 2 running configuration

1.3 Evaluation

The different methods of routing and IP address classes was explored in this element. This explored the simple routing method, which was static routing, and it ended with DHCP routing, which was the most complex to implement initially, however it was then far easier to use when adding devices as the IP address was being assigned to them automatically. It also allowed for excluding certain addresses if it was desired.

Evaluation is only brief

The tables below show the addressing scheme used for the CIDR/28 subnetting for the LANs on this system.

Subnet mask is identified

CIDR/28 – London – Network 3 – LAN Address 16											
Subnet mask	1	1	1	1	0	0	0	0	255.255.255.240	/28	
	0	0	0	0	0	0	0	0	Not used		
Net Address 16	0	0	0	1	0	0	0	1	Lowest host address 17		192.168.7.17
Net Address 16	0	0	0	1	0	0	1	0	Next host address 18		192.168.7.18
Default gateway	0	0	0	1	1	1	1	0	FastEthernet 0/0 address 30		192.168.7.30
	0	0	0	1	1	1	1	1	Broadcast 31		

Table 1 - CIDR/28 London LAN Addressing

Correct address config

CIDR/28 – Rome – Network 1 – LAN Address 32											
Subnet mask	1	1	1	1	0	0	0	0	255.255.255.240	/28	
	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		192.168.7.33
Net Address 32	0	0	1	0	0	0	1	0	Next host address 34		192.168.7.34
Default gateway	0	0	1	0	1	1	1	0	FastEthernet 0/0 address 46		192.168.7.46
	0	0	1	0	1	1	1	1	Broadcast 47		

Table 2 - CIDR/28 Rome LAN Addressing

Table is correct for net address 32

CIDR/28 – Paris – Network 2 – LAN Address 48											
Subnet mask	1	1	1	1	0	0	0	0	255.255.255.240	/28	
	0	0	0	0	0	0	0	0	Not used		
Net Address 48	0	0	1	1	0	0	0	1	Lowest host address 49		192.168.7.49
Net Address 48	0	0	1	1	0	0	1	0	Next host address 50		192.168.7.50
Default gateway	0	0	1	1	1	1	1	0	FastEthernet 0/0 address 62		192.168.7.62
	0	0	1	1	1	1	1	1	Broadcast 63		

Table 3 - CIDR/28 Paris LAN Addressing

Correct for address 48

The tables below show the addressing scheme used for the CIDR/28 subnetting for the WANs on this system.

London – Rome Network 6 Network address 64										
London Router	0	1	0	0	1	1	0	1	Serial 0/0 address 77	192.168.7.77
Rome Router	0	1	0	0	1	1	1	0	Serial 0/1 address 78	192.168.7.78

Correct for address 64

Table 4 - CIDR/28 London - Rome WAN Addressing

Rome – Paris Network 4-Network address 80										
Paris Router	0	1	0	1	1	1	0	1	Serial 0/0 address 93	192.168.7.93
Rome Router	0	1	0	1	1	1	1	0	Serial 0/0 address 94	192.168.7.94

Correct for address 80

Table 5 - CIDR/28 Rome - Paris WAN Addressing

Paris –London Network 5 Network address 96										
Paris Router	0	1	1	0	1	1	0	1	Serial 0/1 address 109	192.168.7.109
London Router	0	1	1	0	1	1	1	0	Serial 0/1 address 110	192.168.7.110

Correct for address 96

Table 6 - CIDR/28 Paris - London WAN Addressing

Tables are correct and follow a logical design

2.2.2 Implementation of CIDR/28 RIP

Below, the images of the routing tables and RIP Routing configuration for each router using CIDR/28 can be seen.

```
Rome(config)#router rip
Rome(config-router)#ver 2
Rome(config-router)#net 192.168.7.32
Rome(config-router)#net 192.168.7.64
Rome(config-router)#net 192.168.7.80
Rome(config-router)#exit
```

Figure 37 - Rome RIP Configuration Element 2

Routing Table for Rome

Type	Network	Port	Next Hop IP	Metric
R	192.168.7.16/28	Serial0/1	192.168.7.77	120/1
C	192.168.7.32/28	FastEthernet0/0	---	0/0
R	192.168.7.48/28	Serial0/0	192.168.7.93	120/1
C	192.168.7.64/28	Serial0/1	---	0/0
C	192.168.7.80/28	Serial0/0	---	0/0
R	192.168.7.96/28	Serial0/0	192.168.7.93	120/1
R	192.168.7.96/28	Serial0/1	192.168.7.77	120/1

Figure 38 - Rome Routing Table CIDR/28

```
Paris(config)#router rip
Paris(config-router)#ver 2
Paris(config-router)#net 192.168.7.48
Paris(config-router)#net 192.168.7.96
Paris(config-router)#net 192.168.7.80
Paris(config-router)#exit
Paris(config)#
Paris(config)#end
```

Figure 39 - Paris RIP Configuration Element 2

Routing Table for Paris

Type	Network	Port	Next Hop IP	Metric
R	192.168.7.16/28	Serial0/1	192.168.7.110	120/1
R	192.168.7.32/28	Serial0/0	192.168.7.94	120/1
C	192.168.7.48/28	FastEthernet0/0	---	0/0
R	192.168.7.64/28	Serial0/0	192.168.7.94	120/1
R	192.168.7.64/28	Serial0/1	192.168.7.110	120/1
C	192.168.7.80/28	Serial0/0	---	0/0
C	192.168.7.96/28	Serial0/1	---	0/0

Figure 40 - Paris Routing Table CIDR/28

```

London(config)#router rip
London(config-router)#ver 2
London(config-router)#net 192.168.7.16
London(config-router)#net 192.168.7.64
London(config-router)#net 192.168.7.96
London(config-router)#exit
London(config)#
London(config)#end

```

Figure 41 - London RIP Configuration Element 2

Routing Table for London

Type	Network	Port	Next Hop IP	Metric
C	192.168.7.16/28	FastEthernet0/0	---	0/0
R	192.168.7.32/28	Serial0/0	192.168.7.78	120/1
R	192.168.7.48/28	Serial0/1	192.168.7.109	120/1
C	192.168.7.64/28	Serial0/0	---	0/0
R	192.168.7.80/28	Serial0/1	192.168.7.109	120/1
R	192.168.7.80/28	Serial0/0	192.168.7.78	120/1
C	192.168.7.96/28	Serial0/1	---	0/0

Dynamic routing is implemented

Figure 42 - London Routing Table CIDR/28

The routing tables prove that they have been configured correctly for the CIDR/28 as each network address ends in /28. This proves that the router identified that CIDR/28 was in use due to the subnet used.

