Eina d'anàlisi de rendiment d'una xarxa

TaFanet v1.0

Jordi Arroyo Caballero

Enginyeria tècnica d'informàtica de sistemes

Data : **13 de Gener de 2013**

Consultora: **Maria Isabel March Hermo**

A la memòria dels meus pares

Juan Francisco i Soledad

RESUM

Com a culminació dels estudis realitzats en Enginyeria Tècnica d'Informàtica de sistemes es presenta aquest Treball de Final de Carrera, que pretén ser una síntesi eminentment pràctica dels coneixements obtinguts. D'acord amb els itineraris seguits durant la carrera s'inclou aquest TFC dins de l'àrea de les Xarxes de Computadors.

El propòsit d'aquest treball és obtindre una eina d'anàlisi de rendiment d'una xarxa. No queda limitat aquest treball en la implementació de l'eina, TaFanet v1.0, siguent part de la mateixa importància, sinó més, la memòria que aquí es presenta. Es pretén documentar el procés de creació i desenvolupament de l'aplicació, esposant les diferents fases que composen el projecte seguint les pautes de l'enginyeria del programari.

Partint de la captura dels paquets en el seu tràfic a la xarxa i de l'estudi dels protocols que encapsulen les trames de dades, l'aplicació mostra informació de l'estat actual de les comunicacions a nivells d'enllaç, xarxa i transport, creant un conjunt de taules, gràfiques i estadístiques. Aquesta informació ha de facilitar l'estudi del comportament de la xarxa, tant per part de l'administrador com per l'usuari final a l'àmbit domèstic o petita organització.

Si bé, els continguts troncals d'aquest treball es basen en l'àrea de coneixement de les xarxes de computadors de les assignatures cursades durant la carrera, els seus requeriments són transversals amb altres àrees de coneixement treballades, especialment a nivell de programació i desenvolupament d'aplicacions.

Índex de Continguts

1. Introducció 1
1.1 Justificació del TFC 1
1.2 Objectius del TFC 1
1.3 Enfocament i mètode seguit 2
1.4 Planificació del projecte 3
1.4.1 Desviacions de la planificació del projecte 5
1.5 Productes obtinguts 6
1.6 Breu descripció dels altres capítols 6
2. Recollida d'informació 7
2.1 TCP/IP 7
2.1.1 Interfície de xarxa 7
2.1.2 Capa de xarxa 8
2.1.3 Capa de transport 8
2.1.4 Capa d'aplicació 9
2.1.5 Encapsulament de la informació 9
2.2 Breu estudi d'eines similars al mercat 10
2.2.1 Whireshark 10
2.2.2 Tcpdump/winDump 10
2.3 Definició de requisits 11
3. Fases d'anàlisi i disseny 12
3.1 Diagrama de Casos d'ús 12
3.2 Especificació formal dels casos d'ús 12
3.3 Diagrama estàtic de classes 15
3.4 Disseny de l'aplicació 16
3.5 Disseny de la interfície d'usuari 17
3.6 Llenguatge de programació 18
4. Implementació de l'eina 19
4.1 Entorn de desenvolupament 19
4.2 Primer Prototip: Captura de paquets 22
4.2.1 Estratègia de captura i classificació de trames 23
4.2.2 Utilització dels packages a la implementació 24
4.2.3 Exemple d'execució del prototip 1 26
4.3 Creació de les interfícies d'usuari 27
4.3.1 WindowsBuilder 27
4.3.2 Finestra de configuració 28
4.3.3 Finestra principal de l'eina 29
4.3.4 Finestres d'error, informació i quadres de diàleg 29
4.4 Segon prototip: Interrelació de les interfícies d'usuari i la classificació de paquets 30
4.4.1 Anàlisi de trames i protocols. 31

4.4.2 Gestió d'esdeveniments 31
4.5 Fase de proves 33
5. Documentació de l'aplicació 34
5.1 Manual d'instal·lació 34
5.2 Manual de d'usuari 35
6. Conclusions 38
7. Glossari 39
8. Bibliografia 40
8.1 Webs de consulta 40
9. Annexos 41
9.1 Annex A. Capçaleres del protocols de xarxa. 41

Índex de figures

Il·lustració 1: Pila TCP/IP 7
Il·lustració 2: Detall encapsulament d'un paquet IP en una trama Ethernet 8
Il·lustració 3: Formació d'una trama Ethernet 9
Il·lustració 4: Analitzador de xarxa i protocols Wireshark 10
Il·lustració 5: Analitzador de xarxa Tcpdump 10
Il·lustració 6: Diagrama de casos d'ús 12
Il·lustració 7: Diagrama estàtic de classes 15
Il·lustració 8: Disseny de la pantalla inicial de configuració 17
Il·lustració 9: Disseny pantalla central de l'aplicació 17
Il·lustració 10: Pagina principal del projecte WinPcap 19
Il·lustració 11: Instruccions d'instal·lació Jpcap 20
Il·lustració 12: Pàgina inicial projecte JFreeChart 21
Il·lustració 13: Detall de biblioteques referides al projecte dins Eclipse 21
Il·lustració 14: Configuració inici prototip 1 22
Il·lustració 15: Sortida mode text captura paquets ARP prototip 1 22
Il·lustració 16: Sortida mode text captura paquets IP prototip 1 22
Il·lustració 17: API llibreria Jpcap 23
Il·lustració 18: Camps de la classe Packet 24
Il·lustració 19: Packages de l'aplicació 25
Il·lustració 20: Exemple d'execució prototip 1 26
Il·lustració 21: Patró Model Vista Controlador 27
Il·lustració 22: Eina gràfica WindowBuilder 28
Il·lustració 23: Finestra de configuració 28
Il·lustració 24: Finestra principal 29
Il·lustració 25: Diagrama de flux de l'eina 30
Il·lustració 26: Capçalera trama Ethernet II. IEEE 802.3 41
Il·lustració 27: Paquet ARP 41
Il·lustració 28: Capçalera IP. RFC 791 41
Il·lustració 29: Capçalera ICMP. Tipus Echo. RFC 792 42
Il·lustració 30: Capçalera protocol TCP. RFC 793 42
Il·lustració 31: Capçalera protocol UDP. RFC 768 42

1. Introducció

En la gestió d'una xarxa de computadors és fa necessari analitzar el rendiment de les comunicacions. Aquest anàlisi possibilita detectar possibles intrusions i/o configuracions deficients als equips de xarxa, tals que poden repercutir de manera negativa a la transmissió dels paquets d'informació, provocant una disminució de l'ample de banda disponible pel tràfic generat pels usuaris de la xarxa, tant a nivell local com en les connexions a internet.

Això fa que un analitzador de rendiment de xarxa sigui una eina indispensable per un administrador de sistemes encarregat de la gestió i manteniment de la xarxa local d'una organització. Com més extensa sigui aquesta, més necessari es fa el suport de l'eina per poder monitoritzar el tràfic a les línies de comunicació, feina totalment inabastable per l'administrador de xarxa, sense el suport d'una eina centralitzada que ofereixi un conjunt d'estadístiques del tràfic present per poder ser analitzat.

Un analitzador de rendiment, no només és útil en la gestió d'una xarxa local de grans dimensions, on mitjançant la programació d'alertes pot alleugerir en gran manera la feina de l'administrador de xarxes, sinó també en la gestió de petites xarxes d'àrea local com poden ser organitzacions petites o inclús en la gestió de xarxes domestiques, on d'un temps ençà trobem la irrupció de nous sistemes connectats a la xarxa, fora del clàssic computador.

Les xarxes domestiques ja no estan formades única i exclusivament per un o dos computadors, sinó que trobem multitud d'aparells que s'interconnecten, i que per tant, utilitzen una part del seu ample de banda. Es poden trobar televisors amb connexió a xarxa, reproductors multimèdia, mòbils, portàtils, impressores, alarmes, etc. Molts d'ells no són aparells optimitzats per a la xarxa, permeten una configuració molt minsa i disposen d'una pràcticament nul·la capacitat de monitorització.

Això fa que un analitzador de rendiment de xarxa no només sigui utilitzable per un administrador de xarxes, sinó fins i tot per a un usuari domèstic, amb uns mínims coneixement del funcionament d'una xarxa tipus Ethernet i dels seus protocols, que vulgui tenir un control sobre quines comunicacions ocupen l'ample de banda, de tal manera que li permeti prendre decisions que optimitzin la xarxa.

1.1 Justificació del TFC

Amb aquest treball, es pretén desenvolupar una eina d'anàlisi de rendiment d'una xarxa que ofereixi a l'usuari un informe de rendiment de la xarxa local, sense produir una sobrecarrega del medi que distorsioni els resultats o afecti al propi funcionament de la xarxa analitzada.

Són diversos els motius que em van portar a decidir-me per l'eina d'anàlisi de rendiment de xarxa, dins de l'àrea de les xarxes de computadors, a l'hora d'escollir la temàtica pel meu Treball Final de Carrera. En primer lloc, l'atracció que em produeix el descobrir el funcionament a nivell intern d'una xarxa de computadors i els protocols en que és sustenta.

En segon lloc, una predilecció innata per les feines més practiques, per sobre d'aquelles més teòriques, sense deixar de gaudir l'estudi i recopilació d'informació necessària per a dur a bon port els reptes que he d'enfrontar en el dia a dia, tant a l'àmbit laboral com acadèmic.

Crec, que aquest treball m'ha aporta justament aquests dos conceptes: un aprofundiment en l'estudi del funcionament d'una xarxa local i dels seus protocols, així com la possibilitat de desenvolupar un producte partint des de la fase inicial de recollida d'informació fins a la implementació d'una primera versió plenament operativa, passant per totes les fases que composen la realització del projecte.

Així també, aquest últim aspecte de dur el control del meu propi projecte i per tant haver de gestionar l'avanç i reconduir aquelles fites que puguin quedar endarrerides per representar una dificultat superior a la prevista a la planificació inicial del projecte, representa un bon repte que de ben segur aportarà uns coneixements molts necessaris a la meva vida laboral.

1.2 Objectius del TFC

Tal i com s'ha presentat a la justificació del treball, l'objectiu general és obtenir un producte de software que permeti analitzar el rendiment de la xarxa, documentant tot el procés de creació en una memòria que descrigui els passos realitzats i els conceptes en que es basa.

Les funcionalitats que es plantegen implementar són les següents:

- Anàlisi del tràfic, capturant els paquets presents a la xarxa mitjançant la interfície de xarxa escollida de les presents en l'equip on s'executi l'aplicació.
- Analitzar i emmagatzemar la informació referent als protocols de xarxa (IP, ARP i ICMP), transport (TCP i UDP) i aplicació dels paquets. Així com, les adreces MAC i IP, tant d'origen com de destí. A nivell d'aplicació es planteja obtenir els ports de comunicació.
- Mostrar en taules o gràfics la informació recollida. L'objectiu és proporcionar a l'usuari una estadística de tipus de paquets presents per cada protocol i dels equips que l'utilitzen tant interns com externs a la xarxa.

Particularment, les taules i gràfics que es pretenen implementar són:

- Vista general de les trames de dades que circulen per la xarxa en temps real, oferint informació d'origen i destí, així com de protocols utilitzats a cada trama als diferents nivells
- Informació detallada d'una trama en particular de les capturades.
- Estadística de protocols presents a la xarxa.
- Estadística d'adreces IP, separades per pertanyents a la xarxa local o externes.
- Anàlisi del volum de dades present a la xarxa per unitat de temps.
- Gràfica de protocols presents a la xarxa

1.3 Enfocament i mètode seguit

Per a la implantació de l'eina d'anàlisi d'una xarxa s'ha escollit utilitzar el paradigma de Programació Orientada a Objectes, utilitzant per modelar el desenvolupament de l'eina l'Unified Modeling Language (UML). La decisió del llenguatge de programació a utilitzar es documenta en un apartat posterior d'aquesta memòria, un cop definits els requisits de l'eina i superades les fases d'anàlisi i disseny.

S'ha establert com a metodologia de treball un cicle de vida iteratiu i incremental basat en el prototipatge per la producció de l'eina. Aquest mètode permet avançar en la producció de l'eina recuperant fases anteriors del projecte en cas de l'aparició de nous requisits o necessitats d'informació, durant el desenvolupament del producte.

D'acord amb aquest mètode s'han establert les següents fases:

1. **Recollida d'informació**. Recerca d'informació dels protocols de xarxa i anàlisi de productes similars al mercat amb l'objectiu de definir els requisits de l'eina.
   - S'investiga inicialment els protocols a analitzar a nivell de xarxa, transport i aplicació, per definir quins camps de cada protocol cal controlar de cara a fer estadístiques del tràfic detectat.
   - Es comparen productes similars al mercat, estudiant el seu comportament i les funcionalitats que ofereixen.
   - Es cerca informació d'eines i llibreries que ofereixin suport al desenvolupament de l'aplicació.
2. **Anàlisi i disseny**. Generar l'arquitectura del programa i definir el llenguatge de programació a utilitzar.
   - Anàlisi dels casos d'ús i especificació formal de cada cas d'ús.
   - Disseny del diagrama estàtic de classes i de les interfícies d'usuari.
   - Amb tota la informació obtinguda prendre la decisió del llenguatge de programació a utilitzar.
3. **Implementació**. Crear l'eina pel mètode de prototips successius que ampliïn les funcionalitats de l'anterior fins a arribar a la versió funcional de l'eina.
   - Desenvolupar un primer prototip, que a partir del disseny de l'aplicació obtingut.
   - Desenvolupar les interfícies d'usuari.

- Desenvolupar un segon prototip, que conjugui les interfícies d'usuari amb les funcionalitats del primer prototip.
- Creació de la versió definitiva de l'eina, implementant els mètodes necessaris per a la creació d'estadístiques a partir de les dades obtingudes.

4. **Proves**. Realització de proves que donin garantia de compliment dels requisits definits.
   - Iniciar el procés de proves en l'entorn de desenvolupament.
   - Realitzar proves de funcionament amb l'arxiu executable directament sobre el sistema operatiu.
   - Fer proves en altres sistemes operatius per acotar la funcionalitat de l'eina a nivell de plataforma.
5. **Documentació de l'aplicació**. Creació de la documentació necessària per a l'usuari de l'aplicació.
   - Manual d'instal·lació, que descrigui els requisits d'execució de l'eina.
   - Manual d'usuari descriptiu de la funcionalitat de l'eina.

1.4 Planificació del projecte

S'ha planificat la realització del treball, donats els objectius generals d'obtenir un producte i una memòria que documenti el procés de creació de l'eina d'anàlisi de rendiment d'una xarxa, de tal manera que ambdós avancin en paral·lel.

Les possibles desviacions de la planificació establerta s'exposarà a la part de la memòria corresponent a la fase que modifiqui la seva planificació inicial, documentant les raons que l'han provocat i els reajust que comporti a la resta de fites del projecte.

A continuació es detalla la planificació del projecte, que ha estat dividit en tasques. Es descriu cada tasca i els seus objectiu definint la seva temporalització dins del projecte, així com, les fites que han d'aportar a la realització del projecte.

Tasca 1
Temporalització: 10 dies (01/10/2012 - 10/10/2012)
Descripció: Definició de requisits. Recollida d'informació que es consideri important pel desenvolupament del projecte, i particularment dels protocols de xarxa. A falta d'algun aspecte que pugui sorgir a la definició de requisits.
Objectius:
- Analitzar eines existents al mercat.
- Definir les funcionalitats de l'aplicació.
- Aprofundir en el coneixement dels protocols de xarxa (IP, ARP i ICMP) i de transport (TCP, UDP). Considerar quins camps analitzarà l'aplicació.

Fites:
- Definició de requisits.
- Plantilla de la memòria.
- Documentar la informació recollida i incloure-la a la memòria.

Tasca 2
Temporalització: 10 dies (11/10/2012 - 21/10/2012)
Descripció: Fase d'Anàlisi.
Objectius:
- Anàlisi de requisits.
- Anàlisi de l'aplicació.

Fites:
- Diagrama de casos d'us.
- Especificació formal dels casos d'us.
- Diagrama estàtic de classes.

Tasca 3
Temporalització: 10 dies (22/10/2012 - 31/10/2012)
Descripció: Fase de disseny.
Objectius:
- Disseny de l'aplicació.

Fites:
- Revisió del digrama estàtic de disseny.
- Disseny de la interfície d'usuari.

