Universitat Oberta de Catalunya

Treball Final de Grau d'Enginyeria Informàtica

Àrea de Xarxes de Computadors

Analitzador Gràfic de Xarxa en entorn GNU

Autor:
Josep Carcelén Manzanilla

Tutor:
Maria Isabel March Hermo

Gener 2015

Gràcies Inma pel teu suport incondicional

Gràcies a la comunitat Linux per fer possible aquesta meravella

Gràcies a les persones que transmeten desinteressadament el seu coneixement per Internet

Resum

Un analitzador de paquets de xarxa (*sniffer*) és un programa que captura les trames d'una xarxa d'ordinadors. L'analitzador desenvolupat en aquest treball de final de grau, a diferència dels analitzadors de xarxa habituals, no pretén mostrar les dades contingudes en les trames dels paquets capturats, sinó mostrar la informació del tràfic de xarxa des d'un nivell d'abstracció per sobre, centrant-se en l'anàlisi dels protocols i els ports de servei que formen les comunicacions en les xarxes d'ordinadors.

Gràcies a això, l'administrador de xarxes disposa d'una eina de suport per optimitzar l'ús de la xarxa d'ordinadors, oferint de manera ràpida i visual un anàlisi de les comunicacions, localitzant quins equips estan fent servir de forma intensiva la xarxa i quins protocols i ports de servei apareixen en les transmissions, en definitiva una eina que permet detectar problemes de configuració, col·lapses i mals usos a la xarxa.

El programari utilitzat en el desenvolupament d'aquest projecte és GNU/Linux, el codi font de l'aplicació s'ha programat en *Python* fent servir entre d'altres les llibreries *GTK*, *Pcapy*, *Scapy*, *Impacket*, *Gnuplot* i *Matplotlib*.

Índex

Índex de figures

Índex de taules

1

Introducció

Un analitzador gràfic de xarxa o *sniffer* és un programa informàtic que intercepta i analitza el tràfic d'una xarxa d'ordinadors segons els *Request for Comments*[2] (RFC) corresponents als protocols de les comunicacions.

Un RFC és una publicació de la *Internet Engineering Task Force*[3] (IETF) i de la *Internet Society*[4] (ISoc), els principals organismes de desenvolupament i establiment de normes tècniques per Internet. Els RFC contenen notes tècniques i organitzatives sobre Internet, i cobreixen aspectes de les xarxes d'ordinadors com els protocols, procediments, programes i conceptes.

La principal funció d'un *sniffer* és convertir el tràfic de xarxa, que no deixa de ser un conjunt de 0 i 1, en un format comprensible pels humans, per permetre analitzar la informació transmesa per la xarxa.

Aquest anàlisi de les transmissions, normalment es realitza a un nivell de trames, per exemple els desenvolupadors d'aplicacions poden comprovar si les trames emeses compleixen les especificacions dels protocols utilitzats, o els administradors de sistemes poden detectar problemes de comunicacions entre equips, si un equip no rep una trama perquè l'emissor no l'envia o perquè el receptor té un tallafocs mal configurat.

Els *sniffers* també es fan servir de forma fraudulenta, els *hackers* han aprofitat la capacitat d'analitzar les transmissions per capturar informació de tercers, com noms d'usuari i contrasenyes que no viatgen encriptades i que són utilitzades per atacar sistemes, o per interceptar informació sensible d'una companyia per vendre-la a la competència, en definitiva per espiar les comunicacions.

Altres utilitats dels *sniffers* són la identificació de les fonts d'anomalies a la xarxa o de problemes de rendiment de les aplicacions, en aquest cas l'anàlisi de la transmissió no es realitza examinant el contingut d'una trama en concret, sinó avaluant el conjunt de les trames. És en aquesta vessant on es pretén enfocar aquest projecte.

1.1 Justificació

Al mercat existeixen una gran varietat d'analitzadors de paquets de xarxa[5]. Dintre dels de llicència GNU i amb interfície gràfica, el més estès per la gran quantitat de funcionalitats i completesa és *Wireshark*[6] (aka *Ethereal*):

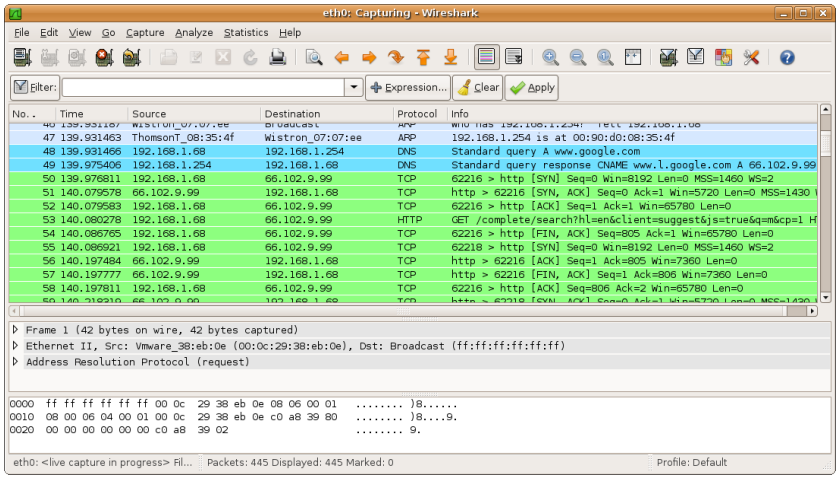


Figura 1.1: Captura de pantalla de Wireshark®

D'altres exemples serien *EtherApe*[7], *Ettercap*[8] i *Xplico*[9]. En tots se sol separar la vessant d'anàlisi del tràfic en altres programes diferenciats, per exemple *Wireshark* ofereix l'aplicació de pagament *SteelCentral Packet Analyzer*[10] per afegir aquesta funcionalitat.

Actualment no hi ha una oferta d'aplicacions GNU que facin una captura i un anàlisi de l'ús de la xarxa, més enllà d'una estadística general de la captura. La majoria de les aplicacions que fan un anàlisi de l'ús de la xarxa són de software propietari com per exemple *CapAnalysis*[11] o *Netflow Traffic Analyzer*[12].

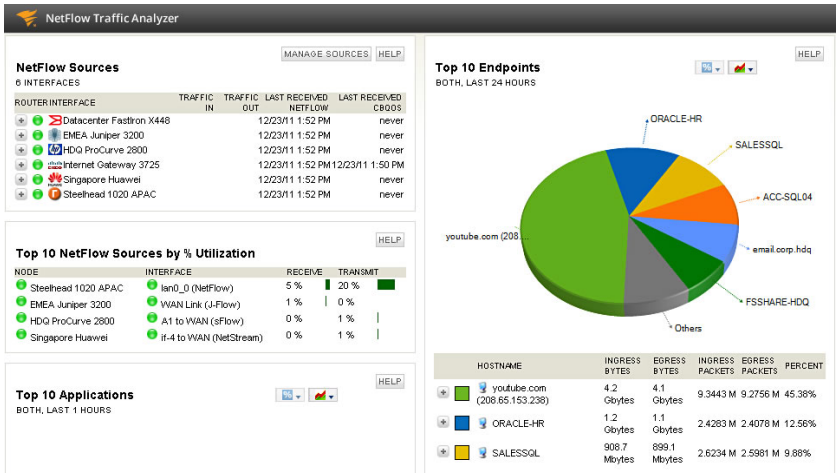


Figura 1.2: Captura de pantalla de Netflow Traffic Analyzer©

Els dispositius de xarxa de la capa 2 del model TCP/IP com són els *switch*, no ofereixen un anàlisi del tràfic que maneguen. Els dispositius de la capa 3 com els *firewalls* i els *routers* permeten bloquejar i encaminar el tràfic de determinats protocols de xarxa, però no solen oferir estadístiques gaire complertes sobre el seu l'ús.

Així doncs, existeix un nínxol de mercat pel programari d'anàlisi de l'ús del tràfic de xarxa en entorn GNU, que fan que el propòsit d'aquest projecte estigui justificat per la falta d'alternatives.

1.2 Objectius

L'objectiu d'aquest projecte és desenvolupar un Analitzador Gràfic de Xarxa (*sniffer*) en entorn GNU. Es pretén realitzar una aplicació gràfica que analitzi l'ús del tràfic d'una xarxa informàtica. L'aplicació ha de permetre filtrar els paquets de la xarxa segons diferents opcions i tractar estadísticament la informació recollida.

L'aplicació capturarà els paquets que circulen en un moment donat per la xarxa i els salvarà en un fitxer en format **.pcap**, compatible amb altres *sniffers*.

Després de la captura de paquets o la càrrega d'un fitxer de captures, el programa ha d'analitzar les dades i mostrar la informació més determinant dels paquets capturats, com l'hora, minuts, segons i microsegons de la trama capturada, el protocol, l'adreça de xarxa IP i MAC tant de l'emissor com del receptor, el port de servei destí i origen, i la grandària de la trama en Bytes.

L'aplicació també ha de mostrar gràfiques amb la informació més rellevant com l'evolució temporal, la mida del tràfic i el número de paquets de xarxa per:

- Protocols de xarxa, inclourem en aquests protocols els de les capes de transport i d'internet del model de referència TCP/IP.
- Adreces de xarxa IP.
- Ports de servei, en aquest cas els protocols inclosos són els de la capa d'aplicació del model de referència TCP/IP. Per diferenciar-los dels protocols de les capes de transport i internet els anomenarem "ports de servei", perquè cada aplicació fa servir un port TCP o UDP diferenciat.

A part de la informació visual en gràfiques, l'aplicació també ha de mostrar taules pels protocols de xarxa de les capes de transport i internet, i pels protocols de la capa d'aplicació o ports de servei, on es veurà el nombre de paquets capturats de cada protocol i la seva grandària. Des d'aquestes taules, també s'ha de poder examinar les adreces IP que intervenen en les comunicacions de cada protocol, mostrant el nombre de paquets capturats per cada IP i la seva grandària.

Per últim, l'aplicació ha de permetre consultar la informació dels protocols i ports de servei de forma individualitzada per cada adreça IP.

Queda fora de l'abast d'aquest projecte desenvolupar una eina que sigui capaç de interpretar tots els protocols de xarxa existents. En la implementació s'han fet servir llibreries de tercers que realitzen aquesta tasca d'interpretació del tràfic capturat, per tant la capacitat d'anàlisi de l'aplicació resultant es veu limitada per les funcionalitats que ofereixen aquestes llibreries.

En concret, el programa *Scapy*[13] que ha estat finalment l'escollit per la captura del tràfic de xarxa, suporta els protocols de les capes de transport i internet ETHER, FDDI, TR, WLAN, IP, IP6, ARP, RARP, DECNET, TCP i UDP.

1.3 Enfocament i mètode seguit

Per la gestió del projecte s'ha seguit una metodologia orientada a objectius, separant la planificació estratègica, marcada per l'entrega de Proves d'Avaluació Continuada (PAC), producte final, memòria i presentacions; de la planificació operativa obtinguda de la descomposició en activitats.

L'elecció del llenguatge de programació ha estat clara des d'un principi, la velocitat d'implementació que ofereix *Python* és superior a la d'altres llenguatges d'entrons GNU com *Java*, però calia una investigació preliminar per verificar l'existència de llibreries en aquest llenguatge que donin suport a la captura de paquets de tràfic de xarxa.

Les alternatives que ofereix *Phyton* per la captura del tràfic de xarxa són:

- Mitjançant *sockets*, capturant el tràfic en brut i fent un *parser* per cada protocol. La gran quantitat de desenvolupament que requereix aquesta opció fa que quedi fora de l'abast del projecte.
- Mitjançant llibreries desenvolupades per tercers, les opcions trobades han estat:
  - *Pcapy*[14] de Core Labs amb llicència *Apache Software* adaptada.
  - *Scapy*[13] desenvolupat per Philippe Biondi amb llicència *GPLv2*.

Tant *Pcapy* com *Scapy* han mostrat problemes durant la implementació del projecte, com s'explica més endavant en aquest document.

A l'hora del desenvolupament de l'aplicació s'ha fet servir una metodologia iterativa. S'ha optat per aquest model pel desconeixement previ de les llibreries de software utilitzat, tant pel tractament del tràfic de xarxa (*Pcapy* / *Scapy*), com per la construcció de la interfície gràfica (*GTK*). Per tant he considerat convenient una aproximació progressiva a la solució enlloc d'un únic anàlisi i disseny més exhaustiu.

En la primera iteració s'ha fet una aproximació a la solució seguint una metodologia estructurada al llarg del cicle de vida amb un primer anàlisi, disseny i implementació. En la segona iteració, s'han completat les fases d'anàlisi, disseny, implementació, afegint les fases de proves i el lliurament.

El model de dades és orientat a objectes i s'ha construït seguint la metodologia UML.

Els principals components de software que es fan servir per la implementació del projecte són:

- GTK+3[15] per la programació de la interfície gràfica, compatible amb la majoria de distribucions GNU/Linux.
- Libpcap[16] llibreria del sistema operatiu per la captura i tractament del tràfic de xarxa.
- Pcapy[14] embolcall Python per Libpcap.
- Scapy[13] programa fet amb Python per la manipulació de paquets de xarxa.
- Impacket[17] classes Python per accedir a la informació dels paquets de xarxa.
- Matplotlib[18] per la representació de la informació en gràfiques.

El programari utilitzat en la resta de fases del projecte també és GNU:

- Kile[19] per l'elaboració de la documentació en LaTeX.
- Planner[20] per la planificació de les tasques.
- Umbrello[21] per la confecció dels diagrames UML.
- Glade[22] pel disseny de la interfície gràfica.
- Pycharm[23] per la implementació del codi font.

1.4 Planificació del projecte

1.4.1 Planificació estratègica

Seguint la metodologia de gestió de projectes orientada a objectius ha estat necessari assolir les següents fites:

1. Lliurament del pla de treball del projecte (*28/09/2014*)
2. Informe de progrés (*12/10/2014*)
3. Informe de progrés (*26/10/2014*)
4. Lliurament PAC1 (*9/11/2014*)
5. Informe de progrés (*23/11/2014*)
6. Informe de progrés (*7/12/2014*)
7. Lliurament PAC2 (*21/12/2014*)
8. Lliurament producte i memòria del projecte (*11/01/2015*)
9. Lliurament presentació del projecte (*18/01/2015*)
10. Lliurament preguntes del tribunal (*25/01/2015*)

1.4.2 Divisió per tasques

Per assolir els objectius del projecte s'han identificat les següents tasques, les seves durades s'han revisat per adequar-les al dia a dia del projecte:

Codi	1
Nom	Gestió i seguiment del projecte
Descripció	Reflectir l'evolució del projecte, detectar els canvis i desviacions de la planificació
Durada	128 dies (tot el projecte)
Resultats	Informes de seguiment quinzenals

Taula 1.1: Gestió i seguiment del projecte

Codi	2
Nom	Documentació
Descripció	Creació de la documentació del projecte
Durada	128 dies (tot el projecte)
Resultats	Documents de memòria, manual, presentació i respostes al tribunal

Taula 1.2: Documentació

Codi	3
Nom	Planificació del projecte
Descripció	Es determinaran els objectius del projecte, les necessitats i estratègies, la divisió de tasques i el seu cronograma
Durada	8 dies (*20/09/2014 - 27/09/2014*)
Resultats	Pla de treball

Taula 1.3: Planificació

La tasca **4 - Anàlisi i disseny lògic** s'ha dividit en les subtasques següents:

Codi	4.1
Nom	Estudi components de software
Descripció	Avaluació de les llibreries GTK+3[15], Libpcap[16] i Matplotlib[18] per determinar les necessitats del model de dades
Durada	3 dies (*28/09/2014 - 30/09/2014*)
Resultats	Requeriments del model de dades

Taula 1.4: Anàlisi i disseny lògic - estudi components de software

Codi	4.2
Nom	Identificació classes
Descripció	Definició de les classes del model de dades orientat a objectes
Durada	3 dies (*1/10/2014 - 3/10/2014*)
Resultats	Casos d'ús i model de dades general

Taula 1.5: Anàlisi i disseny lògic - identificació classes

Codi	4.3
Nom	Creació diagrames UML
Descripció	Primera aproximació al model de dades i als casos d'ús
Durada	5 dies (*4/10/2014 - 8/10/2014*)
Resultats	Diagrames casos d'ús i model de dades amb atributs i mètodes

Taula 1.6: Anàlisi i disseny lògic - creació diagrames UML

La tasca **5 - Implementació** s'ha dividit en les subtasques:

Codi	5.1
Nom	Model de dades
Descripció	Codificació del model de dades
Durada	9 dies (*9/10/2014 - 17/10/2014*)
Resultats	Fitxers amb la implementació del model de dades

Taula 1.7: Implementació - model de dades

Codi	5.2
Nom	Interfase d'adquisició
Descripció	Codificació de la interfase gràfica d'adquisició dels paquets de xarxa
Durada	14 dies (*18/10/2014 - 31/10/2014*)
Resultats	Fitxers amb la implementació de la interfase gràfica d'adquisició

Taula 1.8: Implementació - interfase d'adquisició

Codi	5.3
Nom	Processament paquets de xarxa
Descripció	Codificació del processament de la informació obtinguda d'acord amb el model de dades
Durada	21 dies (*1/11/2014 - 21/11/2014*)
Resultats	Fitxers amb la implementació del processament de les captures

Taula 1.9: Implementació - processament paquets de xarxa

Codi	5.4
Nom	Interfase de resultats
Descripció	Codificació de la interfase gràfica amb el resultat del processament
Durada	30 dies (*1/11/2014 - 30/11/2014*)
Resultats	Fitxers amb la implementació de la interfase gràfica de resultats

Taula 1.10: Implementació - interfase de resultats

Codi	6
Nom	Proves
Descripció	Proves de funcionament de l'aplicació
Durada	5 dies (*1/12/2014 - 5/12/2015*)
Resultats	Fitxers de l'aplicació finalitzada

