

TFC-Simulació i optimització de sistemes

Modelatge i simulació del sistema informàtic que  
dóna suport al Campus Digital – Balancejadors de  
càrrega

Memòria final

Jordi Paloma Albesa  
27/12/2012

Consultors:  
Juan Pérez  
Alberto García Villoria

## Resum

El campus virtual de la UOC ha de complir una sèrie de requeriments per poder donar els millors serveis als seus usuaris. Per a tal fi va aparèixer el projecte Castelldefels, a partir del qual es dissenyaria l'escenari que serviria per a simular el comportament dels equips que componen el sistema. Aquestes simulacions permetrien saber, de manera aproximada, quina seria la millor topologia de xarxa.

En aquest projecte s'intenta recopilar els avenços fins ara realitzats i continuar el treball realitzat per Arquimbau i Lahuerta pel que fa al comportament dels balancejadors dins del sistema Campus Virtual. Pel que fa al treball d'Arquimbau s'intentaran realitzar simulacions amb un miler de subxarxes amb un client i un router. I per la banda de Lahuerta, s'estudiarà el comportament del balancejador fent servir el seu model WAN.

Durant el projecte es veurà com una sèrie de limitacions en el software disposat no han permès obtenir l'objectiu proposat, però si s'han extret conclusions i s'han deixat plantejada noves línies de treball prou interessants.

## Índex

1.- Introducció .....	5
2.- Estat de l'art .....	6
3.- Objectius i metodologia .....	7
4.- Planificació ... ..	8
5.- OPNET .....	9
5.1 Introducció .....	9
5.2 Disseny d'esceanris amb OPNET.....	9
5.3 Aplicacions .....	10
5.4 Perfils .....	11
5.5 Simulació d'esdeveniments discrets .....	14
5.6 OPNET IT Guru .....	16
6 Balancejadors de càrrega .....	17
7 Modelatge i simulació .....	18
7.1 Model pas a pas .....	18
7.2 Model bàsic I .....	18
7.2.1 Escenari .....	18
7.2.2 Aplicacions .....	19
7.2.3 Perfils .....	19
7.2.4 Simulació .....	20
7.3 Model bàsic II .....	21
7.3.1 Escenari .....	21
7.3.2 Aplicacions .....	21
7.3.3 Perfils .....	22
7.3.4 Simulació .....	22

7.4 Model bàsic III .....	23
7.4.1 Escenari .....	23
7.4.2 Aplicacions .....	23
7.4.3 Perfils .....	24
7.4.4 Simulació .....	24
7.5 Model Balancejat I .....	25
7.5.1 Escenari .....	25
7.5.2 Simulació .....	25
7.6 Model Balancejat II .....	27
7.6.1 Escenari .....	27
7.6.2 Simulació .....	27
7.8 Model Balancejat III.....	29
7.8.1 Escenari.....	29
7.8.2 Simulació.....	29
7.9 Model final.....	31
7.9.1 Modelatge dels nodes .....	31
7.9.2 Connexió nodes i simulació.....	32
8 Estudi del balancejador amb el model WAN .....	33
9 Proposició del model WAN – Workstation .....	36
9.1 Problema d'interconnexió al model Workstation.....	36
9.2 Solució a l'interconnexió del model Workstation.....	37
10 Conclusions i línies de recerca.....	39
11 Bibliografia .....	40

## 1.- Introducció

El treball que aquí s'està punt d'iniciar es basarà en projectes realitzats per altres companys en semestres anteriors. Sobretot es tindran molt presents la feina feta per en Freire, Lahuerta i Arquimbau. Aquest darrer serà la base del projecte, ja que continuarem la seva investigació sobre el comportament del balancejador dins del sistema pensat per al "Projecte Castelldefels".

El projecte neix amb el fi de modelar i simular l'entorn de treball en el que es troba a diari el Campus Digital de la UOC. Estudiarem com afecten els milers de visites diàries que reben els servidors de correu, ftp i http i veurem que podem modificar amb la intenció de millorar i fer més eficients els sistemes i dispositius que hi intervenen.

Per això haurem de recrear de manera teòrica l'escenari en el que ens trobem mitjançant el software OPNET. Aquesta eina permetrà modelar el Campus digital i, un cop modelat, es podran fer proves per tal de simular el comportament dels equips implicats en el funcionament del Campus Virtual.

Per a recrear l'escenari s'hauran de tenir en compte molts factors. Factors temporals com els dies d'entrega de PACs o les hores del dia en el que es produeixen pics de visites. Factors com els perfils d'usuaris que hi accedeixen, no serà el mateix el personal docent que necessitarà més recursos dedicats a l'accés FTP per a la descàrrega massiva de PACs o un alumne que requerirà més recursos a la navegació web. I d'altres factors que s'aniran comentant al llarg del projecte.

La finalitat de la simulació és agafar tots aquests factors i crear un model que ens garanteixi fiabilitat, seguretat, estabilitat i escalabilitat. Això últim és molt important ja que s'ha de comprendre que el campus virtual està en continua expansió i requereix de nous equips de manera gradual. S'ha de dir que el modelat no és gens senzill i la simulació no assegura un comportament idèntic al que es tindria en la realitat, però si dona una idea dels problemes que es poden trobar i com solucionar-los.

Com ja s'ha dit el principi, la investigació d'aquest treball es centrarà en un dels dispositius principals d'aquesta arquitectura. Es tracta del balancejador de càrrega, el qual s'encarrega de distribuir possible el tràfic als diferents servidors de serveis de la xarxa. L'objectiu serà trobar les polítiques i algoritmes que millor s'adaptin a les nostres necessitats, modelar-les i simular-les.

## 2.- Estat de l'art

Abans de presentar l'objectiu i començar a treballar-hi cal fer una mica d'història per veure en quin estat es troba el projecte i quins han estat els avenços fets fins ara. Per a tal propòsit s'ha de citar els precursors i, per així dir-ho, instigadors a realitzar el projecte que ens ocupa.

Les dues persones que van iniciar l'estudi dels balancejadors de càrrega van ser en Miguel Ángel Arquimbau i en Hugo Freire. Destacarem la feina feta per el primer d'ells, ja que en Freire es va trobar una sèrie de problemes que no el van deixar continuar el treball. Aquests problemes tenien a veure amb les llicències del programa OPNET i els resultats obtinguts, que segons comenta al seu treball no tenien lògica.

Per tant, em basaré principalment en el treball realitzat per Arquimbau. Els resultats que va extreure de les seves simulacions eren molt interessant i deixaven obertes moltes incògnites pendents de resoldre. Una d'elles deia que la manera més fidel de recrear l'escenari dels usuaris que fan servir el campus virtual és la de modelar un nombre elevat de subxarxes amb una única estació de treball connectada a un router. D'aquesta manera tindríem un escenari bastant exacte al que ens trobem dia a dia a la UOC i a més estaríem incloent dispositius 3G.

En treballs posteriors dels companys Lahuerta i Freire van demostrar que es podia simular l'escenari amb total fidelitat fent servir nodes LAN. El que va descobrir Lahuerta és que Arquimbau havia errat en la configuració dels nodes LAN i que hi ha un paràmetre que permet indicar quin és el servidor que administra el tràfic de la xarxa, un cop indicat quin es el servidor de la xarxa el node LAN es comporta de manera similar a tenir varies màquines penjant de routers de manera individual. Però afecta això el comportament dels balancejadors? Mai va provar-ho.

Lahuerta va anar més enllà i va trobar errors en la configuració de la part WAN que feien servir Arquimbau i els seus predecessors, una configuració per defecte proporcionava per OPNET que era errònea i generava un coll d'ampolla que desvirtuava els resultats obtinguts. En Lahuerta en va configurar una de nova que salvava aquests problemes.

És evident que l'escalabilitat del projecte és molt més senzilla si fem servir nodes LAN en comptes d'una infinitat d'estacions de treball. Però em desmarcaré i intentaré reproduir el que deia Arquimbau en el seu projecte, centrant-me en el funcionament del balancejador intentant aprofitar els descobriments fets per en Lahuerta que milloren el model WAN dissenyat per Arquimbau.

### 3.- Objectius i metodologia

L'objectiu principal és l'estudi del comportament dels balancejadors de càrrega dins de l'arquitectura UOC. Per a arribar a modelar, estudiar i entendre el funcionament d'aquests equips haurem de passar diverses etapes que es complementen:

- **Instal·lació, configuració i estudi de OPNET.**

Opnet serà l'eina que es farà servir per al disseny i simulació de la xarxa UOC. El primer pas serà instal·lar, configurar i aprendre com funciona aquest software per a poder assolir els objectius del projecte. Centrant-nos, sobretot, en el funcionament dels balancejadors de càrrega.

