



ESCOLA
POLITÈCNICA SUPERIOR
UNIVERSITAT DE LLEIDA

Universitat de Lleida

ESCOLA POLITÈCNICA SUPERIOR

PRÀCTICA 3 : REDUNDÀNCIA, BALANCEIG I SNMP

Lab Work #3

Autors:

Àlex Codina Bracerós - 49750825P

Mario Fernández Rodríguez - 21042310R

Pol Triquell Lombardo - 48054396J

June 19, 2023

Index of Contents

1	Introduction	2
2	Part 1. Full redundant SLB	2
2.1	<i>Key Configuration Issues.</i>	4
2.2	<i>Conducted tests to probe SLB and GLBP correct operation.</i>	4
2.3	<i>Determine the least connections threshold to switch between servers.</i>	8
3	Part 2. SNMP	9
3.1	<i>Project Installation.</i>	9
3.2	<i>Project Description.</i>	9
3.3	<i>Objectives Accomplished.</i>	9
3.4	<i>Monitor the network using CISCO traps.</i>	10

List of Figures

1	Esquema de la topologia de xarxa a implementar a la Part 1	2
2	Topologia de xarxa a implementar representada al GNS3	3
3	Connexió abans de tancar el router SLB1	5
4	Connexió després de tancar el router SLB1	6
5	Connexió després de reobrir el router SLB1	7
6	Connexió després de tancar el <i>real server</i> S1	8
7	Resultat després de capturar un trap després d'haver-lo transformat	12

List of Tables

1 Introduction

En aquesta pràctica se'ns ha presentat fer un l'anàlisi de dades SNMP i fer una configuració d'una granja amb dos servidors, un per a HTTP i un altre per a HTTPS. La pràctica consta de dues parts principals: l'anàlisi de dades SNMP i la configuració de la granja de servidors.

Així, la posada a prova dels coneixements apresos en les sessions de teoria utilitzant l'eina d'anàlisi i visualització Wireshark.

Aquesta pràctica està estructurada perquè sigui de fàcil accés per a tots els lectors i s'ajuda de diagrames per poder mostrar una visió més esquemàtica de com funcionen les diferents fases d'implementació d'aquesta. Seguidament, s'explicaran els diferents punts de vista per resoldre els problemes sorgits.

Cal destacar que aquesta pràctica ens ha ajudat molt a l'hora de veure quin és el funcionament dels diferents protocols i entendre'l millor. Com hem comentat anteriorment, hem hagut d'utilitzar els diferents conceptes apresos a classe per poder realitzar la pràctica.

2 Part 1. Full redundant SLB

Design and configure the following network, using CISCO 7200.