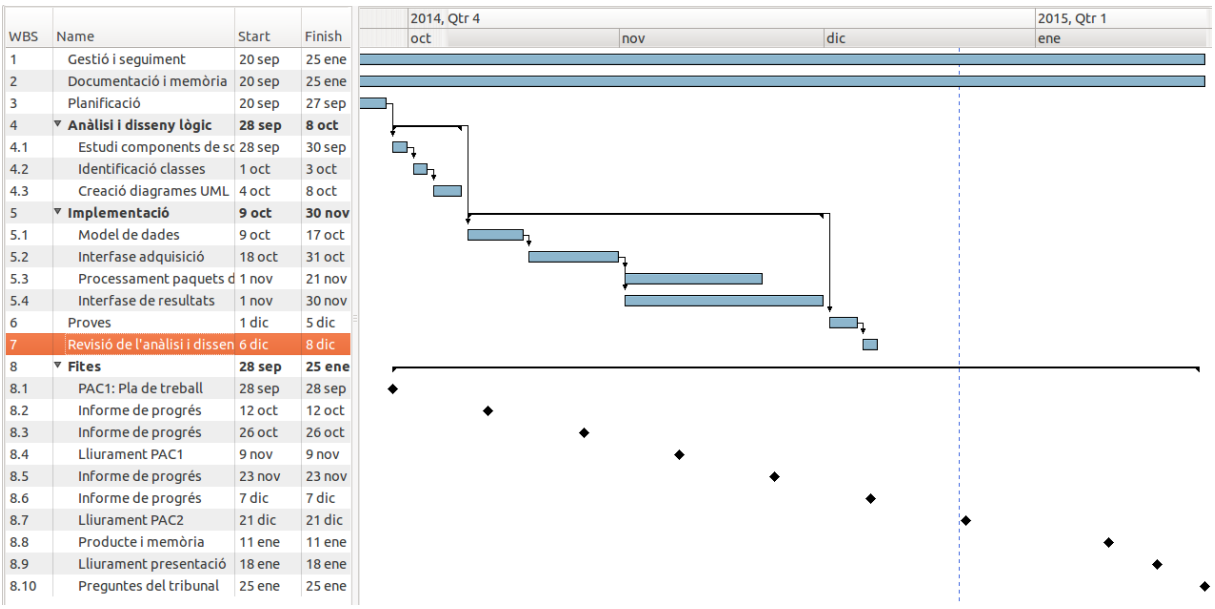
Taula 1.11: Proves

Codi	7
Nom	Revisió de l'anàlisi i el disseny lògic
Descripció	Adequació de l'anàlisi i el disseny a les fases iteratives del desenvolupament
Durada	3 dies (*6/12/2014 - 8/12/2015*)
Resultats	Diagrames casos d'ús i model de dades amb atributs i mètodes revisats

Taula 1.12: Revisió de l'anàlisi i el disseny lògic

1.4.3 Planificació operativa

Segons la descomposició de tasques anterior i la seva seqüenciació es va establir el següent calendari:



WBS	Name	Start	Finish
1	Gestió i seguiment	20 sep	25 ene
2	Documentació i memòria	20 sep	25 ene
3	Planificació	20 sep	27 sep
4	**Anàlisi i disseny lògic**	**28 sep**	**8 oct**
4.1	Estudi components de sc	28 sep	30 sep
4.2	Identificació classes	1 oct	3 oct
4.3	Creació diagrames UML	4 oct	8 oct
5	**Implementació**	**9 oct**	**30 nov**
5.1	Model de dades	9 oct	17 oct
5.2	Interfase adquisició	18 oct	31 oct
5.3	Processament paquets d	1 nov	21 nov
5.4	Interfase de resultats	1 nov	30 nov
6	Proves	1 dic	5 dic
7	Revisió de l'anàlisi i dissen	6 dic	8 dic
8	**Fites**	**28 sep**	**25 ene**
8.1	PAC1: Pla de treball	28 sep	28 sep
8.2	Informe de progrés	12 oct	12 oct
8.3	Informe de progrés	26 oct	26 oct
8.4	Lliurament PAC1	9 nov	9 nov
8.5	Informe de progrés	23 nov	23 nov
8.6	Informe de progrés	7 dic	7 dic
8.7	Lliurament PAC2	21 dic	21 dic
8.8	Producte i memòria	11 ene	11 ene
8.9	Lliurament presentació	18 ene	18 ene
8.10	Preguntes del tribunal	25 ene	25 ene

Figura 1.3: Diagrama de Gantt amb la planificació del projecte

1.5 Productes obtinguts

El producte principal desenvolupat és una aplicació que consta d'una sèrie de fitxers amb el codi font en el llenguatge de programació Python.

Al codi font de l'aplicació també cal afegir el fitxer *gnagnu.glade* que conté la definició de la interfície gràfica de l'aplicació en llenguatge XML.

El producte obtingut es complementa amb un manual d'ús en un fitxer amb format PDF de nom *gnagnu-user-manual.pdf*

1.6 Descripció dels altres capítols de la memòria

Al segon capítol d'aquesta memòria, **Conceptes tècnics**, s'intenta resumir el marc teòric on es situen els analitzadors de xarxa, presentant les principals arquitectures de comunicacions com són els models de referència OSI i TCP/IP, així com els principals protocols de les diferents capes d'aquest últim model.

Al tercer capítol, **Disseny**, es veurà en detall l'anàlisi dels casos d'ús desitjables, així com el diagrama d'activitats que reflecteix el funcionament de l'aplicació, i l'arquitectura de software dissenyada amb les seves classes, segons la metodologia UML.

Al quart capítol, **Implementació**, s'explica el procés de codificació del disseny, com també els principals problemes trobats en aquesta fase i la seva resolució.

Al cinquè capítol, **Producte obtingut**, es detallen els requeriments per l'execució de l'aplicació, els fitxers que la formen, i els apartats dels que consta el manual d'ús, que s'inclou a l'apèndix A d'aquest document.

Al sisè capítol, **Proves**, s'indiquen les proves realitzades sobre l'execució de l'aplicació i els seus resultats.

Per últim al setè capítol, **Conclusions**, consta de les reflexions personals extretes després del desenvolupament d'aquest projecte.

2

Conceptes tècnics

El sistema centralitzat existent a les primeres dècades de la computació es va substituir per un model on diversos computadors interconnectats entre ells realitzaven les mateixes tasques, però amb una millor optimització de la gestió i dels recursos. Aquestes xarxes de computadors es poden classificar de diferents maneres: basant-se en la seva topologia (bus, anell, estrella, arbre o mallada), en el tipus de commutació (de circuits, de paquets o de missatges), en el seu abast (de gran abast *WAN*, d'abast local *LAN*) o en la seva tecnologia (cablada o sense fils)[24].

Les primeres xarxes de computadors es van dissenyar enfocades al hardware, pensant que el procés recauria en el maquinari i en protocols propietaris, però aquesta estratègia no es va mantenir i en l'actualitat el software de xarxa està àmpliament estructurat.

Per reduir la complexitat del disseny, les xarxes s'organitzen com una pila de capes o nivells, cadascun construït sobre la de sota. El propòsit de cada capa és oferir certs serveis a les capes superiors mentre es protegeix les capes dels detalls de com s'implementen realment els serveis oferts[25].

En el model de capes, cada nivell és responsable de resoldre una part específica de les tasques pròpies de la comunicació de xarxes. Perquè aquestes tasques es duguin a terme correctament, l'equip emissor afegeix a les dades que transmet informació de comandament i control a cada capa individual de la pila de protocols. A l'equip que rep aquesta informació, el programari de protocol corresponent a cada capa tracta la informació suplementària per tal d'assegurar la correcta recepció de les dades[1].

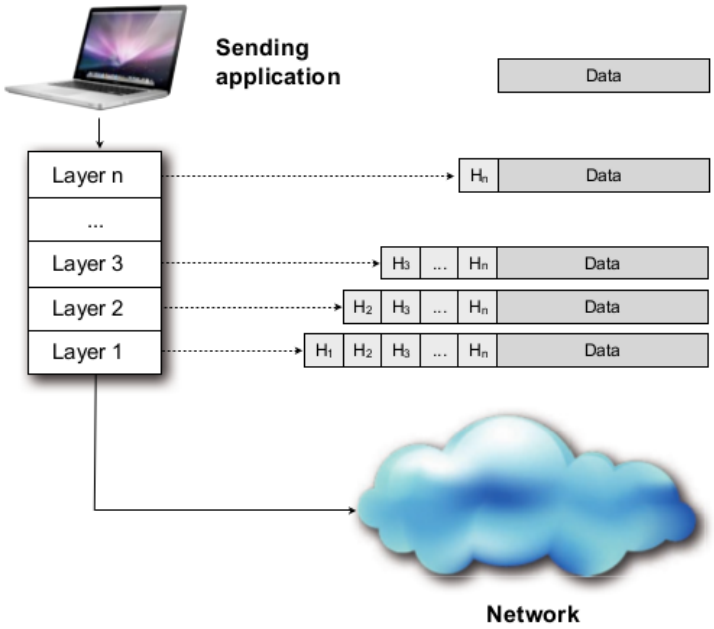


Figura 2.1: Cada capa de la pila de protocols afegeix la seva pròpia informació de comandament i control a les dades transmeses[1]

Les principals arquitectures de comunicació de xarxes són les descrites al model de referència *Open Systems Interconnection*[26] (OSI) de la *International Standards Organization* (ISO), i al model de referència TCP/IP[27], inicialment desenvolupat per la *Defense Advanced Research Projects Agency* (DARPA) del Departament de Defensa dels Estats Units.

El **model de referència OSI** aporta la definició teòrica d'un model arquitectònic de xarxes per promoure una sèrie d'estàndards que especifiquin uns protocols independents dels fabricants i que facilitin la inter-operativitat dels dispositius. Aquest model està compost per 7 capes[25]:

1. Capa física: transmet en brut els bits sobre un canal de comunicació.
2. Capa d'enllaç de dades: transforma una transmissió en brut en una línia que sembla lliure d'errors de transmissió no detectats.
3. Capa de xarxa: determina com els paquets són encaminats des de l'origen a la destinació.
4. Capa de transport: accepta dades de les capes per sobre d'ella, si cal les divideix en unitats més petites, les passa a la capa de xarxa, i assegura que totes les peces arribin correctament a l'altre extrem.
5. Capa de sessió: permet als usuaris de diferents equips controlar el diàleg establert, gestionar l'accés a operacions simultànies, i sincronitzar les transmissions per tal de recuperar-les en cas d'accident.
6. Capa de presentació: s'ocupa de la sintaxi i la semàntica de la informació transmesa.
7. Capa d'aplicació: conté una sèrie de protocols que utilitzen les aplicacions o processos dels usuaris.

D'altra banda, el **model de referència TCP/IP** és l'utilitzat a la xarxa mundial Internet. Els seus principals objectius de disseny són la capacitat per connectar múltiples xarxes de manera transparent, i la capacitat per restablir-se de fallades a parts de la xarxa. El model actual es compon de 4 capes més la capa física[1]:

1. Capa física: hardware.
2. Capa d'enllaç: es centra en la transmissió segura dels paquets de dades en seqüencies de bits agrupats. Els protocols més importants d'aquesta capa són:
   - ATM (*Asynchronous Transfer Mode*).
   - ARP (*Address Resolution Protocol*) RFC 826.
   - RARP (*Reverse Address Resolution Protocol*) RFC 903.
   - NDP (*Neighbor Discovery Protocol*).
   - LLTD (*Link Layer Topology Discovery*).
   - SLIP (*Serial Line Interface Protocol*) RFC 1055.
   - PPP (*Point to Point Protocol*) amb les variants PPPoE (RFC 2516) i PPPoA (RFC 2364).
   - STP (*Spanning Tree Protocol*) establert a l'estàndard IEEE 802.1D.
3. Capa d'internet: permet la comunicació de dos sistemes d'extrem a extrem en ubicacions heterogènies de la xarxa. Els protocols més importants d'aquesta capa són:
   - IP (*Internet Protocol*) amb dos variants, IPv4 (RFC 791) i IPv6 (RFC 2460).
   - ICMP (*Internet Control Message Protocol*) també amb dos variants, IPv4 (RFC 792) i IPv6 (RFC 4443).
   - IPsec (*Internet Protocol Security*) RFC 4835.
   - IGMP (*Internet Group Management Protocol*) descrit als RFC 1112, RFC 2236, RFC 3376.
   - OSPF (*Open Shortest Path First*) RFC 2328.
   - ST 2+ (*Internet Stream Protocol, Version 2*) RFC 1819.
4. Capa de transport: permet que dos programes d'usuari en equips diferents de la xarxa realitzin un intercanvi de dades fiable i orientat a la connexió. Els protocols més importants d'aquesta capa són:
   - TCP (*Transport Control Protocol*) descrit al RFC 793 és el protocol més popular.
   - UDP (*Universal Datagram Protocol*) estandarditzat al RFC 768 és el segon protocol més important.
   - DCCP (*Datagram Congestion Control Protocol*) RFC 4340.
   - RSVP (*Resource Reservation Protocol*) RFC 2205.
   - TLS (*Transport Layer Security*) RFC 2246, RFC 4346 i RFC 5246.
   - SCTP (*Stream Control Transmission Protocol*) RFC 4960.

5. Capa d'aplicació: serveix com interfície per les aplicacions que necessiten comunicar-se entre si a través de la xarxa. Els protocols més importants d'aquesta capa són:
   - TELNET (*TELetype NETwork*) RFC 854.
   - FTP (*File Transfer Protocol*) RFC 959.
   - SMTP (*Simple Mail Transfer Protocol*) RFC 821.
   - HTTP (*Hypertext Transport Protocol*) RFC 2616 entre d'altres.
   - RPC (*Remote Procedure Call*) RFC 1057 i RFC 5531.
   - DNS (*Domain Name System*) RFC 1123.
   - SNMP (*Simple Network Management Protocol*) RFC 3411.
   - RTP (*Real-time Transport Protocol*) RFC 1889.

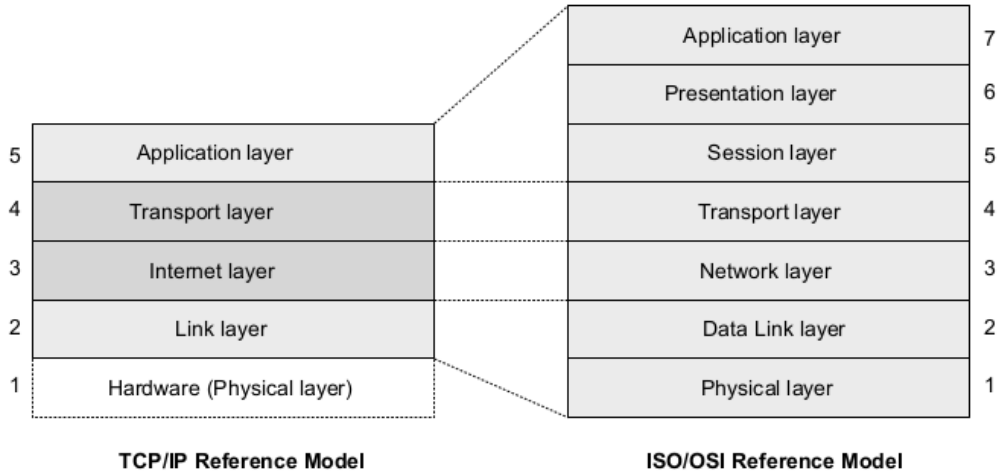


Figura 2.2: Comparativa dels models de referència TCP/IP i OSI[1]

3

Disseny

En la fase de disseny s'ha seguit la metodologia UML, primer identificant el comportament desitjable de l'aplicació i reflectint-lo en un conjunt de casos d'ús. Seguidament s'ha detallat la seqüència d'activitats que ha de seguir l'execució de l'aplicació, per acabar amb la definició de les classes que formen l'arquitectura del software dissenyada.

3.1 Casos d'ús

Descripció del comportament observable del sistema on hi haurà un actor principal, que serà l'usuari de l'aplicació, i la xarxa de computadors com actor secundari.

3.1.1 Mòdul adquisició

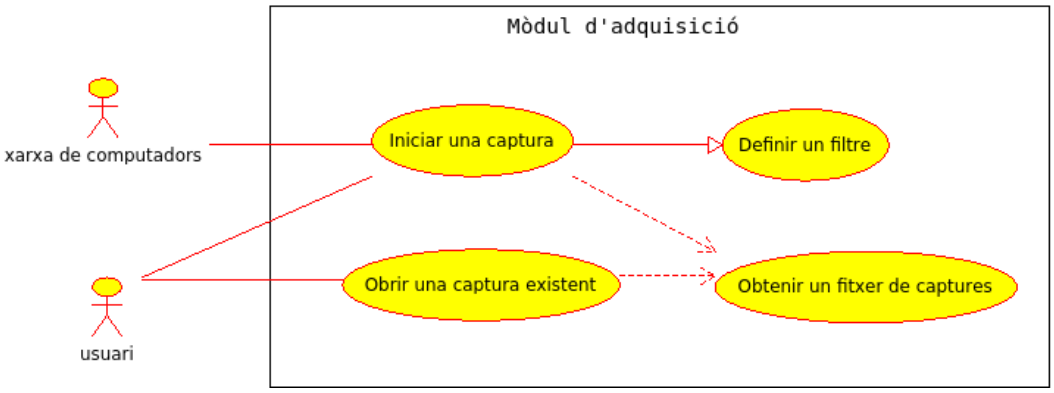


Figura 3.1: Casos d'ús del mòdul d'adquisició

Codi	CU-ADQ-01
Nom	Iniciar una captura
Actor principal	Usuari
Actor secundari	Xarxa de computadors
Àmbit	Mòdul d'adquisició
Precondicions	Usuari ha d'obrir la interfície d'adquisició
Garanties en cas d'èxit	Ruta del fitxer de captures dins del sistema d'arxius.
Escenari principal	L'usuari defineix els paràmetres que s'aplicaran a la captura. L'usuari inicia la captura dels paquets que circulen per la xarxa de computadors. La captura es guarda a un fitxer del sistema d'arxius.

Taula 3.1: Cas d'ús CU-ADQ-01 (Iniciar una captura)

Codi	CU-ADQ-02
Nom	Definir un filtre
Actor principal	Usuari
Àmbit	Mòdul d'adquisició
Precondicions	Usuari ha d'obrir la interfície d'adquisició.
Garanties en cas d'èxit	La captura de paquets rep els filtres a aplicar.
Escenari principal	L'usuari indica per quins paràmetres vol filtrar la captura: IP/MAC, Protocol o Port.

Taula 3.2: Cas d'ús CU-ADQ-02 (Definir un filtre)

Codi	CU-ADQ-03
Nom	Obrir una captura existent
Actor principal	Usuari
Àmbit	Mòdul d'adquisició
Precondicions	Ha d'existir un fitxer amb format *.pcap*
Garanties en cas d'èxit	Ruta del fitxer de captures dins del sistema d'arxius.
Escenari principal	L'usuari executa l'aplicació i obre un fitxer capturat prèviament.

Taula 3.3: Cas d'ús CU-ADQ-03 (Obrir una captura)

Codi	CU-ADQ-04
Nom	Obtenir un fitxer de captures
Actor principal	Sistema
Àmbit	Mòdul d'adquisició
Precondicions	Captura de paquets o selecció de fitxer *.pcap*.
Garanties en cas d'èxit	L'aplicació disposa d'un fitxer *.pcap* vàlid.
Escenari principal	El sistema comprova que el fitxer capturat o seleccionat per l'usuari sigui vàlid.