Tasca 4
Temporalització: 1 setmana (01/11/2012 - 07/11/2012)
Descripció: Recollida d'informació del llenguatge a utilitzar.
Objectius:
- Definir llenguatge de programació a utilitzar.
- Definir biblioteques complementaries a utilitzar.

Fites:
- Tenir enllestit l'entorn de desenvolupament.
- Actualitzar la memòria.

Tasca 5
Temporalització: 2 setmanes (08/11/2012 - 21/11/2012)
Descripció: Implementació primer prototip.
Objectius: Implementar la detecció d'interfícies de xarxa i capturar paquets.
Fites:
- Detectar les interfícies de xarxa i escollir una.
- Capturar i mostrar paquets per pantalla.
- Documentar el procés a la memòria.

Tasca 6
Temporalització: 2 setmanes (22/11/2012 - 05/12/2012)
Descripció: Implementació segon prototip.
Objectius: Analitzar i classificar paquets capturats.
Fites:
- Classificar els paquets per diferents filtres.
- Mostrar estadística per pantalla.
- Documentar el procés a la memòria.

Tasca 7
Temporalització: 2 setmanes (06/12/2012 - 19/12/2012)
Descripció: Implementació tercer prototip.
Objectius: Crear la interfície d'usuari.
Fites:
- Crear les diferents pantalles d'usuari.
- Crear les pantalles per mostrar les estadístiques.
- Documentar el procés a la memòria.

Tasca 8
Temporalització: 1 setmana (20/12/2012 - 26/12/2012)
Descripció: Prototip complert de l'eina.
Objectius: Corregir desajustos en la integració de l'aplicació amb la interfície d'usuari.
Fites:
- Prototip operatiu de l'eina.
- Esbós del manual d'usuari.
- Actualitzar la memòria.

Tasca 9
Temporalització: 10 dies (27/12/2012 - 06/01/2013)
Descripció: Fase de proves.
Objectius:
- Realitzar proves a l'aplicació per garantir el compliment dels requisits.
- Corregir les deficiències trobades a les probes.

Fites:
- Passar els protocols de proves.
- Eina corregida
- Documentar les proves a la memòria.
- Manual d'usuari a incloure a la memòria

Tasca 10
Temporalització: 1 setmana (07/01/2013 - 13/01/2013)
Descripció: Versió 1.0
Objectius: Retocs estètics a l'eina i finalitzar la memòria
Fites:
- Eina d'anàlisi de rendiment d'una xarxa.
- Memòria TFC.

Tasca 11
Temporalització: 1 setmana (14/01/2013 - 20/01/2013)
Descripció: Presentació.
Objectius: Presentar el projecte.
Fites: Crear la presentació virtual.

1.4.1 Desviacions de la planificació del projecte

Durant el desenvolupament de tot projecte poden sorgir situacions donades per condicionants externs al projecte o inherents a la pròpia planificació, que calgui analitzar per determinar si són assumibles dins de la planificació prevista o cal efectuar modificacions de la mateixa.

En aquest cas, es va detectar en iniciar la implementació de l'aplicació que el temps previst per la implementació de les interfícies d'usuari, donat que era un àrea del que no es disposava de la necessària experiència, podria provocar retards en la implementació final de l'eina.

Per aquesta raó, es va decidir intercanviar l'ordre de les tasques 6 (2º prototip: analitzar i classificar paquets capturats) i 7 (3º prototip: Crear la interfície d'usuari). De tal manera la tasca 6 es dedicarà a la creació de les interfícies d'usuari i la tasca 7 contindrà la creació d'un segon prototip que interrelacioni les interfícies creades amb la classificació de paquets i estadístiques sobre la captura realitzada.

1.5 Productes obtinguts

Com a resultat d'aquest TFC s'obtenen els següents productes:

- TaFanet v1.0. Eina d'anàlisi de rendiment d'una xarxa, propòsit d'aquest TFC.
- Memòria. El document que ens ocupa que descriu el procés de desenvolupament de l'eina.
- Manual d'instal·lació. Document que descriu els requisits previs d'instal·lació per poder executar l'aplicació. S'inclou com a capítol d'aquesta memòria i com a fitxer dins del paquet de software.
- Manual d'usuari. Document que descriu el funcionament de l'aplicació. S'inclou com a capítol d'aquesta memòria i com a fitxer dins del paquet de software..
- Presentació. Transparències que descriuen el procés de l'elaboració del treball.

1.6 Breu descripció dels altres capítols

Els següents capítols d'aquesta memòria es centren a les diferents fases descrites per al desenvolupament de l'eina a l'apartat "enfocament i mètode seguit", documentant a cada capítol el procés seguit a cadascuna de les diferents fases:

- Recollida d'informació.
  Centrat en el procés de cerca d'informació, documentant les conclusions obtingudes referents als protocols de xarxa a analitzar i d'eines similars existents al mercat.
- Fases d'anàlisi i disseny.
  Aquest apartat descriu de manera formal els requisits de l'aplicació mitjançant el diagrama de casos d'ús, així com s'especifica el disseny de l'aplicació i de les interfícies d'usuari que satisfacin aquests requisits.
- Implementació de l'eina.
  El capítol mostra el procés de programació de l'aplicació a partir de la creació de successius prototips.
- Documentació de l'aplicació.
  Conté el manual d'instal·lació i d'usuari de l'eina.

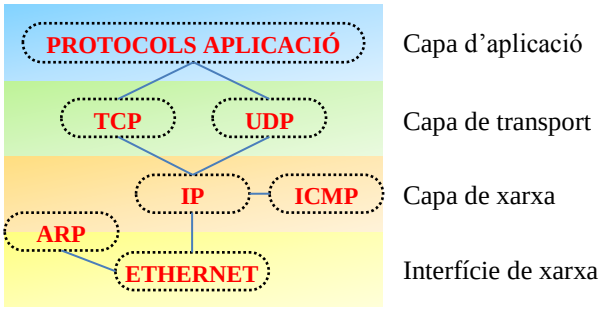
Per últim, es presenten les conclusions del treball, el glossari i la bibliografia consultada.

2. Recollida d'informació

L'objectiu principal d'aquesta fase és documentar tots aquells aspectes que es considerin necessaris a nivell teòric de cara a definir els requisits de l'eina i la seva posterior implementació. El primer objecte d'estudi ha estat la pila de protocols TCP/IP i el procés d'encapsulament de la informació en la comunicació entre equips d'una xarxa local (LAN, Local Area Network), o de gran abast (WAN, Wide Area Network). Són precisament les capçaleres dels protocols de xarxa l'objecte d'anàlisi per part de l'eina a implementar.

2.1 TCP/IP

La pila de protocols TCP/IP es compon de 4 capes: interfície de xarxa, xarxa, transport i aplicació, però s'ha centrat la recerca d'informació als principals protocols[1] de la capa de xarxa i transport que seran les capes analitzades per l'eina. Donat que l'eina a implementar haurà de ser executada en un equip connectat físicament a la xarxa, es centra l'estudi a nivell d'interfície de xarxa en les trames Ethernet.



Il·lustració 1: Pila TCP/IP

- Interfície de xarxa o nivell d'enllaç, és el nivell encarregat de la transmissió dels senyals elèctrics entre els diferents equips.
- Capa de xarxa o nivell IP, identifica els equips de la xarxa i és responsable de l'adreçament i encaminament dels paquets.
- Capa de transport, responsable de la fiabilitat i seguretat en la transmissió del paquets d'informació per la xarxa.
- Capa d'aplicació, connecta les aplicacions d'usuari amb la pila de protocols TCP/IP.

2.1.1 Interfície de xarxa

Depenent de l'autor consultat es troben diferents noms per aquesta capa o nivell que englobaria la capa física i la capa d'enllaç de dades de la torre OSI. La pila de protocols TCP/IP no defineix els protocols a utilitzar en aquesta capa, nomes defineix que ha de ser un protocol que permeti la implementació del protocol IP. A l'actualitat, el protocol Ethernet s'ha establert com un estàndard de facto per aquest capa i és el protocol en que ens centrarem en aquest nivell.

Ens interessen tres camps de la capçalera del protocol Ethernet: MAC de destinació, MAC d'origen i tipus de protocol encapsulat en la trama. L'adreça MAC ─ de l'anglès Media Access Control ─ identifica una interfície de xarxa d'un maquinari de manera quasi única (no entrarem a valorar ara la facilitat per realitzar suplantació d'adreça MAC) i està formada per 6 bytes que formen 6 grups de dos dígits expressats en sistema hexadecimal.

A una xarxa local les trames de dades enviades per un equip són rebudes per tots els equips connectats a la xarxa i són els receptors els responsables de rebutjar aquelles trames de les que no són destinataris.

[1] Es detallen les capçaleres dels diferents protocols a l'annex A.

Aquest funcionament de les xarxes locals permet, utilitzant el modo promiscu de la tarja de xarxa de l'equip on s'executi l'eina, capturar totes les trames que circulin per la xarxa.

2.1.2 Capa de xarxa

El principal protocol de capa de xarxa és el protocol IP, *internet protocol*, i es defineix com un protocol no orientat a la connexió que proporciona les funcions necessàries per enviar paquets de dades des d'un equip origen a un equip destí. Com a protocol no orientat a la connexió, no implementa cap mecanisme de control de flux ni garanteix la fiabilitat de la connexió, delegant aquesta feina als protocols de capes superiors.

Els paquets IP, encapsulats dins de les trames Ethernet, especifiquen a la capçalera les adreces IP de l'equip origen i de l'equip destí. Són aquests dos camps de la capçalera, junt amb el camp protocol que indica el protocol de transport encapsulat dins del paquet IP, el que ens interessa capturar. Les adreces IP estan formades per 4 bytes que formen quatre grups de tres dígits en representació decimal.

IP utilitza el protocol auxiliar ARP, *Address Resolution Protocol*, per obtenir les adreces MAC dels equips de destinació. Amb aquesta finalitat, el protocol ARP utilitza les funcionalitats de la capa d'interfície de xarxa per aconseguir aquest propòsit. És per aquesta raó, que molts autors consideren un protocol a cavall entre les capes d'interfície de xarxa i la pròpia capa de xarxa.

Són quatre els camps que ens interessen de la capçalera del protocol ARP. Emissor HA, adreça hardware, que correspon a l'adreça MAC de l'emissor; emissor IP, adreça IP de l'emissor del paquet ARP; receptor HA, adreça hardware del receptor del paquet que al cas de ser desconegut correspon a l'adreça de broadcast (FF:FF:FF:FF:FF:FF), i receptor IP, adreça IP del receptor del paquet ARP.

Per subministrar informació al protocol IP de determinats errors a l'entorn de xarxa, es pot utilitzar un missatge ICMP, *Internet Control Message Protocol*, encapsulat dins d'un paquet IP. Ens interessa el camp tipus de la capçalera ICMP que indica el tipus de missatge enviat. Pel desenvolupament de l'eina són objecte d'estudi els missatges *echo request*, *echo reply* i *destination unreachable*. Els dos primers per la detecció d'atacs per ping i l'últim per la detecció de connexions fallides per especificació errònia de destinació.

Il·lustració 2: Detall encapsulament d'un paquet IP en una trama Ethernet

2.1.3 Capa de transport

Els principals protocols de la capa de transport són TCP, *Transmission Control Protocol*, i UDP, *User Datagram Protocol*. La diferencia fonamental entre els dos protocols rau en el fet que, TCP és un protocol orientat a la connexió que garanteix la transmissió dels segments entre l'equip origen i l'equip destinació mitjançant l'establiment d'una connexió prèvia entre els dos equips.

A diferencia d'això, UDP està destinat a la transmissió de datagrames entre aplicacions on sigui més important garantir la rapidesa en l'entrega de les dades, enfront de garantir la recepció de les dades, delegant la responsabilitat de control de les dades a les aplicacions que l'utilitzen.

Com a tret comú dels dos protocols, i que serà objecte d'anàlisi per part de l'eina, tots dos protocols inclouen a la seva capçalera els port origen i destinació de la comunicació. Tant pel port d'origen com pel port destinació ambdós protocols utilitzen 2 bytes per identificar amb un número en representació decimal el port assignat.

2.1.4 Capa d'aplicació

No s'aprofundirà en aquesta primera versió de l'eina en l'anàlisi dels protocols d'aplicació, més enllà de l'anàlisi dels ports d'origen i destinació als protocols de la capa de transport. La IANA, Internet Assigned Numbers Authority**,** defineix els ports estàndards per la comunicació entre aplicacions. Ara bé, aquesta assignació de ports és només una recomanació i ha de ser corroborat per l'anàlisi de la capçalera del pertinent protocol d'aplicació.

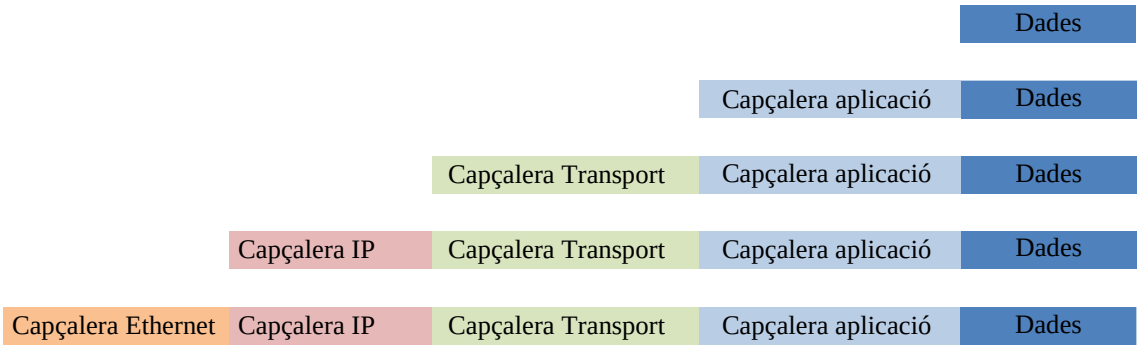
A nivell d'exemple es mostra el número de port d'algunes aplicacions comuns, podent-se consultar un llista complert a l'adreça:

http://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xml

Port	Aplicació
20	FTP - File Transfer protocol (Dades)
21	FTP - File Transfer protocol (Control)
23	Telnet
25	SMTP - Simple Mail Transfer Protocol
53	DNS - Domain Name System
80	HTTP - Hypertext Transfer Protocol
110	POP3 - Post Office Protocol

2.1.5 Encapsulament de la informació

L'encapsulament de les dades permet la comunicació de manera transparent entre les capes de la pila TCP/IP d'equips diferents. Quan una aplicació vol comunicar-se amb altre equip encapsula la informació a transmetre en el protocol d'aplicació pertinent, afegint davant de les dades la seva capçalera. La capa d'aplicació serveix el paquet a la cap de transport que afegeix a les dades la capçalera del protocol de transport escollit, com pot ser TCP o UDP. Un cop preparat el paquet, o datagrama segons el cas, passa a la cap de xarxa on s'afegeix la capçalera IP abans de passar al nivell d'interfície de xarxa on s'afegirà la capçalera Ethernet. Un cop formada la trama, és enviada per la xarxa d'on serà recollirà per l'equip destinació. Els diferents nivells de la pila TCP/IP desfaran el procés d'encapsulament de les dades fins arribar a l'aplicació destinatària la informació enviada per l'equip d'origen.



Il·lustració 3: Formació d'una trama Ethernet

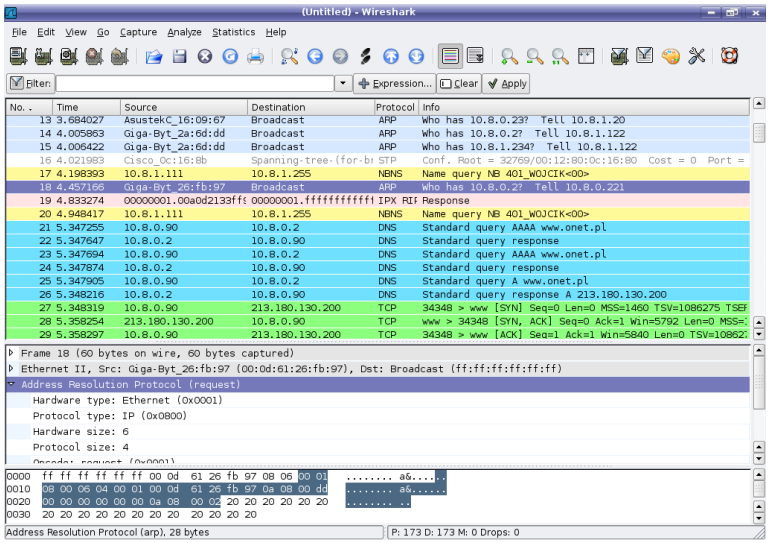
2.2 Breu estudi d'eines similars al mercat

Un cop recollida la informació dels protocols de xarxa, s'ha fet un breu estudi de les eines similars al mercat amb el propòsit de veure quines són les opcions més usuals que es troben en aquest tipus d'eines. Són moltes les que es poden trobar al mercat i no és objecte d'estudi d'aquest treball fer un anàlisi exhaustiu de totes elles.