Aquesta serà una etapa d'aprenentatge que s'allargarà durant tot el projecte, ja que possiblement serà necessari aprendre de manera progressiva a mida que es vagin fent simulacions.

- **Disseny del model a simular.**

Un cop es tinguin els coneixements necessaris, es passarà a la implementació de l'escenari a simular.

Durant aquesta etapa s'hauran d'estudiar els escenaris desenvolupats per Arquimbau al seu treball, ja que partirem de la feina feta en la seva investigació. Un cop estiguin assimilats passarem a fer les modificacions pertinents per a obtenir nous resultats més aproximats a la realitat del "Campus virtual".

- **Simulacions del nou disseny.**

Tan bon punt s'hagin fet les modificacions en l'escenari, es podrà passar a la simulació i obtenció de resultats.

Durant aquesta etapa ens endinsarem de ple en el funcionament dels balancejadors de càrrega dins de l'entorn virtual de la UOC i començarem a obtenir els resultats que ens aproximaran al nostre objectiu final. Serà un procés en els que es trobaran problemes de configuració i s'hauran d'anar solventant a mida que vagin sorgint.

- **Anàlisi de resultats.**

Aquesta serà la part final del projecte, es tractarà d'analitzar les proves realitzades i de treure'n conclusions. També s'haurà de proposar noves línies d'investigació per a futurs projectes.

## 4.- Planificació

Per la planificació s'ha utilitzat un software lliure anomenat Gantt Project i que permet realitzar diagrames Gantt, aquesta és la proposta:

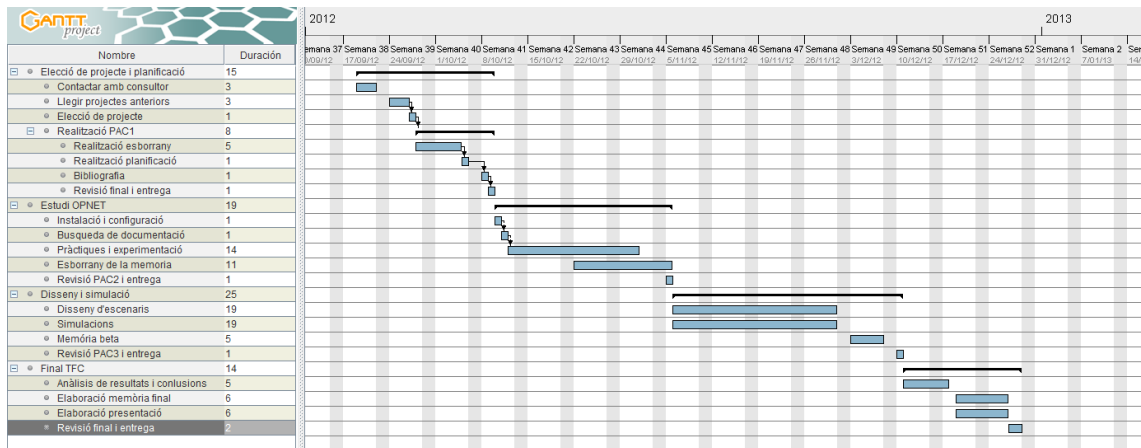


fig. 1 Planificació projecte

S'ha agafat com a base els dies d'entrega de les PAC i a partir d'aquestes dates s'ha dissenyat la planificació de la feina a realitzar.

S'inicia el projecte amb la elecció del tema a investigar i la realització de la primera PAC, la qual consisteix en una introducció al treball a realitzar. Cada un dels passos d'aquesta part de planificació depèn de l'acabament d'un per al començament del següent.

El segon pas consta de l'aprenentatge del software a utilitzar per a la elaboració del treball. Segurament aquesta tasca s'allargui fins al final del projecte, però hi farem més émfasi durant aquesta etapa de treball. Es veu com les pràctiques i la realització de l'esborrany es sobreposen en el temps, això és degut a que ho haurem de realitzar conjuntament per a que ambdues tasques resultin productives.

La tercera etapa consistirà en el disseny i simulació dels escenaris a treballar. És la part més important del projecte, ja que d'aquí sortiran els resultats que ens portaran a l'objectiu final del treball. Es veu com el disseny i la simulació es superposen en el temps, això és degut a que les simulacions ens portaran a canvis en el disseny, per tant serà quelcom que farem a la vegada.

La última part consta de la elaboració de la memòria final i la presentació en PowerPoint. S'han superposat la elaboració de la memòria final i el PowerPoint ja que d'aquesta manera resulta molt més eficient en el temps.



## 5.- OPNET

### 5.1.- Introducció

L'objectiu que perseguim en aquest treball és entendre quin serà el comportament dels balancejadors dins dels diversos escenaris que ens podem trobar. Evidentment, portar a terme aquestes proves en laboratoris equipats amb tot de dispositius i sistemes seria molt costós i complicat. Per tant, és necessari buscar la manera de fer les proves recreant un entorn similar al que podríem tenir en un laboratori.

D'uns anys ençà, han anat apareixent softwares que permeten replicar aquests entorns a nivell teòric i realitzar totes les proves que faríem en un laboratori físic. Amb els avenços que hi han hagut en el món informàtic i aprofitant que les computadores són cada cop més potents, aquests softwares s'han anat complicant i són capaços de recrear la realitat de manera fidedigna.

En el cas que ens ocupa, farem servir el software OPNET, un software que permet dissenyar escenaris telemàtics complexos i simular quin seria el seu comportament en cas de portar-los a la pràctica.

### 5.2.- Disseny d'escenaris amb OPNET

El primer pas que s'ha de fer abans de realitzar les proves és el de recrear l'escenari. Per a tal fi OPNET ens abasteix d'una ampla llista de dispositius, cal remarcar que molts dels equips de les primeres marques mundials estan recreats amb precisió i en podem fer ús a nivell teòric.

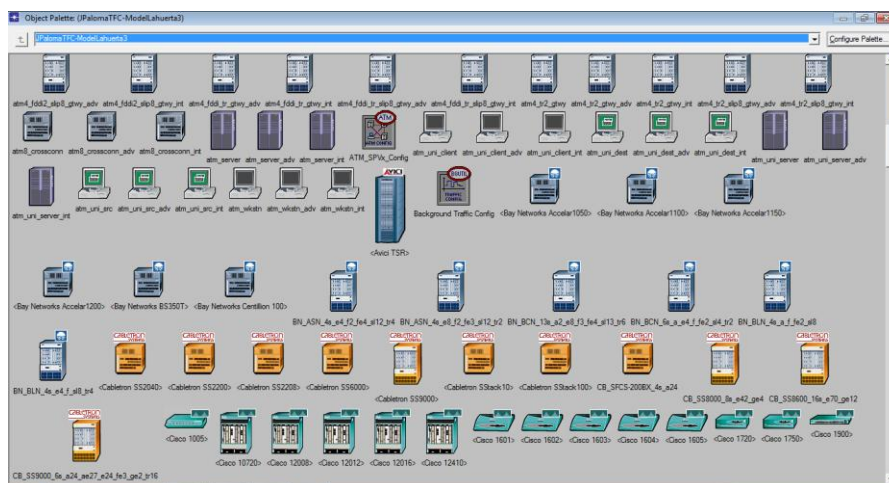


fig. 2 Paleta d'objectes OPNET

En la imatge superior es pot veure una petita mostra dels dispositius dels que disposem per a realitzar els escenaris que volem posar a prova. Aquests equips els

podem disposar en plans pre-dissenyats a la pròpia aplicació i que recreen el mapa mundial.

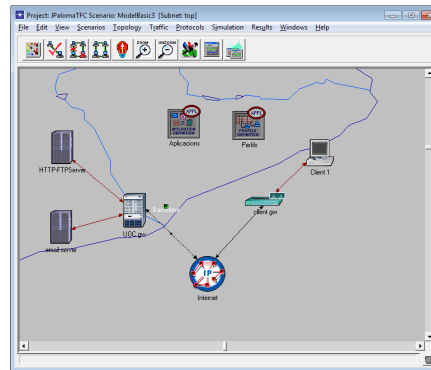


fig. 3 Exemple d'escenari

En la imatge superior es mostra un disseny senzill per fer-nos una idea de com queda la modelització d'un escenari. La manera de dissenyar aquests escenaris és molt senzill, ja que l'únic que em de fer és escollir l'objecte que necessitem i senyalar dins del mapa on el volem ubicar.