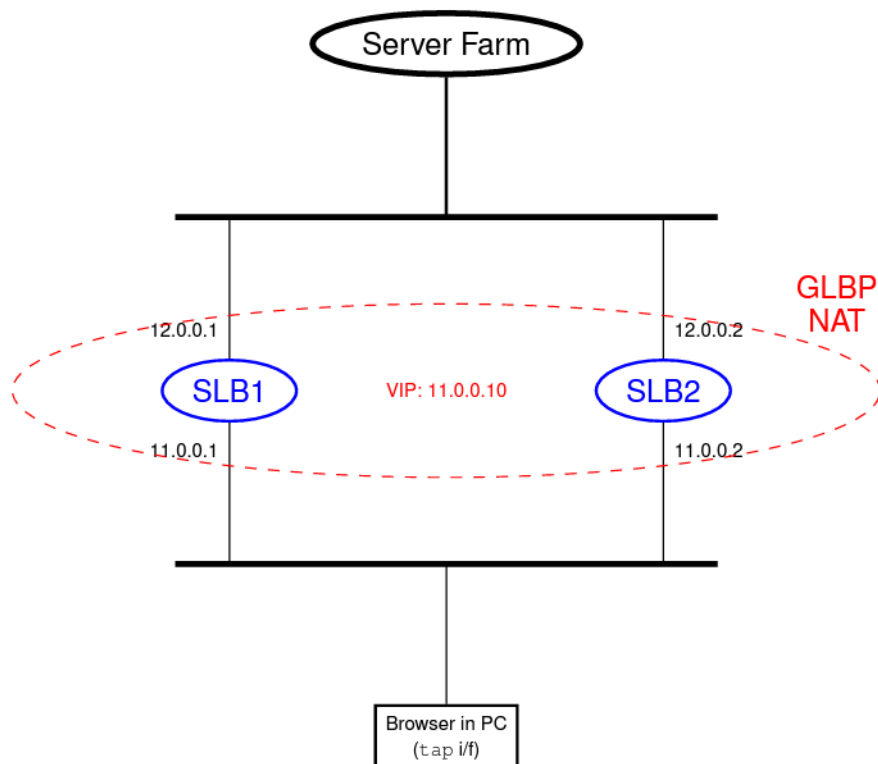


Figure 1: Esquema de la topologia de xarxa a implementar a la Part 1

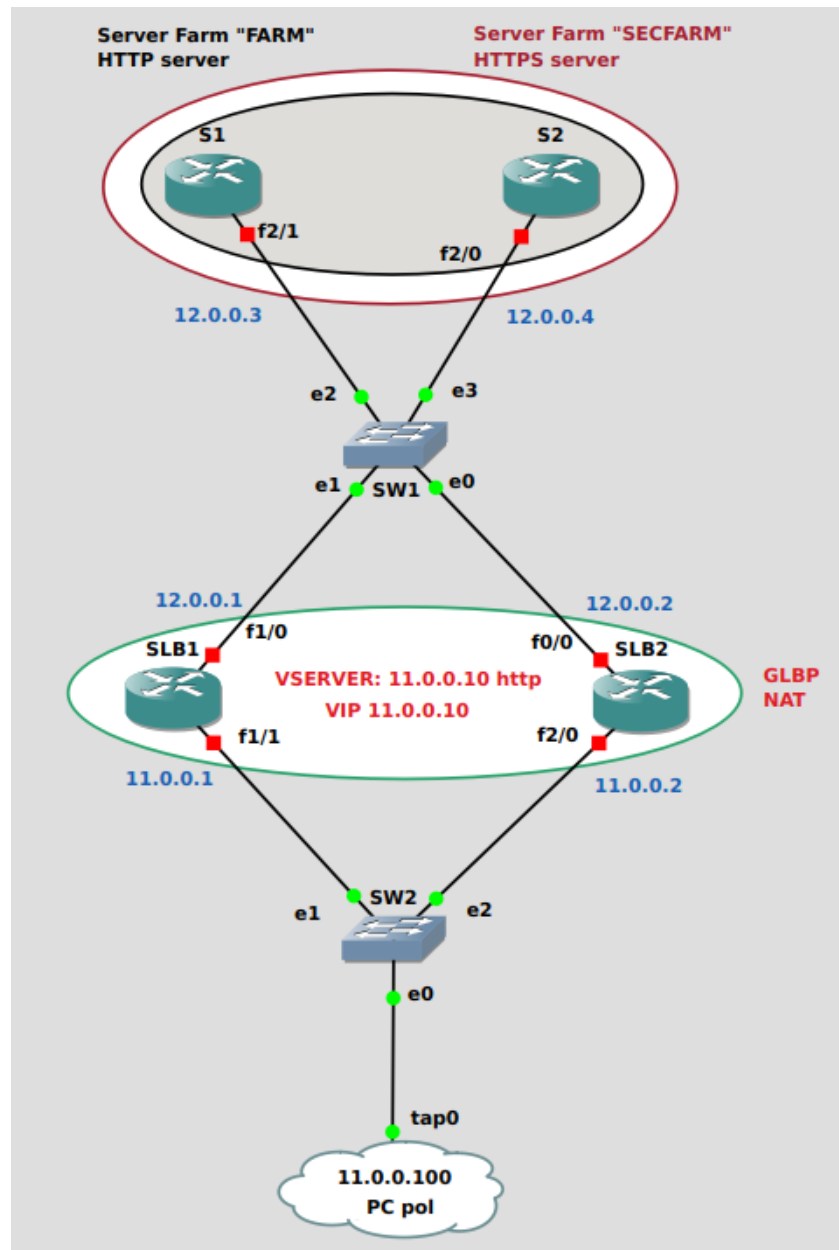


Figure 2: Topologia de xarxa a implementar representada al GNS3

2.1 *Key Configuration Issues.*

Durant la realització de les proves, vam trobar un problema en generar trànsit en un dels nostres equips a causa de la configuració del navegador web Mozilla Firefox. La versió del navegador estava configurada per utilitzar TLSv1.2 i deshabilitava TLSv1.0, el que causava la incapacitat de generar trànsit adequat.