The table below proves that the RIP configuration was correct as all PCs on the different networks were able to communicate successfully with each other using the messaging tool on Cisco Packet Tracer. This is a simulation of what a 'ping' command would do on a PC.

PDU List Window									
Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit
	Successful	PC0	PC2	ICMP		0.000	N	0	(edit)
	Successful	PC3	PC0	ICMP		0.000	N	1	(edit)
	Successful	PC3	PC4	ICMP		0.000	N	2	(edit)
	Successful	PC5	PC2	ICMP		0.000	N	3	(edit)
	Successful	PC1	PC4	ICMP		0.000	N	4	(edit)
	Successful	PC5	PC0	ICMP		0.000	N	5	(edit)

Figure 43 - CIDR/28 Testing RIP Routing

Tests prove the configuration

The 'show run' command for each router was used, and the results can be seen in Appendix 2.1.

2.2.4 CIDR/27 and CIDR/30 Networking

CIDR/27 is used for the LANs on the system, and CIDR/30 is used for the Wide Area Network (WAN). The addresses used can be seen in the tables below.

Subnet mask 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	192.168.7.33
Net Address 32	0	0	1	0	0	0	1	0	Next Host Address 34 Rome 2	192.168.7.34
Net Address 32	0	0	1	1	1	1	1	0	Highest Router Address 62 Rome Router	192.168.7.62
Net Address 64	0	1	0	0	0	0	0	1	Lowest Host Address 65 London 1	192.168.7.65
Net Address 64	0	1	0	0	0	0	1	0	Next Host Address 66 London 2	192.168.7.66
Net Address 64	0	1	0	1	1	1	1	0	Highest Router Address 94 London Router	192.168.7.94
Net Address 96	0	1	1	0	0	0	0	1	Lowest Host Address 97 Paris 1	192.168.7.97
Net Address 96	0	1	1	0	0	0	1	0	Next Host Address 98 Paris 2	192.168.7.98
Net Address 96	0	1	1	1	1	1	1	0	Highest Router Address 126 Paris Router	192.168.7.126

Correct table of addresses for /27

Table 7 - CIDR/27 Addressing

CIDR/30 is used for WANs as there are several networks used, but less devices, which suits the CIDR/30 better than the CIDR/27.

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	192.168.7.5
Net Address 4	0	0	0	0	0	1	1	0	Highest interface Address 6 Paris serial 0/0	192.168.7.6
Net Address 8	0	0	0	0	1	0	0	1	Lowest interface Address 9 Rome serial 0/1	192.168.7.9
Net Address 8	0	0	0	0	1	0	1	0	Highest interface Address 10 London serial 0/0	192.168.7.10
Net Address 12	0	0	0	0	1	1	0	1	Lowest interface Address 13 London serial 0/1	192.168.7.13
Net Address 12	0	0	0	0	1	1	1	0	Highest interface Address 14 Paris serial 0/1	192.168.7.14

Table is correct for /30

Table 8 - CIDR/30 Addressing

2.2.5 Implementation of CIDR/27 and CIDR/30 RIP

The RIP routing configuration was the same for this task as was used for the CIDR/28, however the CIDR class changes were detected automatically from the subnet mask used.

The Routing Tables for each router prove that this is correct. *Good significant point highlighted*

Routing Table for Rome				
Type	Network	Port	Next Hop IP	Metric
C	192.168.7.4/30	Serial0/0	---	0/0
C	192.168.7.8/30	Serial0/1	---	0/0
R	192.168.7.12/30	Serial0/0	192.168.7.6	120/1
R	192.168.7.12/30	Serial0/1	192.168.7.10	120/1
C	192.168.7.32/27	FastEthernet0/0	---	0/0
R	192.168.7.64/27	Serial0/1	192.168.7.10	120/1
R	192.168.7.96/27	Serial0/0	192.168.7.6	120/1

Figure 44 - Rome Routing Table CIDR/27 and CIDR/30

Routing Table for Paris				
Type	Network	Port	Next Hop IP	Metric
C	192.168.7.4/30	Serial0/0	---	0/0
R	192.168.7.8/30	Serial0/0	192.168.7.5	120/1
R	192.168.7.8/30	Serial0/1	192.168.7.13	120/1
C	192.168.7.12/30	Serial0/1	---	0/0
R	192.168.7.32/27	Serial0/0	192.168.7.5	120/1
R	192.168.7.64/27	Serial0/1	192.168.7.13	120/1
C	192.168.7.96/27	FastEthernet0/0	---	0/0

Figure 45 - Paris Routing Table CIDR/27 and CIDR/30

Type	Network	Port	Next Hop IP	Metric
R	192.168.7.4/30	Serial0/0	192.168.7.9	120/1
R	192.168.7.4/30	Serial0/1	192.168.7.14	120/1
C	192.168.7.8/30	Serial0/0	---	0/0
C	192.168.7.12/30	Serial0/1	---	0/0
R	192.168.7.32/27	Serial0/0	192.168.7.9	120/1
C	192.168.7.64/27	FastEthernet0/0	---	0/0
R	192.168.7.96/27	Serial0/1	192.168.7.14	120/1

Tables are correct for /27 and /30

Figure 46 - London Routing Table CIDR/27 and CIDR/30

The testing table below shows that the communication between devices on each LAN was successful.

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num
	Successful	PC0	PC2	ICMP		0.000	N	0
	Successful	PC0	PC4	ICMP		0.000	N	1
	Successful	PC5	PC3	ICMP		0.000	N	2
	Successful	PC2	PC4	ICMP		0.000	N	3
	Successful	PC4	PC1	ICMP		0.000	N	4
	Successful	PC5	PC2	ICMP		0.000	N	5

Figure 47 - CIDR/27 and CIDR/30 Testing

Tests prove the /27 and /30 configurations

2.3 Design Requirements for VLAN

2.3.1 VLAN for CIDR/27 sub network

Virtual Local Area Networks (VLANs) are used for the transmitting and receiving of data packets across devices connected to the LAN. A VLAN is several networks that communicate with each other, just like in a LAN, but the VLAN is in one or several different segments of a LAN. It is a form of separating certain areas of a network but keeping them on the same LAN. They are separated using a router or a switch, where all devices on that LAN can receive information from each device, but none outside of the VLAN. It is used to prevent needless traffic on the network.