Taula 3.4: Cas d'ús CU-ADQ-04 (Obtenir un fitxer de captures)

3.1.2 Mòdul resultats

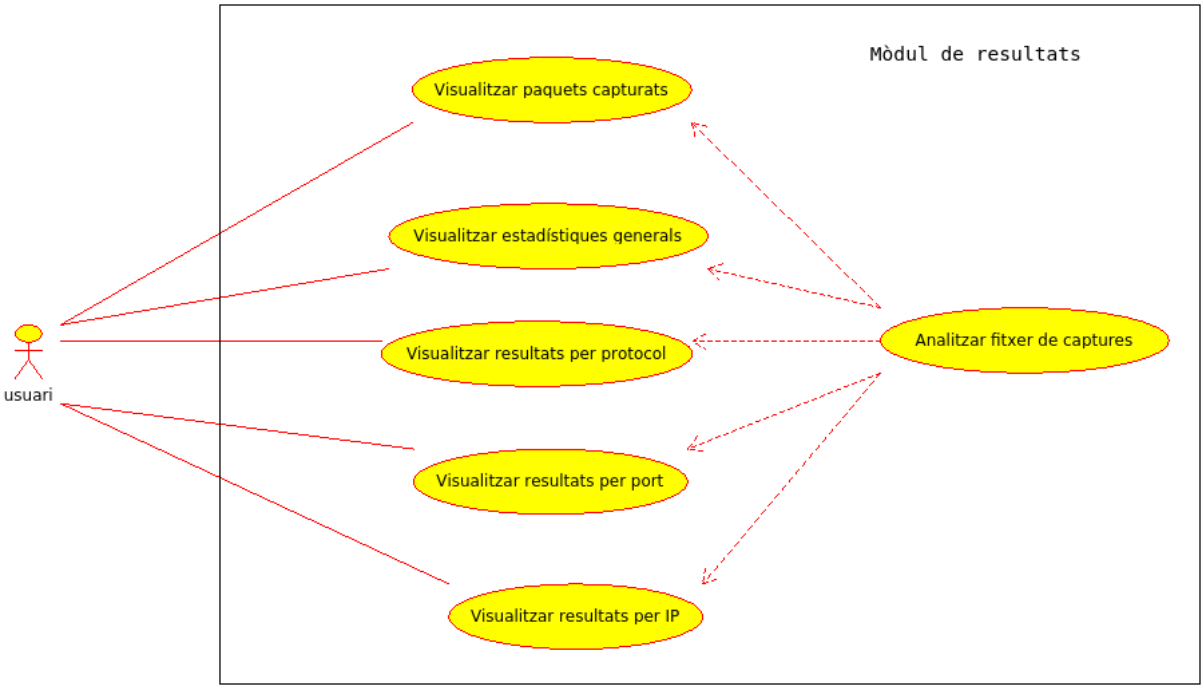


Figura 3.2: Casos d'ús del mòdul de resultats

Codi	CU-RES-01
Nom	Analitzar un fitxer de captures
Actor principal	Sistema
Àmbit	Mòdul de resultats
Precondicions	Mòdul d'adquisició ha de retornar un fitxer de captures vàlid.
Garanties en cas d'èxit	Model de dades amb la informació de la captura.
Escenari principal	A partir del fitxer *.pcap* obtingut al mòdul d'adquisició, l'aplicació carrega el model de dades amb la informació necessària.

Taula 3.5: Cas d'ús CU-RES-01 (Analitzar un fitxer de captures)

Codi	CU-RES-02
Nom	Visualitzar paquets capturats
Actor principal	Usuari
Àmbit	Mòdul de resultats
Precondicions	Mòdul d'adquisició ha de retornar un fitxer de captures vàlid.
Garanties en cas d'èxit	Visualització dels paquets capturats.
Escenari principal	L'usuari visualitza en una taula els paquets capturats, mostrant una taula amb una fila per paquet capturat i columnes amb la data del paquet, el protocol, la MAC i IP d'origen, la MAC i IP de destinació, el port d'origen i destinació, i la grandària del paquet.

Taula 3.6: Cas d'ús CU-RES-02 (Visualitzar paquets capturats)

Codi	CU-RES-03
Nom	Visualitzar estadístiques generals
Actor principal	Usuari
Àmbit	Mòdul de resultats
Precondicions	Mòdul d'adquisició ha de retornar un fitxer de captures vàlid.
Garanties en cas d'èxit	Visualització dades generals de la captura.
Escenari principal	L'usuari visualitza la data i hora d'inici, la data i hora de la finalització, el nombre de paquets i la grandària total de la captura.

Taula 3.7: Cas d'ús CU-RES-03 (Visualitzar estadístiques generals)

3.1.2.1 Cas ús visualitzar resultats per protocol

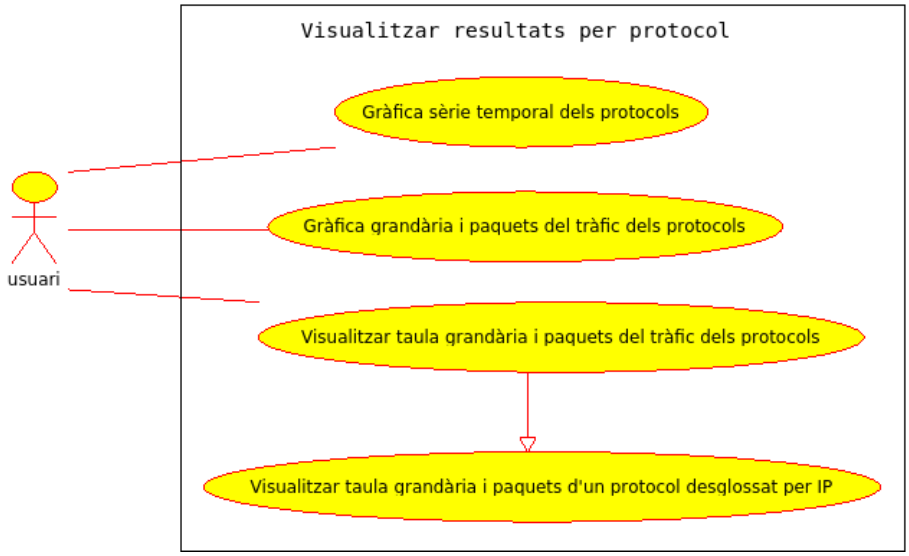


Figura 3.3: Casos d'ús visualitzar resultats per protocol

Codi	CU-RES-04
Nom	Visualitzar resultats per protocol
Actor principal	Usuari
Àmbit	Mòdul de resultats
Precondicions	Mòdul d'adquisició ha de retornar un fitxer de captures vàlid.
Garanties en cas d'èxit	Visualització estadístiques dels protocols.
Escenari principal	L'usuari selecciona la informació dels protocols que vol visualitzar.

Taula 3.8: Cas d'ús CU-RES-04 (Visualitzar resultats per protocol)

Codi	CU-RES-04.1
Nom	Gràfica sèrie temporal dels protocols
Actor principal	Usuari
Àmbit	Mòdul de resultats
Precondicions	Mòdul d'adquisició ha de retornar un fitxer de captures vàlid.
Garanties en cas d'èxit	Visualització gràfica amb l'evolució temporal dels protocols.
Escenari principal	L'usuari visualitza una gràfica amb l'evolució dels paquets i la grandària del tràfic dels diferents protocols per segon de temps.

Taula 3.9: Cas d'ús CU-RES-04.1 (Gràfica sèrie temporal dels protocols)

Codi	CU-RES-04.2
Nom	Gràfica grandària i paquets del tràfic dels protocols
Actor principal	Usuari
Àmbit	Mòdul de resultats
Precondicions	Mòdul d'adquisició ha de retornar un fitxer de captures vàlid.
Garanties en cas d'èxit	Visualització gràfica grandària i nombre de paquets per protocol.
Escenari principal	L'usuari visualitza una gràfica de barres amb un desglòs de la grandària i nombre de paquets del diferents protocols de la captura.

Taula 3.10: Cas d'ús CU-RES-04.2 (Gràfica grandària i paquets del tràfic dels protocols)

Codi	CU-RES-04.3
Nom	Visualitzar taula grandària i paquets del tràfic dels protocols
Actor principal	Usuari
Àmbit	Mòdul de resultats
Precondicions	Mòdul d'adquisició ha de retornar un fitxer de captures vàlid.
Garanties en cas d'èxit	Visualització estadística de la grandària i el nombre de paquets per protocol.
Escenari principal	L'usuari visualitza una taula amb un desglòs de la grandària i nombre de paquets dels diferents protocols de la captura.

Taula 3.11: Cas d'ús CU-RES-04.3 (Visualitzar grandària i paquets del tràfic dels protocols)

Codi	CU-RES-04.4
Nom	Visualitzar taula grandària i paquets d'un protocol desglossat per IP
Actor principal	Usuari
Àmbit	Mòdul de resultats
Precondicions	Visualització estadística de la grandària i el nombre de paquets per protocol.
Garanties en cas d'èxit	Visualització estadística de la grandària i el nombre de paquets del protocol escollit desglossat per IP.
Escenari principal	L'usuari fa doble click sobre el protocol desitjat i visualitza una taula amb un desglòs de la grandària i nombre de paquets de les diferents IP amb aquest tràfic.

Taula 3.12: Cas d'ús CU-RES-04.4 (Visualitzar grandària i paquets d'un protocol per IP)

3.1.2.2 Cas ús visualitzar resultats per port de servei

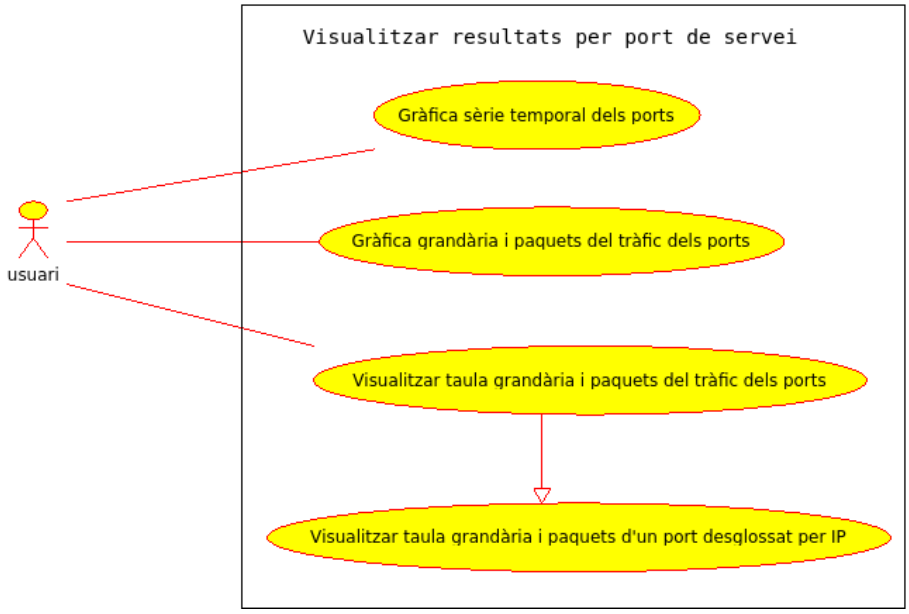


Figura 3.4: Casos d'ús visualitzar resultats per port de servei

Codi	CU-RES-05
Nom	Visualitzar resultats per port de servei
Actor principal	Usuari
Àmbit	Mòdul de resultats
Precondicions	Mòdul d'adquisició ha de retornar un fitxer de captures vàlid.
Garanties en cas d'èxit	Visualització estadístiques dels ports de servei.
Escenari principal	L'usuari selecciona la informació dels ports de servei TCP i UDP que vol visualitzar.

Taula 3.13: Cas d'ús CU-RES-05 (Visualitzar resultats per port de servei)

Codi	CU-RES-05.1
Nom	Gràfica sèrie temporal dels ports de servei
Actor principal	Usuari
Àmbit	Mòdul de resultats
Precondicions	Mòdul d'adquisició ha de retornar un fitxer de captures vàlid.
Garanties en cas d'èxit	Visualització gràfica amb l'evolució temporal del tràfic per port de servei.
Escenari principal	L'usuari visualitza una gràfica amb l'evolució dels paquets i la grandària del tràfic dels diferents ports de servei TCP i UDP per segon de temps.

Taula 3.14: Cas d'ús CU-RES-05.1 (Gràfica sèrie temporal dels ports de servei)

Codi	CU-RES-05.2
Nom	Gràfica grandària i paquets del tràfic dels ports de servei
Actor principal	Usuari
Àmbit	Mòdul de resultats
Precondicions	Mòdul d'adquisició ha de retornar un fitxer de captures vàlid.
Garanties en cas d'èxit	Visualització gràfica grandària i nombre de paquets per port.
Escenari principal	L'usuari visualitza una gràfica de barres amb un desglòs de la grandària i nombre de paquets del diferents ports de servei TCP i UDP del tràfic capturat.

Taula 3.15: Cas d'ús CU-RES-05.2 (Gràfica grandària i paquets del tràfic dels ports de servei)

Codi	CU-RES-05.3
Nom	Visualitzar taula grandària i paquets del tràfic dels ports de servei
Actor principal	Usuari
Àmbit	Mòdul de resultats
Precondicions	Mòdul d'adquisició ha de retornar un fitxer de captures vàlid.
Garanties en cas d'èxit	Visualització estadística de la grandària i el nombre de paquets per port.
Escenari principal	L'usuari visualitza una taula amb un desglòs de la grandària i nombre de paquets dels diferents ports de servei TCP i UDP del tràfic capturat.

Taula 3.16: Cas d'ús CU-RES-05.3 (Visualitzar grandària i paquets del tràfic dels ports de servei)

Codi	CU-RES-05.4
Nom	Visualitzar taula grandària i paquets d'un port de servei desglossat per IP
Actor principal	Usuari
Àmbit	Mòdul de resultats
Precondicions	Visualització estadística de la grandària i el nombre de paquets per port de servei.
Garanties en cas d'èxit	Visualització estadística de la grandària i el nombre de paquets del port de servei escollit desglossat per IP.
Escenari principal	L'usuari fa doble click sobre el port de servei TCP i UDP desitjat i visualitza una taula amb un desglòs de la grandària i nombre de paquets de les diferents IP amb tràfic d'aquest port.

Taula 3.17: Cas d'ús CU-RES-05.4 (Visualitzar grandària i paquets d'un port de servei per IP)

3.1.2.3 Cas ús visualitzar resultats per adreça IP

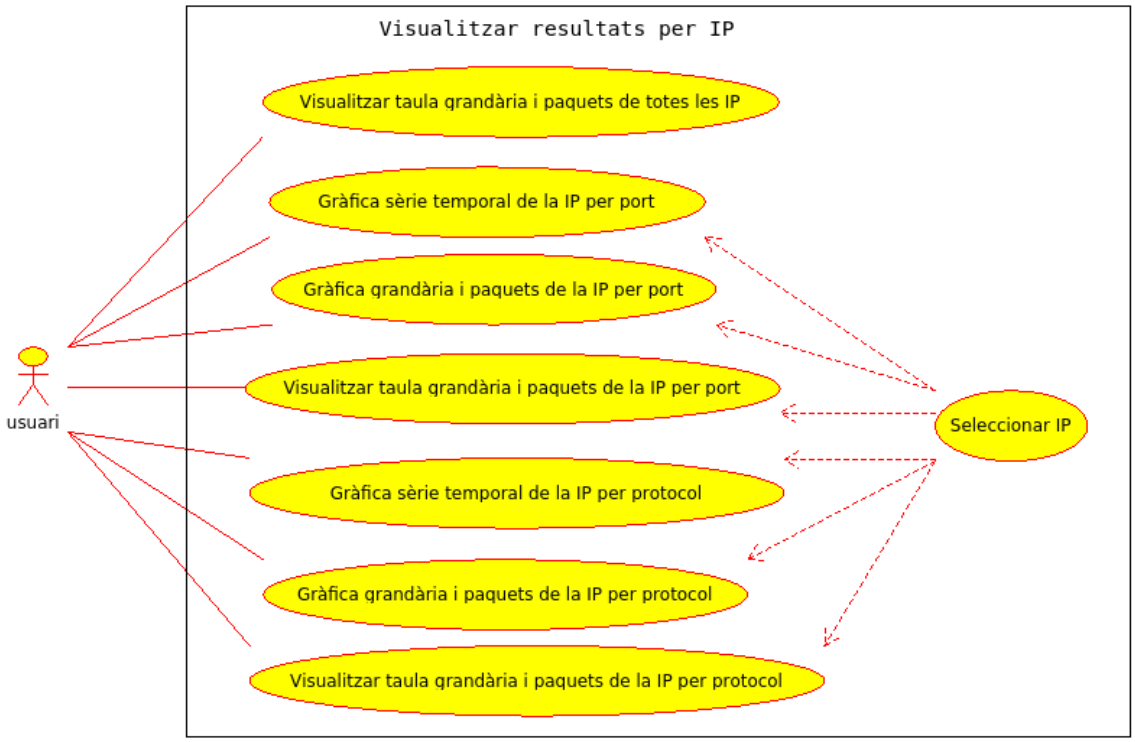


Figura 3.5: Casos d'ús visualitzar resultats per adreça IP

Codi	CU-RES-06
Nom	Visualitzar resultats per adreça IP
Actor principal	Usuari
Àmbit	Mòdul de resultats
Precondicions	Mòdul d'adquisició ha de retornar un fitxer de captures vàlid.
Garanties en cas d'èxit	Visualització estadístiques de les adreces IP.
Escenari principal	L'usuari visualitza la informació del tràfic d'una adreça IP.

Taula 3.18: Cas d'ús CU-RES-06 (Visualitzar resultats per adreça IP)

Codi	CU-RES-06.1
Nom	Visualitzar taula grandària i paquets de totes les IP
Actor principal	Usuari
Àmbit	Mòdul de resultats
Precondicions	Mòdul d'adquisició ha de retornar un fitxer de captures vàlid.
Garanties en cas d'èxit	Visualització taula estadística de la grandària i el nombre de paquets del tràfic capturat agrupat per IP.
Escenari principal	L'usuari visualitza una taula amb un desglòs de la grandària i nombre de paquets del tràfic capturat segons les diferents adreces IP.

Taula 3.19: Cas d'ús CU-RES-06.1 (Visualitzar grandària i paquets de totes les IP)

Codi	CU-RES-06.2
Nom	Seleccionar IP
Actor principal	Usuari
Àmbit	Mòdul de resultats
Precondicions	Mòdul d'adquisició ha de retornar un fitxer de captures vàlid.
Garanties en cas d'èxit	Seleccionada una adreça IP que apareix al tràfic capturat.
Escenari principal	L'usuari d'una llista una adreça IP del conjunt de IP que formen el tràfic capturat.