Per resoldre aquest problema, vam emprar una màquina amb sistema operatiu Ubuntu i vam canviar la configuració del navegador per permetre l'ús de TLSv1.0. Això ens va permetre generar trànsit i capturar-lo amb Wireshark, confirmant així que la configuració dels servidors era correcta.

2.2 *Conducted tests to probe SLB and GLBP correct operation.*

Per assegurar-nos que el funcionament del *SLB* (*Server Load Balancing*) i del *GLBP* (*Gateway Load Balancing Protocol*) era correcte, es van realitzar les següents proves:

- Apagar el *router* actiu.

Es va apagar el router que estava actiu i que responia a les peticions del client. Això va forçar l'altre servidor a assumir el seu rol i respondre a les peticions. Aquesta prova tenia com a objectiu verificar que el canvi de servidors es produïa de manera adequada quan un estava inactiu.

- Prova de SLB: Refrescar la pàgina web.

Es va actualitzar la pàgina web (HTTP) per assegurar-se que es mostraven tant el *real server* S1 com el *real server* S2. Aquesta prova tenia com a objectiu verificar que el trànsit es repartia de manera equitativa entre els servidors actius.

- Prova de SLB: Apagar un dels *real servers*.

Es va apagar un dels *real servers* per assegurar-se que l'altre servidor continuava resolent les peticions. Aquesta prova tenia com a objectiu comprovar que, en cas de fallada d'un servidor, l'altre era capaç de gestionar el trànsit de manera adequada i proporcionar els serveis necessaris als clients.

Amb aquestes proves, s'ha comprovat que la configuració del SLB i el GLBP funciona correctament i és capaç de gestionar el trànsit de manera eficient i fiable entre els servidors actius.

Per reforçar les nostres exhaustives proves, hem considerat convenient adjuntar quatre captures de Wireshark per mostrar el trànsit al fer aquestes proves.

Capturing from - [SW1 Ethernet2 to S2 FastEthernet2/0]

No.	Time	Source	Destination	Protocol	Length	Info
13	79.997007	ca:04:cf:3e:00:38	ca:04:cf:3e:00:38	LOOP	60	Reply
14	86.918909	ca:03:cf:0e:00:39	ca:04:cf:3e:00:38	CDP	349	Device ID: S1 Port ID: F
15	89.998750	ca:04:cf:3e:00:38	ca:04:cf:3e:00:38	LOOP	60	Reply
16	92.076481	ca:02:cd:ee:00:00	ca:04:cf:3e:00:38	CDP	360	Device ID: SLB2 Port ID:
17	97.096165	ca:04:cf:3e:00:38	ca:04:cf:3e:00:38	CDP	349	Device ID: S2 Port ID: F
18	99.994821	ca:04:cf:3e:00:38	ca:04:cf:3e:00:38	LOOP	60	Reply
19	101.982969	ca:01:cd:b6:00:1c	ca:04:cf:3e:00:38	CDP	360	Device ID: SLB1 Port ID:
20	110.000582	ca:04:cf:3e:00:38	ca:04:cf:3e:00:38	LOOP	60	Reply
21	119.998349	ca:04:cf:3e:00:38	ca:04:cf:3e:00:38	LOOP	60	Reply

Frame 2: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface -, id 0
 Ethernet II, Src: ca:04:cf:3e:00:38 (ca:04:cf:3e:00:38), Dst: ca:04:cf:3e:00:38 (ca:04:cf:3e:00:38)

Capturing from - [SW1 Ethernet3 to S1 FastEthernet2/1]

No.	Time	Source	Destination	Protocol	Length	Info
164	255.200785	12.0.0.3	12.0.0.1	TCP	179	80 → 39360 [ACK] Seq=1974
165	255.211049	12.0.0.3	12.0.0.1	TCP	1514	80 → 39360 [ACK] Seq=2099
166	255.221224	12.0.0.3	12.0.0.1	HTTP	912	HTTP/1.1 200 OK (text/ht
167	255.234232	12.0.0.1	12.0.0.3	TCP	54	39360 → 80 [ACK] Seq=421
168	255.234362	12.0.0.1	12.0.0.3	TCP	54	39360 → 80 [ACK] Seq=421
169	255.246521	12.0.0.1	12.0.0.3	TCP	54	39360 → 80 [FIN, ACK] Seq
170	255.262040	12.0.0.3	12.0.0.1	TCP	60	80 → 39360 [ACK] Seq=4418
171	259.999659	ca:03:cf:0e:00:39	ca:03:cf:0e:00:39	LOOP	60	Reply
172	269.992021	ca:03:cf:0e:00:39	ca:03:cf:0e:00:39	LOOP	60	Reply