Per aquest raó es presenten breument les dues aplicacions més conegudes i utilitzades com a eines bàsiques, tant a nivell professional com acadèmic, donada la seva qualitat i ser totes dues aplicacions de software lliure.

2.2.1 Whireshark

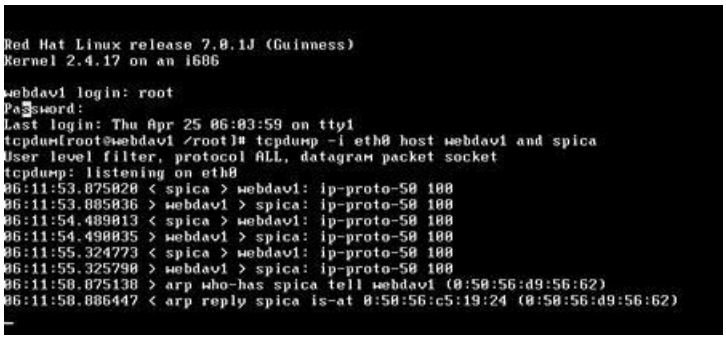
És un potent analitzador de tràfic de xarxa i de protocols de xarxa. Posseeix una interfície gràfica molt intuïtiva que fa molt senzilla la seva utilització. En versions anteriors era conegut com Ethereal. Es pot executar en més de 20 plataformes i és capaç d'analitzar més de 480 protocols. Degut a la seva potencia i facilitat d'us s'ha convertit en una de les eines més utilitzades tant a nivell professional com didàctic.



Il·lustració 4: Analitzador de xarxa i protocols Wireshark

2.2.2 Tcpdump/winDump

És una eina de línia de comades amb capacitat per analitzar el tràfic que circula per la xarxa. Funciona a la majoria de sistemes operatius Unix i existeix una versió pels sistemes operatius Windows: WinDump. L'usuari pot aplicar filtres a fi de limitar els paquets capturats, i el format Tcpdump s'ha convertit en un estàndard per altres aplicacions, tant per l'aplicació de filtres com per guardar les dades en fitxers.



Il·lustració 5: Analitzador de xarxa Tcpdump

2.3 Definició de requisits

Analitzant la informació recollida, s'arriba a la següent definició de requisits per l'eina a implementar:

- Capacitat per detectar les diferents interfícies de xarxa presents a l'equip i oferir a l'usuari la possibilitat d'escollir una de les interfícies detectades.
- Oferir a l'usuari definir el temps d'anàlisi de la xarxa, i la possibilitat d'aturar-lo malgrat no s'esgoti el temps.
- Oferir a l'usuari criteris de selecció dels protocols a detectar (tots o una selecció dels tipus que s'implementi, tant a nivell de xarxa, transport i alguns d'aplicació).
- Capturar les trames a la interfície de xarxa escollida per l'usuari.
- Analitzar les trames capturades.
- Classificar i emmagatzemar la informació recollida.
- Mostrar la informació recollida, segons els criteris de selecció, sense necessitar aturar la captura.
- Mostrar informació detallada en taules i/o gràfics.
- Mostrar un informe de trames capturades en temps real amb la següent informació: origen i destí, tant d'adreça MAC com adreça IP, protocols utilitzats a la trama i volum de dades.
- Estadística de protocols presents a la xarxa.
- Estadística d'adreces IP, tant de la xarxa local com externes a la xarxa.
- Anàlisi del volum de dades present a la xarxa per unitat de temps.
- Gràfica de protocols presents a la xarxa

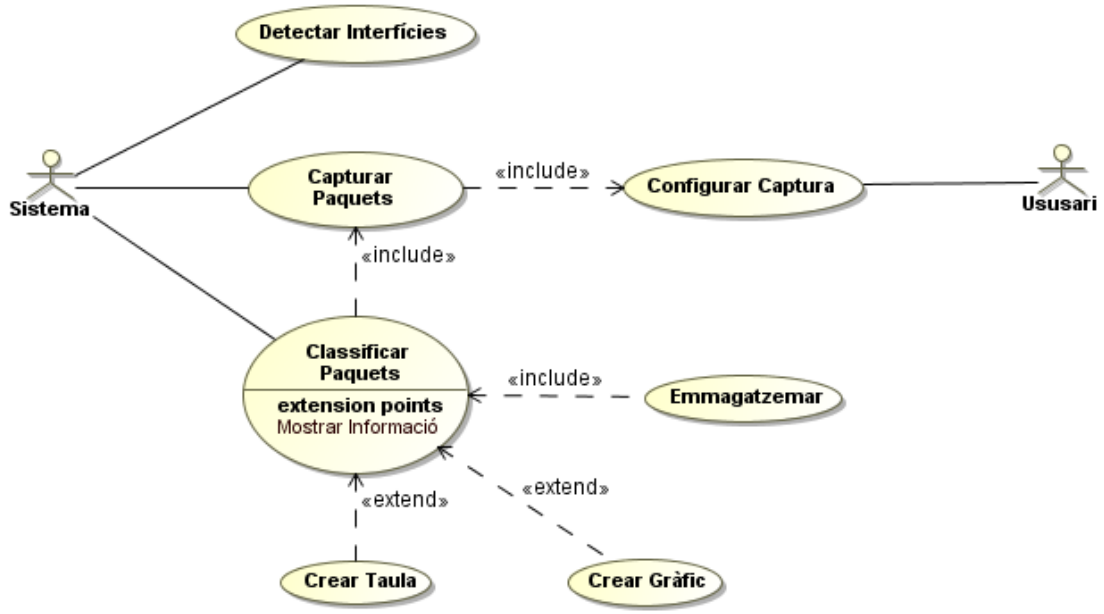
3. Fases d'anàlisi i disseny

A les fases d'anàlisi i disseny es crea l'arquitectura pròpia de l'eina d'anàlisi de rendiment d'una xarxa amb la finalitat de garantir els requisits definits a la fase de recollida d'informació.

3.1 Diagrama de Casos d'ús

S'identifiquen dos actors: la pròpia eina, identificada com a Sistema i l'Usuari. El Sistema haurà de detectar les interfícies de xarxa presents a l'equip, amb aquesta informació l'usuari decidirà quina informació vol rebre del tràfic de xarxa present a la interfície de xarxa escollida, quins protocols i el temps durant el qual serà analitzada la xarxa.

Aquest decisions configuren l'eina que s'encarrega de capturar els paquets, classificar-los i mostrar la informació per pantalla en forma de taula i/o gràfic, així com del seu emmagatzematge per a possibilitar l'anàlisi per part de l'usuari de captures anteriors.



Il·lustració 6: Diagrama de casos d'ús

3.2 Especificació formal dels casos d'ús

A continuació es descriuen formalment els casos d'ús identificats al diagrama de casos d'ús, on es descriu la funcionalitat pròpia de cada cas d'us.

Cas d'us: Detectar interfície

Resum de la funcionalitat: El sistema s'encarrega de detectar les interfícies de xarxa presents a l'equip on s'executa l'eina.

Actors: Sistema
Casos d'us relacionats: Cap.
Precondició: Cap.
Postcondició: S'han detectat les interfícies de xarxa.
Procés normal principal:

1. El sistema detecta la presencia de una o més interfícies de xarxa.

Alternatives de procés i excepcions:

1.1 No es detecta cap interfície de xarxa, o es produeix un error en la detecció.
1.2 El sistema informa a l'usuari de la incidència.

Cas d'us: Capturar Paquets

Resum de la funcionalitat: El sistema captura els paquets presents a la xarxa.
Actors: Sistema.
Casos d'us relacionats: Configurar Captura, Classificar Paquets.
Precondició: L'usuari ha configurat els paràmetres per la captura.
Postcondició: S'han capturat els paquets presents a la xarxa.
Procés normal principal:

1. Es captura el paquet present a la interfície de xarxa.
2. Es passen les dades del paquet al classificador de paquets.

Alternatives de procés i excepcions:

1.1 No s'han configurat els paràmetres de la captura. Es mostra l'error per pantalla.

Cas d'us: Classificar Paquets

Resum de la funcionalitat: Es classifiquen els paquets capturats segons els paràmetres definits per l'usuari.
Actors: Sistema
Casos d'us relacionats: Capturar Paquets, Crear Taula, Crear Gràfic, Emmagatzemar.
Precondició: S'ha capturat almenys un paquet.
Postcondició: S'ha establert una classificació dels paquets capturats.
Procés normal principal:

1. Es reben les dades d'un paquet capturat
2. Es classifica segons els paràmetres definits

Alternatives de procés i excepcions:

2.1 Es produeix un error en la classificació.

Cas d'us: Crear Taula

Resum de la funcionalitat: Crea una taula que mostra els tipus de paquets capturats.
Actors: Sistema
Casos d'us relacionats: Classificar Paquets.
Precondició: S'han rebut i classificat almenys un paquet.
Postcondició: Es mostra a l'usuari una taula amb els tipus de paquets rebuts.
Procés normal principal:

1. Demana les dades de paquets capturat.
2. Mostrar la taula per pantalla.

Alternatives de procés i excepcions:

1.1 No hi ha dades de paquets.

Cas d'us: Crear Gràfic

Resum de la funcionalitat: Crea un gràfic que mostra els tipus de paquets capturats.
Actors: Sistema
Casos d'us relacionats: Classificar Paquets.
Precondició: S'han rebut i classificat almenys un paquet.
Postcondició: Es mostra a l'usuari un gràfic amb els tipus de paquets rebuts.
Procés normal principal:

1. Demana les dades de paquets capturat.
2. Mostrar el gràfic per pantalla.

Alternatives de procés i excepcions:

1.1 No hi ha dades de paquets.

Cas d'us: Emmagatzemar

Resum de la funcionalitat: Emmagatzema en un fitxer les dades de les captures
Actors: Sistema
Casos d'us relacionats: Classificar Paquets.
Precondició: Ha finalitzat la captura.
Postcondició: S'han emmagatzema les dades de les captures.
Procés normal principal:

1. Demana les dades de paquets capturat.
2. Crea un fitxer amb les dades.

Alternatives de procés i excepcions:

1.1 No hi ha dades de paquets. No es crea el fitxer.
2.1 Es produeix un error al crear el fitxer. S'avisa a l'usuari.

Cas d'us: Configurar captura

Resum de la funcionalitat: L'usuari configura els paràmetres de la captura.

- Interfície de xarxa.
- Protocols de xarxa.
- Temps de durada de la captura.

Actors: Usuari
Casos d'us relacionats: Capturar Paquets
Precondició: El sistema ha detectat les interfícies de xarxa
Postcondició: S'han configurat els paràmetres de la captura
Procés normal principal:

1. L'usuari escull una interfície de xarxa d'una llista.
2. L'usuari escull els protocols de xarxa.
3. L'usuari defineix la durada de la captura

Alternatives de procés i excepcions:

1.1 No es mostra cap interfície de xarxa.

3.3 Diagrama estàtic de classes

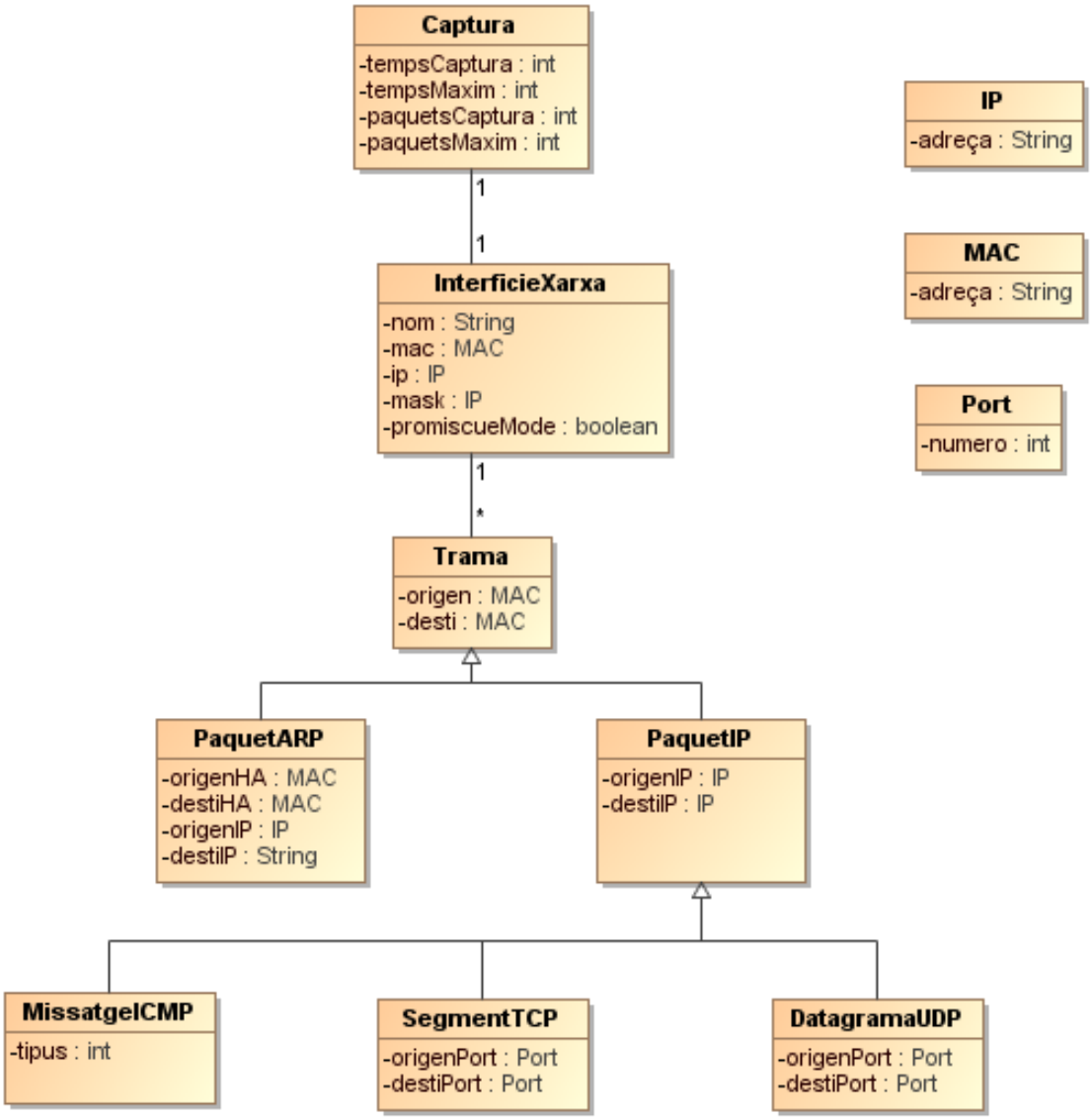
Donat que el propòsit de l'eina ha de ser capturar les trames de dades que circulen per la interfície de xarxa, escollida de les presents a l'equip on s'executi, s'identifiquen tres classes principals: Captura, InterficieXarxa i Trama.

La classe Captura emmagatzema les configuracions inicial de l'usuari, en quant a duració de l'anàlisi i sobre quina interfície de xarxa realitza la captura de trames.

La classe InterficieXarxa conté les dades de la interfície de xarxa escollida i emmagatzema les trames de dades capturades.

La classe Trama és una superclasse de la que hereten les classes PaquetARP, PaquetIP, MissatgeICMP, SegmentTCP i DatagramaUDP que contenen les dades a analitzar de les trames de dades segons el tipus de protocol.

Es preveuen també tres classes (IP, MAC i Port) que contindran els mètodes de control d'aquest tipus de dades.