Taula 3.20: Cas d'ús CU-RES-06.2 (Seleccionar IP)

Codi	CU-RES-06.3
Nom	Gràfica sèrie temporal de la IP per protocol
Actor principal	Usuari
Àmbit	Mòdul de resultats
Precondicions	Fitxer de captures vàlid i IP seleccionada.
Garanties en cas d'èxit	Visualització gràfica amb l'evolució temporal dels protocols que formen el tràfic d'una adreça IP.
Escenari principal	L'usuari visualitza una gràfica amb l'evolució dels paquets i la grandària del tràfic dels diferents protocols associats a una adreça IP per segon de temps.

Taula 3.21: Cas d'ús CU-RES-06.3 (Gràfica sèrie temporal de la IP per protocol)

Codi	CU-RES-06.4
Nom	Gràfica grandària i paquets de la IP per protocol
Actor principal	Usuari
Àmbit	Mòdul de resultats
Precondicions	Fitxer de captures vàlid i IP seleccionada.
Garanties en cas d'èxit	Visualització gràfica grandària i nombre de paquets dels protocols que formen el tràfic d'una adreça IP.
Escenari principal	L'usuari visualitza una gràfica de barres amb la grandària i nombre de paquets del diferents protocols associats a una adreça IP.

Taula 3.22: Cas d'ús CU-RES-06.4 (Gràfica grandària i paquets de la IP per protocol)

Codi	CU-RES-06.5
Nom	Visualitzar taula grandària i paquets de la IP per protocol
Actor principal	Usuari
Àmbit	Mòdul de resultats
Precondicions	Fitxer de captures vàlid i IP seleccionada.
Garanties en cas d'èxit	Visualització taula estadística de la grandària i el nombre de paquets dels protocols que formen el tràfic d'una adreça IP.
Escenari principal	L'usuari visualitza una taula amb la grandària i nombre de paquets dels diferents protocols associats a una adreça IP.

Taula 3.23: Cas d'ús CU-RES-06.5 (Visualitzar grandària i paquets de la IP per protocol)

Codi	CU-RES-06.6
Nom	Gràfica sèrie temporal de la IP per port
Actor principal	Usuari
Àmbit	Mòdul de resultats
Precondicions	Fitxer de captures vàlid i IP seleccionada.
Garanties en cas d'èxit	Visualització gràfica amb l'evolució temporal dels ports de servei que formen el tràfic d'una adreça IP.
Escenari principal	L'usuari visualitza una gràfica amb l'evolució dels paquets i la grandària del tràfic dels diferents ports de servei TCP i UDP associats a una adreça IP per segon de temps.

Taula 3.24: Cas d'ús CU-RES-06.6 (Gràfica sèrie temporal de la IP per port)

Codi	CU-RES-06.7
Nom	Gràfica grandària i paquets de la IP per port
Actor principal	Usuari
Àmbit	Mòdul de resultats
Precondicions	Fitxer de captures vàlid i IP seleccionada.
Garanties en cas d'èxit	Visualització gràfica grandària i nombre de paquets dels ports de servei que formen el tràfic d'una adreça IP.
Escenari principal	L'usuari visualitza una gràfica de barres amb la grandària i nombre de paquets del diferents ports de servei TCP i UDP associats a una adreça IP.

Taula 3.25: Cas d'ús CU-RES-06.7 (Gràfica grandària i paquets de la IP per port)

Codi	CU-RES-06.8
Nom	Visualitzar taula grandària i paquets de la IP per port
Actor principal	Usuari
Àmbit	Mòdul de resultats
Precondicions	Fitxer de captures vàlid i IP seleccionada.
Garanties en cas d'èxit	Visualització taula estadística de la grandària i el nombre de paquets dels ports de servei que formen el tràfic d'una adreça IP.
Escenari principal	L'usuari visualitza una taula amb la grandària i nombre de paquets dels diferents ports de servei TCP i UDP associats a una adreça IP.

Taula 3.26: Cas d'ús CU-RES-06.8 (Visualitzar grandària i paquets de la IP per port)

3.2 Diagrama d'activitats

El funcionament de l'aplicació es reflecteix al següent diagrama d'activitats on s'aprecia l'ordre d'execució de les diferents activitats, i les 2 opcions per obtenir un fitxer de captures: mitjançant una captura en viu, o mitjançant l'apertura d'un fitxer d'una captura prèvia, realitzada des de la mateixa aplicació o des d'un altre *sniffer*.

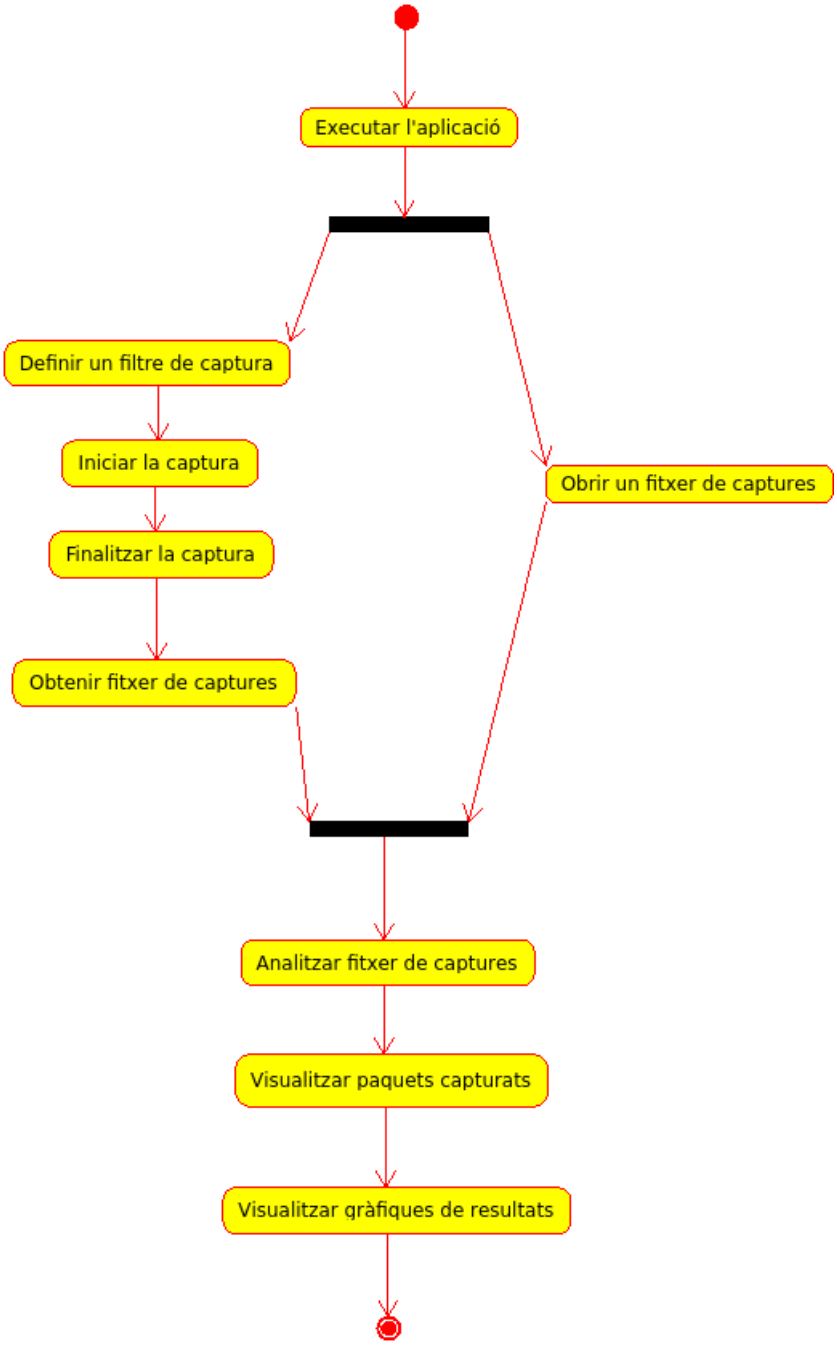


Figura 3.6: Diagrama d'activitats

3.3 Arquitectura de software

El patró de disseny de les classes creades intenten seguir una arquitectura Model-Vista-Controlador (MVC) que separa les dades, la lògica del negoci i la interfície d'usuari.

Les classes que componen el Model de dades són:

- **PortsDict**: classe de suport per incorporar les definicions dels ports de servei dels protocols TCP i UDP[28] segons la *Internet Assigned Numbers Authority* (IANA)[29].
- **ProtocolsDict**: classe de suport per incorporar les definicions dels protocols d'Internet[30] assignats segons la IANA[29].
- **Capture**: classe principal que conté les dades analitzades de la captura i els mètodes per gestionar-les.
- **Protocol**: classe amb la informació relacionada amb un protocol, conté el número identificatiu del protocol i un *dictionary* de classes Traffic amb la sèrie temporal del protocol al llarg de la captura. Aquesta classe és utilitzada per emmagatzemar l'evolució d'un protocol de la captura, però també per registrar l'evolució d'un protocol relacionat amb una IP.
- **Port**: classe amb la informació relacionada amb un port de servei, conté el número identificatiu del port de servei i un *dictionary* de classes Traffic amb la sèrie temporal del port al llarg de la captura. Aquesta classe és utilitzada per emmagatzemar l'evolució d'un port de servei de la captura, però també per registrar l'evolució d'un port relacionat amb una IP.
- **Ip**: classe per emmagatzemar la informació relativa a una IP del fitxer de captures.
- **Traffic**: classe utilitzada per construir les sèries temporals dels protocols i ports de servei guardant per cada segon de temps el número de paquets capturats i la seva grandària.

En quant a les classes del component Controlador serien:

- **CaptureThread**: classe que implementa el *thread* que realitza la captura del tràfic de xarxa.
- **EnableThread**: classe que implementa un *thread* que permet actualitzar la barra de progrés de la captura a la GUI.
- **Signals**: classe que defineix la resposta als esdeveniments de la GUI, només conté mètodes i variables internes.
- **gnagnu**: classe que incorpora les dades introduïdes per l'usuari a la GUI respecte el filtre de captura. També genera alguns diàlegs de la GUI que no s'han definit amb Glade per millorar la personalització, quedant aquesta classe en part al Controlador i en part a la Vista.

Les classes del component Vista es generen dinàmicament a partir del fitxer XML **gnagnu.glade**, creat amb el software Glade[22] i que conté les definicions dels elements que formen la interfície gràfica.

Al capítol següent es dóna una explicació més detallada dels mètodes i atributs d'aquestes classes.

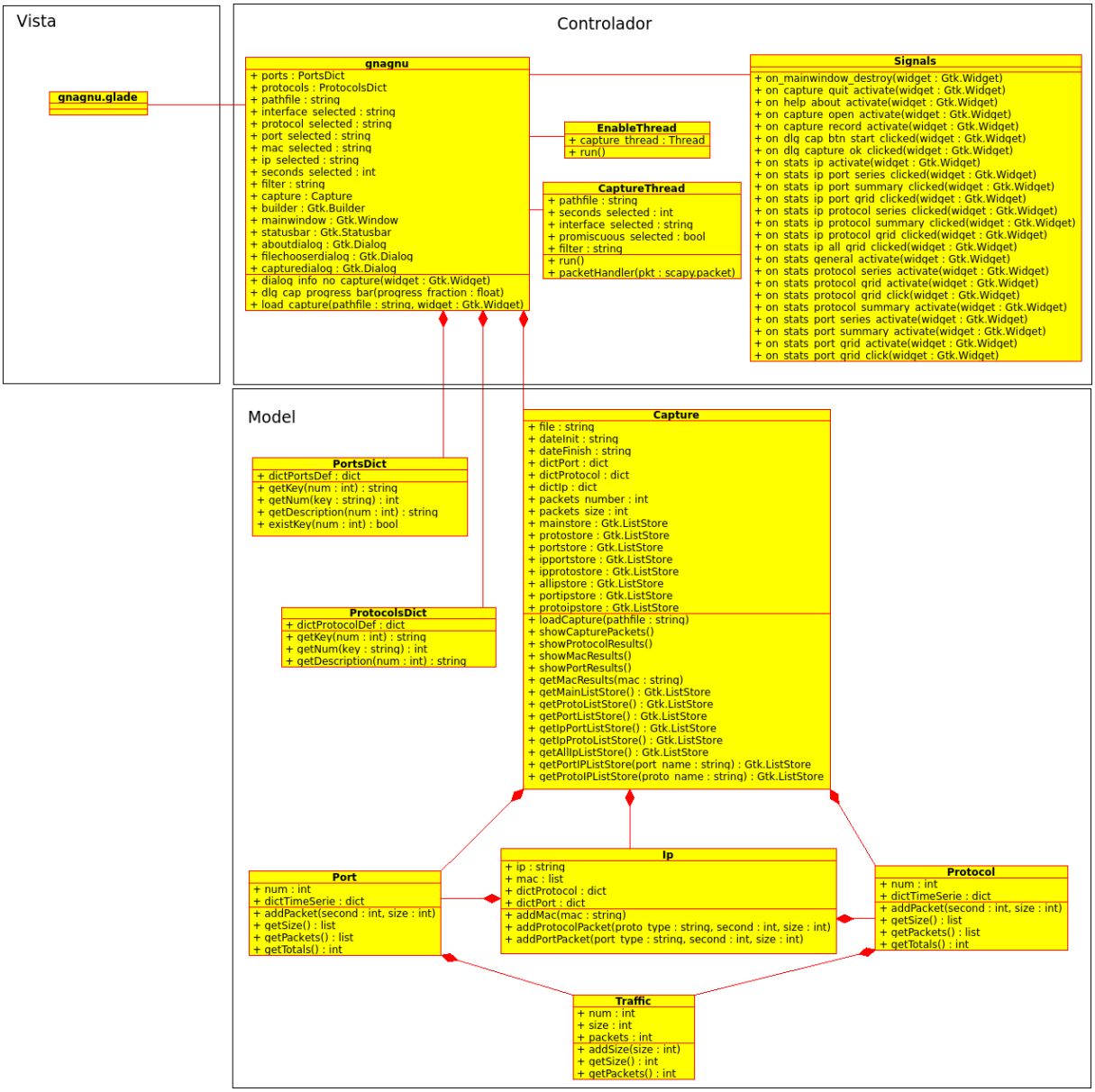


Figura 3.7: Diagrama de classes

4

Implementació

4.1 Component Model

Per implementar el model de dades s'ha codificat cada classe en un fitxer Python.

4.1.1 Classe PortsDict

Aquesta classe s'implementa al fitxer *PortsDict.py* que llegeix el fitxer *service-names-port-numbers.csv* amb format CSV que conté les definicions dels ports TCP i UDP de la IANA[28].

Els valors llegits es guarden en un objecte natiu de tipus *dictionary* on la clau és el número de port i el valor és un objecte *list* amb el mnemotècnic del port i la seva descripció.

La classe implementa els mètodes:

- *getKey* per obtenir el mnemotècnic a partir del número de port de servei.
- *getNum* per obtenir el número de port de servei a partir del mnemotècnic.
- *existKey* per comprovar si està definit un número de port de servei.
- *getDescription* per obtenir la descripció a partir del número de port.

4.1.2 Classe ProtocolsDict

Aquesta classe s'implementa al fitxer *ProtocolsDict.py* que llegeix el fitxer *protocol-numbers.csv* amb format CSV que conté les definicions dels protocols de la capa d'Internet definits per la IANA[30].

Els valors llegits es guarden en un objecte natiu de tipus *dictionary* on la clau és el número de port i el valor és un objecte *list* amb el mnemotècnic del protocol i la seva descripció.

La classe implementa els mètodes:

- *getKey* per obtenir el mnemotècnic a partir del número de protocol.
- *getNum* per obtenir el número de protocol a partir del mnemotècnic.
- *getDescription* per obtenir la descripció a partir del número de protocol.

4.1.3 Classe Traffic

Aquesta classe s'implementa al fitxer *Traffic.py* i consta de 3 atributs:

- Un *int* amb la unitat de temps en segons on s'ha capturat un paquet de tràfic de xarxa.
- Un *int* amb la grandària en Bytes dels paquets capturats en la unitat de temps.
- Un *int* amb el número de paquets capturats en la unitat de temps.

Els seus mètodes són:

- *addSize* per afegir un paquet i la seva grandària a la unitat de temps.
- *getSize* per obtenir la grandària en Bytes dels paquets capturats en la unitat de temps.
- *getPackets* per obtenir el nombre de paquets capturats en la unitat de temps.

4.1.4 Classe Port

Aquesta classe s'implementa al fitxer *Port.py* i consta de 2 atributs:

- Un *int* amb el número de port de servei TCP o UDP.
- Un *dictionary* format d'objectes *Traffic* per representar la sèrie temporal del port de servei.

Els seus mètodes són:

- *addPacket* per afegir un paquet del port de servei i la seva grandària a la unitat de temps corresponent.
- *getSize* obté dues llistes ordenades, una amb les unitats de temps on hi ha paquets del port de servei, i una altra amb les grandàries a cada unitat de temps.
- *getPackets* obté dues llistes ordenades, una amb les unitats de temps on hi ha paquets del port de servei, i una altra amb el nombre de paquets a cada unitat de temps.
- *getTotals* per obtenir la grandària total i el nombre de paquets del port de servei a la captura.

4.1.5 Classe Protocol

Aquesta classe s'implementa al fitxer *Protocol.py* i consta de 2 atributs:

- Un *int* amb el número de protocol de la capa Internet.
- Un *dictionary* format d'objectes *Traffic* per representar la sèrie temporal del protocol.

Els seus mètodes són:

- *addPacket* per afegir un paquet del protocol i la seva grandària a la unitat de temps corresponent.
- *getSize* obté dues llistes ordenades, una amb les unitats de temps on hi ha paquets del protocol, i una altra amb les grandàries a cada unitat de temps.
- *getPackets* obté dues llistes ordenades, una amb les unitats de temps on hi ha paquets del protocol, i una altra amb el nombre de paquets a cada unitat de temps.
- *getTotals* per obtenir la grandària total i el nombre de paquets del protocol a la captura.

4.1.6 Classe Ip

Aquesta classe s'implementa al fitxer *Ip.py* i consta de 4 atributs:

- Un *str* amb l'adreça Internet Protocol.
- Un objecte tipus *list* amb les adreces MAC dels paquets capturats corresponents a la IP, si un equip no canvia de IP l'equivalència serà 1 a 1, però pot donar-se el cas que una mateixa IP s'assigni a diferents MAC al llarg de la captura.
- Un objecte *dictionary* per emmagatzemar la informació dels protocols relacionats amb la IP, la clau és el número de protocol i el valor un objecte *Protocol*.
- Un objecte *dictionary* per emmagatzemar la informació dels ports de servei relacionats amb la IP, la clau és el número de port i el valor un objecte *Port*.