Frame 1: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface -, id 0
 Ethernet II, Src: ca:03:cf:0e:00:39 (ca:03:cf:0e:00:39), Dst: ca:03:cf:0e:00:39 (ca:03:cf:0e:00:39)
 Configuration Test Protocol (loopback)
 Data (40 bytes)

Figure 3: Connexió abans de tancar el router SLB1

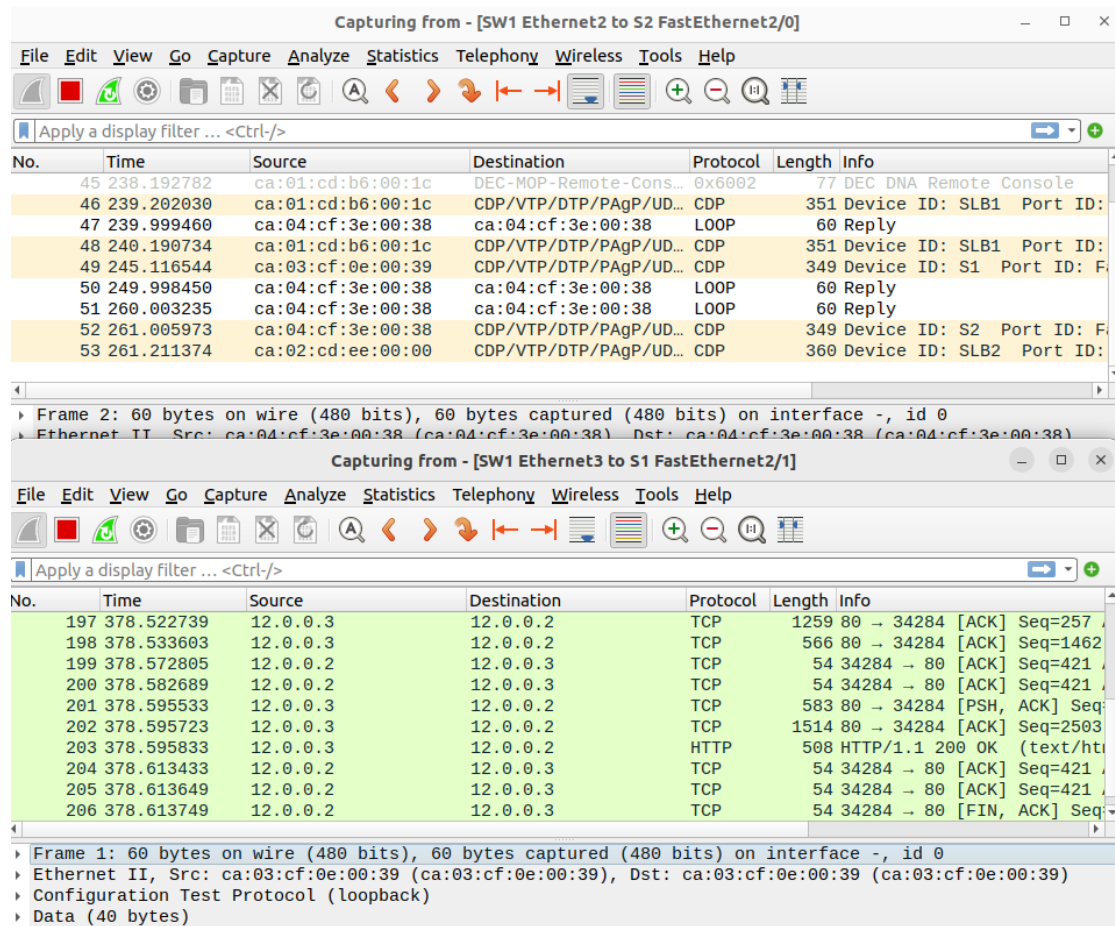


Figure 4: Connexió després de tancar el router SLB1

Capturing from - [SW1 Ethernet2 to S2 FastEthernet2/0]