The switch is the main component for VLAN. The router is only for trunk protocol config

The switch configuration can be seen below.

```
London#show vlan
```

VLAN Name Status Ports

```
-----
1 default active Fa0/1, Fa0/2, Fa0/3, Fa0/4
Fa0/5, Fa0/6, Fa0/7, Fa0/8
Fa0/9, Fa0/10, Fa0/11, Fa0/12
Fa0/13, Fa0/14, Fa0/15, Fa0/16
Fa0/17, Fa0/18, Fa0/19, Fa0/20
Fa0/21, Fa0/22, Fa0/23, Fa0/24
2 vlan2 active
3 vlan3 active
1002 fddi-default active
1003 token-ring-default active
1004 fddinet-default active
1005 trnet-default active
```

Report shows VLANs active but not configured

VLAN Type SAID MTU Parent RingNo BridgeNo Stp BrdgMode Trans1 Trans2

```
-----
1 enet 100001 1500 ----- 0 0
2 enet 100002 1500 ----- 0 0
3 enet 100003 1500 ----- 0 0
1002 fddi 101002 1500 ----- 0 0
1003 tr 101003 1500 ----- 0 0
1004 fdnet 101004 1500 --- ieee - 0 0
1005 trnet 101005 1500 --- ibm - 0 0
```

VLAN Type SAID MTU Parent RingNo BridgeNo Stp BrdgMode Trans1 Trans2

Remote SPAN VLANs

Primary Secondary Type Ports

Figure 48 - London Switch VLAN Config

```

London#conf t
Enter configuration commands, one per line. End with CNTL/Z.
London(config)#int f0/2
London(config-if)#switchport mode access
London(config-if)#switchport access vlan 2
London(config-if)#exit
London(config)#int f0/3
London(config-if)#switchport mode access
London(config-if)#switchport access vlan 3
London(config-if)#exit
London(config)#^Z
London#
%SYS-5-CONFIG I: Configured from console by console

London#show vlan

VLAN Name Status Ports
-----
1 default active Fa0/1, Fa0/4, Fa0/5, Fa0/6
  Fa0/7, Fa0/8, Fa0/9, Fa0/10
  Fa0/11, Fa0/12, Fa0/13, Fa0/14
  Fa0/15, Fa0/16, Fa0/17, Fa0/18
  Fa0/19, Fa0/20, Fa0/21, Fa0/22
  Fa0/23, Fa0/24
2 vlan2 active Fa0/2
3 vlan3 active Fa0/3
1002 fddi-default active
1003 token-ring-default active
1004 fddinet-default active
1005 trnet-default active

VLAN Type SAID MTU Parent RingNo BridgeNo Stp BrdgMode Trans1 Trans2
-----
1 enet 100001 1500 ---- 0 0
2 enet 100002 1500 ---- 0 0
3 enet 100003 1500 ---- 0 0
1002 fddi 101002 1500 ---- 0 0
1003 tr 101003 1500 ---- 0 0
1004 fdnet 101004 1500 --- ieee - 0 0
1005 trnet 101005 1500 --- ibm - 0 0

VLAN Type SAID MTU Parent RingNo BridgeNo Stp BrdgMode Trans1 Trans2
-----

Remote SPAN VLANs
-----

Primary Secondary Type Ports

```

VLANs configured and ports assigned

Ports assigned

Figure 49 - London Switch VLAN Config (1)

```

London#show interface trunk
Port Mode Encapsulation Status Native vlan
Fa0/24 on 802.1q trunking 1

Port Vlans allowed on trunk
Fa0/24 1-1005

Port Vlans allowed and active in management domain
Fa0/24 1,2,3

Port Vlans in spanning tree forwarding state and not pruned
Fa0/24 none

```

Figure 50 - London Switch VLAN Config (2)

The London router was then configured for VLAN. This is shown below.

```

London>en
London#conf t
Enter configuration commands, one per line. End with CNTL/Z.
London(config)#int f0/0
London(config-if)#no ip address
London(config-if)#exit
London(config)#int f0/0.2
London(config-subif)#
%LINK-5-CHANGED: Interface FastEthernet0/0.2, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0.2, changed state to
up

London(config-subif)#encapsulation dot1Q 2
London(config-subif)#ip address 192.168.7.94 255.255.255.224
London(config-subif)#no shutdown
London(config-subif)#exit
London(config)#
London(config)#int f0/0.3
London(config-subif)#
%LINK-5-CHANGED: Interface FastEthernet0/0.3, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0.3, changed state to
up

London(config-subif)#encapsulation dot1Q 3
London(config-subif)#ip address 192.168.7.158 255.255.255.224
London(config-subif)#no shutdown
London(config-subif)#exit
London(config)#Z
London#
%SYS-5-CONFIG_I: Configured from console by console

London#show ip int brie
Interface IP-Address OK? Method Status Protocol
FastEthernet0/0 unassigned YES manual up up
FastEthernet0/0.2 192.168.7.94 YES manual up up
FastEthernet0/0.3 192.168.7.158 YES manual up up
FastEthernet0/1 unassigned YES unset administratively down down
Serial0/0 192.168.7.10 YES manual up up
Serial0/1 192.168.7.13 YES manual up up

```

Two gateways are configured with protocol

Figure 51 - London Router VLAN Config

The routing table below shows the VLAN routing setup.

Routing Table for London

Type	Network	Port	Next Hop IP	Metric
R	192.168.7.4/30	Serial0/0	192.168.7.9	120/1
R	192.168.7.4/30	Serial0/1	192.168.7.14	120/1
C	192.168.7.8/30	Serial0/0	---	0/0
C	192.168.7.12/30	Serial0/1	---	0/0
R	192.168.7.32/27	Serial0/0	192.168.7.9	120/1
C	192.168.7.64/27	FastEthernet0/0.2	---	0/0
R	192.168.7.96/27	Serial0/1	192.168.7.14	120/1
C	192.168.7.128/27	FastEthernet0/0.3	---	0/0

Trunk config evidence is missing and no explanation for the trunk connection

Figure 52 - VLAN Routing Table

To test the VLAN network, messages were sent between PCs on the VLAN. This is proven to be successful below.