Cada un d'aquests equips tenen els seus propis atributs de funcionament, a la següent imatge es mostra una finestra de configuració d'un router.

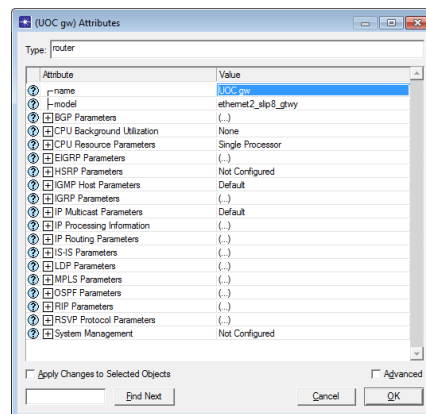


fig. 4 Edició d'atributs

La manera d'indicar el comportament de cada un dels equips a OPNET és creant perfils de comportament i ús d'aplicacions.

### 5.3.- Aplicacions

OPNET té una sèrie d'aplicacions per defecte que es poden activar o desactivar amb facilitat. Així com també es podran configurar amb uns certs valors per defecte que el programa facilita (també poden ser configurades manualment, però les opcions preconfigurades són més que suficients per al cas que ens ocupa). Aquestes poden ser:

- **Custom:** seran aplicacions customitzables per l'usuari.

- **Database:** es tracta de tràfic generat a partir de consultes a BBDD.
- **Email:** tràfic generat per l'enviament/recepció de correu electrònic.
- **FTP:** paquets de dades generats a partir d'una descàrrega de fitxers.
- **HTTP:** tràfic generat a partir de clients de navegació web.
- **Print:** informació generada per l'ús d'una impressora.
- **Remote login:** tràfic generat per l'autenticació d'usuaris.
- **Video Conferencing:** paquets de dades generats a partir d'una videoconferència.
- **Voice:** tràfic de VoIP.

Per a totes aquestes aplicacions podem indicar quin és el nivell de tràfic que es generarà per a cada un dels protocols. Aquestes aplicacions es podran assignar després als perfils de comportament.

## 5.4.- Perfils

Com ja s'ha comentat anteriorment, els perfils ens permetran crear patrons de comportament de cada un dels equips representats en el nostre disseny. Les opcions que tenim a l'hora de poder crear perfils són molt àmplies, tant que difícilment no trobem la manera de representar el comportament que perseguim de la manera més fidel i aproximada de la realitat possible.

En treballs anteriors al meu, s'han aconseguit recrear els comportaments dels usuaris del Campus Virtual, però no hi aprofundiré perquè no em faran falta per arribar a l'objectiu buscat en aquest treball. Quedarà pendent per treballs pròxims barrejar els perfils modelats amb l'escenari plantejat en aquest treball.

Per explicar com funcionen els perfils citaré Arquimbau, no crec que trobi una manera millor de resumir el funcionament dels perfils:

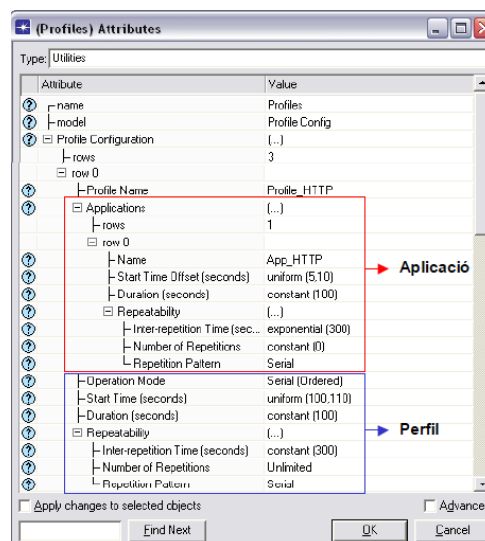


fig. 5 Edició de perfils

En la figura superior es mostra el panell de configuració dels perfils. Podem definir tants perfils com vulguem indicant-ho al paràmetre rows que hi ha just després de Profile Configuration (en aquest exemple, el 3 significa que definim tres perfils).

Si despleguem row0 ens surten els paràmetres de configuració del primer dels perfils.

- **Profile Name:** Nom del perfil. Serà el nom que identificarà aquest perfil.

A partir d'aquí els paràmetres queden agrupats en dos conjunts: els que fan referència al perfil en conjunt, i els que fan referència al comportament de cada una de les aplicacions que componen el perfil.

Paràmetres relatius al perfil en conjunt:

- **Operation Mode:** Determina el mode en que inicien les diferents aplicacions pertanyents al perfil, i pot prendre aquests tres valors:
  - **Serial (ordered).** Les aplicacions comencen ordenadament des de la primera (row0) fins la última. Una aplicació no comença fins que no ha acabat l'anterior.
  - **Serial (random):** L'ordre de començament de les aplicacions és aleatori, però sempre esperant a finalitzar una per començar la següent.
  - **Simultaneou.** Totes les aplicacions poden començar al mateix temps.
- **Start Time (seconds):** El temps que triga en començar el perfil una vegada començada la simulació.
- **Duration (seconds):** Temps de duració del perfil. Pot assignar-se una distribució estadística sobre un valor, però també pot rebre els valors End of Simulation (el perfil dura fins el final de la simulació), o End of Last Application (el perfil acaba quan l'última aplicació ho fa).
- **Repeatability:** Determina si el perfil es repeteix al llarg de la simulació. Consta de tres paràmetres per configurar el patró de repetició:
  - **Inter-repetition Time (seconds):** Temps que triga en començar el següent perfil.
  - **Number of repetitions.** Nombre de vegades que es repeteix el perfil. Pot ser unlimited, none o una distribució estadística sobre un valor.
  - **Repetition pattern.** Determina quan comença la següent sessió del perfil. Pot ser Serial o Concurrent. Si és Serial, la següent sessió del perfil comença després de Inter-repetition Time segons d'haver acabat la sessió anterior del perfil. Si és Concurrent, cada Inter-repetition Time segons comença una nova sessió del perfil.

Paràmetres relatius a cada una de les aplicacions que componen el perfil:

- **Name.** Nom de l'aplicació. Cal haver-lo definit abans al mòdul d'aplicacions.
- **Start Time Offset (seconds).** Aquest valor té dues interpretacions en funció del valor del paràmetre Operation Mode del perfil. Si Operation Mode

es Simultaneous, indica el temps que triguen en començar totes les aplicacions des de que comença el perfil. Si Operation Mode es Serial, llavors indica el temps que triga en començar la primera aplicació, i el Inter-repetition Time entre el final d'una aplicació i el començament de la següent.

- **Duration (seconds).** Es el temps de duració de l'aplicació. Pot assignar-se una distribució estadística sobre un valor, però també pot rebre els valors End of Profile (l'aplicació dura fins el final del perfil), o End of Last Task (l'aplicació dura fins que acaba l'última tasca).
- **Repeatability.** Determina si l'aplicació es repeteix al llarg del perfil. Consta de tres paràmetres per configurar el patró de repetició:
  - **Inter-repetition Time (seconds).** Temps que triga en començar la següent sessió de l'aplicació.
  - **Number of repetitions.** Nombre de vegades que es repeteix l'aplicació. Pot ser unlimited o una distribució estadística sobre un valor.
  - **Repetition pattern.** Determina quan comença la següent sessió de l'aplicació. Pot ser Serial o Concurrent. Si és Serial, la següent sessió de la aplicació comença després de Inter-repetition Time segons d'haver acabat la sessió anterior de la aplicació. Si és Concurrent, cada Interrepetition Time segons comença una nova sessió de la aplicació.