Il·lustració 7: Diagrama estàtic de classes

3.4 Disseny de l'aplicació

Partint del diagrama estàtic de classes i un cop normalitzats els noms de classes i d'atributs, com és el cas del diagrama presentat, resulta senzill obtenir un esquelet de les classes a implementar, bé manualment o aprofitant les funcionalitats de les eines CASE un cop definit el llenguatge de programació. Com és el cas l'aplicació MagicDraw[2] amb que ha sigut realitzat tant el diagrama de casos d'ús com el diagrama estàtic de classes d'aquest treball. Es presenta a continuació, sense els mètodes accesors dels atributs, el codi Java resultant del disseny de l'aplicació.

```
public class Captura {
        private int tempsCaptura;
        private int tempsMaxim;
        private int paquetsCaptura;
        private int paquetsMaxim;
        private InterficieXarxa interficieXarxa;
}

public class InterficieXarxa {
        private String nom;
        private MAC mac;
        private IP ip;
        private IP mask;
        private boolean promiscueMode;
        private Trama[] trames;
}

public class Trama {
        private MAC origen;
        private MAC desti;
}

public class PaquetARP extends Trama {
        private MAC origenHA;
        private MAC destiHA;
        private IP origenIP;
        private String destiIP;
}

public class PaquetIP extends Trama {
        private IP origenIP;
        private IP destiIP;
}

public class SegmentTCP extends PaquetIP {
        private Port origenPort;
        private Port destiPort;
}

public class DatagramaUDP extends PaquetIP {
        private Port origenPort;
        private Port destiPort;
}

public class MissatgeICMP extends PaquetIP {
        private int tipus;
}

public class MAC {
        private String adreça;
}

public class IP {
        private String adreça;
}

public class Port {
        private int numero;
}
```

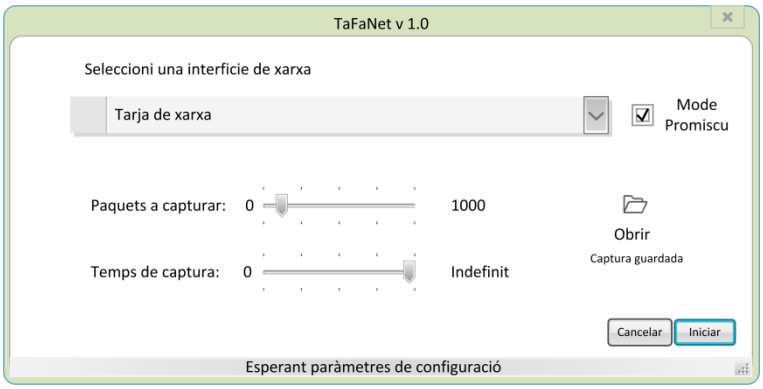
[2] MagicDraw és propietat de No Magic, Inc. http://www.nomagic.com/

3.5 Disseny de la interfície d'usuari

És important dissenyar les interfícies d'usuari en aquest punt, quan es té clar quins requisits te l'aplicació i quin és el disseny de l'aplicació. Es presenta el disseny de la pantalla inicial de configuració i el disseny de la pantalla central de l'aplicació.

La pantalla inicial de configuració ha de permetre a l'usuari escollir una de les interfícies de xarxa presents a l'equip, que es mostren en una llista desplegable. Es configura per defecte en mode promiscu, que pot deseleccionar l'usuari. De igual manera, l'usuari ha d'escollir temps/nombre de paquets a capturar, tenint com a opció per defecte indefinit, que farà que l'aplicació capturi paquets fins que s'aturi per part de l'usuari.

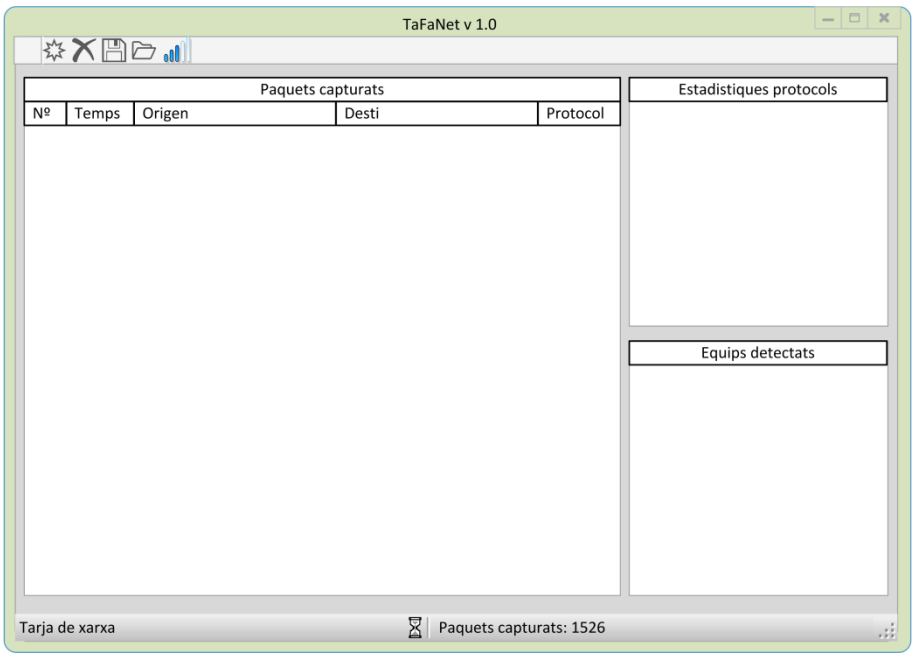
Es preveu també una icona *Obrir* que permetrà obrir un arxiu de captura guardar anteriorment.



Il·lustració 8: Disseny de la pantalla inicial de configuració

A la pantalla principal de l'aplicació es preveu mostrar per pantalla les dades pels paquets capturats en temps real, una estadística dels protocols detectats i un llistat de les adreces IP dels equips detectats.

Es preveu dotar a l'aplicació de diverses icones que permetin iniciar la captura (deshabilitat, en cas d'estar activa la captura), aturar la captura, guardar la captura en un fitxer, recuperar un fitxer guardat amb anterioritat i generació de gràfiques.



Il·lustració 9: Disseny pantalla central de l'aplicació

3.6 Llenguatge de programació

Malgrat el propòsit d'aquest treball és el desenvolupament d'una eina de rendiment de xarxa plenament operativa, és cert que seran molts els aspectes en que pugui ser ampliada i millorada. És aquesta una de les raons per les que el disseny de l'aplicació s'ha realitzat tenint com a premissa utilitzar en la implementació de l'aplicació un llenguatge orientat a objectes, pels avantatges que proporciona en quant a reutilització i extensió del codi. Dins de la varietat de llenguatges basats en la programació orientada a objectes, s'ha escollit Java amb la finalitat de afavorir la portabilitat del software a diferents plataformes, amb la mínima adaptació del codi.

En l'apartat d'interfície gràfica, el Package Swing forma part de les biblioteques de Java i proporciona el mètodes necessaris pel disseny de la interfície gràfica, tant en la creació de finestres, possibilitat d'entrada de dades i control d'esdeveniments, moviments del ratolí, etc. Un gran avantatge que proporciona és que s'executa en un fil d'execució (thread) diferent del programa principal per poder ser actualitzat en temps real. Com a inconvenient, no té suport per a la creació de gràfics. Per aquesta part de la implementació s'utilitza la llibreria JFreeChart que ofereix una amplia gama de classes per a la creació de gràfics estadístics.

Java no ofereix cap llibreria del sistema amb l'accés necessari a les interfícies de xarxa per poder analitzar les trames de dades. S'utilitzarà la llibreria Jpcap que disposa de mètodes per la captura i enviament de paquets a la xarxa. Jpcap consta de dos Packages, jpcap que conte les classes per gestionar les interfícies de xarxa i jpcap.packet amb classes que representen els paquets capturats. Malgrat existir classes especifiques per cada tipus de paquet que analitzarà l'eina (ARP, IP, ICMP, TCP i UDP), només s'utilitzaran els mètodes de la classe Packet, donat el caràcter d'estudi d'aquest treball de final de carrera, i per tant, aprofundir en el seu coneixement.

4. Implementació de l'eina

Un cop finalitzada les fases d'anàlisi i disseny, s'inicia la implementació de l'eina de rendiment d'una xarxa, a partir d'aquí TaFanet. El nom és un petit joc de paraules a partir de la funcionalitat de l'eina: "tafaner de la xarxa", i es remarquen en majúscules T i F com a sigles de Treball Final. En aquest capítol es descriu el procés d'implementació de TaFanet des de la preparació de l'entorn de desenvolupament fins a l'obtenció de la primera versió del producte v1.0, passant pels diferents prototips de l'eina.

4.1 Entorn de desenvolupament

L'entorn de desenvolupament està format per la versió 4.2.1(Juno) del programa Eclipse[3] versió de 32 bits, instal·lat a un PC amb processador Intel® i7 3.06 GHz i 6 GB de RAM amb sistema operatiu Microsoft Windows 7 Profesional SP1 de 64 bits.

Les raons de la utilització de la versió de 32 bits del software Eclipse en comptes de la versió de 64 bits ve donat per no existir versió de 64 bits de la llibreria Jpcap i donar errors de compilació Jpcap amb Eclipse 64 bits. Per la utilització de la llibreria Jpcap sota sistema operatiu Windows és necessària la instal·lació de WinPcap disponible a www.winpcap.org . Jpcap no necessita cap suport pels sistemes operatius tipus Unix, com ara Linux Ubuntu 12.04.

Il·lustració 10: Pagina principal del projecte WinPcap

La llibreria Jpcap en les diferents versions per Windows i Linux, així com les instruccions d'instal·lació estan disponibles a www.netresearch.ics.uci.edu/kfujii/Jpcap/doc/index.html . La instal·lació per defecte de Jpcap no és particularment efectiva i cal revisar que la llibreria jpcap.jar estigui present al directori de llibreries Java del sistema o afegir la llibreria al projecte dins d'Eclipse.

Desgraciadament, el projecte sembla estar en un punt mort de desenvolupament, com es pot observar a les instruccions d'instal·lació on és recomana la instal·lació de JRE 6 (Java Runtime Environment) per l'execució d'aplicacions basades en la llibreria Jpcap, o JDK 6 (Java Development Kit) pels desenvolupadors, fent molt de temps que estan operatives les versions 7.

Igualment, com s'ha comentat anteriorment, no existeix suport oficial per arquitectures de 64 bits. Si be, en el cas de Windows pot trobar-se algun projecte d'implementació de la biblioteca jpcap.dll en versió de 64 bits, no estan continuats i les biblioteques existents no estan lliures d'errors.

Es mostra a continuació les instruccions d'instal·lació per Jpcap a diversos sistemes operatius.

[3] El projecte Eclipse està disponible a www.eclipse.org

How to Install

<Microsoft Windows>

1. Download and install Java Runtime Environment 6 (if you want to simply run Jpcap-based applications) or JDK 6 (if you want to develop Jpcap-based applications).
2. Download and install the latest WinPcap.
3. Download and run Jpcap Self Installer for windows.

> If you have the previous version of Jpcap, please uninstall it first before installing a new version.
> If you forgot to uninstall the previous version, search '**Jpcap.dll**' and '**jpcap.jar**' from your system, delete them, and then install a new version.

> It is recommended to use JRE/JDK 6 or higher because Jpcap installer assumes JRE/JDK 6.
> If you are using JRE/JDK 5, set your '**CLASSPATH**' to include '***%SystemRoot%\Sun\Java\lib\ext***'

<Linux>

Use RPM package (Fedora, RedHat)

Download and install Jpcap RPM package.
For some distributions (e.g., Mandriva), JDK6 is also installed automatically.
However, for some distributions (e.g., Fedora Core), you have to manually install Sun's JDK6 for Linux as a RPM package before installing Jpcap RPM package.

Use Debian package (Ubuntu, GNU/Debian)

Download and install Jpcap Debian package.
For some distributions (e.g., Ubuntu), JDK6 is also installed automatically.
For some distributions (e.g., Debian GNU/Lunux), you may have to edit apt-line (e.g., include "no-free") before installing Jpcap Debian package. Also, you may have to update your GLIBC to version 2.4 or higher.

If you cannot install using RPM/Debian package, you can try to build from the source.

> It is recommended to use Sun's JRE/JDK 6 or higher because Jpcap package assumes JRE/JDK 6.
> If you are using JRE/JDK 5, set your '**CLASSPATH**' to include '***/usr/java/packages/lib/ext***'
> Also, Jpcap currently does not work with gcj / gij, so please use Sun's JRE/JDK.

<Mac OS X>

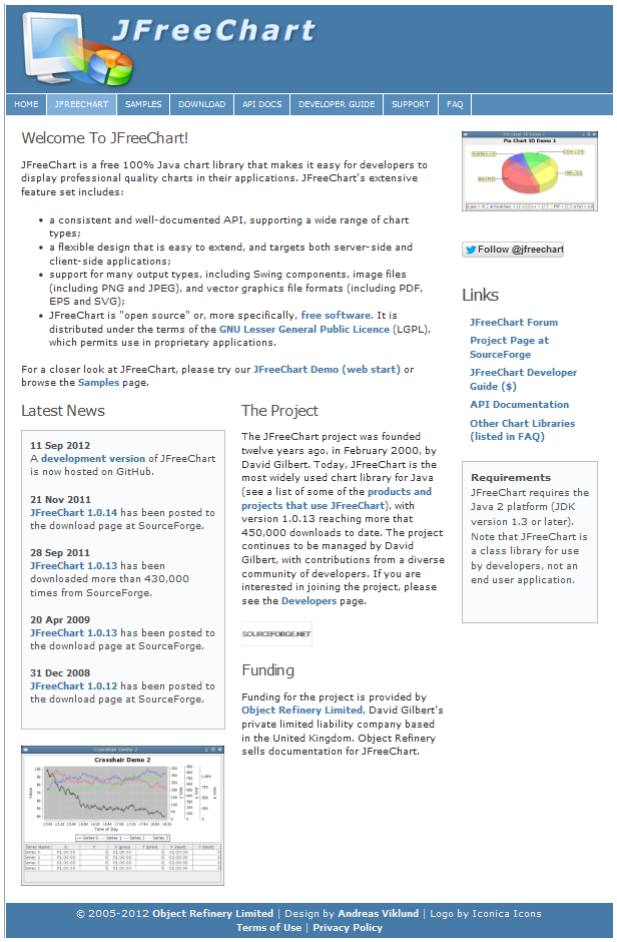
1. Both Java and libpcap are preinstalled on Mac OS X.
   If any of them is missing you should be able to install them from the Mac OS X install DVD.
2. Download and install Xcode.
   The default installation of Xcode should provide you with the toolchain required for compiling Jpcap.
3. Download and extract Jpcap source build.
4. Go to '***[Jpcap extracted directory]/src/c***' directory.
5. Run '**make**'.
6. Copy '**libjpcap.jnilib**' to '***/Library/Java/Extensions/***' directory.
7. Copy '***[Jpcap extracted directory]/lib/***jpcap.jar' to '***/Library/Java/Extensions/***'
   Or, place '*jpcap.jar*' to any directory and include it to your CLASSPATH.

<Source build (Linux/FreeBSD/Solaris)>

1. Make sure you have '**gcc**' and '**make**' installed to compile Jpcap.
   Other software/packages may be necessary (for example, you need '**build-essential**' package to install on Ubuntu).
2. Download and install libpcap (ver.0.9.4 or later) if not installed.
3. Download and extract Jpcap source build.
4. Go to '***[Jpcap extracted directory]/src/c***' directory.
5. Run '**make**'.
6. Copy '**libjpcap.so**' to '***[Java installed directory]/jre/lib/<arch>***'. <arch> is either '***i386***' or '***sparc***'.
   Or, if you are using Java 6, copy '**libjpcap.so**' to '***/usr/java/packages/lib/ext***'.
   Or, place '**libjpcap.so**' in the directory where your application is located.
7. Copy '***[Jpcap extracted directory]/lib/***jpcap.jar' to '***/usr/lib***' or '***[Java installed directory]/jre/lib/ext***'.
   Or, place '**jpcap.jar**' to any directory and include it in your CLASSPATH.