No.	Time	Source	Destination	Protocol	Length	Info
73	374.240871	ca:04:cf:3e:00:38	CDP/VTP/DTP/PagP/UD...	CDP	349	Device ID: S2 Port ID: F
74	379.997395	ca:04:cf:3e:00:38	ca:04:cf:3e:00:38	LOOP	60	Reply
75	389.992226	ca:04:cf:3e:00:38	ca:04:cf:3e:00:38	LOOP	60	Reply
76	399.419371	ca:01:cd:b6:00:1c	CDP/VTP/DTP/PagP/UD...	CDP	360	Device ID: SLB1 Port ID:
77	400.000483	ca:04:cf:3e:00:38	ca:04:cf:3e:00:38	LOOP	60	Reply
78	404.732294	ca:03:cf:0e:00:39	CDP/VTP/DTP/PagP/UD...	CDP	349	Device ID: S1 Port ID: F
79	409.998833	ca:04:cf:3e:00:38	ca:04:cf:3e:00:38	LOOP	60	Reply
80	416.552789	ca:02:cd:ee:00:00	CDP/VTP/DTP/PagP/UD...	CDP	360	Device ID: SLB2 Port ID:
81	419.995647	ca:04:cf:3e:00:38	ca:04:cf:3e:00:38	LOOP	60	Reply

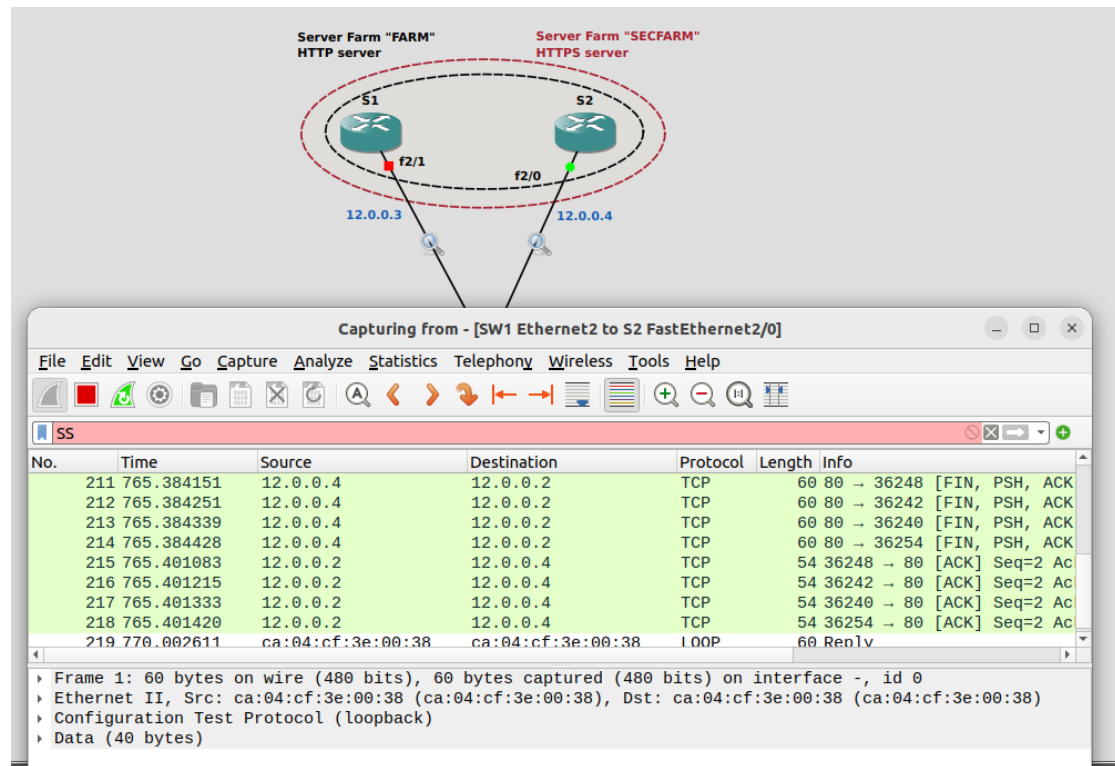
Frame 2: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface -, id 0
 Ethernet II, Src: ca:04:cf:3e:00:38 (ca:04:cf:3e:00:38), Dst: ca:04:cf:3e:00:38 (ca:04:cf:3e:00:38)

Capturing from - [SW1 Ethernet3 to S1 FastEthernet2/1]