El gràfic següent mostra un exemple d'una simulació amb perfils en Operation Mode establert a Serial, i aplicacions i perfils amb Repetition pattern també Serial:

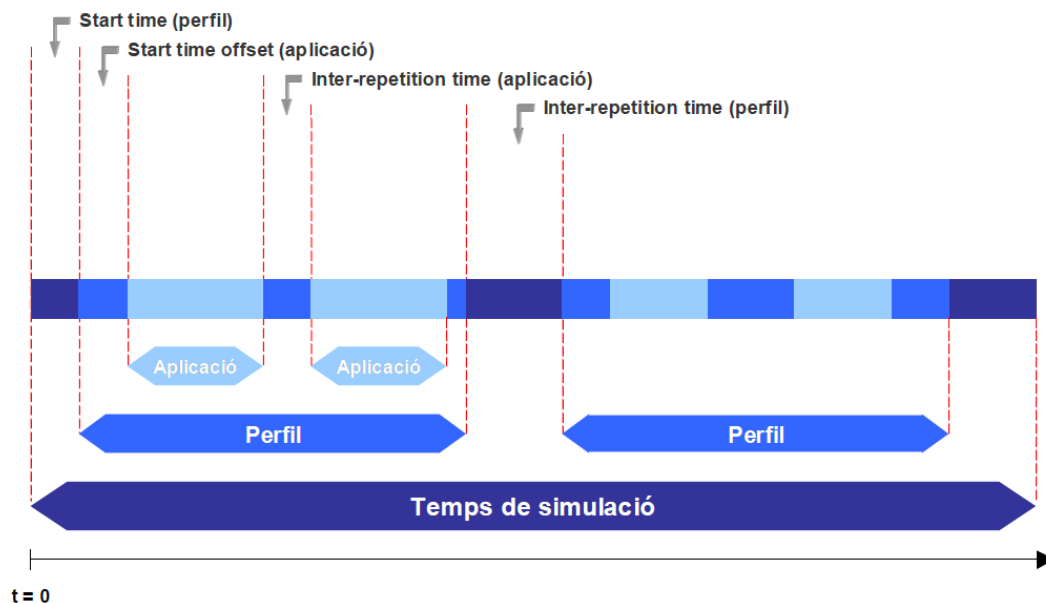


fig. 6 Evolució temporal de perfils i aplicacions

## 5.5.- Simulació d'esdeveniments discrets

Per entendre el concepte “Simulació d'esdeveniments discrets” hem de fer primer una distinció entre el que són sistemes estocàstics i deterministes:

- **Sistema estocàstic:** És aquell en el que el seu comportament depèn de comportaments aleatoris.
  - **Estàtic:** Prenem el resultat en un instant en concret.
  - **Dinàmic:** Prenem el resultat al llarg d'un període de temps.
    - **Continua:** Si prenem els resultats de manera continuada.
    - **Discreta:** Si agafem resultats puntuals a una certa freqüència i amb la unió dels resultat obtenim una predicció molt aproximada del resultat al llarg del temps.
- **Sistema determinista:** No depèn de variables aleatòries. En aquest tipus de sistemes sabem quin serà el resultat futur estudiant únicament l'estat actual del sistema.

Doncs be, un cop tenim clars els conceptes anteriors podem entendre quin tipus de simulació realitzarà OPNET dins del nostre sistema i perquè.

Recapitem, el que necessitem simular en aquest projecte són les connexions que rebrà el balancejador al llarg del temps i com seran tractades. Aquestes connexions ens arribaran de manera aleatòria, és a dir, mai podrem predir quina quantitat d'usuaris i en quin moment estaran enviant peticions de connexió. Podem dir llavors que es tracta d'un sistema estocàstic ja que dependrem de variables aleatòries.

Segon ens interessarà saber com es comporta el balancejador i, en general, tot el nostre sistema al llarg del temps. Per tant parlem de que les nostres proves estaran basades en un sistema estocàstic dinàmic.

I tercer, la millor manera de saber com es comporta el sistema és agafar resultats particulars al llarg del temps per tractar-los i obtenir un resultat lineal del comportament del sistema. Per a que sigui més aclaridor, el que farà OPNET és agafar fotografies cada cert temps de l'estat del sistema per després posar-les una darrera de l'altre i obtenir un resultat continu al llarg del temps. Tal i com passa en les pel·lícules, que el que veiem realment són imatges una darrera de l'altre donant sensació de moviment i linealitat.

Per tant podem concloure que els sistemes que simularem seran **Sistemes dinàmics discrets**.

Arquimbau explicà el procés de simulació de la següent manera:

- **Formulació** del problema a resoldre, determinació dels objectius globals i els recursos que cal assignar per assolir-los en el temps previst.
- **Modelatge** del sistema, creant un model lògic-matemàtic que representi convenientment l'estructura interna del sistema i les seves característiques, de manera que els resultats derivats de l'experimentació amb el model siguin enterament extrapolables al propi sistema real.
- **Implementació** del model al sistema informàtic, be codificant-lo per mitjà d'algun llenguatge de programació, be mitjançant l'ús d'algun programa de simulació específic.
- **Verificació** del programa, consistent en la comprovació no sols de la inexistència de errors sintàctics o lògics en el codi, sinó també la coherència dels resultats del programa per les diferents combinacions d'*inputs* possibles.
- **Validació** del model, després de comprovar que el model lògic-matemàtic representa convenientment el model real, comparant els resultats del model amb les dades observables del sistema real.
- **Disseny de la simulació** i proves pilot. Una vegada validat el model, cal decidir els paràmetres dels experiments de simulació (iteracions, *inputs*...). En aquesta fase pot ser de gran utilitat la realització de proves pilot (simulacions curtes).
- **Simulació**. Execució del programa amb els paràmetres establits abans. Normalment aquesta fase consistirà en la realització de diverses execucions del programa amb diferents paràmetres (escenaris) per tal de comparar com afecta la modificació d'algun/s dels paràmetres al funcionament del model.
- **Anàlisi dels resultats** derivats de la simulació. Sol ser necessari un anàlisi estadístic no trivial per obtenir conclusions sòlides sobre el comportament del sistema analitzat.
- **Documentació** de l'experiment per tal de fomentar la credibilitat i validesa de les conclusions assolides.

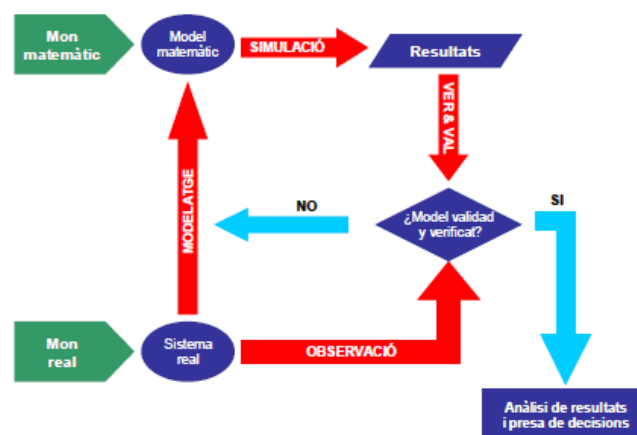


fig. 7 Procés de simulació

## 5.6.- OPNET IT Guru



*fig. 8 OPNET IT Guru*

Per a realitzar el projecte faré servir la versió acadèmica del OPNET, que s'obté de manera gratuïta. El més adient seria fer servir OPNET Modeler, que és la versió completa i no té cap limitació, però per motius de llicències no la puc fer servir.



## 6.- Balancejadors de càrrega

Aquest equip pot ser un equip hardware:



fig. 9 Balancejador hardware

O be pot ser un PC amb un software amb capacitat per tractar el tràfic que li arriba. El més utilitzat és iptables, un software open source molt potent.

<http://www.iptables.org/>

En base, un balancejador és un dispositiu que poden treballar entre les capes 3 i 7 de la torre OSI i es situa entre dos o més equips, normalment servidors, i la xarxa externa que hi accedeix. El que farà aquest equip serà tractar els paquets de dades que li arriben i repartir-los per els servidors segons una política preestablerta.

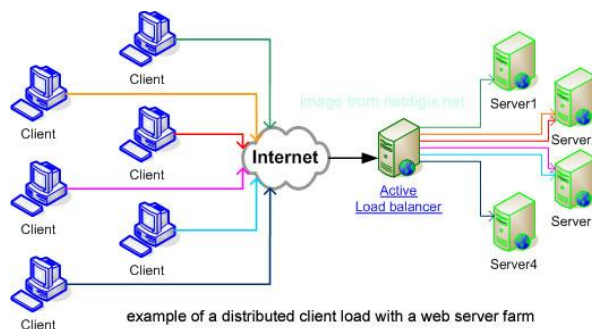


fig. 10 Exemple topologia amb balancejador

Les polítiques més comunes i que tenim implementades a OPNET són:

- **Aleatoria:** El balancejador reparteix el tràfic de manera aleatòria entre els servidors destí, sense tenir cap factor en compte. Amb aquesta política podem assignar prioritats mitjançant ponderacions, és a dir, establir una probabilitat més alta o baixa de rebre tràfic a un servidor o un altre.
- **Seqüencial:** El balancejador reparteix el tràfic seqüencialment, és a dir, un servidor radera de l'altra seguint un ordre. Com en la política anterior, podem definir ponderacions.
- **Menys càrrega:** El balancejador enviarà el tràfic al servidor que té menys càrrega de treball.
- **Menys connexions:** El balancejador enviarà el tràfic al servidor que hagi tingut menys connexions fins el moment. Per a tal cosa, portarà un registre de les connexions rebudes per cada servidor.