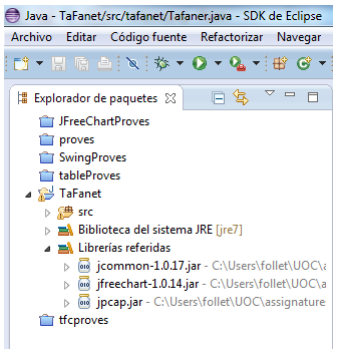
Il·lustració 11: Instruccions d'instal·lació Jpcap

Per utilitzar la llibreria JFreeChart caldrà afegir les llibreries corresponents al projecte dins d'Eclipse, en particular *jcommon-1.0.17.jar* i *jfreechart-1.0.14.jar* (versions actuals) que es poden descarregar des de la pagina del projecte JFreeChart www.jfree.org/jfreechart/index.html, on a més hi ha disponible documentació i exemples.



Il·lustració 12: Pàgina inicial projecte JFreeChart

I un cop preparat l'entorn, és el moment d'iniciar el desenvolupament de l'eina.



Il·lustració 13: Detall de biblioteques referides al projecte dins Eclipse

4.2 Primer Prototip: Captura de paquets

El propòsit d'aquest primer prototip és implementar una aplicació en mode text amb entrada i sortida per consola amb capacitat de captura dels paquets que circulin per la xarxa. Es preveu un primer pas de detecció de les interfícies de xarxa presents a l'equip, per que l'usuari pugui escollir la que més l'interessi i confirmar si la captura s'ha de realitzar en mode promiscu, és a dir capturant totes les trames de dades presents a la xarxa, malgrat l'equip no sigui el receptor.

```
************************************************************
**                                                        **
**                TaFanet prototip 1                      **
**                                                        **
**              Treball Final de Carrera                  **
**      Enginyeria Tècnica Informàtica de Sistemes        **
**                                                        **
**                                                        **
**               Jordi Arroyo Caballero                   **
************************************************************

Llista d'interfícies de xarxa presents a l'equip
0: Sun
1: VMware Virtual Ethernet Adapter
2: Realtek PCIe GBE Family Controller
3: VMware Virtual Ethernet Adapter

Escollir interficie (0 - 3)
2
mode promiscu (S/N)
s
nombre de paquets a capturar
1000
S'inicia la captura de paquets a la interficie:
NOM:  Realtek PCIe GBE Family Controller
MAC:  1c:6f:65:c0:de:d7
IP:   192.168.0.192
MASK: 255.255.255.0
mode: promiscu
```

Il·lustració 14: Configuració inici prototip 1

Un cop marcat un nombre de paquets a capturar, cada trama present a la xarxa serà analitzada per l'aplicació, mostrant per pantalla les dades pertinents per facilitar el seu estudi. Les dades a mostrar de cada trama varien depenent del tipus de protocol que utilitza, però amb trets comuns ates la tècnica d'encapsulament de les dades que s'utilitza en les trames de dades enviades per la xarxa.

Serà comú a totes les dades la mida de la trama de dades. De la trama Ethernet es mostrarà l'adreça MAC tant d'origen com de destinació. Aquests tres camps seran comuns a totes les trames. Encapsulat a la trama Ethernet analitzarem dos protocols: IP i ARP. Del protocol ARP es mostrarà el contingut del missatge ARP compost per tipus de missatge (petició o resposta), adreça hardware de l'emissor i la seva IP i adreça hardware (en cas de ser una petició ARP serà 00:00:00:00:00:00) del receptor i la seva IP.

```
42 bytes 1c:6f:65:c0:de:d7 -> ff:ff:ff:ff:ff:ff (ARP) petició  ARP de 1c:6f:65:c0:de:d7 (  192.168.0.192) a 00:00:00:00:00:00 (    192.168.0.1)
60 bytes 00:26:b6:08:db:5c -> 1c:6f:65:c0:de:d7 (ARP) resposta ARP de 00:26:b6:08:db:5c (    192.168.0.1) a 1c:6f:65:c0:de:d7 (  192.168.0.192)
```

Il·lustració 15: Sortida mode text captura paquets ARP prototip 1

Del protocol IP es mostrarà l'adreça IP d'origen i de destinació, així com el protocol que encapsula. Encapsulat en el protocol IP, s'analitzaran els protocols ICMP, TCP i UDP. Del protocol ICMP es mostrarà el tipus de missatge ICMP (ECHO petició, ECHO resposta, destí inaccessible. Dels protocols TCP i UDP es mostraran els ports d'origen i de destinació.

```
  74 bytes 1c:6f:65:c0:de:d7 -> 00:1e:8c:e4:9c:37 (IP)   192.168.0.192 ->   192.168.0.199 ICMP: ECHO petició
  74 bytes 00:1e:8c:e4:9c:37 -> 1c:6f:65:c0:de:d7 (IP)   192.168.0.199 ->   192.168.0.192 ICMP: ECHO resposta

  66 bytes 1c:6f:65:c0:de:d7 -> 00:26:b6:08:db:5c (IP)   192.168.0.192 ->  173.194.41.247 (TCP) 50584 ->    80

 154 bytes 00:26:b6:08:db:5c -> 1c:6f:65:c0:de:d7 (IP)         8.8.8.8 ->   192.168.0.192 (UDP)    53 -> 50550
```

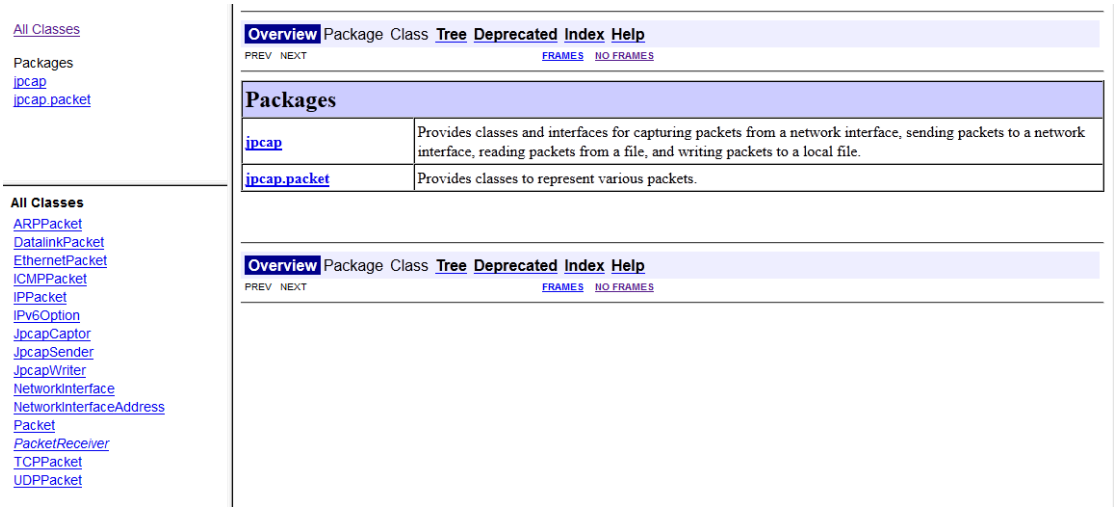
Il·lustració 16: Sortida mode text captura paquets IP prototip 1

No es contemplarà en aquesta primera versió de l'eina de rendiment de xarxa, i per tant tampoc en aquest primer prototip de l'aplicació, altres protocols de la capa de xarxa com són els corresponents al IPv6. Les trames que encapsulin protocols de xarxa que no siguin ARP o IP (IPv4) és comptabilitzaran simplement com a trama Ethernet.

Paral·lelament, es porta una estadística de protocols i es comprova si l'emissor o el receptor de la trama pertanyen a la xarxa local, a fi de mostrar una estadística al finalitzar la captura d'equips detectats. Junt amb l'estadística d'equips detectats es mostra una estadística dels protocols detectats.

4.2.1 Estratègia de captura i classificació de trames

Com s'ha comentat a l'apartat sobre llenguatge de programació escollit, java no disposa de cap llibreria del sistema amb accés suficient a les interfícies de xarxa que permeti realitzar l'anàlisi de les trames de dades que circulen per la xarxa, i amb aquesta finalitat s'utilitza la llibreria Jpcap, que està documentada a la seva API.



Il·lustració 17: API llibreria Jpcap

El primer pas serà detectar les interfícies de xarxa presents a l'equip on s'executa l'aplicació. Amb aquesta finalitat la classe *JpcapCaptor* ofereix el mètode *getDeviceList()*, que retorna una llista d'objectes *NetworkInterface* que conté la representació d'una interfície de xarxa amb informació del nom de la interfície, la seva descripció, l'adreça MAC i l'adreça IP. Un cop es decideix la interfície de xarxa de la que és capturaran paquets s'utilitza el mètode *openDevice*, al que se li passen per paràmetre la interfície escollida, el màxim de bytes a capturar de cada trama, modo promiscuo (*booleà*) i temps de timeout. L'execució d'aquest mètode obre la interfície i retorna un objecte *JcapCaptor* sobre el que es podrà iniciar la captura.

Són diversos els mètodes que permeten capturar paquets: *getPacket*, *processPacket* i *loopPaquet*. S'ha escollit el mètode *loopPaquet(int count, PacketReceiver handler)* per què permet definir el nombre de paquets a capturar i quina serà la classe que implementi *PacketReceiver* i que s'encarregarà de rebre aquets paquets. Això permet definir que la classe principal implementa la interfície *PacketReceiver* i definir el mètode *recievePacket (Packet p)*. La funcionalitat que proporciona aquesta implementació és que un cop s'executa el mètode *loopPacket* es crea un nou fil d'execució (thread), que crida al mètode *recievePacket (Packet p)* cada cop que una nova trama de dades es capturada i li arriba com un objecte *Packet*, també definit a la llibreria Jpcap.

Un cop els paquets arriben al mètode *recievePacket* ja es pot iniciar el procés d'anàlisi i classificació de les trames. La classe *Packet* pertany al package jpcap.packet de la llibreria Jpcap, i dins del package es troben també classes que hereten de la classe Packet, com pot ser IPPacket, TCPPacket, etc que faciliten

l'anàlisi de les trames de dades, però donat el caràcter d'investigació sobre xarxes de computadors i els seus protocols d'aquest Treball de final de Carrera s'ha desestimat la seva utilització.

La classe *Packet* ofereix una sèrie de camps públics on emmagatzema les dades de la trama de dades capturades:

Field Summary	
int	caplen Captured length
byte[]	data Packet data (excluding the header)
DatalinkPacket	datalink Datalink layer header
static Packet	EOF Returned by JpcapCaptor.getPacket() when EOF was reached while reading from an offline file.
byte[]	header Header data
int	len Length of this packet
long	sec Captured timestamp (sec)
long	usec Captured timestamp (micro sec)

Il·lustració 18: Camps de la classe Packet

Com es pot veure a la documentació de la llibreria Jpcap el camp *header* de la classe *Packet* és un array del tipus byte de les dades de capçalera de les trames de dades, fet que permetrà analitzar cada trama tal i com si s'analitzés directament les dades sobre la xarxa. L'anàlisi que caldrà fer sobre cada byte[] header és anar al byte corresponent segons cada protocol de xarxa, llegir les dades i anar omplint els camps de cada tipus de paquets.

4.2.2 Utilització dels packages a la implementació

Per clarificar la implementació es creen diversos packages que classifiquen les classes per la seva funcionalitat. A partir de l'esquema de classes sorgit a les fases d'anàlisi i disseny es creen les classes troncals de l'aplicació que es complementen amb noves classes que es presenten a continuació.

Com es pot observar a l'arbre de la Il·lustració 18, la classe principal de l'aplicació *Tafaner* s'inclou dins del package *tafanet* junt amb les classes *Captura* i *InterficieXarxa*. A Captura s'emmagatzemen totes les dades de la captura actual, temps de captura, nombre de paquets a capturar, sobre quina interfície de xarxa es realitza, etc. La classe InterficieXarxa representa la interfície de xarxa sobre la que es realitza la captura i emmagatzema les trames un cop classificades.

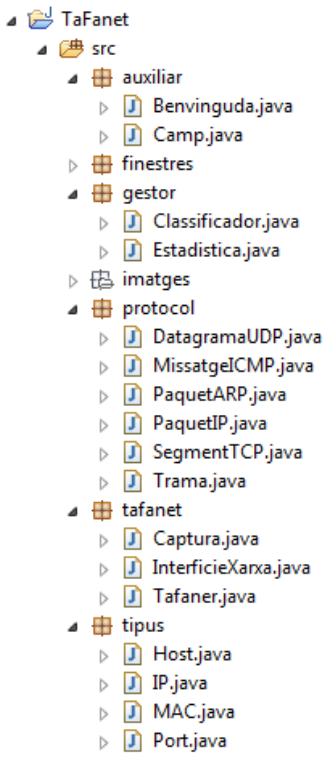
Al package *protocol* es troben les classes que representen els diferents protocols presents a la xarxa. Com va sorgir de la fase d'anàlisi les diferents classes hereten de la classe *trama*, fet que facilita posteriorment la creació d'estadístiques utilitzant el mètode *instanceof* sobre un objecte *Trama* obtenint directament quin protocol ha generat l'objecte.

El package *gestor* conté les diferents classes encarregades de classificar (*Classe Classificador*) els paquets i per tant generar els objectes d'acord al tipus de trama capturat i de generar estadístiques sobre les dades obtingudes (*Classe Estadistica*).

A fi de facilitar l'emmagatzematge d'informació es creen diferents classes a modo d'estructures de dades. Les classes *IP*, *MAC*, *Port* i *Host* (*IP* i *MAC* d'un mateix equip) s'agrupen dins del packages *tipus*.

El package *auxiliar* està destinat a encabir les *interface* estàtiques de dades. La interface *Camp* conté les posicions dels diferents camps del protocols dins de la capçalera de les trames de dades.

Per últim es creen dos packages més, *finestres* on s'inclouran les classes que representin les interfícies d'usuari i *imatges* on s'inclouran les imatges corresponent a les icones de les interfícies d'usuari.