Els seus mètodes són:

- *addMac* per afegir una adreça MAC a l'adreça IP.
- *addProtocolPacket* per afegir a la IP la informació d'un paquet corresponent a un protocol en una unitat de temps.
- *addPortPacket* per afegir a la IP la informació d'un paquet corresponent a un port de servei en una unitat de temps.

4.1.7 Classe Capture

Classe principal del model de dades que s'implementa al fitxer *Capture.py* i consta dels atributs:

- Un *str* amb el nom del fitxer de la captura al sistema d'arxius.
- Un *str* amb la data i hora d'inici de la captura.
- Un *str* amb la data i hora final de la captura.
- Un *int* amb el nombre de paquets que componen la captura.
- Un *int* amb la grandària total en Bytes de la captura.

- Un objecte tipus *dictionary* per emmagatzemar la informació corresponent als protocols, amb clau el número de protocol i valor un objecte *Protocol*.
- Un objecte tipus *dictionary* per emmagatzemar la informació corresponent als ports de servei, amb clau el número de port i valor un objecte *Port*.
- Un objecte tipus *dictionary* per emmagatzemar la informació corresponent a les IP, amb clau la IP i valor un objecte *IP*.
- Una sèrie d'objectes (*GTK.ListStore*) que s'utilitzen als elements *Grid* de la GUI per representar la informació corresponent als protocols, ports de servei i IP del tràfic capturat.

Els seus mètodes són:

- *loadCapture* carrega la informació al model de dades a partir del fitxer de captures. A l'hora de descodificar els paquets de xarxa m'he trobat amb el **problema** que l'objecte *ImpactDecoder* de la llibreria *Impacket* no obtenia correctament alguns número de protocols. Per corregir aquest problema he obert el fitxer de captura amb el programa *Wireshark* i he fixat al codi font la relació correcta d'aquests protocols:
  - Els protocols de *Impacket* 17920 i 18112 corresponen al protocol 2 (IGMP).
  - Els protocols de *Impacket* 24576, 24580, 24582, 24586 i 24590 corresponen al protocol 58 (IPv6-ICMP).
- *getMainListStore* obté un objecte *GTK.ListStore* amb la informació de la captura per la pantalla principal de l'aplicació.
- *getProtoListStore* obté un objecte *GTK.ListStore* amb la informació general de tots els protocols.
- *getPortListStore* obté un objecte *GTK.ListStore* amb la informació general de tots els ports de servei.
- *getIpPortListStore* obté un objecte *GTK.ListStore* amb la informació dels ports de servei relativa a una IP.
- *getIpProtoListStore* obté un objecte *GTK.ListStore* amb la informació dels protocols relativa a una IP.
- *getAllIpListStore* obté un objecte *GTK.ListStore* amb el tràfic de totes les IP.
- *getPortIPListStore* obté un objecte *GTK.ListStore* amb les IP que intervenen en el tràfic d'un port de servei determinat.
- *getProtoIPListStore* obté un objecte *GTK.ListStore* amb les IP que intervenen en el tràfic d'un protocol determinat.

4.2 Component Controlador

Les classes d'aquest component s'han implementat totes en el fitxer *gnagnu.py*.

4.2.1 Classe EnableThread

Aquesta classe deriva de *threading.Thread* i només té el mètode *run* que espera a la finalització del Thread *CaptureThread* per habilitar els botons de la GUI que es deshabiliten durant el temps que l'aplicació està capturant el tràfic de xarxa.

4.2.2 Classe CaptureThread

Classe que deriva de *threading.Thread*, els seus atributs són:

- Un *str* amb el nom del fitxer de la captura al sistema d'arxius.
- Un *int* amb el nombre de segons que l'usuari ha seleccionat per la duració de la captura.
- Un *str* amb el nom de la interfície de xarxa seleccionada per l'usuari on es capturarà el tràfic de xarxa.
- Un *bool* indicant que l'usuari ha seleccionat el mode promiscu en la interfície de xarxa.
- Un *str* amb el valor del filtre que s'aplicarà al fer la captura.

Els seus mètodes són:

- *packetHandler* funció *callback* del programa *Scapy*, en aquest cas només treu el resum del paquet capturat al fitxer de log quan l'aplicació s'executa en mode de depuració.
- *run* realitza la captura del tràfic de xarxa al cridar la funció *sniff* del programa *Scapy*, després guarda la captura en fitxer *.pcap* mitjançant la funció *wrpcap* també d'*Scapy*.

Al fer la captura la llibreria *Pcapy* era la primera opció per ser la més estesa, però ha presentat problemes importants amb el sistema operatiu Linux que l'han descartada en favor del programa *Scapy*.

Un **problema** és que un cop iniciada la captura amb *Pcapy* no atén a la comanda d'aturada i només s'interromp si es fa un *kill* del procés que l'ha iniciada, fins i tot matant el *thread* que executa la captura, aquesta continua en execució.

L'altre **problema** és que la funció de captura de *Pcapy* no fa cas al paràmetre *timeout*, que hauria d'aturar la captura després dels segons indicats, en canvi aquesta continua en execució fins que és mata el procés que l'ha iniciada.

Per aquests motius, realitzar una captura amb *Pcapy* sota el sistema operatiu Linux, que hagi d'aturar-se després d'un temps o sota petició de l'usuari sense matar el procés, és inviable.

4.2.3 Classe Signals

Aquesta classe només conté mètodes i variables internes que responen als esdeveniments de la GUI:

- *on_mainwindow_destroy* tanca l'aplicació.
- *on_capture_quit_activate* tanca la finestra de la captura.
- *on_help_about_activate* mostra el diàleg d'informació de l'aplicació.
- *on_capture_open_activate* diàleg per obrir un fitxer de captura existent. Implementa els casos d'ús CU-ADQ-03, CU-ADQ-04 i CU-RES-01.
- *on_capture_record_activate* diàleg definir un filtre de captura. Implementa el cas d'ús CU-ADQ-02.
- *on_dlg_cap_btn_start_clicked* inicia la captura del tràfic de xarxa. Implementa els casos d'ús CU-ADQ-01 i CU-ADQ-04.
- *on_dlg_capture_ok_clicked* processa el fitxer de la captura per carregar el model de dades. Implementa el cas d'ús CU-RES-01.
- *on_stats_ip_activate* obre una finestra i permet seleccionar una IP de les existents al tràfic capturat per mostrar la seva informació. Implementa el cas d'ús CU-RES-06.2.
- *on_stats_ip_port_series_clicked* obre una finestra i mostra la sèrie temporal dels ports de servei d'una IP seleccionada. Implementa el cas d'ús CU-RES-06.6.
- *on_stats_ip_port_summary_clicked* obre una finestra i mostra el resum del tràfic dels ports de servei d'una IP seleccionada. Implementa el cas d'ús CU-RES-06.7.
- *on_stats_ip_port_grid_clicked* obre una finestra i mostra una taula amb el tràfic dels ports de servei d'una IP seleccionada. Implementa el cas d'ús CU-RES-06.8.
- *on_stats_ip_protocol_series_clicked* obre una finestra i mostra la sèrie temporal dels protocols d'una IP seleccionada. Implementa el cas d'ús CU-RES-06.3.
- *on_stats_ip_protocol_summary_clicked* obre una finestra i mostra el resum del tràfic dels protocols d'una IP seleccionada. Implementa el cas d'ús CU-RES-06.4.
- *on_stats_ip_protocol_grid_clicked* obre una finestra i mostra una taula amb el tràfic dels protocols d'una IP seleccionada. Implementa el cas d'ús CU-RES-06.5.
- *on_stats_ip_all_grid_clicked* obre una finestra i mostra una taula amb la informació del tràfic de la captura de totes les IP. Implementa el cas d'ús CU-RES-06.1.
- *on_stats_general_activate* obre una finestra i mostra la informació general del tràfic de la captura. Implementa el cas d'ús CU-RES-03.
- *on_stats_protocol_series_activate* obre una finestra i mostra la sèrie temporal dels protocols del tràfic de la captura. Implementa el cas d'ús CU-RES-04.1.
- *on_stats_protocol_summary_activate* obre una finestra i mostra el resum del tràfic dels protocols del tràfic de la captura. Implementa el cas d'ús CU-RES-04.2.

- *on_stats_protocol_grid_activate* obre una finestra i mostra una taula amb la informació dels protocols existents al tràfic capturat. Implementa el cas d'ús CU-RES-04.3.
- *on_stats_protocol_grid_click* obre una finestra i mostra una taula amb la informació de la captura segons protocol seleccionat i desglossat per IP. Implementa el cas d'ús CU-RES-04.4.
- *on_stats_port_series_activate* obre una finestra i mostra la sèrie temporal dels ports de servei del tràfic de la captura. Implementa el cas d'ús CU-RES-05.1.
- *on_stats_port_summary_activate* obre una finestra i mostra el resum del tràfic dels ports de servei del tràfic de la captura. Implementa el cas d'ús CU-RES-05.2.
- *on_stats_port_grid_activate* obre una finestra i mostra una taula amb la informació dels ports de servei existents al tràfic capturat. Implementa el cas d'ús CU-RES-05.3.
- *on_stats_port_grid_click* obre una finestra i mostra una taula amb la informació de la captura segons el port de servei seleccionat i desglossat per IP. Implementa el cas d'ús CU-RES-05.4.

4.2.4 Classe gnagnu

Aquesta classe s'encarrega d'inicialitzar els objectes de l'aplicació, construir la interfície gràfica a partir del fixter XML *gnagnu.glade* i connectar els esdeveniments amb la classe *Signals*. Té els següents atributs:

- Un objecte de tipus *PortsDict* amb les definicions dels ports de servei.
- Un objecte de tipus *ProtocolsDict* amb les definicions dels protocols.
- Un *str* amb el nom del fitxer de la captura al sistema d'arxius.
- Un *int* amb el nombre de segons que l'usuari ha seleccionat per la duració de la captura.
- Un *str* amb el nom de la interfície de xarxa seleccionada per l'usuari on es capturarà el tràfic de xarxa.
- Un *str* amb el valor del filtre que s'aplicarà al fer la captura.
- Un *str* amb el protocol seleccionat al filtre de captura.
- Un *str* amb el port de servei seleccionat al filtre de captura.
- Un *str* amb l'adreça MAC seleccionada al filtre de captura.
- Un *str* amb l'adreça IP seleccionada al filtre de captura.
- Un objecte de tipus *Capture* amb el model de dades de la captura.
- Un objecte de tipus *GTK.Builder* que construeix la interfície gràfica.
- Un objecte de tipus *GTK.Statusbar* que representa la barra d'estat de l'aplicació.
- Un objecte de tipus *GTK.Dialog* pel diàleg d'ajuda de l'aplicació.
- Un objecte de tipus *GTK.Dialog* pel diàleg per seleccionar una captura existent.
- Un objecte de tipus *GTK.Dialog* pel diàleg de captura del tràfic de xarxa.

Els seus mètodes són:

- *dialog_info_no_capture* mostra un missatge d'error si no s'ha obert un fitxer de captura prèvia o si no s'ha capturat cap tràfic.
- *dlg_cap_progress_bar* actualitza la barra de progrés del diàleg de captura durant el procés de captura del tràfic de xarxa.
- *load_capture* s'encarrega d'inicialitzar la taula principal de l'aplicació i cridar la càrrega del model de dades a partir del fitxer de la captura.

4.3 Component Vista

La implementació de la interfície gràfica s'ha realitzat mitjançant el software Glade[22], una eina pel desenvolupament ràpid d'aplicacions GTK+ per l'entorn d'escriptori Gnome[31].

La interfície dissenyada es troba al fitxer XML *gnagnu.glade* i es carrega dinàmicament amb l'objecte GtkBuilder de la llibreria GTK+.

Els principals components de la GUI són:

- Finestra principal de l'aplicació, amb el menú d'opcions.

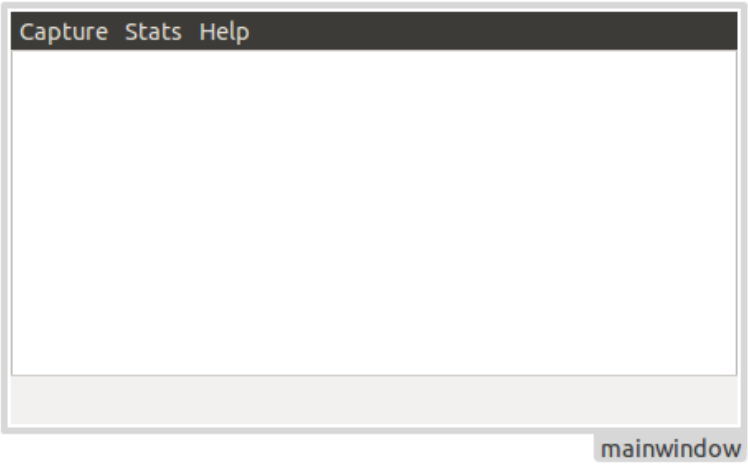


Figura 4.1: Finestra principal

- Submenú *Capture* amb opcions sobre la captura:
  - *Open*, diàleg per obrir una captura existent.

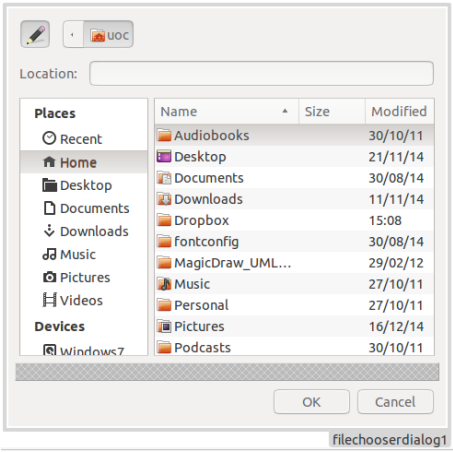


Figura 4.2: Diàleg seleccionar captura existent

- *Record*, diàleg per iniciar una nova captura.

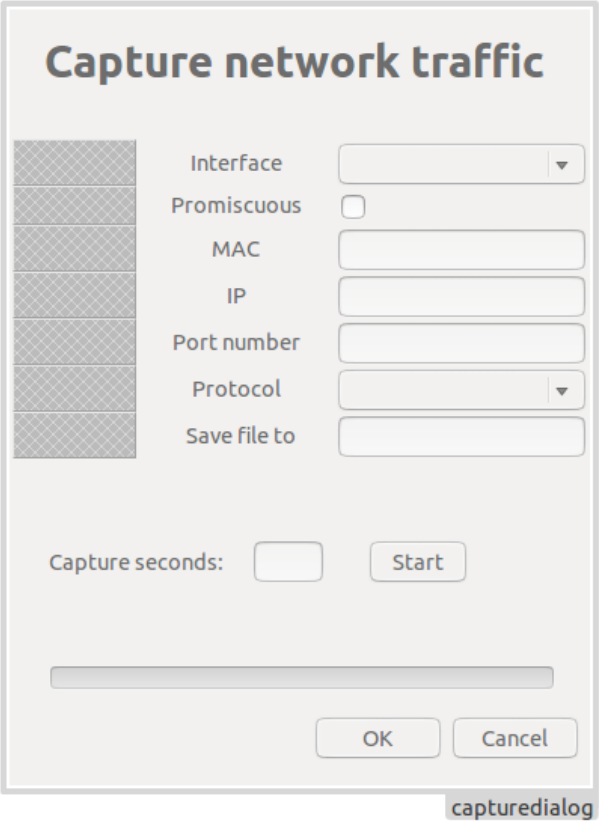


Figura 4.3: Diàleg per iniciar una una nova captura

- *Quit*, per tancar l'aplicació.

- Submenú *Stats*
  - *Capture information*, informació general de la captura.
  - *Protocols*, informació dels protocols que formen el tràfic capturat, les opcions són:
    * *Times series*, obre una gràfica amb la sèrie temporal dels protocols que formen el tràfic capturat.
    * *Tràffic summary*, obre una gràfica de barres resum dels paquets i grandària del tràfic capturat dels principals protocols.
    * *Traffic data grid*, taula amb el desglòs dels paquets i grandària del tràfic de xarxa de tots els protocols capturats.
  - *Ports*, informació dels ports de servei que formen el tràfic capturat, les opcions són:
    * *Times series*, obre una gràfica amb la sèrie temporal dels ports de servei que formen el tràfic capturat.
    * *Tràffic summary*, obre una gràfica de barres amb el resum dels paquets i grandària del tràfic capturat dels principals ports de servei.
    * *Traffic data grid*, taula amb el desglòs dels paquets i grandària del tràfic de xarxa de tots els ports de servei capturats.

– *IP*, informació sobre el tràfic de les IP que formen la captura. Aquesta opció obre un diàleg on es pot consultar una taula amb el tràfic de totes les IP, o seleccionar una IP i veure la seva informació en detall.

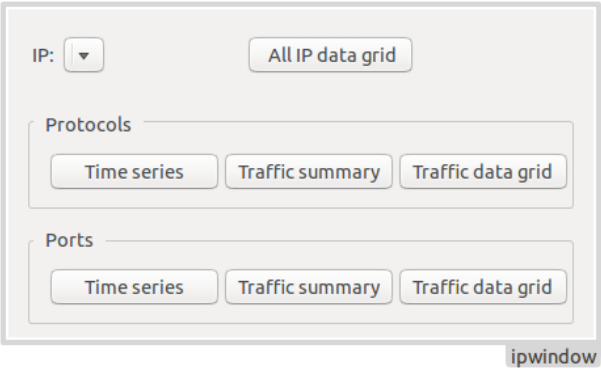


Figura 4.4: Diàleg resultats IP

- Submenú *Help*, mostra el diàleg *About* amb informació sobre l'autoria de l'aplicació.