## 7.- Modelatge i simulació

### 7.1.- Model pas a pas

El més adequat seria iniciar les proves i models partint dels models dissenyats per Arquimbau. Però degut a que aquest va fer servir una versió de OPNET de la que no dispo, hauré de modelar un nou escenari desde zero.

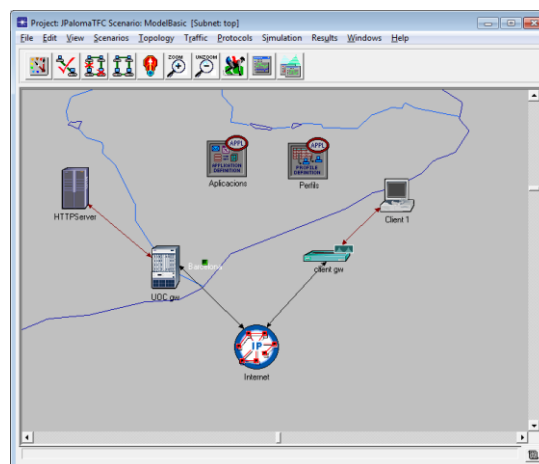
Amb el fi de deixar constància del procediment que vaig seguint per modelar l'escenari final sobre el qual realitzaré les simulacions, aniré creant petits escenaris que serviran per anar avançant i comprovant que tots els passos realitzats són correctes i funcionen satisfactòriament.

Començaré modelant escenaris molt bàsics amb client servidor i mica en mica els aniré completant fins arribar al model complex que ha de servir per extreure simulacions i conclusions.

### 7.2.- Model bàsic I

#### 7.2.1.- Escenari

S'ha de tenir present que l'objectiu final del treball és arribar a analitzar els balancejadors de càrrega dins del campus virtual de la UOC. Per a arribar a aquest objectiu haurem de fer arribar tràfic http, ftp i d'email d'una sèrie de màquines client als servidors del Campus. Per tant el primer que he fet és crear un petit escenari en el que tindrem un client i un servidor que rebran i enviaran tràfic http, per ara només s'enviaran dades a través d'una sola aplicació per a entendre el que fem.



*fig. 11 Escenari model bàsic I*

Per a modelar els escenaris, OPNET aporta una sèrie d'elements de xarxa més que suficients per a poder dissenyar qualsevol tipus de topologia. Tots aquests elements són configurables.

El que es representa en aquesta imatge és el model més senzill client servidor que és pot dissenyar. Hi ha una màquina desde la qual es generarà tràfic http. Aquest tràfic generat passarà a través d'un router cap a internet i arribarà a la màquina servidor de serveis web (per exemple apache) a través d'un altre router. Per ara no donaré gaire importància al tipus de connexió que faig servir, ja me'n preocuparé més endavant.

### 7.2.2.- Aplicacions

El següent pas es indicar quins són les aplicacions y protocols que es faran servir en l'escenari.

Com s'ha indicat anteriorment, en aquest cas el que es farà es crear un escenari el més senzill possible amb el fi d'entendre quin és el procediment a seguir per a modelar i simular un escenari de xarxa amb OPNET. En aquest cas només caldrà activar l'aplicació http, ja que l'únic que farem en el model bàsic és fer arribar peticions web desde un suposat navegador a un servidor apache.

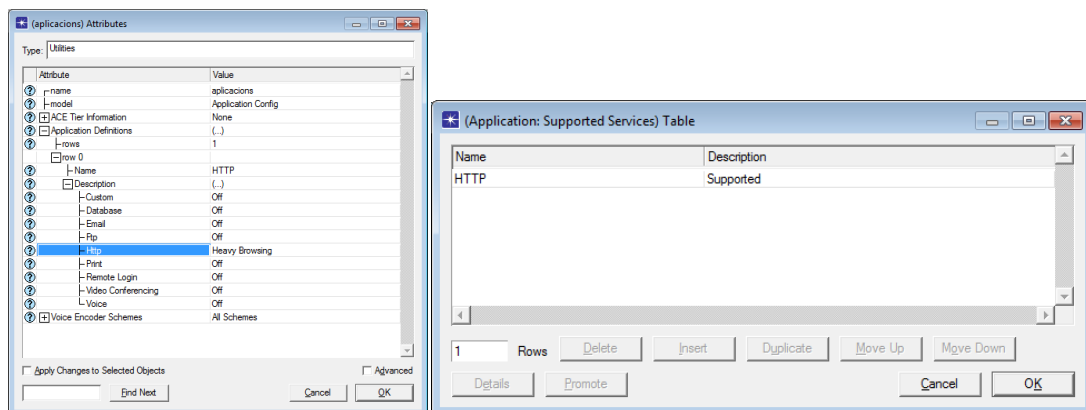


fig. 12 Aplicacions model bàsic 1

Es configura l'aplicació http en mode "Heavy browsing" que indica que hi haurà un nivell alt de peticions web entre el client i el servidor i l'anomenem HTTP mitjançant el camp "Name" i li assigno al servidor.

### 7.2.3.- Perfils

Els perfils de OPNET ens permeten definir els comportaments de les màquines del nostre disseny. Aquests faran referència a les aplicacions abans definides i, mitjançant una sèrie de paràmetres, podrem configurar quines aplicacions fan servir cada una de les màquines i de quina manera.

Seguint amb l'exemple anterior, passem a definir el perfil per a una màquina que estarà navegant per una web allotjada al servidor abans comentat.

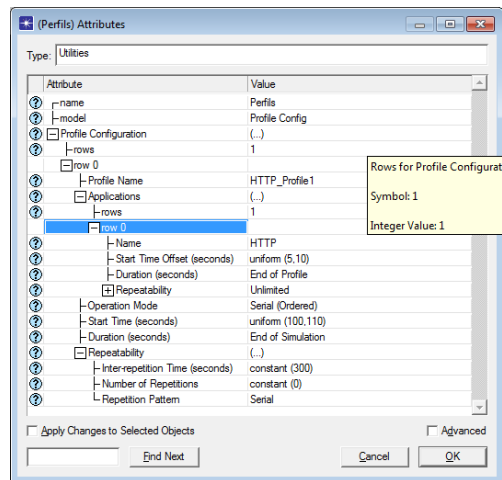


fig. 13 Perfil model bàsic I

Ara per ara no donaré importància a com estiguin definits els perfils. L'únic que busco és que hi hagi comunicació client-servidor, ja que aquesta es la meva pressa de contacte amb el programa OPNET i estic agafant pràctica mica en mica. L'anomeno HTTP\_Profile1.

#### 7.2.4.- Simulació

Per a comprovar que l'escenari està ben configurat procediré a fer una simulació i analitzaré els resultats per veure si realment hi ha connectivitat client-servidor.

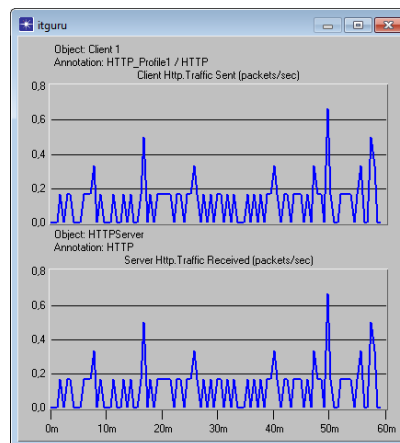


fig. 14 Simulació model bàsic I

La simulació dura poc més d'un segon, això ja és normal per els pocs clients que hi han configurats. Les dues gràfiques anteriors indiquen la quantitat de tràfic generat per la màquina client (gràfica superior) i el tràfic rebut per el servidor (gràfica inferior). Veiem que ambdues gràfiques són iguals i que per tant el servidor està rebent tot el tràfic enviat per el client.

## 7.3.-Model bàsic II

### 7.3.1.- Escenari

El que faig al segon escenari és afegir un nou servidor. Es tracta d'un servidor de correu destinat a donar servei SMTP als clients. El que intento amb aquest escenari és comprovar que a un mateix client puc crear-li un perfil en el que generi tràfic tant SMTP com HTTP.

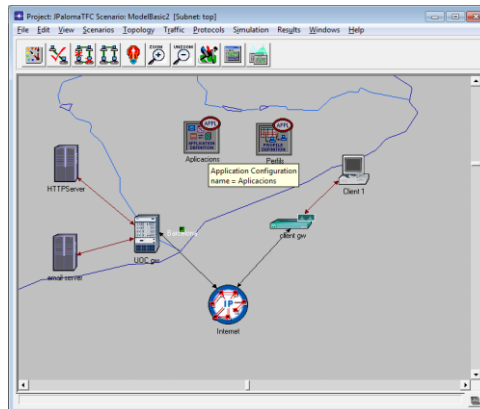


fig. 15 Escenari model bàsic II

Com es comprova a l'escenari de dalt, el nou servidor penjarà del mateix gateway que els servidor HTTP i la resta de topologia continua igual.