Il·lustració 19: Packages de l'aplicació

4.2.3 Exemple d'execució del prototip 1

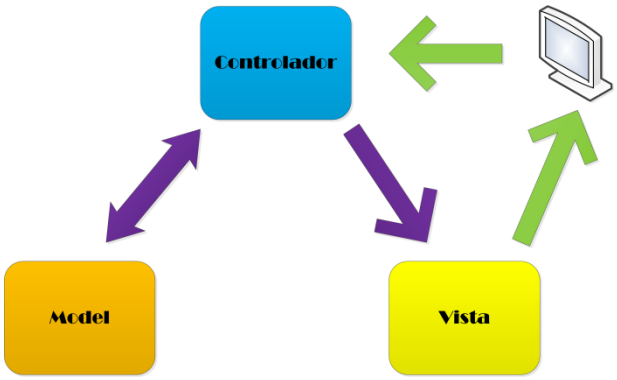
Es presenta a continuació un exemple d'execució de l'aplicació un cop finalitzat el primer prototip. L'instant correspon a un inici de sessió al campus de la UOC.

```
*********************************************************
**                                                     **
**                TaFanet prototip 1                   **
**                                                     **
**              Treball Final de Carrera               **
**      Enginyeria Tècnica Informàtica de Sistemes     **
**                                                     **
**                                                     **
**               Jordi Arroyo Caballero                **
*********************************************************

Llista d'interfícies de xarxa presents a l'equip
0: Sun
1: VMware Virtual Ethernet Adapter
2: Realtek PCIe GBE Family Controller
3: VMware Virtual Ethernet Adapter

Escollir interficie (0 - 3)
2
mode promiscu (S/N)
s
nombre de paquets a capturar
25
S'inicia la captura de paquets a la interficie:
NOM:  Realtek PCIe GBE Family Controller
MAC:  1c:6f:65:c0:de:d7
IP:   192.168.0.192
MASK: 255.255.255.0
mode: promiscu
     1: 09-01-2013 19:01:05:471  165 bytes 00:1e:8c:e4:9c:37 -> 01:00:5e:7f:ff:fa (IP)   192.168.0.199 -> 239.255.255.250 (UDP)  2336 ->  1900
     2: 09-01-2013 19:01:07:585   70 bytes 1c:6f:65:c0:de:d7 -> 00:26:b6:08:db:5c (IP)   192.168.0.192 ->         8.8.8.8 (UDP) 61915 ->    53
     3: 09-01-2013 19:01:07:639   86 bytes 00:26:b6:08:db:5c -> 1c:6f:65:c0:de:d7 (IP)         8.8.8.8 ->   192.168.0.192 (UDP)    53 -> 61915
     4: 09-01-2013 19:01:07:640   66 bytes 1c:6f:65:c0:de:d7 -> 00:26:b6:08:db:5c (IP)   192.168.0.192 ->   213.73.40.246 (TCP) 50611 ->   443
     5: 09-01-2013 19:01:07:704   66 bytes 00:26:b6:08:db:5c -> 1c:6f:65:c0:de:d7 (IP)   213.73.40.246 ->   192.168.0.192 (TCP)   443 -> 50611
     6: 09-01-2013 19:01:07:705   54 bytes 1c:6f:65:c0:de:d7 -> 00:26:b6:08:db:5c (IP)   192.168.0.192 ->   213.73.40.246 (TCP) 50611 ->   443
     7: 09-01-2013 19:01:07:705  221 bytes 1c:6f:65:c0:de:d7 -> 00:26:b6:08:db:5c (IP)   192.168.0.192 ->   213.73.40.246 (TCP) 50611 ->   443
     8: 09-01-2013 19:01:07:755 1514 bytes 00:26:b6:08:db:5c -> 1c:6f:65:c0:de:d7 (IP)   213.73.40.246 ->   192.168.0.192 (TCP)   443 -> 50611
     9: 09-01-2013 19:01:07:756 1514 bytes 00:26:b6:08:db:5c -> 1c:6f:65:c0:de:d7 (IP)   213.73.40.246 ->   192.168.0.192 (TCP)   443 -> 50611
    10: 09-01-2013 19:01:07:756   54 bytes 1c:6f:65:c0:de:d7 -> 00:26:b6:08:db:5c (IP)   192.168.0.192 ->   213.73.40.246 (TCP) 50611 ->   443
    11: 09-01-2013 19:01:07:758 1514 bytes 00:26:b6:08:db:5c -> 1c:6f:65:c0:de:d7 (IP)   213.73.40.246 ->   192.168.0.192 (TCP)   443 -> 50611
    12: 09-01-2013 19:01:07:803  347 bytes 00:26:b6:08:db:5c -> 1c:6f:65:c0:de:d7 (IP)   213.73.40.246 ->   192.168.0.192 (TCP)   443 -> 50611
    13: 09-01-2013 19:01:07:803   54 bytes 1c:6f:65:c0:de:d7 -> 00:26:b6:08:db:5c (IP)   192.168.0.192 ->   213.73.40.246 (TCP) 50611 ->   443
    14: 09-01-2013 19:01:07:804  364 bytes 1c:6f:65:c0:de:d7 -> 00:26:b6:08:db:5c (IP)   192.168.0.192 ->   213.73.40.246 (TCP) 50611 ->   443
    15: 09-01-2013 19:01:07:805   79 bytes 1c:6f:65:c0:de:d7 -> 00:26:b6:08:db:5c (IP)   192.168.0.192 ->         8.8.8.8 (UDP) 54403 ->    53
    16: 09-01-2013 19:01:07:857   97 bytes 00:26:b6:08:db:5c -> 1c:6f:65:c0:de:d7 (IP)   213.73.40.246 ->   192.168.0.192 (TCP)   443 -> 50611
    17: 09-01-2013 19:01:07:865  126 bytes 00:26:b6:08:db:5c -> 1c:6f:65:c0:de:d7 (IP)         8.8.8.8 ->   192.168.0.192 (UDP)    53 -> 54403
    18: 09-01-2013 19:01:07:866   66 bytes 1c:6f:65:c0:de:d7 -> 00:26:b6:08:db:5c (IP)   192.168.0.192 ->    178.255.83.1 (TCP) 50612 ->    80
    19: 09-01-2013 19:01:08:009   60 bytes 00:26:b6:08:db:5c -> 1c:6f:65:c0:de:d7 (IP)    178.255.83.1 ->   192.168.0.192 (TCP)    80 -> 50612
    20: 09-01-2013 19:01:08:009   54 bytes 1c:6f:65:c0:de:d7 -> 00:26:b6:08:db:5c (IP)   192.168.0.192 ->    178.255.83.1 (TCP) 50612 ->    80
    21: 09-01-2013 19:01:08:010  533 bytes 1c:6f:65:c0:de:d7 -> 00:26:b6:08:db:5c (IP)   192.168.0.192 ->    178.255.83.1 (TCP) 50612 ->    80
    22: 09-01-2013 19:01:08:061   54 bytes 1c:6f:65:c0:de:d7 -> 00:26:b6:08:db:5c (IP)   192.168.0.192 ->   213.73.40.246 (TCP) 50611 ->   443
    23: 09-01-2013 19:01:08:157   60 bytes 00:26:b6:08:db:5c -> 1c:6f:65:c0:de:d7 (IP)    178.255.83.1 ->   192.168.0.192 (TCP)    80 -> 50612
    24: 09-01-2013 19:01:08:173  912 bytes 00:26:b6:08:db:5c -> 1c:6f:65:c0:de:d7 (IP)    178.255.83.1 ->   192.168.0.192 (TCP)    80 -> 50612
    25: 09-01-2013 19:01:08:173   60 bytes 00:26:b6:08:db:5c -> 1c:6f:65:c0:de:d7 (IP)    178.255.83.1 ->   192.168.0.192 (TCP)    80 -> 50612


  Total de trames:           25
     Paquets ARP:             0
     Paquets IP:             25
        Missatges ICMP:       0
        Segments TCP:        20
        Datagrames UDP:       5

Hosts Xarxa Local                                Hosts Externs a la xarxa
IP: 192.168.0.199    MAC: 00:1e:8c:e4:9c:37      IP: 239.255.255.250  MAC: 01:00:5e:7f:ff:fa
                                                 IP: 8.8.8.8          MAC: 00:26:b6:08:db:5c
                                                 IP: 213.73.40.246    MAC: 00:26:b6:08:db:5c
                                                 IP: 178.255.83.1     MAC: 00:26:b6:08:db:5c
```

Il·lustració 20: Exemple d'execució prototip 1

4.3 Creació de les interfícies d'usuari

Per la creació de les interfícies d'usuari java ofereix la llibreria swing que permet utilitzar el patró MVC (model-vista-controlador). La utilització del patró MVC separa la lògica de l'aplicació de la interfície d'usuari, fet que facilita l'evolució per separat tant de la lògica de l'aplicació com del desenvolupament de la interfície, aquest fet facilita la reutilització i la flexibilitat de l'aplicació.



Il·lustració 21: Patró Model Vista Controlador

El patró MVC defineix el model com la part de l'aplicació encarregada de realitzar la part lògica, al cas de l'aplicació que ens ocupa seria tota la part implementada al primer prototip, captura de paquets, anàlisi dels paquets, classificació i la generació d'estadístiques a partir de les dades obtingudes. La vista correspon a les interfícies d'usuari generades mitjançant les classes de la liberià Swing, encarregada de mostrar a l'usuari les dades de programa que rep del controlador.

El controlador s'encarrega de rebre les entrades de part de l'usuari per comunicar-les al model, un cop rep dades del model les passa a vista que les mostra a l'usuari. El controlador s'executa com un nou fil d'execució, el que permet que l'aplicació continuï executant-se sense veure's afectada. Per implementar el controlador ofereix diferents classes de gestió d'esdeveniments al package *event* com poden ser *MouseListener*, *MouseAdapter*, *MouseEvent*, etc.

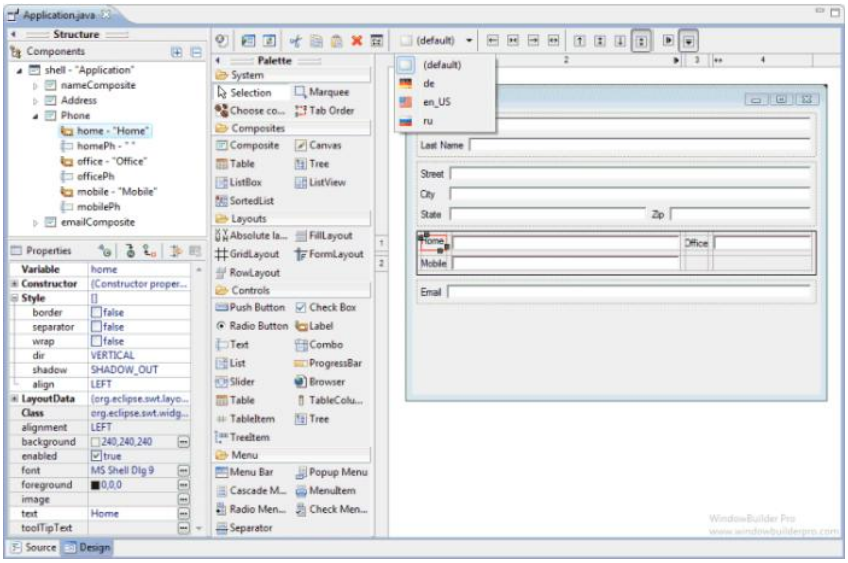
4.3.1 WindowsBuilder

A l'hora de facilitar la creació de les interfícies d'usuari el projecte Eclipse ofereix una eina gràfica anomenada *WindowBuilder*[4]. No forma part, de les distribucions estàndards d'Eclipse, però fàcilment descarregable i instal·lable des de la web del projecte http://www.eclipse.org/windowbuilder/ , on es troben també les instruccions per a la instal·lació.

Com es pot veure a la imatge adjunta WindowBuilder proporciona una potent eina gràfica que facilita en gran mesura la creació de les interfícies d'usuari. Permet afegir elements tant en format gràfic, com a la vegada afegir codi directament sense cap restricció, detectat automàtic i per tant modificable gràficament a posteriori.

Igualment, que es creen gràficament les interfícies d'usuari que formen la vista del patró MVC, també permet incloure la detecció dels esdeveniments que configuren el controlador del patró i la interrelació amb el model format per l'aplicació, sempre podent passar de l'entorn gràfic al mode text en qualsevol moment.

[4] Imatge adjunta pertany a la web del projecte WindowBuilder.

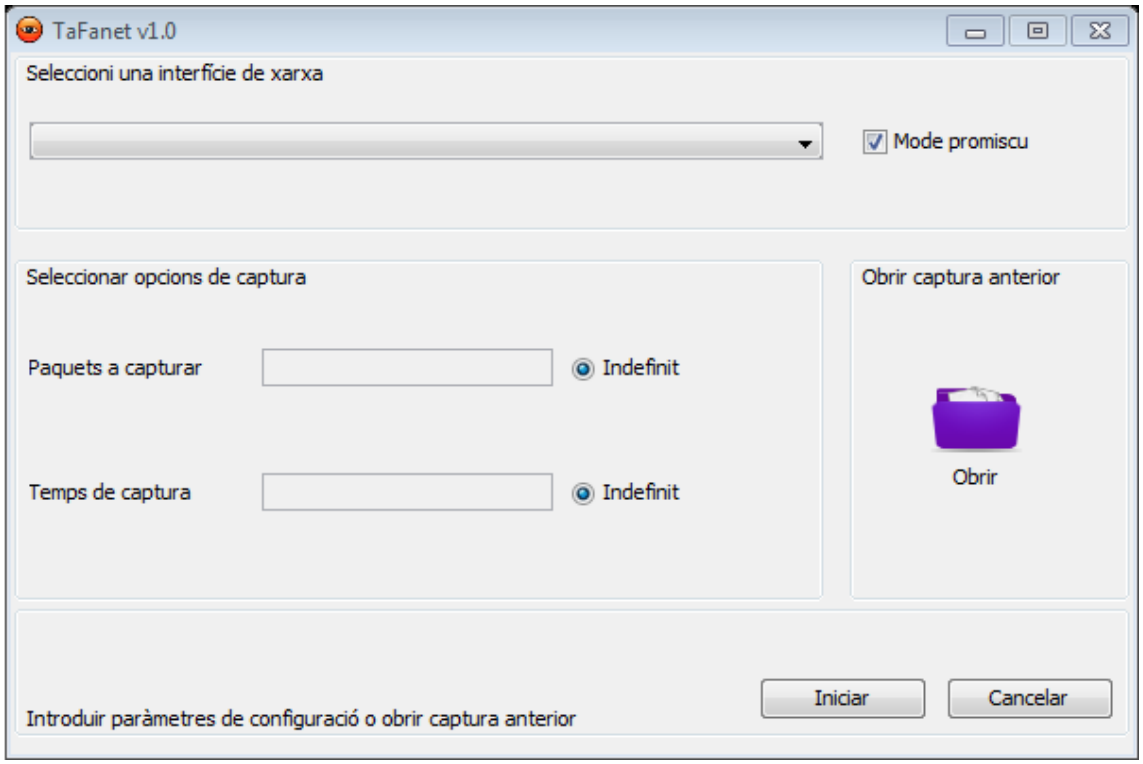


Il·lustració 22: Eina gràfica WindowBuilder

4.3.2 Finestra de configuració

La primera interfície d'usuari que s'ha creat és la pantalla de configuració, on es configuren els paràmetres de la captura. La pantalla permet escollir una interfície de xarxa de les presents a l'equip, així com el nombre de paquets a capturar i la duració de la captura. Tant el temps, com el nombre de paquets, per defecte s'ofereix indefinit, el que permet una captura indefinida en el temps i el nombre de paquets.

Per defecte ofereix obrir la interfície de xarxa en mode promiscu, podent-se deshabilitar l'opció i permet escollir visualitzar una captura guardada amb anterioritat, per facilitar el seu anàlisi.

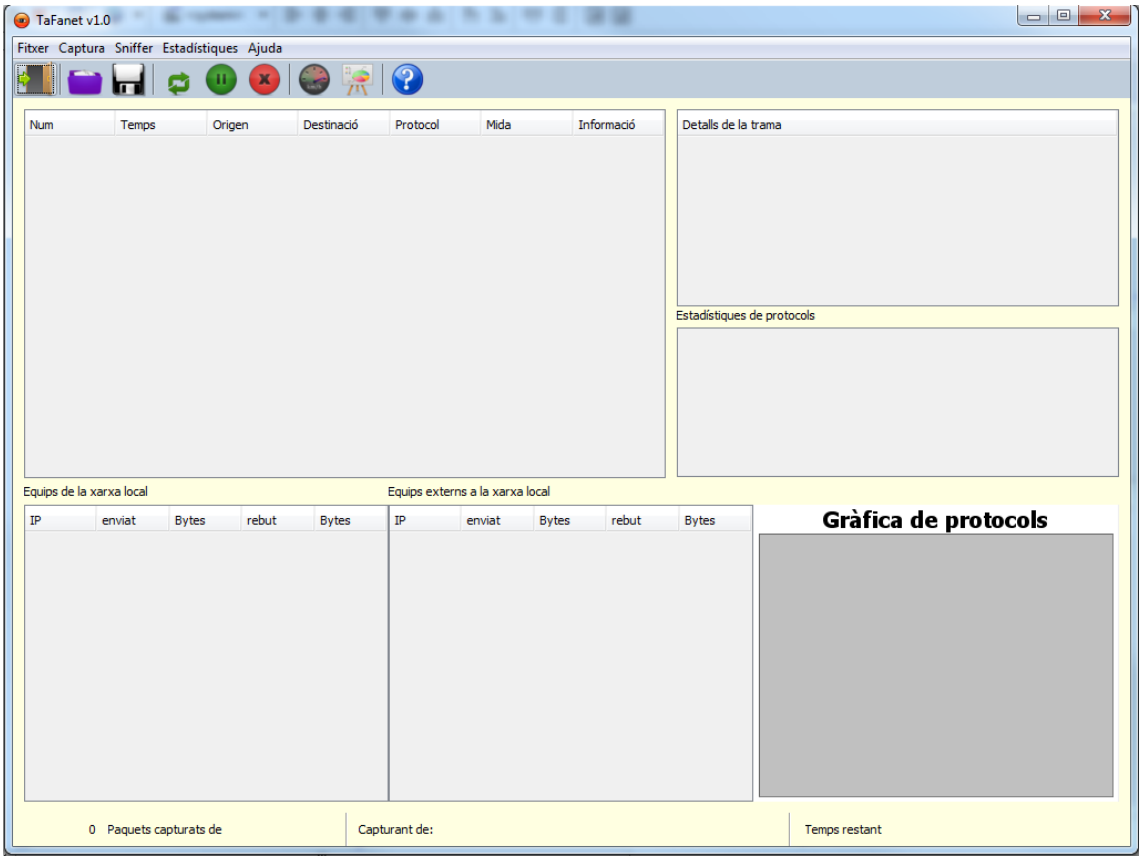


Il·lustració 23: Finestra de configuració

4.3.3 Finestra principal de l'eina

La finestra principal de l'eina, queda dividida en diferents parts, i cadascuna mostrarà una informació diferenciada. Es composa de les següents parts:

- Barra de menús.
- Barra d'icones.
- Taula principal, on es mostra informació de les trames capturades a modo sniffer.
- Marc d'informació de les trames, que mostra informació detallada de la trama seleccionada.
- Dos taules de Hosts, que mostren els equips detectats durant la captura.
- Marc d'estadístiques de protocols, on es mostra informació del protocols detectats.
- Gràfica de protocols o ample de banda, on es mostra una gràfica en porcions del protocols detectats o un diagrama de barres de l'ample de banda utilitzat.