PDU List Window

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num
	Successful	PC2	PC3	ICMP		0.000	N	0
	Successful	PC2	PC4	ICMP		0.000	N	1
	Successful	PC2	PC1	ICMP		0.000	N	2
	Successful	PC3	PC4	ICMP		0.000	N	3
	Successful	PC3	PC0	ICMP		0.000	N	4

Test proves the configuration is working

Figure 53 - VLAN Testing

2.4 ACL

An Access Control List (ACL) was set up in the Paris router to prevent receiving signals from a VLAN network that was configured in one of the previous tasks. In this task, the Paris router is configured to restrict access from the London devices to the Paris PCs. The CLI configuration for the Paris router can be seen below, where it is denying access to 192.168.7.129.

```
Paris>en
Paris#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Paris(config)#hostname Paris
Paris(config)#access-list 1 ?
deny Specify packets to reject
permit Specify packets to forward
remark Access list entry comment
Paris(config)#access-list 1 deny 192.168.7.129 0.0.0.0
Paris(config)#access-list 1 permit any
Paris(config)#
Paris#
%SYS-5-CONFIG_I: Configured from console by console
```

ACL config is correct

Figure 54 - Paris Router ACL Config

```
Paris#show run
Building configuration...

Current configuration : 756 bytes
!
version 12.2
no service timestamps log datetime msec
no service timestamps debug datetime msec
no service password-encryption
!
hostname Paris
!
no ip cef
no ipv6 cef
!
interface FastEthernet0/0
ip address 192.168.7.126 255.255.255.224
duplex auto
speed auto
!
interface FastEthernet0/1
no ip address
duplex auto
speed auto
shutdown
!
interface Serial0/0
ip address 192.168.7.6 255.255.255.252
!
interface Serial0/1
ip address 192.168.7.14 255.255.255.252
clock rate 64000
!
router rip
```

Figure 55 - Paris Router ACL Config (1)

```

version 2
network 192.168.7.0
!
ip classless
!
ip flow-export version 9
!
access-list 1 deny host 192.168.7.129
access-list 1 permit any
!
line con 0
!
line aux 0
!
line vty 0 4
login
!
end
Paris#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Paris(config)#int f0/0
Paris(config-if)#ip access-group 1
% Incomplete command.
Paris(config-if)#ip access-group 1?
in inbound packets
out outbound packets
Paris(config-if)#ip access-group 1 out
Paris(config-if)#
Paris#
%SYS-5-CONFIG_I: Configured from console by console

```

Report confirms configuration

Figure 56 - Paris Router ACL Config (2)

Once the ACL was configured on the Paris router, testing was carried out to confirm that the VLANs that shouldn't have access were denied access, and that the rest were allowed access. The ping command was used to test this, as can be seen below.

Pinging from pc4 to London PC's

```
C:\>ping 192.168.7.129
```

Pinging 192.168.7.129 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 192.168.7.129:

Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

```
C:\>ping 192.168.7.65
```

Pinging 192.168.7.65 with 32 bytes of data:

Reply from 192.168.7.65: bytes=32 time=7ms TTL=126
Reply from 192.168.7.65: bytes=32 time=1ms TTL=126
Reply from 192.168.7.65: bytes=32 time=17ms TTL=126
Reply from 192.168.7.65: bytes=32 time=10ms TTL=126

Test proves ACL firewall

Ping statistics for 192.168.7.65:

Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),

Approximate round trip times in milli-seconds:

Minimum = 1ms, Maximum = 17ms, Average = 8ms

Figure 57 - ACL Testing PC4 to London PCs

Pinging from PC5 to London PC'S

C:\>ping 192.168.7.129

Pinging 192.168.7.129 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 192.168.7.129:

Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\>ping 192.168.7.65

Pinging 192.168.7.65 with 32 bytes of data:

Reply from 192.168.7.65: bytes=32 time=13ms TTL=126
Reply from 192.168.7.65: bytes=32 time=7ms TTL=126
Reply from 192.168.7.65: bytes=32 time=1ms TTL=126
Reply from 192.168.7.65: bytes=32 time=1ms TTL=126

Ping statistics for 192.168.7.65:

Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),

Approximate round trip times in milli-seconds:

Minimum = 1ms, Maximum = 13ms, Average = 5ms

Figure 58 - ACL Testing PC5 to London PCs

Pinging from PC5 to Rome PC:

C:\>ping 192.168.7.33

Pinging 192.168.7.33 with 32 bytes of data:

Reply from 192.168.7.33: bytes=32 time=13ms TTL=126
Reply from 192.168.7.33: bytes=32 time=13ms TTL=126
Reply from 192.168.7.33: bytes=32 time=10ms TTL=126
Reply from 192.168.7.33: bytes=32 time=10ms TTL=126

Tests show access and denial of service

Ping statistics for 192.168.7.33:

Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),

Approximate round trip times in milli-seconds:

Minimum = 10ms, Maximum = 13ms, Average = 11ms

Figure 59 - ACL Testing PC5 to Rome PCs

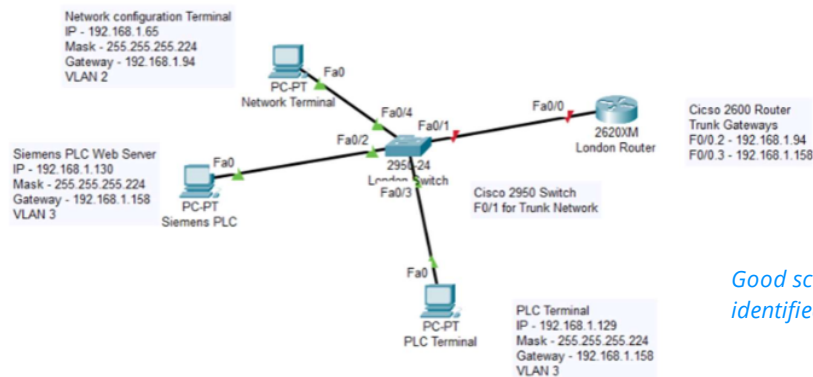
2.5 Evaluation

The understanding and implementation of CIDR, VLAN and ACL networking types have been proven in this element by building the networks on Cisco and then testing them using the ping commands to ensure the integrity of the network is correct as per specification.

Limited evaluation

3. Element 3

The network topology for this element can be seen below. This shows that a cisco router and a cisco switch was used, as well as the PC for configuring the network, a PC for PLC web server, and the PLC terminal.