### 7.3.2.- Aplicacions

Afegeixo una segona aplicació de Email i la aplico al servidor de correu.

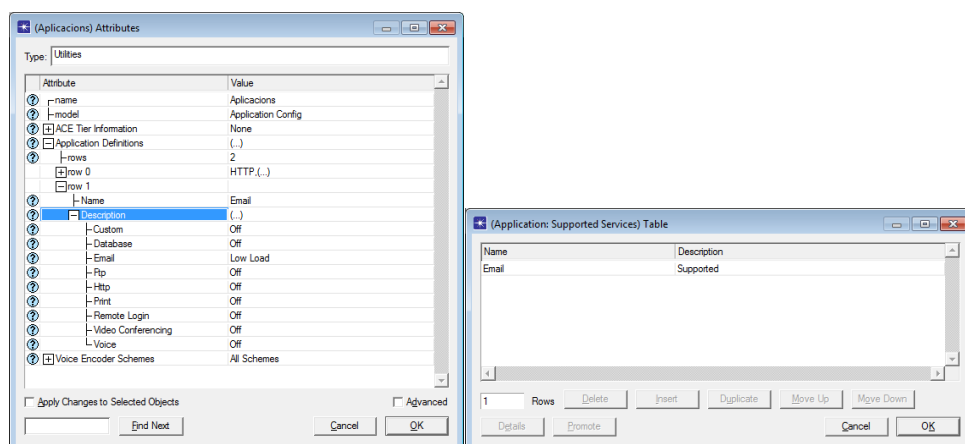


fig. 16 Aplicacions model bàsic II

Es configura l'aplicació Email en mode "Low load" que indica que hi haurà un nivell mig de peticions email entre el client i el servidor i l'anomenem Email mitjançant el camp "Name", seguidament li assigno al servidor. Es pot observar que els serveis els estic creant en "rows" diferents, ho faig d'aquesta manera per poder assignar els serveis de manera individual.

### 7.3.3.- Perfils

En aquest punt li haig de dir al client que generi tràfic HTTP i SMTP.

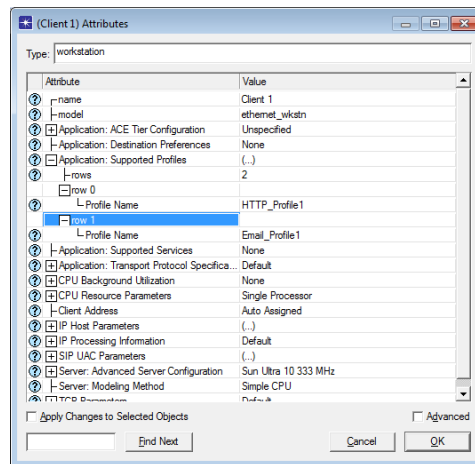


fig. 17 Perfil model bàsic II

### 7.3.4.- Simulació

L'objectiu que buscava amb aquest escenari era aconseguir que un terminal client generi tràfic Email i HTTP contra els servidors del Campus Virtual. Faig la simulació i comprovo que així és.

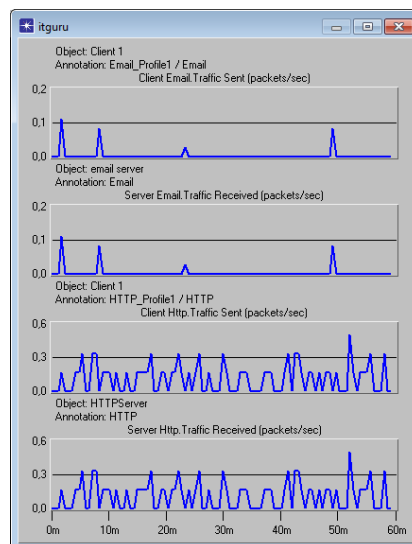


fig. 18 Simulació model bàsic II

Les dues primeres gràfiques indiquen l'intercanvi de tràfic SMTP entre el servidor i el client. Es pot comprovar com les gràfiques són iguals i que, per tant, el tràfic generat per el client és absorbit per el servidor. Les dues darreres gràfiques indiquen el tràfic de dades generat i es comprova de nou com les gràfiques són iguals, confirmant així que el sistema està funcionant correctament.

## 7.4.- Model bàsic III

### 7.4.1.- Escenari

En aquest nou escenari afegiré un nou servei, en aquest cas el servei FTP. Aquest protocol és el que es fa servir per a la transferència d'arxius. La diferència en aquest cas és que aquest servei estarà suportat per el mateix servidor que suporta el tràfic HTTP, fent d'aquesta manera que la màquina hagi de suportar una major càrrega de treball.

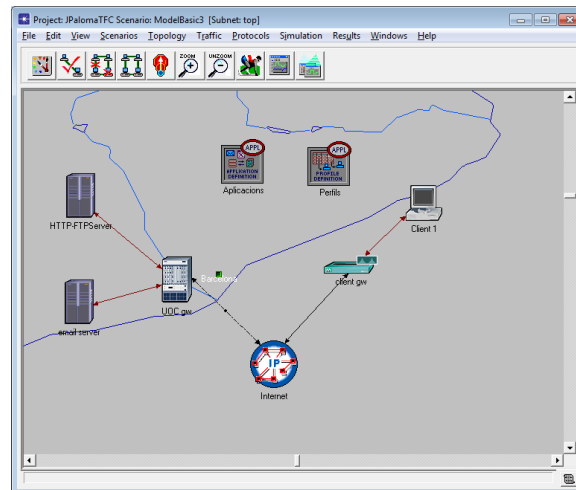


fig. 19 Escenari model bàsic III

### 7.4.2.- Aplicacions

En aquest punt no canvia res, tot el que s'ha de fer és crear la nova aplicació i assignar-la al servidor HTTP que ja tenia.

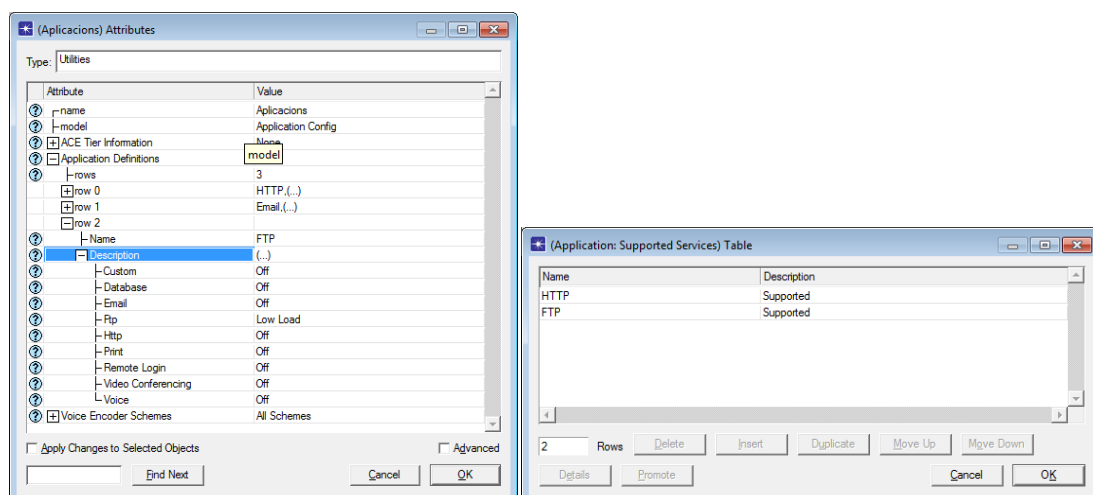


fig. 20 Aplicacions model bàsic III

Creo una nova aplicació FTP en mode "Low load" i la assigno al al servidor HTTP.

### 7.4.3.- Perfils

Res nou que no hagi fet fins ara. Afegir la nova aplicació al perfil client.