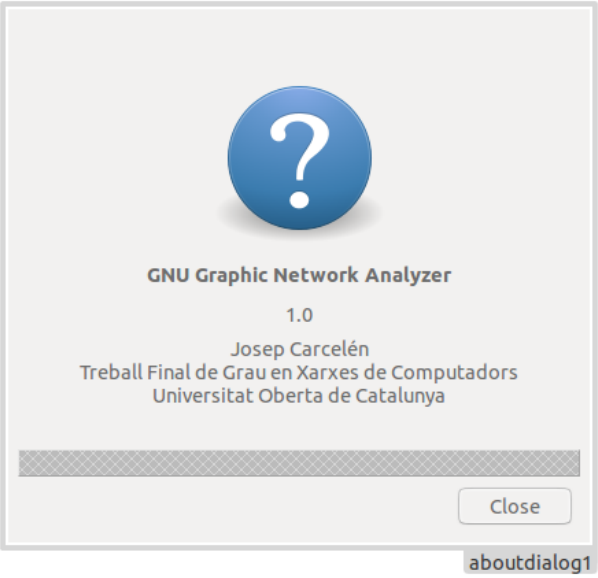


Figura 4.5: Diàleg *About*

5

Producte obtingut

El producte principal desenvolupat és una aplicació que consta d'una sèrie de fitxers amb el codi font en el llenguatge de programació *Python*, aquests fitxers corresponen a les classes de l'apartat arquitectura de software i són:

- *Capture.py*
- *gnagnu.py*
- *Ip.py*
- *Ports.py*
- *PortsDict.py*
- *Protocol.py*
- *ProtocolsDict.py*
- *Traffic.py*

Al codi font de l'aplicació també cal afegir el fitxer **gnagnu.glade** que conté la definició de la interfície gràfica de l'aplicació en llenguatge XML.

El producte obtingut es complementa amb un manual d'ús en un fitxer amb format PDF de nom **gnagnu-user-manual.pdf** que acompanya al codi font. Es pot consultar una versió adaptada d'aquest document a l'apèndix d'aquesta memòria.

Dependències

L'aplicació necessita per la seva execució les següents llibreries de *Python*, que es poden trobar als repositoris de software oficials de les principals distribucions Linux:

- python-pcapy
- python-scapy
- python-impacket
- python-matplotlib

6

Proves

A la fase d'implementació s'han realitzat proves de funcionament per comprovar que el comportament de l'aplicació és l'esperat. Un cop finalitzada l'aplicació s'han fet proves més exhaustives per validar els resultats obtinguts i comprovar el funcionament de totes les opcions del programa.

L'aplicació informa a l'usuari d'errors recuperables, com per exemple intentar visualitzar estadístiques sense un fitxer de captures, però en el cas d'errors no previstos, l'aplicació mostrarà el missatge per consola per tal que l'usuari tingui tota la informació disponible de l'error.

Captura de tràfic de xarxa

Id	Descripció de la prova	Resultat esperat	Correctes	Errònies	Avaluació
1	Captura 10s en xarxa petita	Captura vàlida	10	0	10
2	Captura 10s en xarxa gran	Captura vàlida	5	0	10
3	Captura 60s en xarxa petita	Captura vàlida	5	0	10
4	Captura 30s en xarxa gran	Captura vàlida	4	1	8
5	Filtre Interfície	Només tràfic de la xarxa seleccionada	2	0	10
6	Filtre Promiscu	Tràfic varis equips	5	0	10
7	Filtre MAC	Només tràfic MAC	2	0	10
8	Filtre IP	Només tràfic IP	2	0	10
9	Filtre Port	Només tràfic port	2	0	10
10	Filtre Protocol	Només tràfic protocol	2	0	10
11	Canviar nom captura	Fitxer captura amb el nom seleccionat	2	0	10
12	Obrir fitxer de captura amb Wireshark	Captura vàlida	2	0	10

Taula 6.1: Proves sobre captura de tràfic de xarxa

En una de les proves amb identificador 4, la llibreria *Scapy* ha llançat un error no esperat d'un índex fora de rang i el diàleg de captura ha quedat sense resposta, essent necessari matar el procés de l'aplicació.

Visualització resultats de captures

Id	Descripció de la prova	Resultat esperat	Correctes	Errònies	Avaluació
1	Veure resultats sense una captura	Avís falta captura	2	0	10
2	Sèrie temporal tràfic ports	Mostra gràfica	10	0	10
3	Sèrie temporal tràfic protocols	Mostra gràfica	10	0	10
4	Sèrie temporal tràfic d'una IP	Mostra gràfica	10	0	10
5	Gràfica resum tràfic ports	Mostra gràfica	10	0	10
6	Gràfica resum tràfic protocols	Mostra gràfica	10	0	10
7	Gràfica resum tràfic d'una IP	Mostra gràfica	10	0	10
8	Estadístiques generals	Mostra diàleg	10	0	10
9	Ordenar taula per columna	Ordenació correcta	10	0	10
10	Taula tràfic dels ports	Mostra taula	10	0	10
11	Relació IP tràfic d'un port	Mostra taula	10	0	10
12	Taula tràfic dels protocols	Mostra gràfica	10	0	10
13	Relació IP tràfic d'un protocol	Mostra taula	10	0	10
14	Taula tràfic ports d'una IP	Mostra taula	10	0	10
15	Taula tràfic protocols d'una IP	Mostra taula	10	0	10
16	Taula tràfic totes les IP	Mostra taula	10	0	10

Taula 6.2: Proves visualització de resultats

Proves generals de la interfície

Id	Descripció de la prova	Resultat esperat	Correctes	Errònies	Avaluació
1	Tancar l'aplicació	Sortida del programa	10	0	10
2	Seleccionar una captura prèvia	Mostra resultats	10	0	10
3	Mostrar diàleg *About*	Mostra informació aplicació	5	0	10

Taula 6.3: Proves generals de la interfície

7

Conclusions

Crec que l'aplicació desenvolupada satisfà els objectius inicials establerts, s'ha aconseguit un analitzador gràfic de xarxa en entorn GNU que aporta un valor afegit al proporcionar una interfície intuïtiva i més informació sobre l'ús del tràfic de xarxa capturat respecte d'altres productes existents.

Durant el desenvolupament del projecte he posat en pràctica coneixements apresos a diverses assignatures del Grau, no tan sols de xarxes de computadors, també de gestió de projectes, d'enginyeria del programari, competència comunicativa i de programació orientada a objectes. També he aprés matèries noves, he pogut aprofundir en el llenguatge de programació *Python* i especialment en el desenvolupament d'interfícies d'usuaris amb *GTK*, quedant gratament sorprès per la seva potència i velocitat de programació.

No puc dir el mateix de les llibreries de xarxa utilitzades. L'última versió de *Pcapy* data de l'any 2010 i com he pogut comprovar, el funcionament en sistemes Linux no està del tot suportat. En quant a *Scapy*, sí que té un desenvolupament continuat, però he apreciat problemes d'estabilitat que potser es corregeixin amb futures versions. En tots dos casos la falta de documentació és considerable, com el suport del protocol IPv6.

L'analitzador gràfic de xarxa desenvolupat és millorable, podria proporcionar informació sobre conversacions entre equips, augmentar el rendiment del procés d'anàlisi per reduir el temps d'espera de l'usuari, o donar més suport a IPv6, entre d'altres millores; però tenint en compte el nombre de crèdits del projecte i el gran esforç que he dedicat, estic molt satisfet del resultat obtingut.

Bibliografia

[1] C. Meinel and H. Sack. *Internetworking*. Springer - ISBN: 978-3-642-35391-8, 2013.

[2] Request for Comments: document series contain technical and organizational notes about the Internet, http://www.ietf.org/rfc.html.

[3] Internet Engineering Task Force: comunitat internacional oberta de dissenyadors de xarxa, operadors, proveïdors i investigadors interessats en l'evolució de l'arquitectura d'Internet i el seu bon funcionament, http://www.ietf.org/about/.

[4] Internet Society: és una organització global unida per una causa comú i governada per una variada Junta de Síndics que es dedica a assegurar que Internet segueixi sent oberta, transparent i definida perquè tots puguem gaudir d'ella, http://www.internetsociety.org/who-we-are.

[5] Comparison of packet analyzers, http://en.wikipedia.org/wiki/Comparison_of_packet_analyzers.

[6] Wireshark, https://www.wireshark.org/.

[7] Etherape a graphical network monitor, http://etherape.sourceforge.net/.

[8] Ettercap, http://ettercap.github.io/ettercap/.

[9] Xplico extract from an internet traffic capture the applications data contained, http://www.xplico.org.

[10] SteelCentral Packet Analyzer, http://www.riverbed.com/products/performance-management-control/network-performance-management/packet-analysis.html.

[11] CapAnalysis©Web pcap file Viewer, http://www.capanalysis.net/site/.

[12] NetFlow Traffic Analyzer de Solarwinds©, http://www.solarwinds.com/netflow-traffic-analyzer.aspx.

[13] Scapy©: programa Python per la manipulació interactiva de paquets de xarxa de Philippe Biondi, http://www.secdev.org/projects/scapy/.

[14] Pcapy: interfície de Python per la llibreria libpcap de *Core Labs*©, http://corelabs.coresecurity.com/index.php?module=Wiki&action=view&type=tool&name=Pcapy.

[15] GTK: toolkit multiplataforma per la creació d'interfases gràfiques, http://www.gtk.org/.

[16] Libpcap: llibreria en C/C++ per la captura de tràfic de xarxa, http://www.tcpdump.org/.

[17] Impacket: classes Python destinades a donar accés a paquets de xarxa, de *Core Labs*©, http://corelabs.coresecurity.com/index.php?module=Wiki&action=view&type=tool&name=Impacket.

[18] Matplotlib: llibreria en Python per la creació de gràfiques 2D i 3D, http://matplotlib.org/.

[19] Kile: integrated LaTeXenvironment, http://kile.sourceforge.net/.

[20] Planner: Gnome projec management tool, https://wiki.gnome.org/Apps/Planner/About.

[21] Umbrello: UML modeller, https://umbrello.kde.org/.

[22] Glade: RAD tool to enable quick and easy development of user interfaces for the GTK toolkit, https://glade.gnome.org/.

[23] Pycharm: Python IDE from JetBrains©, https://www.jetbrains.com/pycharm/.

[24] X. Vilajosana, M. Font, E. Lara, and R. Serral. *Material docent de l'assignatura Estructura de xarxes de computadors*. Fundació Universitat Oberta de Catalunya, 2012.

[25] A. Tanenbaum and D. Wetherall. *Computer Networks 5th edition*. Prentice Hall - ISBN: 978-0-13-212695-3, 2011.

[26] Open systems interconnection (OSI), http://www.iso.org/iso/iso_catalogue/catalogue_ics/catalogue_ics_browse.htm?ICS1=35&ICS2=100.

[27] D. Clark. The Design Philosophy of the DARPA Internet Protocols. *SIGCOMM '88 Symposium proceedings on Communications architectures and protocols*, pages 106–114, 1998.

[28] Service Name and Transport Protocol Port Number Registry by Internet Assigned Numbers Authority (IANA), http://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml.

[29] The Internet Assigned Numbers Authority (IANA) is responsible for the global coordination of the DNS Root, IP addressing, and other Internet protocol resources, http://www.internetassignednumbersauthority.org/.

[30] Protocol Numbers by Internet Assigned Numbers Authority (IANA), http://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml.

[31] Gnome: entorn d'escriptori per a sistemes X-Window que forma part oficial del projecte GNU, http://www.gnome.org/.

[32] Llenguatge de programació Python, https://www.python.org/.

[33] Ubuntu operating system, http://www.ubuntu.com/desktop.

[34] Fedora operating system, https://getfedora.org/.

[35] GNU General Public License, http://www.gnu.org/licenses/gpl.html.

Apèndix - Manual d'ús

Instal·lació

L'aplicació no requereix una instal·lació, però sí és necessari disposar de les llibreries *Python*[32] que s'utilitzen.

Al directori de l'aplicació han d'estar els 2 fitxers CSV amb les definicions dels ports i protocols de la IANA[29] que venen amb el codi font:

- *protocol-numbers.csv*[30]
- *service-names-port-numbers.csv*[28]

Al sistema operatiu *Ubuntu 14*[33] amb una instal·lació per defecte, cal executar les següents comandes per obtenir les llibreries dels repositoris de software oficials:

```
sudo apt-get install python-pcapy python-scapy python-impacket
sudo apt-get install python-matplotlib
```

Al sistema operatiu *Fedora 21*[34] amb una instal·lació per defecte, cal executar les següents comandes com a *root* per obtenir les llibreries dels repositoris de software oficials:

```
su -
yum install pcapy scapy python-matplotlib python-impacket
```

En aquest últim cas, també és necessari corregir un *bug* de la llibreria *matplotlib* comentant les línies 86 i 87 del fitxer `/usr/lib/python2.7/site-packages/matplotlib/pyplot.py`, i afegint el següent:

```
# import gobject
# if gobject.MainLoop().is_running():

from gi.repository import GObject
if GObject.MainLoop().is_running():
```

Execució

Si es vol capturar tràfic de xarxa, l'aplicació necessita executar-se amb permisos de *root* per poder adquirir amb privilegis la interfície de xarxa, per tant des d'un terminal cal situar-se al directori on es troba el codi font de l'aplicació i executar-la amb la comanda "**sudo**" o "**su -**", entrant la *password* d'administració:

```
# Ubuntu 15
sudo python gnagnu.py

# Fedora 21
su -
python gnagnu.py
```

Si només es vol analitzar una captura de xarxa existent, no cal executar l'aplicació amb privilegis de *root* i es pot executar sense la comanda "**sudo**" o "**su -**":

```
python gnagnu.py
```

Després de l'execució la finestra de l'aplicació ha d'aparèixer buida només amb el menú principal:

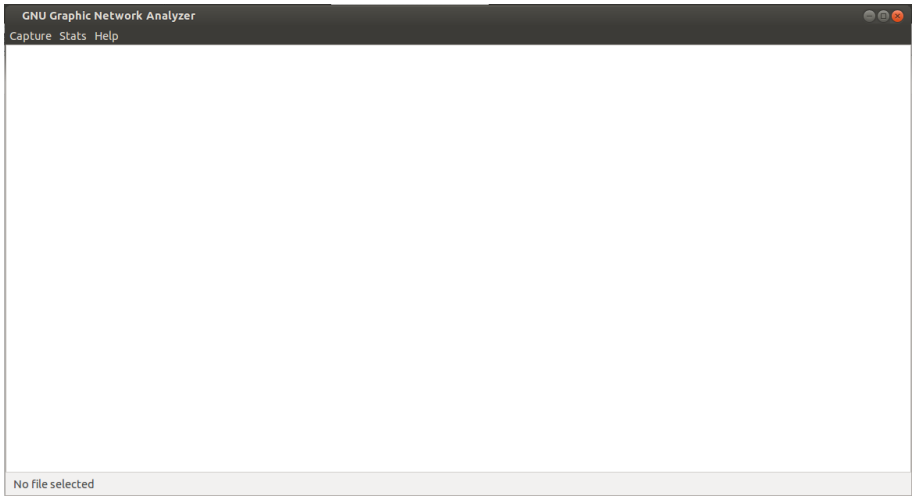


Figura 1: Finestra principal de l'aplicació

Obtenir una captura

Hi ha dues formes d'obtenir una captura, iniciant una captura del tràfic de xarxa durant l'execució de l'aplicació, o bé obrint un fitxer amb el tràfic de xarxa capturat prèviament.

Iniciar una captura

Per iniciar una nova captura del tràfic de xarxa, cal anar al menú *Capture*, opció *Record*, on s'obrirà el diàleg de captura:

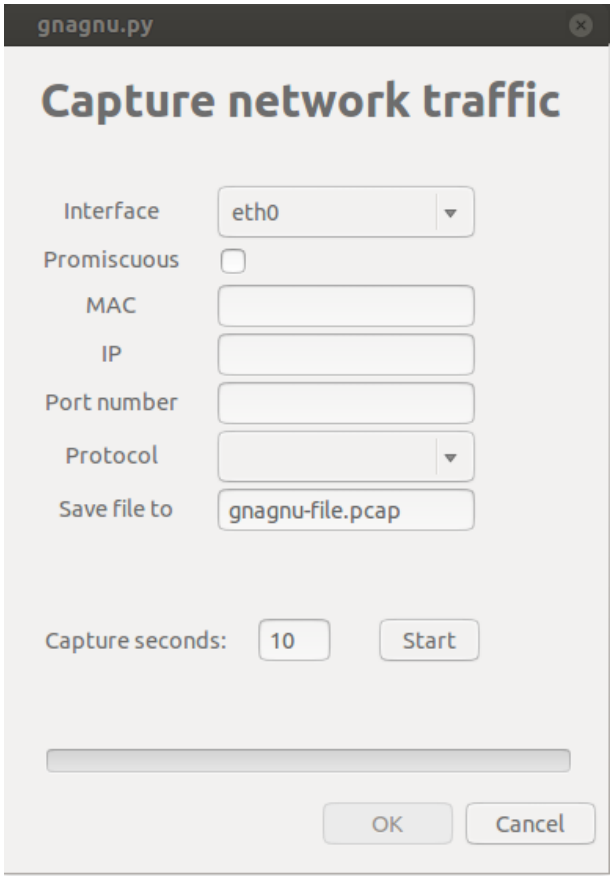


Figura 2: Diàleg de captura

En aquest diàleg podem indicar els filtres de la captura:

- **Interface**: la interfície de xarxa on volem capturar el tràfic, el desplegable mostrarà les interfícies disponibles de l'equip.
- **Promiscuous**: si volem que el tràfic de xarxa sigui capturat en mode promiscu, tot el tràfic que la interfície pugui captar, o només el tràfic amb origen o destí a l'equip que fa la captura.
- **MAC**: per indicar que només volem el tràfic d'una determinada adreça MAC. El format ha de ser del tipus xx:xx:xx:xx:xx:xx.
- **IP**: per indicar que només volem el tràfic d'una determinada adreça IP.
- **Port number**: per filtrar el tràfic per un determinat número de port de servei.
- **Protocol**: per indicar que només volem el tràfic d'un determinat protocol. El desplegable mostra els protocols disponibles: ARP, DECNET, ETHER, FDDI, IP, IP6, RARP, TCP, TR, UDP i WLAN.

Al camp **Save file to** podem indicar el nom del fitxer on es guardarà la captura amb format *.pcap*, al directori on s'executa l'aplicació. D'aquesta manera es pot recuperar la captura posteriorment. Per defecte l'aplicació proposa el nom *gnagnu-file.pcap*.

Al camp **Capture seconds** indicarem els segons que volem que l'equip estigui capturant el tràfic de xarxa, per defecte 10 segons. Hem de tenir en compte que com més gran sigui aquest paràmetre més espai de disc ocuparà la captura i més trigarà el procés d'anàlisi del tràfic capturat.

Una vegada seleccionats els paràmetres desitjats, iniciarem la captura mitjançant el botó **Start**. Durant el procés de captura una barra de progrés a la part inferior del diàleg indicarà l'evolució de la captura.

Si ho desitgem podem realitzar una nova captura tornant a prémer el botó **Start** o tancar el diàleg sense analitzar la captura mitjançant el botó **Cancel**.

Finalitzada la captura, amb el botó **OK** l'aplicació passarà a analitzar el tràfic de xarxa capturat. Es mostrarà un missatge d'avís perquè el procés d'anàlisi es pot allargar segons la grandària de la captura. Per continuar cal prémer el botó **OK** de l'avís i esperar que finalitzi l'anàlisi.