Good schematic diagram with VLANs identified

Figure 60 - Element 3 Network Topology

3.1 Router and Switch Configuration

The VLAN configuration can be seen below.

```
Switch>enable
Switch#conf t
Switch(config)#vlan 2
Switch(config-vlan)#name vlan2
Switch(config-vlan)#exit
Switch(config)#vlan 3
Switch(config-vlan)#name vlan3
Switch(config-vlan)#exit
Switch(config)#^Z
Switch#show vlan
////////////////////////////////report on vlan config////////////////////////////////
Switch#conf t
Switch(config)#int f0/2
Switch(config-if)#switchport mode access
Switch(config-if)#switchport access vlan 3
Switch(config-if)#exit
```

Figure 61 - VLAN Configuration

```
Switch(config)#int f0/3
Switch(config-if)#switchport mode access
Switch(config-if)#switchport access vlan 3
Switch(config-if)#exit
```

```
Switch(config)#int f0/4
Switch(config-if)#switchport mode access
Switch(config-if)#switchport access vlan 2
Switch(config-if)#exit
Switch(config)#^Z
Switch#
```

Figure 62 - VLAN Configuration (1)

```
Switch#show vlan
//////////report shows vlans with port config//////////
End of VLAN configuration
```

Figure 63 - VLAN Configuration (2)

Inter VLAN routing

Switch config:

```
Switch#conf t
Switch(config)#int f0/1
Switch(config-if)#switchport mode trunk Trunk is set
Switch(config-if)#^Z
Switch#
Switch#show interface trunk
//////////report shows trunk config//////////
//////////might have to show this again after router config//////////
```

Router config:

```
Router>enable
Router#conf t
Router(config)#int f0/0
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#int f0/0
Router(config-if)#no ip address
Router(config-if)#exit

Router(config)#int f0/0.2
Router(config-subif)#encapsulation dot1Q 2
Router(config-subif)#ip address 192.168.1.94 255.255.255.224 Gateway configured
Router(config-subif)#no shutdown
```

Figure 64 - Inter VLAN Router/Switch Config

```
Router(config-subif)#exit
```

```
Router(config)#int f0/0.3
```

```
Router(config-subif)#encapsulation dot1Q 3
```

```
Router(config-subif)#ip address 192.168.1.158 255.255.255.224
```

Gateway configured

```
Router(config-subif)#no shutdown
```

```
Router(config-subif)#exit
```

```
Router(config)#^Z
```

```
Router#show ip interfaces brie
```

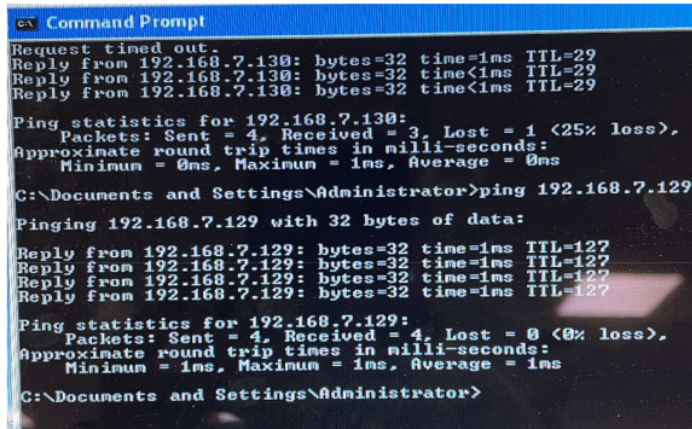
```
//////////report int status for f0/0.2 and f0/0.3//////////
```

```
Router#show run
```

```
//////////running config for router//////////
```

Figure 65 - Inter VLAN Router Config

Once the router and switch were configured, the physical connections between the devices were tested using the 'ping' function in the command prompt. Both images below prove that it was all correct.



```
c:\ Command Prompt
Request timed out.
Reply from 192.168.7.130: bytes=32 time=1ms TTL=29
Reply from 192.168.7.130: bytes=32 time<1ms TTL=29
Reply from 192.168.7.130: bytes=32 time<1ms TTL=29
Ping statistics for 192.168.7.130:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms
C:\Documents and Settings\Administrator>ping 192.168.7.129
Pinging 192.168.7.129 with 32 bytes of data:
Reply from 192.168.7.129: bytes=32 time=1ms TTL=127
Reply from 192.168.7.129: bytes=32 time=1ms TTL=127
Reply from 192.168.7.129: bytes=32 time=1ms TTL=127
Reply from 192.168.7.129: bytes=32 time=1ms TTL=127
Ping statistics for 192.168.7.129:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 1ms, Average = 1ms
C:\Documents and Settings\Administrator>
```

Test confirms config

Figure 66 - Ping Testing

```
Command Prompt
Ping statistics for 192.168.7.65:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),

C:\Users\Leila>ping 192.168.7.130

Pinging 192.168.7.130 with 32 bytes of data:
Reply from 192.168.7.130: bytes=32 time=3ms TTL=30
Reply from 192.168.7.130: bytes=32 time=1ms TTL=30
Reply from 192.168.7.130: bytes=32 time<1ms TTL=30
Reply from 192.168.7.130: bytes=32 time<1ms TTL=30

Ping statistics for 192.168.7.130:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 3ms, Average = 1ms

C:\Users\Leila>ping 192.168.7.65

Pinging 192.168.7.65 with 32 bytes of data:
Reply from 192.168.7.65: bytes=32 time=2ms TTL=127
Reply from 192.168.7.65: bytes=32 time<1ms TTL=127
Reply from 192.168.7.65: bytes=32 time<1ms TTL=127
Reply from 192.168.7.65: bytes=32 time<1ms TTL=127

Ping statistics for 192.168.7.65:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 2ms, Average = 0ms

C:\Users\Leila>
```

Tests proves comms across VLANs

Figure 67 - Ping Testing (1)

3.2 Siemens PLC Web Server Configuration

The web server had to be activated for the PLC on TIA Portal. This is shown below.

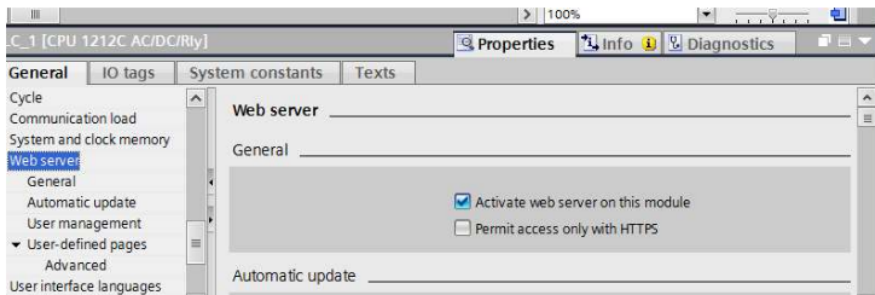


Figure 68 - Web Server Activation

The file extensions needed to be added next. The file extensions used were htm, html and js. Where HTM is an allowable short version of HTML, HTML is Hypertext Markup Language and js is javascript.

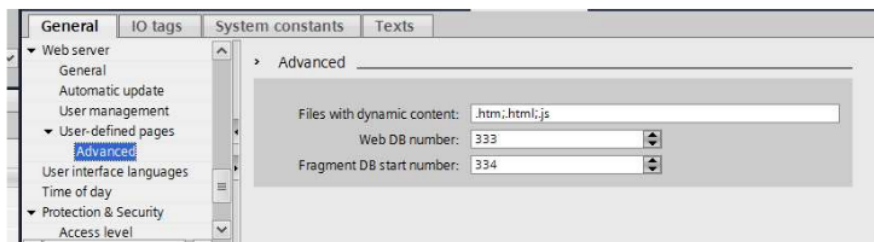


Figure 69 - PLC File Extensions

The tags were created for the motor control and monitoring.