Il·lustració 24: Finestra principal

4.3.4 Finestres d'error, informació i quadres de diàleg

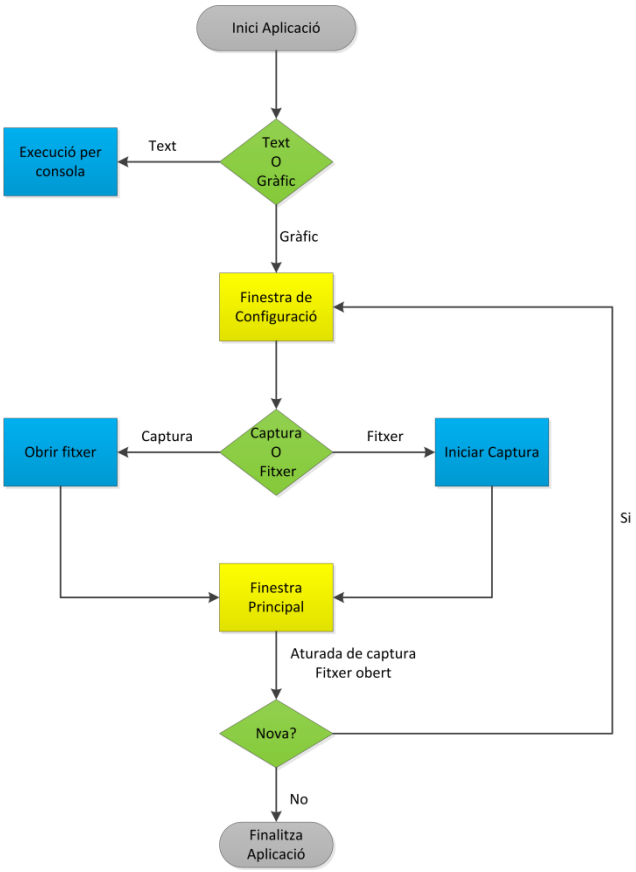
De cara a facilitar la implementació de les finestres d'error, informació i quadres de diàleg en general el package swing ofereix la classe JOptionPane. Aquesta classe permet implementar en poques línies de codi tot el necessari per recollir la resposta de l'usuari al missatge mostrat. Ofereix diferents mètodes com per exemple ShowMessageDialog i ShowOptionDialog que configuren fàcilment la finestra a mostrar i les possibles respostes esperades per part de l'usuari, que es recullen en variables que ofereixen els propis mètodes de cara a la gestió de l'aplicació.

Igualment per obrir i guardar fitxers la classe FileChosser del propi package swing facilita la implementació de les finestres que permeten accedir a l'arbre de directoris del usuari, de cara a escollir el possible fitxer a obrir o definir el nom d'un fitxer a guardar.

4.4 Segon prototip: Interrelació de les interfícies d'usuari i la classificació de paquets

Com s'ha comentat a l'apartat anterior, s'han dissenyat dues finestres: la finestra de configuració, interfície d'usuari que demana configurar la captura a l'usuari o obrir un fitxer d'una captura anterior, i la finestra principal dissenyada per ser la interfície que mostri a l'usuari totes les dades que s'obtinguin de les trames de dades analitzades en la captura configurada o mostrar les dades d'una captura anterior, si s'ha seleccionat l'opció d'obrir fitxer a la finestra de configuració.

Per donar a la finestra principal la capacitat de reiniciar captures, obrir una nova captura o visualitzar un fitxer un cop aturada la captura configurada, es defineix un bucle perpetu que permetrà tornar a mostrar la finestra de configuració si es donen les condicions adequades, fonamentalment en el cas d'estar realitzant-se una captura que sigui aturada. Es presenta a continuació el diagrama de flux que defineix el citat bucle.



Il·lustració 25: Diagrama de flux de l'eina

Com es pot observar al diagrama de flux, l'eina permet dos formats d'execució: mode text i mode gràfic. El mode text correspon al primer prototip de l'eina i permet una captura simplificada de paquets de xarxa que es visualitzen per consola. Per iniciar en mode consola s'ha d'executar l'eina afegint text com argument: "java -jar TaFanet.jar text".

L'execució en mode gràfic, inicia l'aplicació mostrant la finestra de configuració, on l'usuari ha d'escollir la interfície de xarxa, el nombre de paquets a captura i temps de captura. També és possible obrir un fitxer d'una captura anterior. A partir d'aquí es mostra la finestra principal, al finalitzar la captura o el visionat del fitxer obert, l'usuari pot demanar iniciar una nova captura. Aquesta acció fa novament visible la finestra de configuració (amb l'opció d'obrir fitxers deshabilitada, al estar disponible a la pròpia finestra principal).

Les diverses funcionalitats de la finestra principal queden descrites al manual d'usuari i es presenta un breu resum en el següent apartat.

4.4.1 Anàlisi de trames i protocols.

S'han implementat diferents anàlisi de les trames capturades. S'analitza els protocols utilitzats, els equips que intervenen i la mida de les trames a fi de comptabilitzar l'ample de banda ocupat a la xarxa. Tot això es mostra a la finestra principal de l'eina a les diferents parts que la formen.

- Taula sniffer. Mostra seqüencialment les trames capturades oferint un resum de les dades característiques de cada trama.
- Dades de la trama. Es possible escollir una de les trames de la taula sniffer, clicant a sobre, i ofereix un anàlisi més detallat de la trama
- Taules de equips detectats. Són dues taules, equips de xarxa local i equips externs a la xarxa, que mostren els equips detectats en les comunicacions capturades a la xarxa això com el nombre de trames en que intervenen, tant com a equip origen com destinació, i la mida en bytes de les trames que han enviat o rebut. S'actualitza a cada nova trama.
- Estadística de protocols. Mostra una estadística dels protocols de xarxa detectats a nivell de xarxa i de transport, actualitzat a cada nova trama capturada
- Gràfica de protocols o diagrama d'ample de banda. Mostra una gràfica en porcions segons el percentatge de cada protocol com a opció predefinida. Seleccionant l'opció ample de banda mostra un diagrama de barres que mostra l'ample de banda ocupat de la xarxa. Mostra quatre barres: total, pujada, baixada i altres.

4.4.2 Gestió d'esdeveniments

Durant l'execució de l'aplicació es produeixen una sèrie d'esdeveniments que han de ser atesos i rebre la corresponent resposta per part de l'eina. Generalitzant, es poden classificar els esdeveniments produïts en tres tipus que són:

<u>Captura d'una nova trama</u>

La resposta d'aquesta esdeveniment es produeix per la pròpia funcionalitat de l'eina jpcap. La classe principal de l'aplicació implementa la interface *PacketReceiver*, que obliga a implementar el mètode *receivePaquet(Packet p)*. Un cop s'obre la interfície de xarxa de la que s'esperen capturar els paquets, es crida al mètode *loopPaquet* de la classe *JcapCaptor*, que inicia el bucle de captura i crea un nou fil d'execució que roman a l'espera de l'arribada de nous paquets. Cada nou paquet capturat es enviat al mètode *receivePaquet(Packet p)*.

Dintre d'aquest mètode s'implementa la lògica de l'aplicació, encarregant-se d'enviar els paquets a les classes encarregades de classificar les trames i extreure la informació oportuna per generar les estadístiques i gràfiques descrites a l'apartat anterior. S'aprofita també aquest mètode per enviar la trama ja classificada a la classe que gestiona la finestra principal, per ser introduïda a la taula sniffer. Aquest fet provoca un esdeveniment a la classe principal que actualitza la resta de taules.

<u>Esdeveniments periòdics o temporals</u>

Hi ha tres objectes de la classe Timer a l'aplicació. Aquests objectes creen un nou fil d'execució que roman inactiu, activant-se cada cert pas de temps. El primer d'ells, es crida des de la classe *Captura* (encarregada d'emmagatzemar les dades de configuració de la captura) en cas de marcar l'usuari un límit de temps. S'activa un cop per segon, decrementa un segon al temps restant d'execució i el comunica a la finestra principal per actualitzar el comptador regressiu.

Un segon objecte de la classe Timer es crea a la classe Classificador. Aquesta classe, s'encarrega de rebre els paquets capturats, comprova els camps de la capçalera referent als protocols i crea un objecte del tipus que correspon a la trama capturada. Aprofita, aquest anàlisi de la capçalera, per llegir la mida de la trama i si l'equip on s'està realitzant la captura és origen o destí de la comunicació. Aquesta informació la utilitza el Timer, que s'activa cada segon per realitzar els càlculs d'ample de banda.

Per últim, la classe que gestiona la interfície principal d'usuari crea un nou objecte de tipus Timer, que també s'activa un cop per segon. Aquest Timer és el fil encarregat de demanar a les classe corresponents la informació referent al nombre de protocols de cada tipus i/o l'ample de banda ocupat. Amb la

informació rebuda actualitza la gràfica de protocols o el diagrama d'ample de banda, depenent de quin dos sigui actiu per pantalla.

Esdeveniments creats per l'usuari

Aquest tipus d'esdeveniment es produeixen cada cop que l'usuari interactua amb l'aplicació, bé clicant alguna de les icones o menús de l'eina o sobre la taula sniffer. Per gestionar aquests esdeveniments, s'afegeixen escoltadors a aquests elements. S'utilitza el mètode de crear subclasses, dins de les classes que gestionen les interfícies per implementar els escoltadors.

4.5 Fase de proves

L'equip on s'ha realitzat el desenvolupament de l'eina, i per tant on s'ha realitzant la gran majoria de proves és un PC amb processador Intel® i7 3.06 GHz i 6 GB de RAM amb sistema operatiu Microsoft Windows 7 Profesional SP1 de 64 bits. S'han realitzat proves complementaries en maquines virtuals instal·lades al propi equip de proves amb els següents sistemes operatius: Windows XP sp3 Profesional i Linux Ubuntu 12.04 versió de 32 bits.

Les proves han servit fonamentalment per comprovar el perfecte funcionament de l'aplicació i per comprovar la seva portabilitat entre les diferents plataformes. L'aplicació ha funcionat de manera similar a tots els sistemes operatius on ha sigut provada, tret de petits ajustos que ha calgut fer al provar l'eina en sistemes operatius Linux, i un problema d'estabilitat en l'execució en plataformes de 64 bits.

Es presenten a continuació algunes de les correccions que ha calgut fer:

La comprovació de la portabilitat a Linux de l'aplicació ha fet detectar diverses excepcions que calien controlar degut al canvi de sistema operatiu i el funcionament particular de Jpcap a cada plataforma. A mode d'exemple, la classe *NetworkInterface* del Jpcap, ofereix el mètode *addresses* que retorna un array de *NetworkInterfaceAddress*. Aquest mètode s'utilitza a l'aplicació per detectar l'adreça IP, l'adreça MAC i la mascara de la interfície de xarxa escollida. Es troba una diferencia entre el funcionament del mètode a Windows i a Linux oferint en ordre diferent l'adreça IPv4 i l'adreça IPv6, per lo que ha calgut afegir un control de tipus d'adreça a l'hora de generar l'objecte *InterficieXarxa*.

La classe *JpcapWriter* que permet guardar en un fitxer els paquets capturats provoca una errada a la maquina virtual Java al executar el mètode *close()* per tancar el fitxer un cop escrit, als sistemes operatius de 64 bits, malgrat executar l'aplicació amb Java 32 bits. Al no tractar-se d'una excepció que es pogués controlar, s'ha optat per no executar el mètode *close()*. El fitxer es tanca un cop finalitza l'execució de l'aplicació.

S'ha detectat un esgotament progressiu de la memòria lliure de la màquina virtual Java al capturar un nombre elevat de paquets. Sobre els 150.000 paquets s'observa l'alentiment de l'eina arribant cap als 170.000 a bloquejar-se per complert. Per evitar aquesta situació s'ha afegit un límit de captura de 100.000 paquets.

La resta de proves que s'han anat realitzant han servit per comprovar la captura dels paquets adequadament, si bé la correcció de les captures va ser provada en profunditat al prototip 1, generant paquets i comprovant que es capturaven els paquets esperats. Per exemple, utilitzar l'eina ping per generar missatges ICMP, iniciar el navegador web per generar peticions DNS a fi de capturar paquets UDP, etc.

S'han realitzat proves de funcionalitat de les interfícies d'usuari, i per últim s'han fet comparatives de captures amb el programari whireshark, obtenint-se captures molt similars, sempre dintre de les limitacions d'anàlisi de protocols de TaFanet v1.0, però que han resultat prou satisfactòries.

5. Documentació de l'aplicació

Junt amb l'eina de rendiment de xarxa s'adjunten dos manuals: el manual d'instal·lació i el manual d'usuari.

5.1 Manual d'instal·lació

Per al funcionament de l'eina d'anàlisi de rendiment de xarxa TaFanet v1.0, cal tenir instal·lat el següent programari:

Sistemes operatius Windows:

- Java Runtime Environment 7 (JRE7) versió 32bits. Si no està instal·lada o actualitzada en el sistema pot descarregar-se de la pagina d'Oracle http://www.java.com/es/download/ . La versió actual de JRE és la versió 7 Update 11.

  Important!!! Degut a que Jpcap només treballa en 32 bits, per a sistemes de 64 bits és necessari tenir instal·lada la versió de 32bits de la Java Runtime Environment. Si amb la instal·lació automàtica de Java de l'enllaç anterior proporciona la versió de 64 bits, està disponible la descarrega manual a l'adreça http://www.java.com/es/download/manual.jsp

- WinPCap. Driver que proporciona accés de baix nivell de xarxa al sistema operatiu. La versió actual és la 4.1.2 i és troba dins del directori *recursos/Windows* del paquet d'instal·lació de l'eina. Es pot accedir a http://www.winpcap.org/ per comprovar si existeix una versió més recent.

- Jpcap. Llibreria java i llibreria dinàmica (DLL) que comunica el driver winPcap amb la maquina virtual Java. Al directori *recursos/Windows* del paquet d'instal·lació de l'eina es troba l'executable "JcapSetup-0.7.exe", ultima versió del Jpcap que cal instal·lar. Es pot comprovar a la pàgina http://netresearch.ics.uci.edu/kfujii/Jpcap/doc/index.html per versions posteriors

Sistemes operatius Linux:

- Java Runtime Environment 7 (JRE7) versió de 32 bits. Si no està instal·lada o actualitzada en el sistema pot descarregar-se de la pagina d'Oracle http://www.java.com/es/download/manual.jsp . La versió actual de JRE és la versió 7 Update 11.

- Jpcap. Llibreria Java que permet capturar paquets amb la maquina virtual Java. Al directori *recursos/Linux* del paquet d'instal·lació de l'eina es troben els paquets "jpcap-0.7.deb", per distribucions Debian o Ubuntu i "jpcap-0.7.tar.gz", per distribucions RedHat o Fedora. Última versió disponible del Jpcap que cal instal·lar. Per versions posteriors, es pot comprovar a la pàgina http://netresearch.ics.uci.edu/kfujii/Jpcap/doc/index.html per versions posteriors

L'aplicació TaFanet v1.0, no precisa de cap instal·lació prèvia i un cop copiada al directori escollit pot ser executada bé des de el gestor gràfic de fitxers clicant al fitxer TaFanet.jar i seleccionant obrir amb Java, o bé des de la consola de comandes.

Per sistemes Windows, introduint la instrucció **javaw -jar TaFanet.jar** o simplement **TaFanet.jar** , situant-se al directori escollit o afegint la ruta del directori.

Per l'execució en mode text, és necessari accedir per consola, des del directori escollit o afegint la ruta del directori, amb la instrucció **java -jar TaFanet.jar text** .

Per sistemes Linux, un cop situats amb la consola de comandes al directori que conté l'eina cal executar **sudo java -jar TaFanet.jar** . L'execució en mode usuari no superusuari nomes permet visualitzar fitxers guardats de captures anteriors. Per l'execució en mode text, cal afegir el paràmetre text i igualment accedir com a superusuari: **sudo java -jar TaFanet.jar text** .