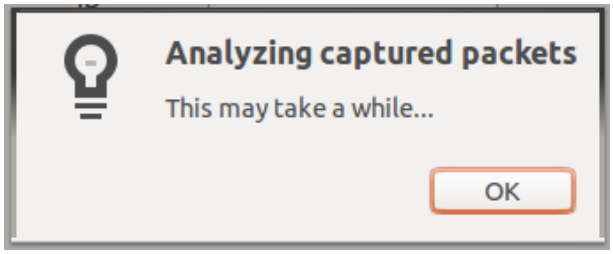


Figura 3: Avís del procés d'anàlisi

Obrir una captura prèvia

Per analitzar un fitxer amb una captura del tràfic de xarxa efectuada amb anterioritat, cal anar al menú *Capture*, opció *Open*, on s'obrirà el diàleg per seleccionar el fitxer al sistema d'arxius:

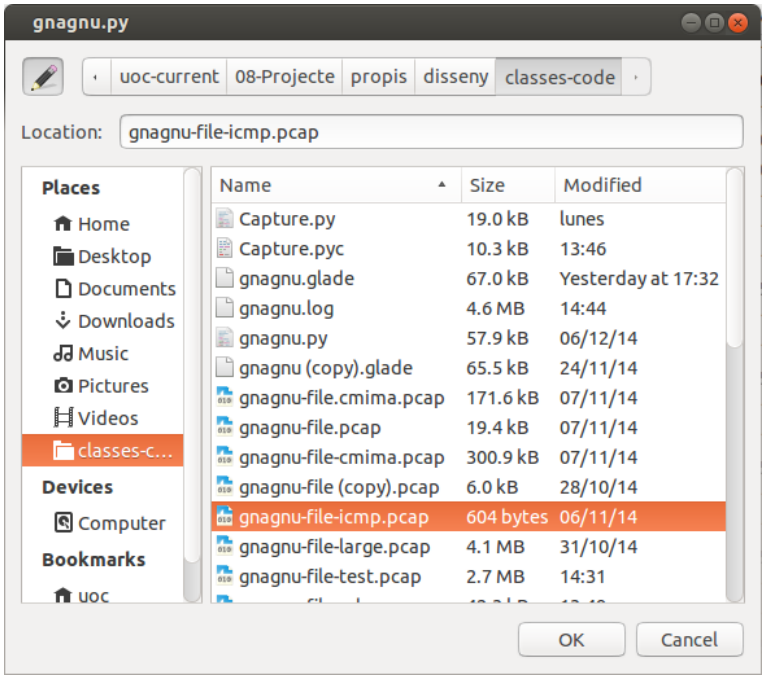


Figura 4: Diàleg per seleccionar una captura prèvia

Un cop seleccionat el fitxer, es passarà a analitzar el tràfic de xarxa que conté. L'aplicació mostrarà un missatge d'avís perquè el procés d'anàlisi es pot allargar segons la grandària de la captura. Per continuar cal prémer el botó **OK** de l'avís i esperar que finalitzi l'anàlisi.

Anàlisi dels resultats

Un cop analitzat el tràfic de xarxa, els paquets capturats es mostren a la finestra principal de l'aplicació.

GNU Graphic Network Analyzer

Capture Stats Help

Order	Time	Protocol	Source IP	Source MAC	Src Port	Destination IP	Destination MAC	Dst Port	Bytes
1	12:43:33.636764	UDP	80.58.61.250	e4:c1:46:a9:95:30	53/DOMAIN	192.168.1.35	08:00:27:78:41:2f	33120	680
2	12:43:33.637015	UDP	127.0.1.1	00:00:00:00:00:00	53/DOMAIN	127.0.0.1	00:00:00:00:00:00	60428	680
3	12:43:33.650191	UDP	80.58.61.250	e4:c1:46:a9:95:30	53/DOMAIN	192.168.1.35	08:00:27:78:41:2f	2080/AUTODESK-NLM	222
4	12:43:33.650295	UDP	127.0.1.1	00:00:00:00:00:00	53/DOMAIN	127.0.0.1	00:00:00:00:00:00	60428	222
5	12:43:33.650297	UDP	127.0.1.1	00:00:00:00:00:00	53/DOMAIN	127.0.0.1	00:00:00:00:00:00	60428	222
6	12:43:33.656655	TCP	173.194.45.194	e4:c1:46:a9:95:30	443/HTTPS	192.168.1.35	08:00:27:78:41:2f	58736	66
7	12:43:33.656675	TCP	173.194.45.194	e4:c1:46:a9:95:30	443/HTTPS	192.168.1.35	08:00:27:78:41:2f	58736	124
8	12:43:33.658757	TCP	173.194.45.194	e4:c1:46:a9:95:30	443/HTTPS	192.168.1.35	08:00:27:78:41:2f	58736	1466
9	12:43:33.658782	TCP	192.168.1.35	08:00:27:78:41:2f	58736	173.194.45.194	e4:c1:46:a9:95:30	443/HTTPS	66
10	12:43:33.658831	TCP	173.194.45.194	e4:c1:46:a9:95:30	443/HTTPS	192.168.1.35	08:00:27:78:41:2f	58736	1466
11	12:43:33.661391	TCP	173.194.45.194	e4:c1:46:a9:95:30	443/HTTPS	192.168.1.35	08:00:27:78:41:2f	58736	1466
12	12:43:33.661408	TCP	192.168.1.35	08:00:27:78:41:2f	58736	173.194.45.194	e4:c1:46:a9:95:30	443/HTTPS	66
13	12:43:33.661446	TCP	173.194.45.194	e4:c1:46:a9:95:30	443/HTTPS	192.168.1.35	08:00:27:78:41:2f	58736	1466
14	12:43:33.664065	TCP	173.194.45.194	e4:c1:46:a9:95:30	443/HTTPS	192.168.1.35	08:00:27:78:41:2f	58736	1466
15	12:43:33.664080	TCP	192.168.1.35	08:00:27:78:41:2f	58736	173.194.45.194	e4:c1:46:a9:95:30	443/HTTPS	66
16	12:43:33.664114	TCP	173.194.45.194	e4:c1:46:a9:95:30	443/HTTPS	192.168.1.35	08:00:27:78:41:2f	58736	1466
17	12:43:33.666870	TCP	173.194.45.194	e4:c1:46:a9:95:30	443/HTTPS	192.168.1.35	08:00:27:78:41:2f	58736	1466
18	12:43:33.666880	TCP	192.168.1.35	08:00:27:78:41:2f	58736	173.194.45.194	e4:c1:46:a9:95:30	443/HTTPS	66
19	12:43:33.666906	TCP	173.194.45.194	e4:c1:46:a9:95:30	443/HTTPS	192.168.1.35	08:00:27:78:41:2f	58736	1466
20	12:43:33.670160	TCP	173.194.45.194	e4:c1:46:a9:95:30	443/HTTPS	192.168.1.35	08:00:27:78:41:2f	58736	1466
21	12:43:33.670178	TCP	192.168.1.35	08:00:27:78:41:2f	58736	173.194.45.194	e4:c1:46:a9:95:30	443/HTTPS	66
22	12:43:33.670213	TCP	173.194.45.194	e4:c1:46:a9:95:30	443/HTTPS	192.168.1.35	08:00:27:78:41:2f	58736	1466
23	12:43:33.672741	TCP	173.194.45.194	e4:c1:46:a9:95:30	443/HTTPS	192.168.1.35	08:00:27:78:41:2f	58736	1466

Using file: /home/uoc/Dropbox/uoc-current/08-Projecte/propis/disseny/classes-code/gnagnu-file-vbox.pcap

Figura 5: Finestra principal de l'aplicació amb el tràfic de xarxa capturat

Els camps de la taula són:

- **Order** amb el número d'ordre del paquet dintre de la captura.
- **Time** és l'hora, minut, segon i microsegon de quan s'ha capturat el paquet.
- **Protocol** de la capa d'Internet del paquet capturat.
- **Source IP** és l'adreça IP de l'equip que ha emès el paquet de xarxa.
- **Source MAC** adreça MAC de l'equip que ha emès el paquet de xarxa.
- **Src Port** port de servei origen a l'equip que ha emès el paquet de xarxa.
- **Destination IP** és l'adreça IP de l'equip que rep el paquet de xarxa.
- **Destination MAC** adreça MAC de l'equip que rep el paquet de xarxa.
- **Src Port** port de servei destí a l'equip que rep el paquet de xarxa.
- **Bytes** és la grandària del paquet capturat.

La taula amb els resultats es pot ordenar per qualsevol camp prement la capçalera de cada columna.

Informació general

La informació global de la captura es pot consultar al menú *Stats* opció *Capture information*.

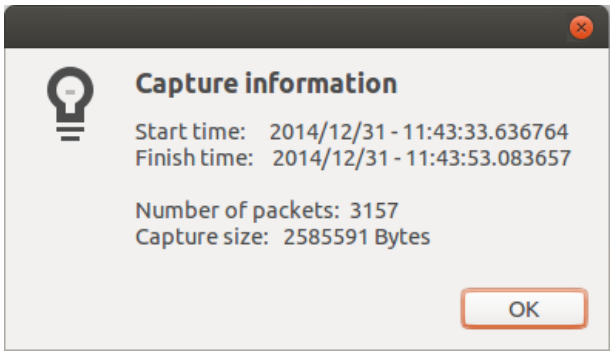


Figura 6: Informació general de la captura

El diàleg mostra la informació següent:

- La data i hora d'inici de la captura.
- La data i hora de finalització de la captura.
- El nombre de paquets del tràfic de xarxa capturat.
- La grandària en Bytes del tràfic de xarxa capturat.

Resultats per protocol

Al menú *Stats* opció *Protocols* podem examinar els protocols que intervenen al tràfic de xarxa capturat.

Sèrie temporal

A l'opció **Time series** podem examinar la gràfica de l'evolució temporal del tràfic capturat segons els protocols que el formen.

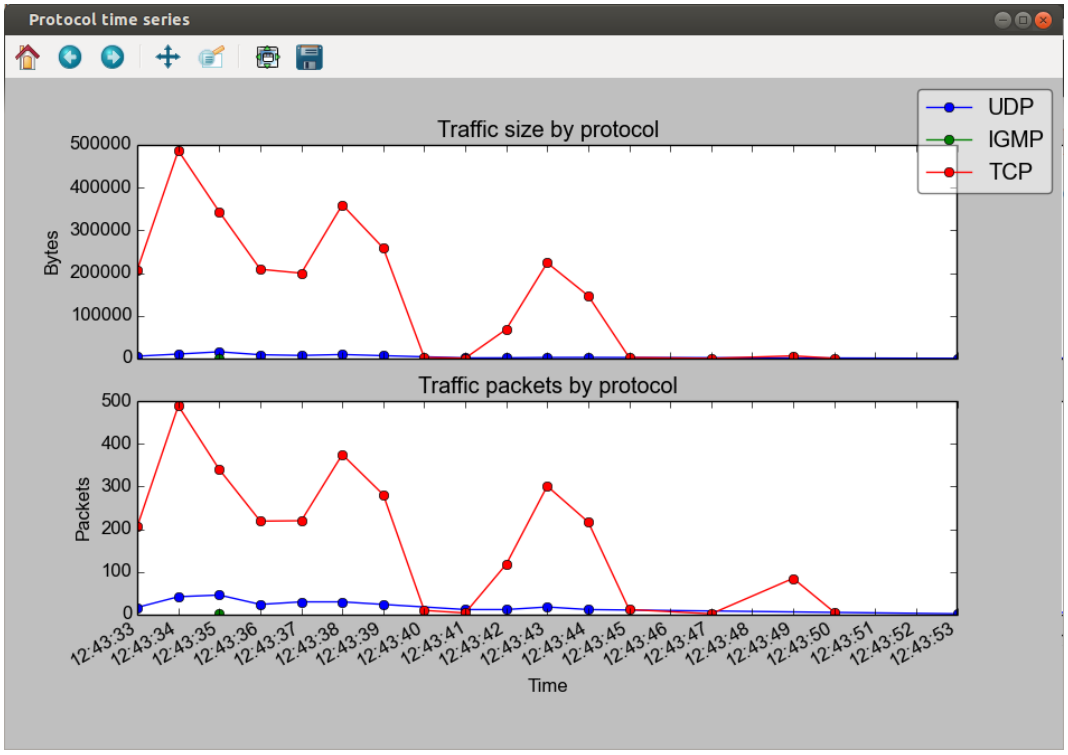


Figura 7: Sèrie temporal dels protocols

La gràfica superior mostra l'evolució en Bytes dels paquets capturats segons el seu protocol, mentre la gràfica inferior mostra l'evolució del nombre de paquets.

Totes les gràfiques disposen d'una barra d'eines a la part superior de la finestra per moure els eixos, fer zoom, configurar els espais interlineals i per guardar la gràfica en un arxiu PNG.

Figura 8: Barra d'eines a les gràfiques

Resum del tràfic

A l'opció **Traffic summary** podem examinar la gràfica de barres dels protocols principals que formen el tràfic capturat, amb la seva mida i la grandària en Bytes.

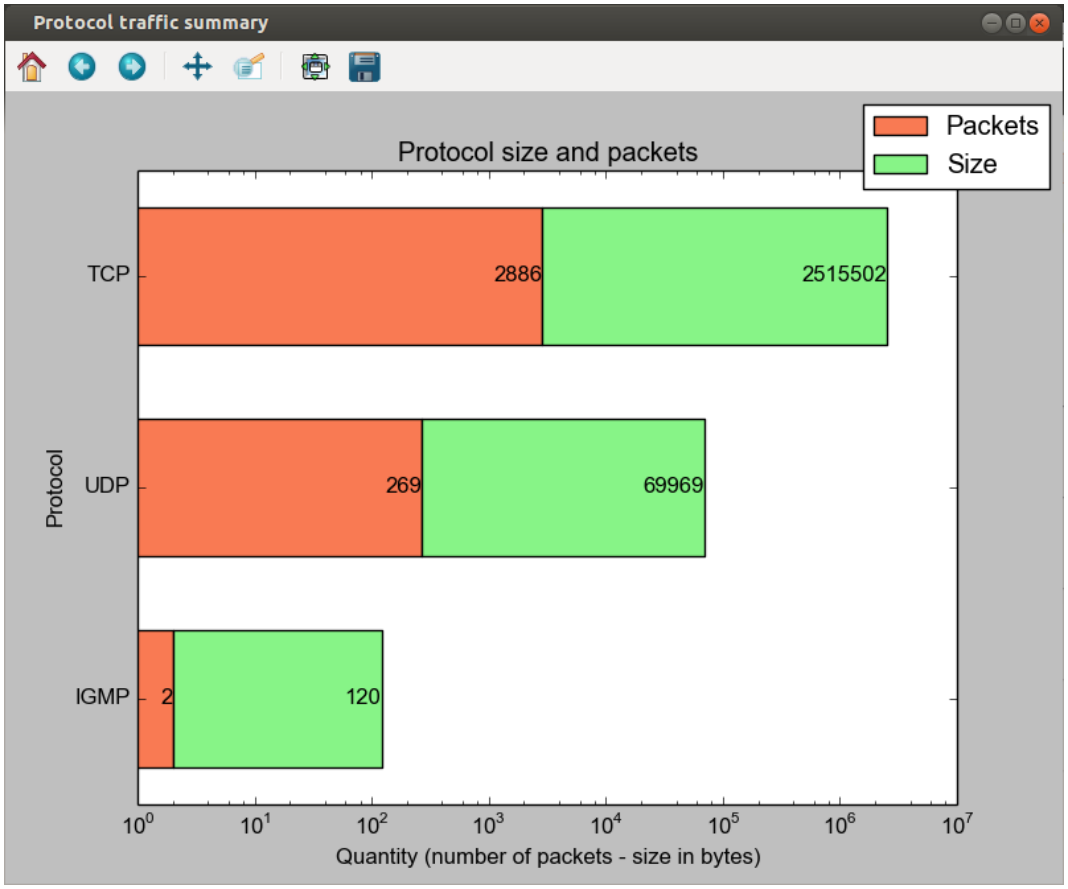


Figura 9: Resum dels protocols de la captura

L'eix X de la gràfica té una escala logarítmica per poder mostrar alhora els paquets y la grandària.

Detall del tràfic

A l'opció **Traffic data grid** podem examinar una taula amb el detall del tràfic per protocol, amb la seva mida i la grandària en Bytes.

Protocol traffic grid

Protocol	Description	Packets	Size (bytes)
UDP	User Datagram	269	69969
IGMP	Internet Group Management	2	120
TCP	Transmission Control	2886	2515502

Figura 10: Detall dels protocols de la captura

Si premem sobre un protocol en concret s'obrirà una nova finestra amb una taula on es mostra la relació de les IP de la captura que intervenen en el tràfic del protocol.

IP traffic detail

Key	IP	Packets	Size (bytes)
UDP	127.0.1.1	177	45774
UDP	127.0.0.1	177	45774
UDP	80.58.61.250	88	23138
UDP	255.255.255.255	1	144
UDP	192.168.1.35	90	23907
UDP	192.168.1.37	2	288
UDP	80.58.61.254	2	769
UDP	192.168.1.255	1	144

Figura 11: Relació de les IP d'un protocol

Les taules es poden ordenar per qualsevol camp prement la capçalera de cada columna.

Resultats per port de servei

Al menú *Stats* opció *Ports* podem examinar els ports de servei que intervenen al tràfic de xarxa capturat.

Sèrie temporal

A l'opció **Time series** podem examinar la gràfica de l'evolució temporal del tràfic capturat segons els ports de servei que el formen.