No.	Time	Source	Destination	Protocol	Length	Info
296	574.346401	12.0.0.2	12.0.0.3	TCP	54	53706 → 80 [ACK] Seq=421
297	574.366692	12.0.0.2	12.0.0.3	TCP	54	53706 → 80 [ACK] Seq=421
298	574.385695	12.0.0.3	12.0.0.2	TCP	130	80 → 53706 [ACK] Seq=1767
299	574.396100	12.0.0.3	12.0.0.2	TCP	1514	80 → 53706 [ACK] Seq=1843
300	574.396228	12.0.0.3	12.0.0.2	HTTP	1168	HTTP/1.1 200 OK (text/ht
301	574.407905	12.0.0.2	12.0.0.3	TCP	54	53706 → 80 [ACK] Seq=421
302	574.417955	12.0.0.2	12.0.0.3	TCP	54	53706 → 80 [ACK] Seq=421
303	574.418102	12.0.0.2	12.0.0.3	TCP	54	53706 → 80 [FIN, ACK] Seq
304	574.436797	12.0.0.3	12.0.0.2	TCP	60	80 → 53706 [ACK] Seq=4418

Frame 1: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface -, id 0
 Ethernet II, Src: ca:03:cf:0e:00:39 (ca:03:cf:0e:00:39), Dst: ca:03:cf:0e:00:39 (ca:03:cf:0e:00:39)
 Configuration Test Protocol (loopback)
 Data (40 bytes)

Figure 5: Connexió després de reobrir el router SLB1

Figure 6: Connexió després de tancar el *real server* S1

2.3 Determine the least connections threshold to switch between servers.

Durant l'experiment realitzat per avaluar l'algorisme de balanceig de càrrega "least connection", s'ha observat que aquest utilitza un mecanisme de "slow-start" per evitar enviar noves connexions als servidors que acaben de gestionar una connexió. En aquest algorisme, el *real server* amb el menor nombre de connexions actives obté la següent sol·licitud de connexió per a la *server farm*.

Es va portar a cap un seguiment detallat de les connexions i es van analitzar les estadístiques corresponents per comprendre el comportament del sistema sota càrregues variables. A partir d'aquestes observacions, es van obtenir les següents conclusions:

- Si una petició HTTP s'obria justament en el moment de portar a cap la commutació, l'algorisme ens redirigia cap a un altre *real server* per tal de poder aprofitar el mecanisme de "slow-start" de TCP. Això es fa per garantir una millor distribució de la càrrega entre els *Active servers*.
- En canvi, si la connexió TCP ja s'havia finalitzat en un *real server* específic, s'observava una tendència a redirigir les sol·licituds posteriors al mateix servidor. S'ha considerat que aquest comportament canvia a l'altre *real server* després d'un nombre màxim d'aprox. 8-9 connexions finalitzades. Això suggereix que l'algorisme prioritza la distribució equitativa de les noves connexions, fins i tot si s'han finalitzat connexions prèvies en un servidor específic.

Així doncs, basant-se en l'experiment dut a terme, es pot concloure que l'algorisme "least connection" commuta entre servidors en funció del nombre de connexions actives, prioritzant la distribució equitativa de les noves connexions i aprofitant el mecanisme de "slow-start" de TCP per millorar el rendiment del sistema.

3 Part 2. SNMP

Develop a management application using SNMP for IP networks of routers. Discover systems, IPs and routes. Plot the network architecture. Monitor OSPF components.

3.1 Project Installation.

Per tal d'instal·lar el projecte i totes les seves dependències de manera senzilla, s'ha proporcionat un fitxer *requirements.txt*. Aquest fitxer conté una llista de les dependències necessàries per al correcte funcionament del projecte.

Per instal·lar aquestes dependències, simplement s'ha d'executar la comanda següent:

```
$ pip install -r requirements.txt
```

Amb aquests passos, totes les dependències requerides per al projecte seran instal·lades de forma automàtica, permetent que l'aplicació funcioni adequadament.

Per provar si funciona el nostre *SNMP Data Analyzer*, hem decidit implementar dues tipologies de xarxa, una simple i una completa que permeten veure el funcionament de la pròpia aplicació al repositori "*GNS3-CISCO-Routers-Network-Examples*".

3.2 Project Description.

Per obtenir més informació sobre com instal·lar i configurar el projecte, consulteu el repositori de GitHub següent: "*SNMP Data Analyzer*".

S'ha desenvolupat una aplicació que utilitza SNMP per gestionar i monitoritzar els dispositius d'una xarxa.