Web server is activated and file extensions declared

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

Figure 70 - Motor Tags

Then the tags that would be used for the web server needed to be added in to the webdata data block.

11	Motor_PWM	Int	0
3	web_ok	Int	

Figure 71 - Webdata Tags

The motor PWM function block was added in to the 'main' block.

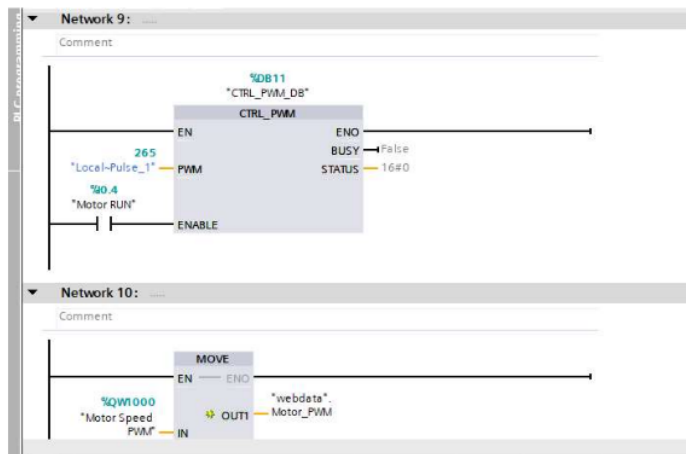


Figure 72 - Motor PWM Main

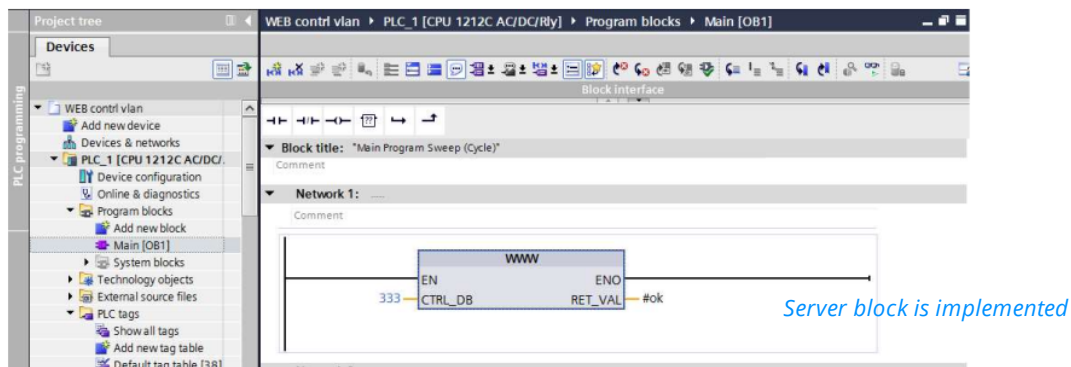


Figure 73 - Web Server in the Main Block

The index .html page with the http:// source for the index page was downloaded to the PLC.

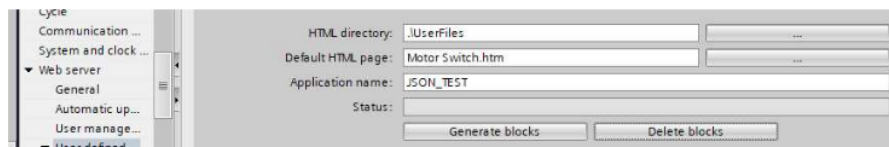


Figure 74 - html index

The webpages are held in the 'userfiles' folder.

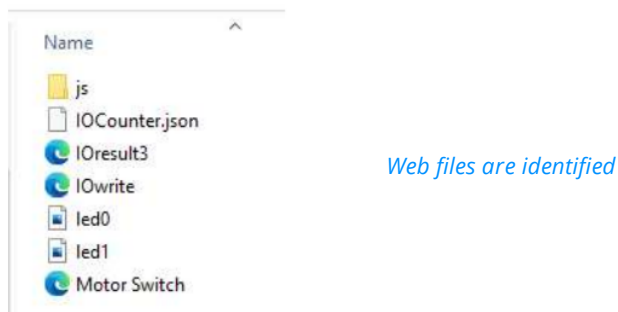


Figure 75 - userfiles folder

In the images below are the explanations for the files in the above image.

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.

Figure 76 - IOwrite file

IOresult3.htm:

:= "webdata".Motor_PWM:

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

Figure 77 - IOresult3 file

IOCounter.json:

```
"Motor Speed PWM" :      "":"Motor Speed PWM":",  
"Motor_PWM" :          "":"webdata".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.

Data tags and ID's are identified

Figure 78 - IOCounter file

All html program code can be seen below.

```
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 <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">  
9  
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     #gl {  
19         width:500px; height:350px;  
20         display: inline-block;  
21         margin: 1em;  
22         title: "Result 1";  
23     }  
24  
25  
26     p {  
27         display: block;  
28         width: 450px;  
29         margin: 2em auto;  
30         text-align: left;  
31     }  
32 </style>  
33
```

Figure 79 - HTML MotorSwitch

```

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 dis
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; font-family: 'calibri'; border-collapse: collapse;">
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/jstgauge.1.0.1.min.js"></script>
60

```

Figure 80 - HTML MotorSwitch (1)

```

60
61
62 <!--Give an id to the gauge dynamic 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 81 - HTML MotorSwitch (2)

```

87
88
89 <!-- This is the JSON function call for reading the Motor_PWM value 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 //change image to green if condition is on
115 $('#img9').html('');
116 }

```

Figure 82 - HTML MotorSwitch (3)

```

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 from 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 83 - HTML MotorSwitch (4)