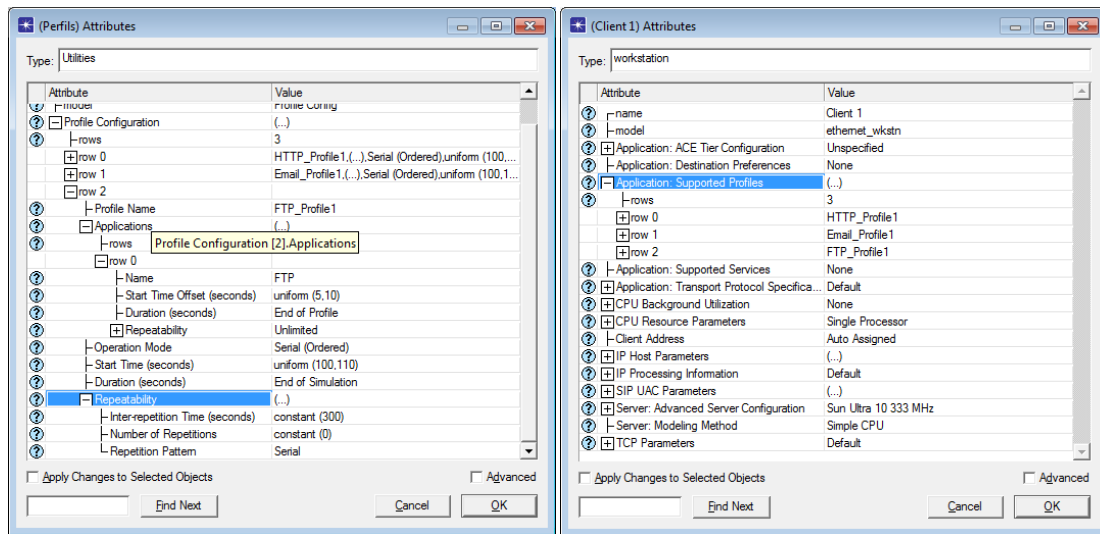


fig. 21 Perfil escenari bàsic III

### 7.4.4.- Simulació

En aquesta simulació espero trobar tràfic HTTP, Email i FTP. Aquests seran els tres protocols amb els que treballaré en l'escenari final que estic construint pas a pas.

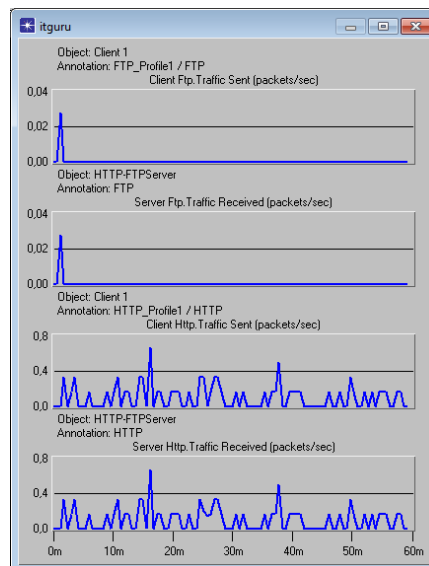


fig. 22 Simulació escenari bàsic III

Analitzant les gràfiques comprovem com el servidor està absorbint tot el tràfic FTP generat per el client a (dues gràfiques superiors) i com està absorbint també tot el tràfic HTTP (dues gràfiques inferiors). Concluïm doncs que el sistema funciona correctament i els servidors estan llestos per treballar amb balancejadors.



## 7.5.- Model Balancejat I

### 7.5.1.- Escenari

Començo a entrar en matèria de veritat. En aquest escenari introdueixo un balancejador. Per a tal fi el que faig es replicar (fer copia-enganxa) del servidor HTTP-FTP fins a tres vegades i els faig penjar d'un balancejador.

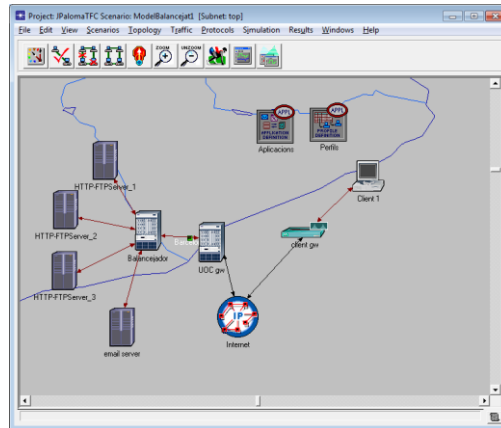


fig. 23 Escenari model balancejat I

El que buscaré en aquest escenari és comprovar que el balancejador realment balanceja tràfic tal i com s'espera d'ell. En aquest punt no tindrè en compte connexions ni polítiques de balanceig, em centraré únicament en la seva funció més bàsica.

### 7.5.2.- Simulació

Com ja he comentat, el que buscaré de la simulació és comprovar que el tràfic enviat per el client es reparteix correctament per els tres servidors (sense oblidar el servidor de correu).

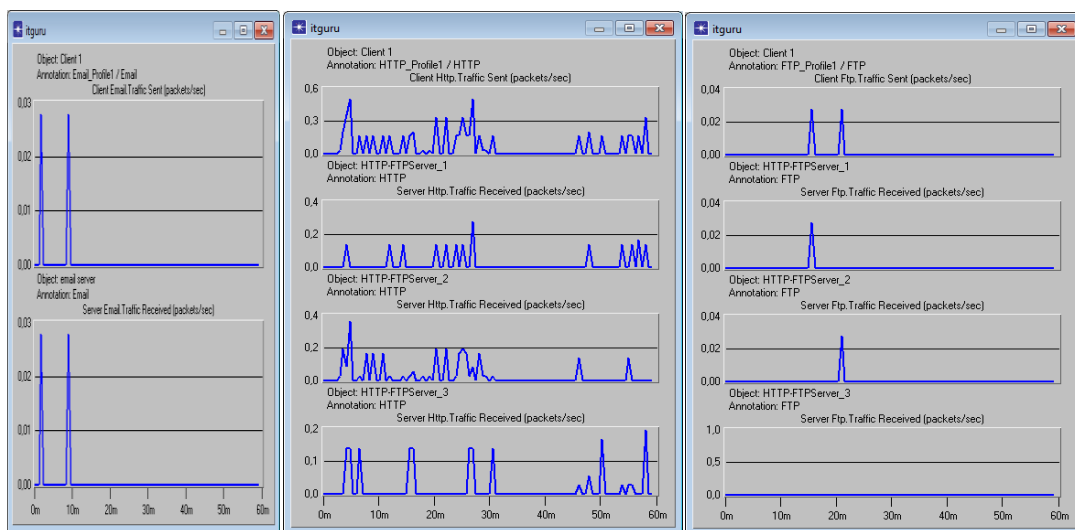


fig. 24 Simulació escenari balancejat I

A la primera figura es comprova com el tràfic SMTP generat per el client (gràfica superior) arriba de manera completa al servidor de correu (gràfica inferior). Per tant el balancejador no afecta al tràfic email.

A la segona figura es comprova que el tràfic està essent balancejat cap els tres servidors. A la gràfica superior veiem el tràfic HTTP generat per el client i a les tres inferiors els tràfic HTTP rebut per el primer, segon i tercer servidor respectivament. Comprovem com tots tres servidors reben tràfic, per tant el balancejador està repartint les dades per tots tres servidors i concluïm que el balancejador està balancejant el tràfic HTTP.

El mateix passa amb el tràfic FTP. A la tercera figura és comprova com el tràfic FTP generat per el client (gràfica superior) és repartir per els tres servidors (tres gràfiques inferiors).

Podem concluir definitivament que l'escenari està ben dissenyat i que el balancejador està complint la seva funció.

## 7.6.- Model Balancejat II

### 7.6.1.- Escenari

Pel que fa a les màquines, en aquest escenari hi tinc exactament les mateixes màquines que en la versió anterior, però aquí incorpore una nova utilitat. Es tracta d'un node anomenat "failure-recovery" que s'encarrega de fer caure o de recuperar màquines caigudes. Em servirà per saber com es comporta el balancejador en cas de que un dels servidors caigui.

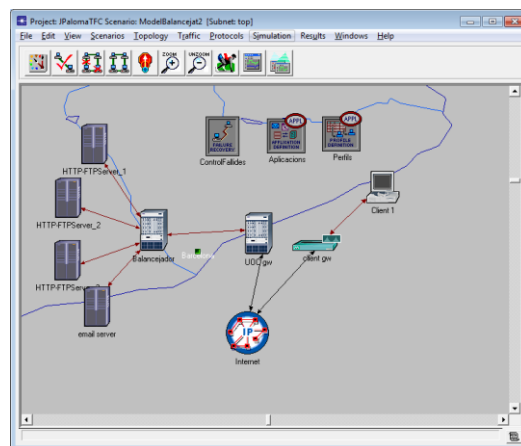


fig. 25 Escenari model balancejat II

### 7.6.2.- Simulació

El que espero trobar-me en aquesta simulació es que en el moment que un dels servidors caigui, el balancejador faci que els altres dos servidors assumeixin el tràfic de la màquina caiguda. Per a tal efecte configuro la utilitat "Control de fallides".