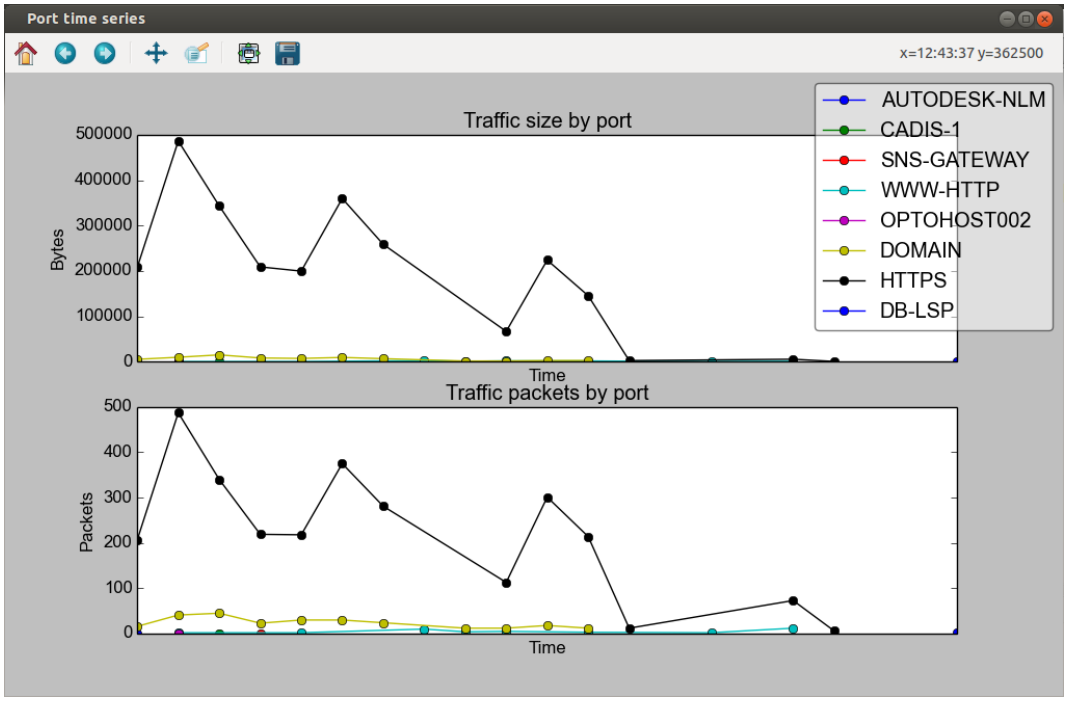


Figura 12: Sèrie temporal dels ports de servei

La gràfica superior mostra l'evolució en Bytes dels paquets capturats segons el seu port, mentre la gràfica inferior mostra l'evolució del nombre de paquets.

Al situar-se amb el ratolí en un punt de la gràfica, a la part superior dreta de la finestra es pot veure les coordenades X i Y del punt.

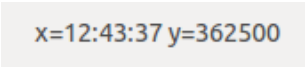


Figura 13: Coordenades d'un punt de la gràfica

Resum del tràfic

A l'opció **Traffic summary** podem examinar la gràfica de barres dels ports de servei principals que formen el tràfic capturat, amb la seva mida i la grandària en Bytes.

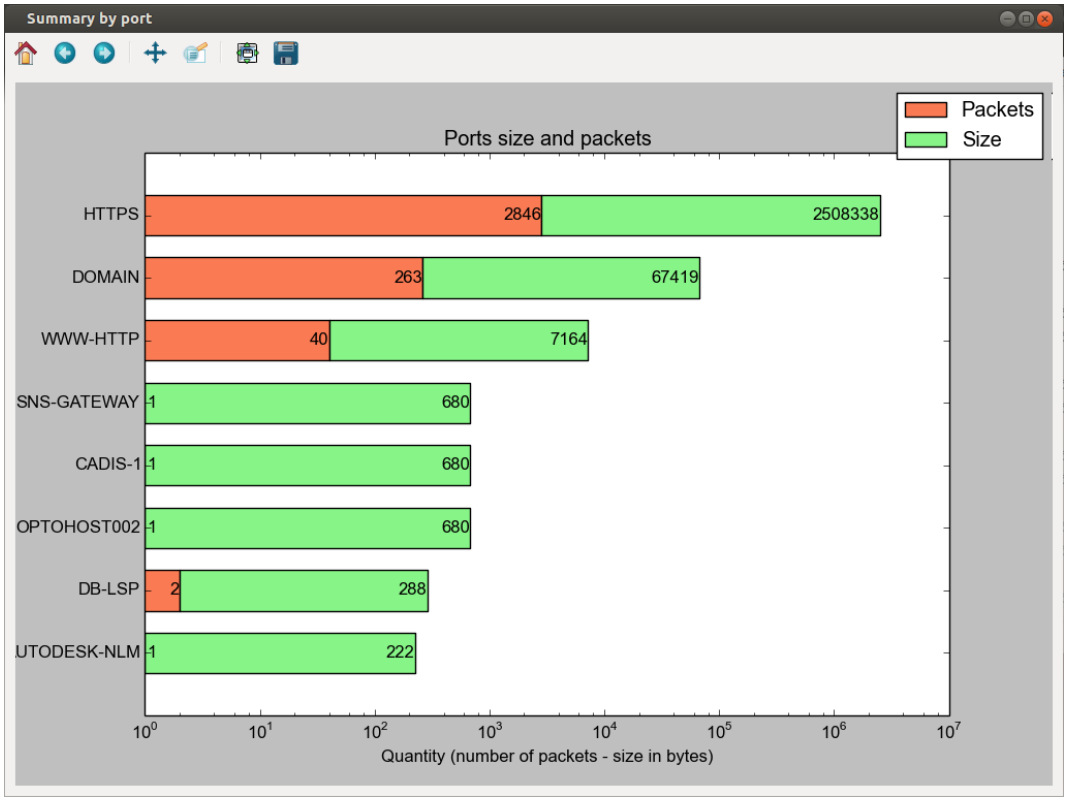


Figura 14: Resum dels ports de servei de la captura

L'eix X de la gràfica té una escala logarítmica per poder mostrar alhora els paquets y la grandària.

Detall del tràfic

A l'opció **Traffic data grid** podem examinar una taula amb el detall del tràfic per port de servei, amb la seva mida i la grandària en Bytes.

Port traffic grid

Port	Description	Packets	Size (bytes)
AUTODESK-NLM	Autodesk NLM (FLEXlm)	1	222
CADIS-1	Cadis License Management	1	680
SNS-GATEWAY	SNS Gateway	1	680
WWW-HTTP	World Wide Web HTTP	40	7164
OPTOHOST002	Opto Host Port 2	1	680
DOMAIN	Domain Name Server	263	67419
HTTPS	http protocol over TLS/SSL	2846	2508338
DB-LSP	Dropbox LanSync Protocol	2	288

Figura 15: Detall dels ports de servei de la captura

Si premem sobre un port de servei en concret s'obrirà una nova finestra amb una taula on es mostra la relació de les IP de la captura que intervenen en el tràfic del port.

IP traffic detail

Key	IP	Packets	Size (bytes)
OPTOHOST002	80.58.61.250	1	680
OPTOHOST002	192.168.1.35	1	680

Figura 16: Relació de les IP d'un port de servei

Les taules es poden ordenar per qualsevol camp prement la capçalera de cada columna.

Resultats per IP

Al menú *Stats* opció *IP* podem examinar les IP que intervenen al tràfic de xarxa capturat.

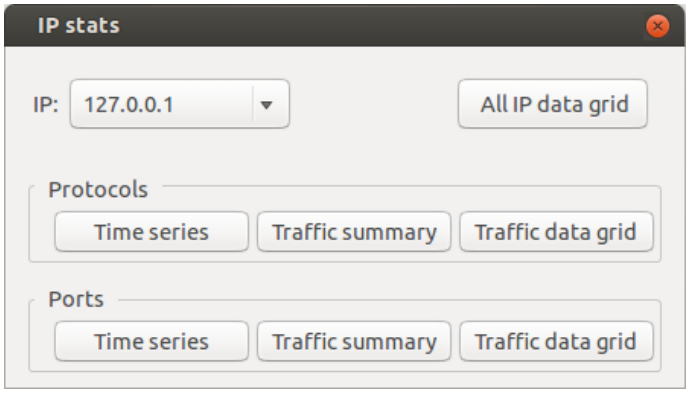
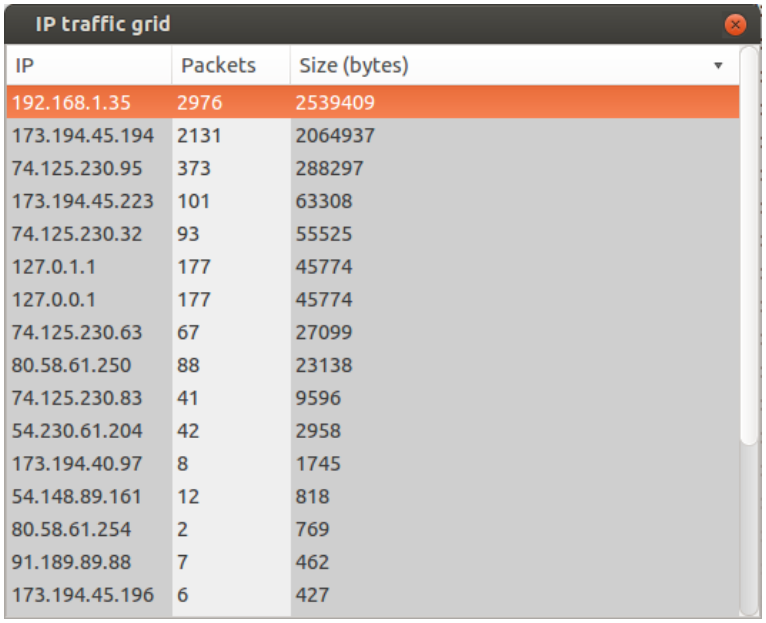


Figura 17: Diàleg de resultats per IP

Al prémer el botó **All IP data grid** s'obrirà una taula amb el nombre de paquets i la grandària del tràfic capturat de totes les IP.



IP traffic grid

IP	Packets	Size (bytes)
192.168.1.35	2976	2539409
173.194.45.194	2131	2064937
74.125.230.95	373	288297
173.194.45.223	101	63308
74.125.230.32	93	55525
127.0.1.1	177	45774
127.0.0.1	177	45774
74.125.230.63	67	27099
80.58.61.250	88	23138
74.125.230.83	41	9596
54.230.61.204	42	2958
173.194.40.97	8	1745
54.148.89.161	12	818
80.58.61.254	2	769
91.189.89.88	7	462
173.194.45.196	6	427

Figura 18: Tràfic de totes les IP

Per examinar el tràfic de xarxa d'una **IP en concret** l'hem de seleccionar al desplegable i escollir quina informació volem veure.

- Informació dels protocols:
  - Sèrie temporal del tràfic dels protocols de la IP amb el botó **Time series**.

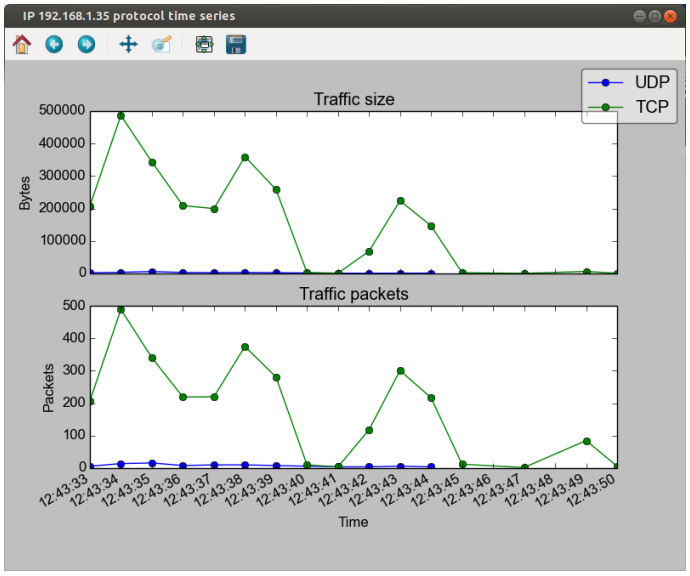


Figura 19: Sèrie temporal dels protocols d'una IP

  - Resum del tràfic del protocols de la IP mitjançant gràfica de barres amb el botó **Traffic summary**.

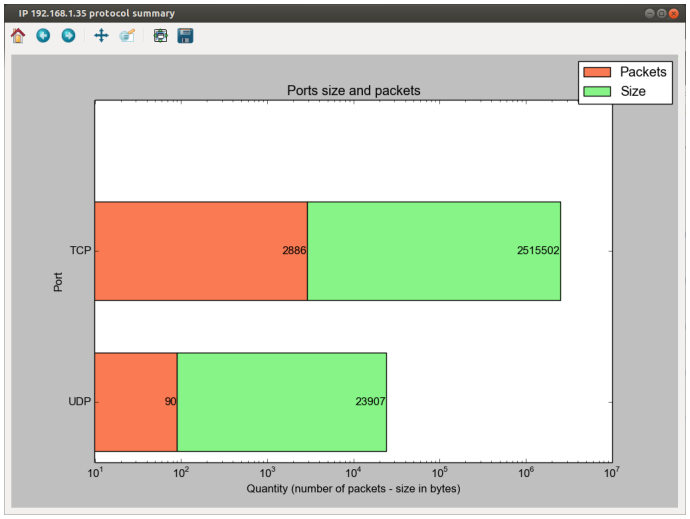
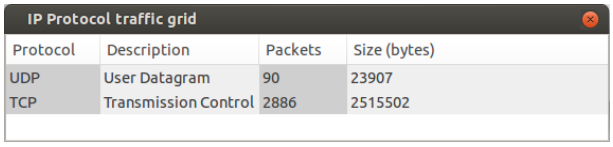


Figura 20: Resum dels protocols d'una IP

– Detall del tràfic del protocols de la IP amb el botó **Traffic data grid**.



IP Protocol traffic grid

Protocol	Description	Packets	Size (bytes)
UDP	User Datagram	90	23907
TCP	Transmission Control	2886	2515502

Figura 21: Detall dels protocols al tràfic d'una IP

- Informació dels ports de servei:

  – Sèrie temporal del tràfic dels ports de la IP amb el botó **Time series**.

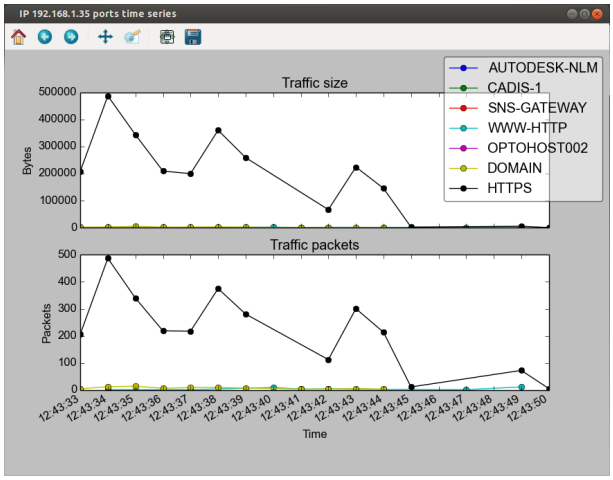


Figura 22: Sèrie temporal dels ports de servei d'una IP

  – Resum del tràfic dels ports de la IP mitjançant gràfica de barres amb el botó **Traffic summary**.

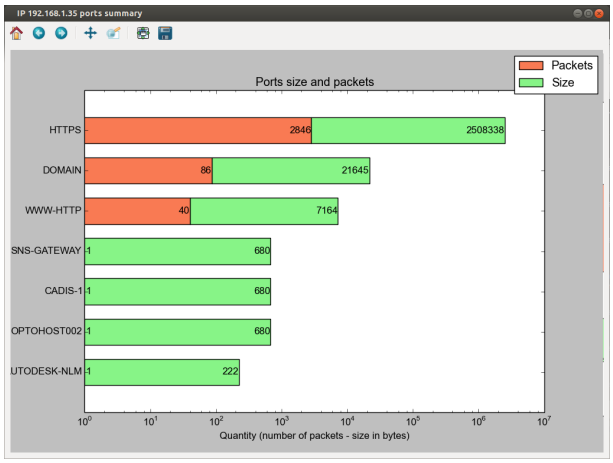


Figura 23: Resum dels ports de servei d'una IP

– Detall del tràfic del ports de la IP amb el botó **Traffic data grid**.

IP Port traffic grid

Port	Description	Packets	Size (bytes)
AUTODESK-NLM	Autodesk NLM (FLEXlm)	1	222
CADIS-1	Cadis License Management	1	680
SNS-GATEWAY	SNS Gateway	1	680
WWW-HTTP	World Wide Web HTTP	40	7164
OPTOHOST002	Opto Host Port 2	1	680
DOMAIN	Domain Name Server	86	21645
HTTPS	http protocol over TLS/SSL	2846	2508338

Figura 24: Detall dels ports de servei al tràfic d'una IP

Llicència

Aquesta aplicació es distribueix sota llicència *GNU General Public License*[35], que permet a l'usuari:

- La llibertat de fer servir l'aplicació per qualsevol propòsit.
- La llibertat de canviar l'aplicació per satisfer les seves necessitats.
- La llibertat de compartir l'aplicació amb amics i veïns.
- La llibertat de compartir els canvis realitzats.

Figura 25: Logotip de la llicència GNU-GPL versió 3

Per protegir aquests drets, si algú distribueix o modifica aquesta aplicació, té la responsabilitat de fer arribar el codi font i el mateix tipus de llicència a l'usuari final.

Al principi de cada fitxer del codi font també s'indica el tipus de llicència:

```
# GNU Graphic Network Analyzer
# Copyright (C) 2015  Josep Carcelen <jose.carcelen@gmail.com>
#
# This program is free software: you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation, either version 3 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program.  If not, see <http://www.gnu.org/licenses/>.
```

Una còpia de la llicència completa es distribueix en format PDF amb el codi font al fitxer *license-gpl-3.0.pdf*

A l'aplicació, al menú *Help* opció *About* també s'indica el tipus de llicència:

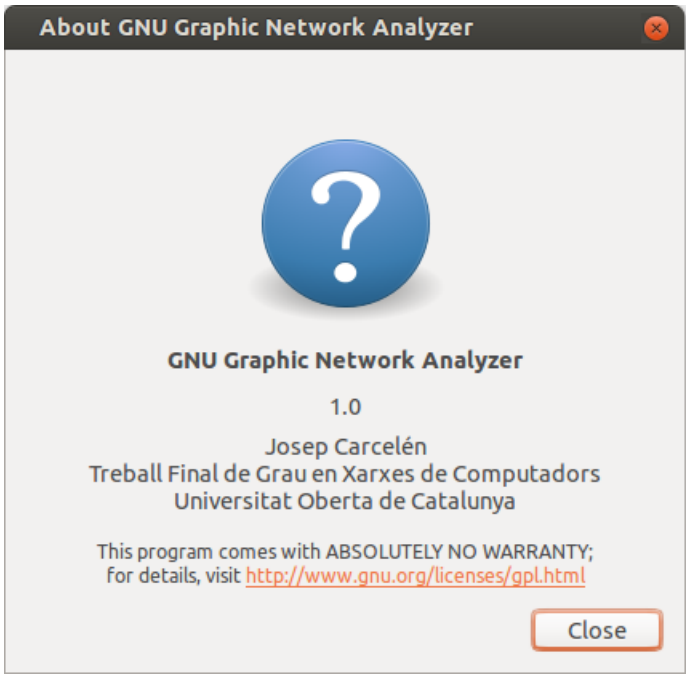


Figura 26: Informació de la llicència a l'aplicació