3.3 Objectives Accomplished.

En aquesta secció de l'informe, proporcionarem una descripció de les estructures de dades principals utilitzades en el projecte, així com un pseudocodi dels procediments implementats.

- *Polling all the routers.*

En aquest punt, hem aconseguit recuperar la informació del sysName i les interfícies de tots els routers utilitzant les biblioteques de Python per SNMP, concretament les biblioteques *net-snmp-python*. Hem fet servir les MIB OSPF i OSPF-MIB per descobrir els veïns i obtenir detalls de la configuració de la xarxa.

Els passos realitzats són:

1. Utilitzant les MIB OSPF, OSPF-MIB, IF-MIB, IP-MIB i IP-FORWARD-MIB, hem consultat la informació del sistema (sysName) i les interfícies (i/f) de cada router.

2. La informació obtinguda inclou l'adreça IP, la màscara de xarxa, la velocitat, així com els bucles descartats i les interfícies inactives.

Hem aconseguit complir aquest requisit, obtenint la informació requerida de tots els routers.

- *Getting the routing tables.*

En aquest punt, hem aconseguit obtenir les *routing tables* de cada router. Per a això, hem utilitzat la *routing table* `ipCidrRouteTable` de la MIB IP-FORWARD-MIB per obtenir informació detallada de les rutes de xarxa.

Els passos realitzats són:

1. Hem consultat la *routing table* `ipCidrRouteTable` per a cada router.
2. La *routing table* inclou informació com ara la xarxa, la màscara de xarxa, el *next-hop* i el tipus de ruta.

Hem aconseguit complir aquest requisit, obtenint les taules de rutatge de tots els routers.

- *Creating route summaries.*

Per a cada parella possible d'adreces IP, hem detallat el camí més curt utilitzant la llibreria `python-netaddr` (inclosa en la biblioteca estàndard). Hem aplicat una operació de màscara a les adreces IP utilitzant l'operador `"&"`.

Hem aconseguit complir aquest requisit, generant *routing summaries* per a totes les parelles d'adreces IP possibles.

- *Plotting the network.*

Hem generat un fitxer de representació gràfica de la xarxa en format PDF utilitzant la biblioteca `Matplotlib`. Hem etiquetat cada node amb el `sysName` i cada enllaç inclou les adreces IP dels nodes adjacents i la velocitat de l'enllaç.

Hem aconseguit complir aquest requisit, generant una representació gràfica llegible de la xarxa.

- *Monitor the network.*

Hem enviat traps SNMP des dels routers utilitzant els traps `CISCOospfstate-change` i `neighbor-state-change`. Hem rebut i processat aquestes traps utilitzant `snmptrapd`, imprimint i decodificant tota la informació de les traps.

Hem aconseguit complir aquest requisit, monitorant la xarxa i processant amb èxit les traps SNMP rebudes.

3.4 Monitor the network using CISCO traps.

En el nostre cas, hem optat per implementar un script en bash per a la captura i anàlisi de les traps que s'envien des dels routers. Aquest script ens permet obtenir la informació dels traps amb una estructura similar als paquets que s'utilitzen en l'sniffer Wireshark.

Les traps que s'envien des dels routers es registren en un arxiu de registre (log) situat al directori `/tmp`. Per ser més específics, la ruta completa és `/tmp/log`.

A més, hem realitzat diverses modificacions als arxius de configuració `snmpd.conf` i `snmptrapd.conf` per a poder implementar la captura i anàlisi de les traps. En l'arxiu `snmpd.conf`, hem afegit la configuració necessària per a habilitar la captura de les traps i hem especificat les destinacions on volem que siguin enviades. D'altra banda, en l'arxiu `snmptrapd.conf`, hem indicat la comunitat d'autenticació i hem afegit `traphandles` per a processar les traps amb el nostre script de bash.

- **`snmpd.conf`**

```
mibs +OSPF-MIB

# trap2sink: A SNMPv2c trap receiver
#           arguments: host [community] [portnum]

trap2sink rocom localhost
trap2sink rocom 10.0.0.3

# authtrapenable: Should send traps when authentication...
#           arguments 1 | 2           (1 = yes, 2 = no)

authtrapenable 2
```