```

144
145 //ajax call function for gl 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 setInterval(function() {
149     $.get("IOresult3.htm", function(result){
150
151         gl.refresh(parseInt(result));
152
153     });
154
155 },1000); //END OF THIS AJAX FUNCTION CALL
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>

```

The function calls need to be explained

Figure 84 - HTML MotorSwitch (5)

Below are images of the WebServer on the PC which shows the HMI for the PLC, indicating the percentage of power the motor is running at and an LED to prove the digital output of the motor being on. The watch tables show the value changing from false to true when the motor is on/off.

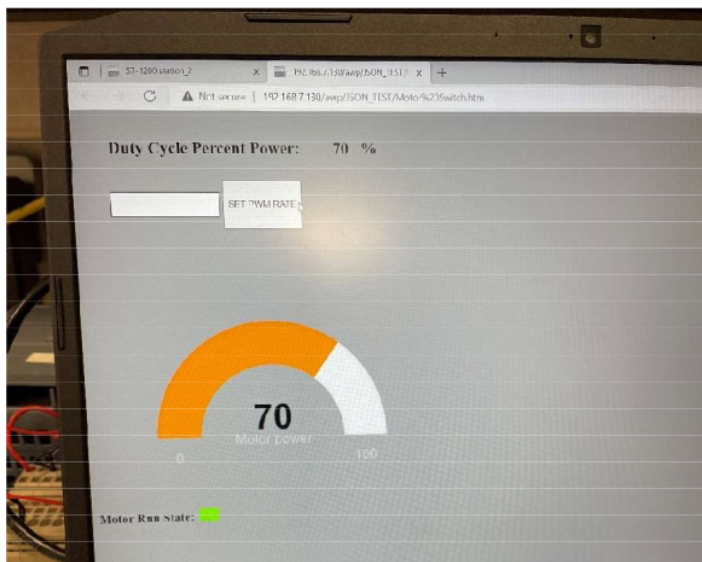


Figure 85 - WebServer HMI Testing

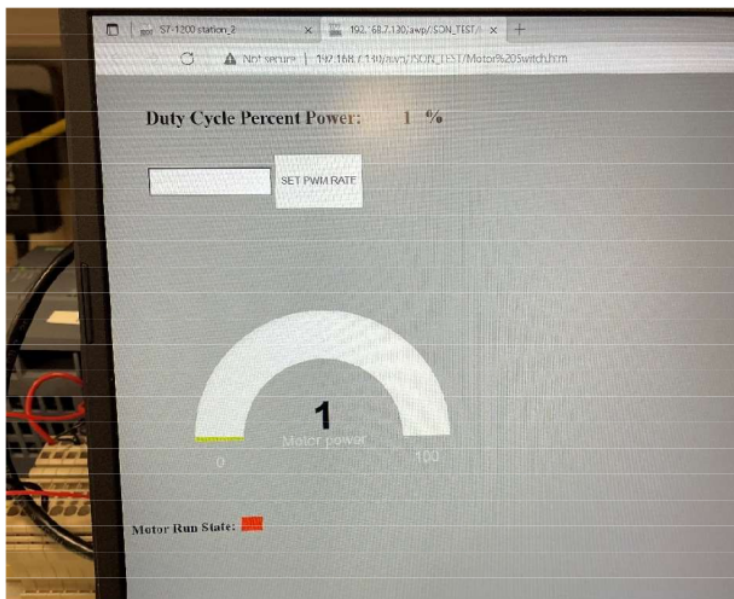


Figure 86 - WebServer HMI Testing (1)

Name	Address	Display format	Monitor value
"Motor Speed ..."	%QW1000	Hex	16#0000
"Motor RUN"	%I0.4	Bool	FALSE
"webdata".Motor...		DEC+/-	0

Figure 87 - Watch table for motor OFF

Name	Address	Display format	Monitor value
"Motor Speed ..."	%QW1000	Hex	16#0014
"Motor RUN"	%I0.4	Bool	TRUE
"webdata".Motor...		DEC+/-	20

Figure 88 - Watch table for motor ON

3.3 Conclusions and Evaluation

The final element of this portfolio proved how a web server can be used as a HMI to show the conditions of the equipment that the PLC is running, with a basic HTML program. The benefit of the web server is that a hard-wired connection doesn't need to be made from the PLC to the HMI, if there is an IP Address then the HMI can be viewed, and the plant could potentially be controlled from a remote place. This is one of the concepts of Industry 4.0. The final task showed aspects of all the learnings from this course and brought it together at the end.

Works Cited

bytesofgigabytes, 2020. *bytesofgigabytes*. [Online]
Available at: <https://bytesofgigabytes.com/networking/what-is-port-number-in-networking/>
[Accessed 29 05 2022].

imperva, 2020. *imperva*. [Online]
Available at: <https://www.imperva.com/learn/application-security/osi-model/>
[Accessed 29 05 2022].

Janssen, T., 2022. *Stackify*. [Online]
Available at: <https://stackify.com/oop-concept-for-beginners-what-is-encapsulation/>
[Accessed 29 05 2022].

Moumita, 2019. *TutorialsPoint*. [Online]
Available at: <https://www.tutorialspoint.com/what-is-broadcasting-in-computer-network#>
[Accessed 29 05 2022].

OmniSecu, 2020. *OmniSecu*. [Online]
Available at: <https://www.omniseку.com/tcpip/tcpip-model.php>
[Accessed 29 05 2022].

studytonight, 2020. *studytonight*. [Online]
Available at: <https://www.studytonight.com/computer-networks/complete-osi-model>
[Accessed 29 05 2022].

Reasonable number of relevant sources

Appendix

Appendix 2.1

```
London>^Z
London>EN
London#SHOW RUN
Building configuration...

Current configuration : 699 bytes
!
version 12.2
no service timestamps log datetime msec
no service timestamps debug datetime msec
no service password-encryption
!
hostname London
!
no ip cef
no ipv6 cef
!
interface FastEthernet0/0
ip address 192.168.7.30 255.255.255.240
duplex auto
speed auto
!
interface FastEthernet0/1
no ip address
duplex auto
speed auto
shutdown
!
interface Serial0/0
ip address 192.168.7.77 255.255.255.240
!
interface Serial0/1
ip address 192.168.7.110 255.255.255.240

router rip
```

Figure 89 - London Show Run CIDR/28 (1)

```
version 2
network 192.168.7.0
network 192.168.30.0

ip classless
ip flow-export version 9
line con 0

line aux 0
line vty 0 4
login
end
```

Figure 90 - London Show Run CIDR/28 (2)

```

Paris>^Z
Paris>EN
Paris#SHOW RUN
Building configuration...

Current configuration : 716 bytes
!
version 12.2
no service timestamps log datetime msec
no service timestamps debug datetime msec
no service password-encryption
!
hostname Paris
!
no ip cef
no ipv6 cef
!
interface FastEthernet0/0
ip address 192.168.7.62 255.255.255.240
duplex auto
speed auto
!
interface FastEthernet0/1
no ip address
duplex auto
speed auto
shutdown
!
interface Serial0/0
ip address 192.168.7.93 255.255.255.240
!
interface Serial0/1
ip address 192.168.7.109 255.255.255.240
clock rate 64000
!

```

Figure 91 - Paris Show Run CIDR/28 (1)

```

router rip
version 2
network 192.168.7.0
network 192.168.30.0
!
ip classless
!
ip flow-export version 9
!
line con 0
!
line aux 0
!
line vty 0 4
login
end

```

Figure 92 - Paris Show Run CIDR/28 (2)

```
Rome>en
Rome#show run
Building configuration...