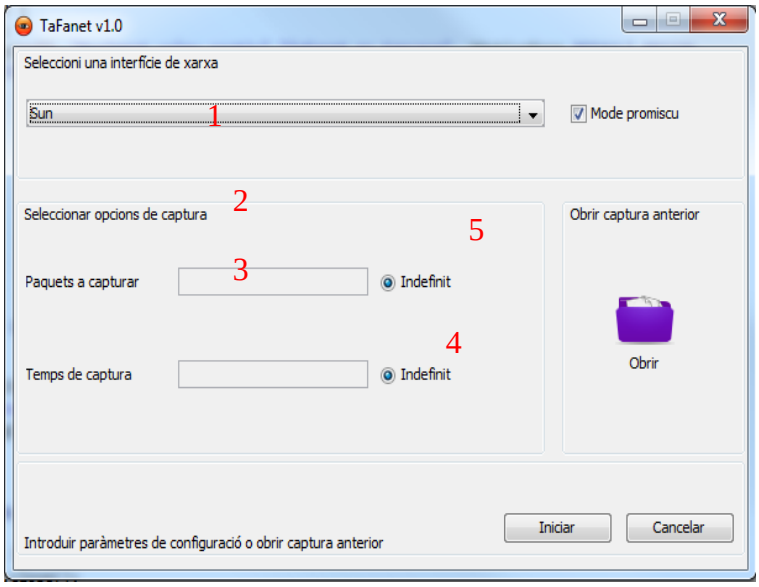
Important!!! Cal configurar les variables d'entorn Java per l'execució en mode consola. Es pot consultar l'adreça http://www.java.com/es/download/help/path.xml on es detalla el procediment per cada sistema operatiu.

Important!!! El fitxer "ajuda.pdf" ha de ser al mateix directori que l'aplicació per a ser accessible des de la icona d'ajuda, un cop executada l'eina.

5.2 Manual de d'usuari

En mode text només cal seguir les instruccions per pantalla.

En mode gràfic un cop executada l'eina és mostra la finestra de configuració.

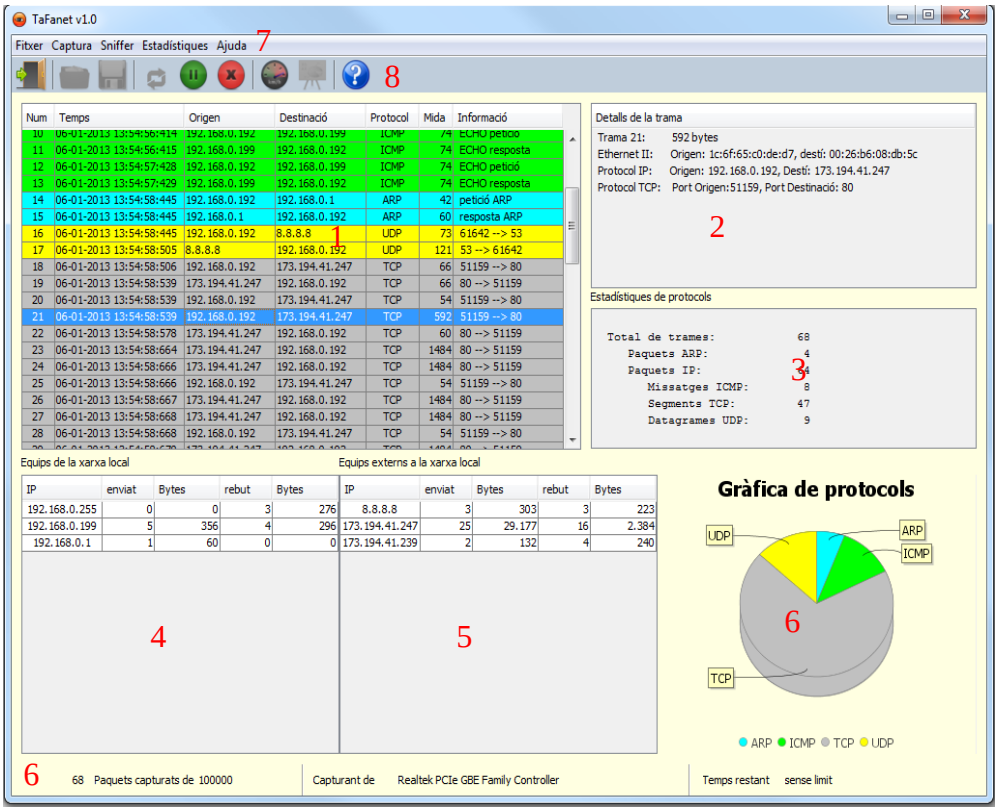


6. Cal seleccionar la interfície de xarxa en la que es produirà la captura de paquets de xarxa i si es desitja una captura en mode promiscu o no. Per defecte, l'opció és mode promiscu.
7. En aquest espai és pot definir el nombre de paquets a capturar. Ha de ser un valor compres entre 1 i 100.000. Per defecte es marca indefinit el que marca un límit de 100.000 paquets.
8. En aquest espai és pot definir un temps màxim de captura expressat en segons. Per defecte, es marca l'opció indefinit que no limita el temps de captura.
9. Un cop escollits els paràmetres de configuració de la captura clicant el botó 'Iniciar', es passa a la finestra principal de l'aplicació i es dona inici a la captura de paquets.
10. Si el que es desitja es visualitzar una captura guardada amb anterioritat, simplement cal premé el botó 'Obrir' que mostrarà un gestor de fitxers del que cal escollir el fitxer a visualitzar. Un cop escollit es passa a la finestra principal de l'aplicació.

Prement el botó 'Cancel·lar' es pot suspendre l'execució de l'aplicació.

Un cop configurada la captura o seleccionat un fitxer a obri es mostra la finestra principal de l'aplicació, que està subdividida en les següents zones:

1. **Taula *sniffer***. Mostra les trames capturades. Cada fila de la taula modifica el seu color depenent del protocol de xarxa detectat.
   - Protocol ARP - color cian.
   - Protocol ICMP - color verd.
   - Protocol TCP - color gris.
   - Protocol UDP - color groc.
2. **Detalls de la trama**. Mostra informació detallada de la trama seleccionada a la taula *sniffer*.
3. **Estadístiques de protocols**. Mostra una estadística en temps real dels protocols de xarxa detectats.
4. **Equips de la xarxa local**. Mostra una taula amb els equips detectats a la xarxa local, així com un còmput de trames i bytes enviats i rebuts.
5. **Equips externs a la xarxa local**. Mostra una taula amb els equips detectats externs a la xarxa local, així com un còmput de trames i bytes enviats i rebuts



6. **Gràfica de protocols**. Mostra una gràfica en porcions de percentatge dels protocols detectats. Al mateix espai de la gràfica de protocols es mostra un diagrama de barres amb informació sobre l'ample de banda ocupat.
7. **Barra inferior**. Queda dividida en tres parts: a l'esquerra mostra el nombre de capturats i el límit de captura, a l'espai central l'estat de la captura i a la dreta el temps restant de captura.
8. **Barra d'icones.** Al següent paràgraf es descriu la funcionalitat de cada icona.

	Sortir	Surt de l'aplicació.
	Obrir	Obre un fitxer guardat amb anterioritat. Desactivat amb captura activa.
	Guardar	Guarda en fitxer la captura. Desactivat amb captura activa.
	Iniciar	Inicia el procés de configuració d'una nova captura. Desactivat amb captura activa.
	Pausa	Pausa el procés de captura. Desactivat en visualització de fitxer.
	Aturar	Atura la captura. Desactivat en visualització de fitxer.
	Ample de Banda	Intercanvia l'espai amb la gràfica de protocols. Desactivat si es mostra el diagrama d'ample de banda.
	Gràfica	Intercanvia l'espai amb el diagrama d'ample de banda. Desactivat si es mostra la gràfica de protocols.
	Ajuda	Mostra el fitxer d'ajuda.

9. Barra de menús. La formen 5 menús: Fitxer, Captura, Sniffer, Estadístiques i Ajuda.

Fitxer	Obrir	Obre un fitxer guardat amb anterioritat. Desactivat amb captura activa.
	Guardar	Guarda en fitxer la captura. Desactivat amb captura activa.
	Sortir	Surt de l'aplicació.
Captura	Iniciar	Inicia el procés de configuració d'una nova captura. Desactivat amb captura activa.
	Pausa	Pausa el procés de captura. Desactivat en visualització de fitxer.
	Aturar	Atura la captura. Desactivat en visualització de fitxer.
Sniffer	Num	Amaga o mostra la columna corresponent de la taula sniffer
	Temps	Amaga o mostra la columna corresponent de la taula sniffer
	Origen	Amaga o mostra la columna corresponent de la taula sniffer
	Destinació	Amaga o mostra la columna corresponent de la taula sniffer
	Protocol	Amaga o mostra la columna corresponent de la taula sniffer
	Mida	Amaga o mostra la columna corresponent de la taula sniffer
	Informació	Amaga o mostra la columna corresponent de la taula sniffer
	Mostra tots	Mostra totes les columnes corresponents de la taula sniffer
Estadístiques	Ample de Banda	Intercanvia l'espai amb la gràfica de protocols. Desactivat si es mostra el diagrama d'ample de banda.
	Gràfica	Intercanvia l'espai amb el diagrama d'ample de banda. Desactivat si es mostra la gràfica de protocols.
Ajuda	Ajuda	Mostra el fitxer d'ajuda.
	Quant al TaFanet	Informació de l'aplicació.

6. Conclusions

La realització d'aquet Treball de Final de Carrera ha sigut una feina àrdua, tot i que satisfactòria tal i com es veien avançar les fases del projecte. Ha requerit de constància i de respectar al màxim els terminis establerts per la planificació que es va marcar en els seus inicis.

S'han aconseguit els requisits que s'havien plantejat, malgrat haver tingut sempre en ment el desig d'implementar funcionalitats que no havien estat previstes. El resultat es l'eina de rendiment de xarxa TaFanet v1.0, que de manera molt intuïtiva ofereix a l'usuari la possibilitat d'analitzar el tràfic present a la xarxa, tot oferint-li, a part de la visió de les trames capturades una sèrie d'estadístiques i gràfiques.

Ha calgut aplicar els coneixements obtinguts durant la carrera, així com ampliar-los en molts casos, principalment de les següents àrees:

- Xarxes de computadors.
- Enginyeria del programari.
- Programació orientada a objectes.
- Sistemes operatius.

La part del projecte que ha requerit d'un esforç complementari per a la seva comprensió i posterior posada en pràctica ha sigut la generació de les interfícies d'usuari i la seva integració amb la lògica de l'aplicació. Malgrat tot, el resultat final ha sigut prou satisfactori i engrescador de cara a iniciar nous projectes en un futur i desenvolupar noves funcionalitats a TaFanet, per que en un temps futur arribi la v2.0.

Són moltes les millores possibles, per començar incloure l'anàlisi de paquets IPv6 que han sigut descartats en aquesta versió de l'eina i s'han contemplat simplement com a trames Ethernet no classificades. Cal aprofundir en l'estudi de les capçaleres dels protocols de transport, especialment TCP, que permeti detectar intents fallits de connexions, o diversos tipus d'atacs. Queda pendent també, analitzar els protocols d'aplicació i seria molt interesant afegir la possibilitat d'aplicar filtres a les captures per concentrar la informació recollida.

Resumint, i a nivell personal, ha estat un gran repte el desenvolupament d'aquest projecte, però tot i les dificultats que han anat sorgint s'han resolt satisfactòriament. He trobat en la seva realització, el goig de comprovar com molts coneixements adquirits durant la carrera agafaven forma en la feina feta, omplint de significat l'esforç realitzat.

7. Glossari

TCP/IP Conjunt de protocols utilitzats per les comunicacions entre equips a la xarxa.

ARP (Address Resolution Protocol) Protocol encarregat de resoldre les adreces hardware (MAC) per la transmissió sobre Ethernet.

IP (Internet Protocol) Protocol no orientat a connexió per les comunicacions equip a equip a nivell de xarxa.

ICMP (Internet Control Message Protocol) Protocol utilitzat pel protocol IP per la transmissió de missatges per la notificació de missatges de control.

TCP (Transmission Control Protocol) Protocol de comunicacions a nivell de transport orientat a connexió.

UDP (User Datagram Protocol) Protocol de comunicacions a nivell de transport no orientat a connexió.

Trama Grup de bits que formen una unitat d'enviament de dades a nivell d'enllaç.

Paquet Unitat d'enviament de dades a nivell de xarxa utilitzat pels protocols IP i ARP.

Segment Unitat d'enviament de dades utilitzat pel protocol TCP.

Datagrama Unitat d'enviament de dades utilitzat pel protocol UDP.

Adreça IP Nombre binari que identifica un equip a la xarxa, format en el cas de IPv4 per 32 bits dividits en 4 octets

Adreça MAC (Media Access Control) Nombre de 48 bits que identifica a cada interfície de xarxa, determinada per la IEEE i el fabricant.

IANA (Internet Assigned Numbers Authority) Entitat supervisora de l'assignació d'adreces IP, noms de domini DNS i assignació de ports per les aplicacions.

RFC (Request for Comments) memoràndums de metodologies relacionades amb Internet.

Package Contenidor de classes que agrupa diferents parts d'una aplicació per la seva funcionalitat.

8. Bibliografia

Font Rosselló, M; Lara Ochoa, E; Serral i Gracià, R; Vilajosana Guillén, X (2011). *Estructura de xarxes de computadors*. Barcelona: Fundació per a la Universitat Oberta de Catalunya.

Campderrich Falgueras, B; Recerca informàtica, SL (2004). *Enginyeria del programari*. Barcelona: Fundació per a la Universitat Oberta de Catalunya.

Bel Puchol, A (2001). Apuntes Lenguaje Java. [en línia]. http://www.arrakis.es/~abelp/ApuntesJava/indice.htm [data de consulta: 10 de novembre de 2012].

8.1 Webs de consulta

RFC https://www.rfc-editor.org/index.html

IANA http://www.iana.org/

Jpcap http://netresearch.ics.uci.edu/kfujii/Jpcap/doc/index.html

9. Annexos

9.1 Annex A. Capçaleres del protocols de xarxa.

A continuació es presenten les capçaleres dels diferents protocols de xarxa que estudia l'eina de rendiment de xarxa.

Les imatges que són originals dels documents RFC inclouen al peu d'imatge el número identificatiu del document RFC corresponent.

64 bits	48 bits	48 bits	16 bits
Preamble	MAC Destination	MAC Source	Type

Il·lustració 26: Capçalera trama Ethernet II. IEEE 802.3

8 bits	8 bits	8 bits	8 bits
Hardware Type		Protocol Type	
HLEN	PLEN	OpCode	
Sender HA (octets 0-3)			
Sender HA (octets 4-5)		Sender IP (octets 0-1)	
Sender IP (octets 2-3)		Target HA (octets 0-1)	
Target HA (octets 2-5)			
Target IP (octets 0-3)			

Il·lustració 27: Paquet ARP

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Version|  IHL  |Type of Service|          Total Length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Identification        |Flags|      Fragment Offset    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Time to Live |    Protocol   |         Header Checksum       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Source Address                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Destination Address                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Options                    |    Padding    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Il·lustració 28: Capçalera IP. RFC 791

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |     Code      |          Checksum             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Identifier          |        Sequence Number        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Data ...
+-+-+-+-+-
```

Il·lustració 29: Capçalera ICMP. Tipus Echo. RFC 792

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Source Port          |       Destination Port        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Sequence Number                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Acknowledgment Number                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Data |           |U|A|P|R|S|F|                               |
| Offset| Reserved  |R|C|S|S|Y|I|            Window             |
|       |           |G|K|H|T|N|N|                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Checksum            |         Urgent Pointer        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Options                    |    Padding    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             data                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Il·lustració 30: Capçalera protocol TCP. RFC 793

```
 0      7 8     15 16    23 24    31
+--------+--------+--------+--------+
|     Source      |   Destination   |
|      Port       |      Port       |
+--------+--------+--------+--------+
|                 |                 |
|     Length      |    Checksum     |
+--------+--------+--------+--------+
|
|          data octets ...
+---------------- ...
```

Il·lustració 31: Capçalera protocol UDP. RFC 768