- **`snmptrapd.conf`**

```
authCommunity log,execute,net rocom

# Aquest traphandle s'activa quan es rep una trap que indica un canvi d'es-
# tat en una interfície OSPF virtual.
traphandle .1.3.6.1.2.1.14.16.2.1 /etc/snmp/scripts/traps\_parser

# Aquest traphandle s'activa quan es rep una trap que indica un canvi
# d'estat en un veí OSPF no virtual.
traphandle .1.3.6.1.2.1.14.16.2.2 /etc/snmp/scripts/traps\_parser

# Aquest traphandle s'activa quan es rep una trap que indica un canvi d'es-
# tat en un veí OSPF virtual.

traphandle .1.3.6.1.2.1.14.16.2.3 /etc/snmp/scripts/traps\_parser

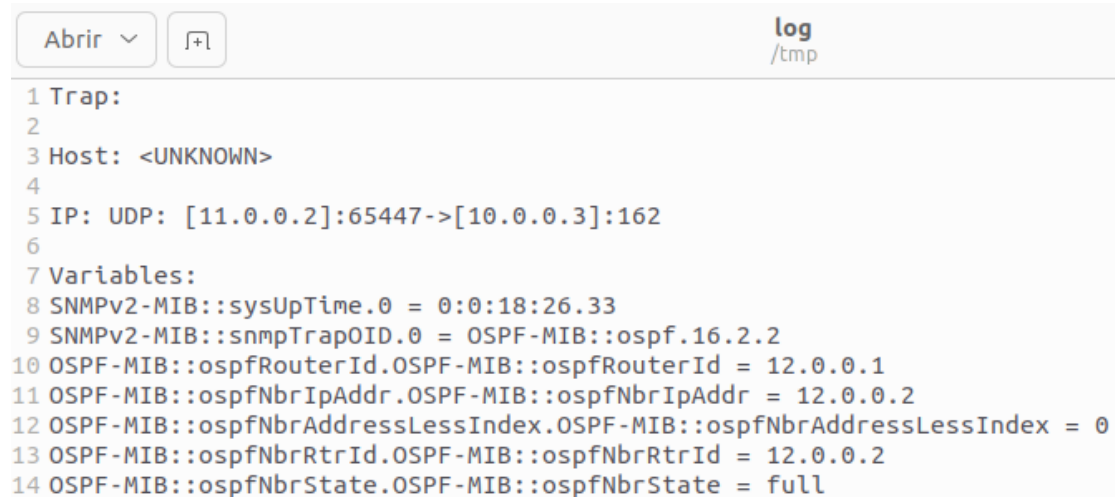
# Aquest traphandle s'activa quan es rep una trap que indica un canvi d'es-
# tat en una interfície OSPF no virtual.
traphandle .1.3.6.1.2.1.14.16.2.16 /etc/snmp/scripts/traps\_parser
```

El nostre script realitza la captura de les traps i després les processa en un format de registre estandarditzat. Aquest format, com es pot observar a la Figura 7, segueix una estructura específica que ens permet identificar i analitzar la informació rellevant de cada trap capturada.

Amb el nostre script i les modificacions als arxius de configuració, hem creat una solució completa per a la captura, registre i anàlisi de les traps. Això ens permetrà monitorar i comprendre millor el comportament de la xarxa, identificar canvis d'estat en les interfícies o veïns OSPF i

prendre les mesures adequades en cas de problemes o anomalies detectades.

En resum, la combinació del nostre script en bash per a la captura i anàlisi de les traps, juntament amb les modificacions als arxius de configuració, ens ofereix una eina valuosa per al seguiment, la resolució de problemes i l'optimització de la xarxa.



```
1 Trap:
2
3 Host: <UNKNOWN>
4
5 IP: UDP: [11.0.0.2]:65447->[10.0.0.3]:162
6
7 Variables:
8 SNMPv2-MIB::sysUpTime.0 = 0:0:18:26.33
9 SNMPv2-MIB::snmpTrapOID.0 = OSPF-MIB::ospf.16.2.2
10 OSPF-MIB::ospfRouterId.OSPF-MIB::ospfRouterId = 12.0.0.1
11 OSPF-MIB::ospfNbrIpAddress.OSPF-MIB::ospfNbrIpAddress = 12.0.0.2
12 OSPF-MIB::ospfNbrAddressLessIndex.OSPF-MIB::ospfNbrAddressLessIndex = 0
13 OSPF-MIB::ospfNbrRtrId.OSPF-MIB::ospfNbrRtrId = 12.0.0.2
14 OSPF-MIB::ospfNbrState.OSPF-MIB::ospfNbrState = full
```

Figure 7: Resultat després de capturar un trap després d'haver-lo transformat