Current configuration : 732 bytes

version 12.2
no service timestamps log datetime msec
no service timestamps debug datetime msec
no service password-encryption

hostname Rome

no ip cef
no ipv6 cef

interface FastEthernet0/0
ip address 192.168.7.46 255.255.255.240
duplex auto
speed auto

interface FastEthernet0/1
no ip address
duplex auto
speed auto
shutdown

interface Serial0/0
ip address 192.168.7.94 255.255.255.240
clock rate 64000

interface Serial0/1
ip address 192.168.7.78 255.255.255.240
clock rate 64000
```

Figure 93 - Rome Show Run CIDR/28 (1)

```
router rip
version 2
network 192.168.7.0
network 192.168.30.0
ip classless
ip flow-export version 9
line con 0
line aux 0
line vty 0 4
login
end
```

Figure 94 - Rome Show Run CIDR/28 (2)

1909365

GRADEMARK REPORT

FINAL GRADE

GENERAL COMMENTS

Instructor

68/100

PAGE 1

PAGE 2

PAGE 3

Text Comment. Good table of contents

PAGE 4

PAGE 5

PAGE 6

Text Comment. Extensive list of figures

PAGE 7

Text Comment. Good table that supports explanation

PAGE 8

Text Comment. Good and accurate explanation

PAGE 9

Text Comment. Image supports explanation

Text Comment. This can be described as the combined high level layer

Text Comment. This explanation is not accurate

PAGE 10

Text Comment. The image is correct but the explanation does not support this

Text Comment. Good supporting image

Text Comment. Explanation is correct, however the domain can be any size but is undesirable. Smaller domains are more reliable and faster

PAGE 11

Text Comment. Port allocation is identified

PAGE 12

Text Comment. Good schematic diagram

PAGE 13

Text Comment. Address scheme is correct and can be used in the implementation

PAGE 14

PAGE 15

Text Comment. Static routing is configured

Text Comment. Static routing protocol test evidence proves network credibility

PAGE 16

PAGE 17

Text Comment. Dynamic routing is configured fir the addresses in the schematic

PAGE 18

Text Comment. Reports show the connected addresses and routed

PAGE 19

Text Comment. Router reports confirm configuration

Text Comment. RIP routing test evidence supports network functionality

Text Comment. DHCP is configured

Text Comment. The address here should be .1 not .11

Text Comment. Test prior to helper addresses configurations indicate the failure

Text Comment. Helper addresses are configured

Text Comment. The test proves the helper addresses solve the problem.

Text Comment. The exclusion configuration is missing

Text Comment. Exclusion addresses are set here but this is missing in the config section. The DHCP should be tested with and without exclusion addresses

Text Comment. Router report show the configuration for each router

Text Comment. Evaluation is only brief

Text Comment. Intro shows the understanding of the brief

Text Comment. Network diagram indicates the address configuration

Text Comment. Subnet mask is identified

Text Comment. Correct address config

Text Comment. Table is correct for net address 32

Text Comment. Correct for address 48

Text Comment. Correct for address 64

Text Comment. Correct for address 80

Text Comment. Correct for address 96

Text Comment. Tables are correct and follow a logical design

Text Comment. Dynamic routing is implemented

Text Comment. Tests prove the configuration

Text Comment. Subnet mask is identified

Text Comment. Correct table of addresses for /27

Text Comment. Table is correct for /30

Text Comment. Good significant point highlighted

Text Comment. Tables are correct for /27 and /30

Text Comment. Tests prove the /27 and /30 configurations

PAGE 38

Text Comment. The switch is the main component for VLAN. The router is only for trunk protocol config

Text Comment. Report shows VLANs active but not configured

PAGE 39

Text Comment. VLANs configured and ports assigned

Text Comment. Ports assigned

PAGE 40

Text Comment. Two gateways are configured with protocol

PAGE 41

Text Comment. Trunk config evidence is missing and no explanation for the trunk connection

Text Comment. Test proves the configuration is working

PAGE 42

Text Comment. ACL config is correct

PAGE 43

Text Comment. Report confirms configuration

PAGE 44

Text Comment. Test proves ACL firewall

PAGE 45

Text Comment. Tests show access and denial of service

Text Comment. Limited evaluation

Text Comment. Good schematic diagram with VLANs identified

Text Comment. Trunk is set

Text Comment. Gateway configured

Text Comment. Gateway configured

Text Comment. Test confirms config

Text Comment. Tests proves comms across VLANs

Text Comment. Web server is activated and file extensions declared

Text Comment. Server block is implemented

Text Comment. Web files are identified

Text Comment. Data tags and ID's are identified

Text Comment. The function calls need to be explained

PAGE 58

PAGE 59

PAGE 60

Text Comment. Reasonable number of relevant sources

PAGE 61

PAGE 62

PAGE 63

GRADING FORM: CSG-RUBIC

MATTHEW DAVIES

68

RESEARCH-INFORMED LITERATURE



Research has parts missing, however relevant ref sources are included. 6/10

6

KNOWLEDGE AND UNDERSTANDING



Element 1 is designed with a good schematic address table and implemented using the chosen address scheme. The exclusion address config is missing. 7.5/10 Element 2 address tables are correct. VLAN Trunk config is missing and not

21

explained. 7/10. Element 3 practical is detailed and each step is explained. The function calls have not been explained and the router address config is missing for the PLC. 6.5/10

ANALYSIS



E1 test evidence is included and explained. The tests are successful and in a good order. Limited evaluation. 3/5 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. 7/10 E3 - Practical VLAN test evidence is provided and indicate the correct functioning of the network. The PLC functions are tested with evidence directly from the PLC and browser matching. A limited evaluation. 7/10

17

PRACTICAL APPLICATION AND DEP.



E1 - Implementation from topology design carried out using address schema table method. Testing procedure tested every device in every direction and proved successful. 4/5 E2 - Implementations for CIDR are successful. Network was further developed for VLAN configuration and ACL functions implemented. 3.5/5 E3 - The practical network has been successfully implemented and tested. The PLC router config evidence is missing and the function calls have not been explained. 6.5/10

14

SKILLS FOR PROFESSIONAL PRAC.



A good report despite some minor errors and some content missing. 10/15

10