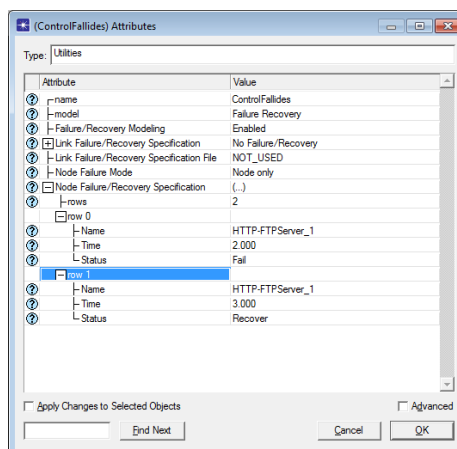
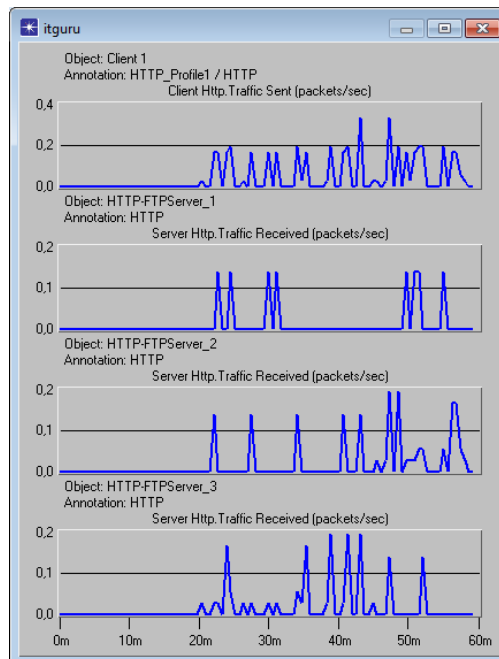


fig. 26 Atributs del control de fallides

El que estic indicant en la imatge superior es que necessito que el servidor HTTP\_FTP\_Server\_1 deixi de funcionar al segon 2000 de la simulació i recuperi el servei al segon 3000 de la simulació. Realitzo la simulació.



*fig. 27 Simulació escenari balancejat II*

A la figura anterior es comprova que la màquina client està enviant tràfic durant tota la estona (gràfica superior). En les tres gràfiques inferiors observem com els tres servidors estan rebent el tràfic generat per el client de manera balancejada i que just arribat el segon 2000 de la simulació, el primer servidor deixa de rebre tràfic i que en el segon 3000 remunta tornant a tenir tràfic un altre cop. Durant aquest interval de temps, els altres dos servidors estan assumint tot el tràfic. Per tant ja estic segur que el node que faré servir per simular les caigudes dels servidors està funcionant perfectament.

## 7.8.- Model Balancejat III

### 7.8.1.- Escenari

En aquest escenari no incorporo cap element destacable. El que faig és incorporar un firewall entre el balancejador i el dispositiu gateway que ens dóna accés a internet. La funció d'aquest equip és la de filtrar tots els paquets i protegir la xarxa interna de possibles atacs externs. En sí no aporta res al projecte, únicament crec que l'haig d'incloure per a representar el campus de la manera més fidedigna possible.

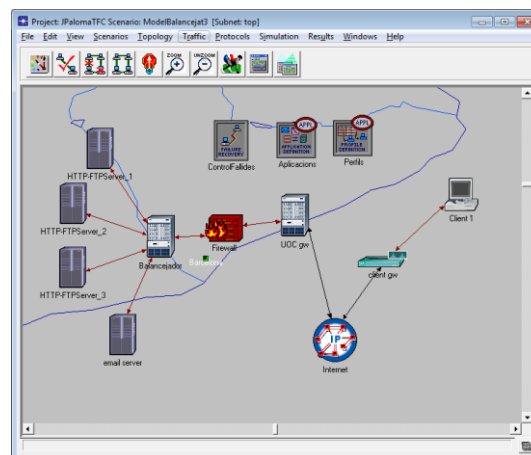


fig. 28 Escenari model balancejat III

### 7.8.2.- Simulació

A la simulació l'únic que haig de buscar és que el firewall no estigui filtrant tràfic i estigui alterant les gràfiques.

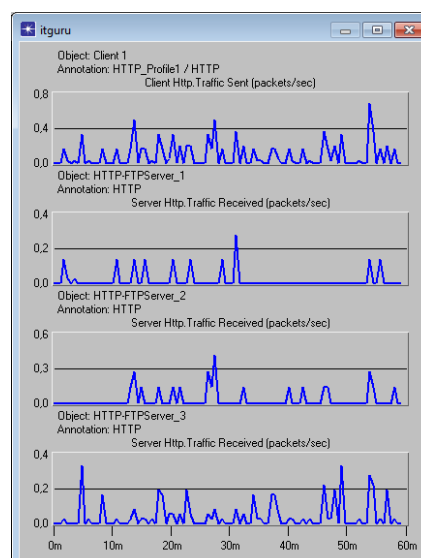


fig. 29 Simulació model balancejat III

Confirmo d'aquesta manera que el firewall no està influenciant en el tràfic de dades. Es veu com el client envia dades http de manera continuada, que els servidors absorbeixen el tràfic de manera balancejada i que el primer servidor cau als 2000 segons de començar la simulació i que es recupera al cap de 3000 segons.

Puc dir que l'escenari bàsic amb balanceig ja el tinc acabat i que serà aquest el que faré servir per a fer les proves client servidor que desitjo.

## 7.9.- Model Final

Tornem de nou a l'objectiu que em vaig marcar a l'inici del projecte i ataquem-lo directament. El que estic buscant és recrear el model dissenyat per Arquimbau i provar el comportament del balancejador si el tràfic que rep està generat per subxarxes formades per un sol equip radera d'un router. Recordem que Arquimbau ho va fer amb subxarxes de un, dos, tres, quatre i cinc equips.

També afegirem al model final els canvis realitzats a la part WAN per en Lahuerta, en el que re definia les connexions WAN dels equips configurats a les subxarxes i solucionava d'aquesta manera el problema de tràfic que es generava a l'entrada del Campus Virtual.

### 7.9.1.- Modelatge dels nodes

El primer pas serà crear un node de varies màquines fàcilment replicable fent servir el copia-enganxa.

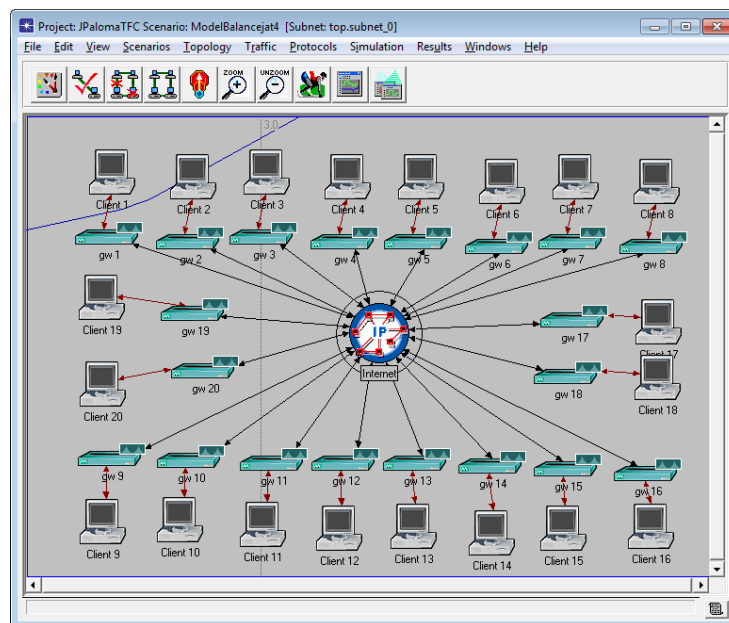


fig. 30 Node del model final

Aquest serà el node base a partir del qual podem crear subxarxes de 20 en 20 equips. La idea serà generar una sèrie de perfils amb diferents comportaments i configurar de manera aleatòria cada una de les màquines client. D'aquesta manera tindrem un comportament bastant semblant al de la realitat.

### 7.9.2.- Connexió dels nodes i simulació

Començaré connectant un sol node a la xarxa per veure si el node funciona i passaré a realitzar la simulació.

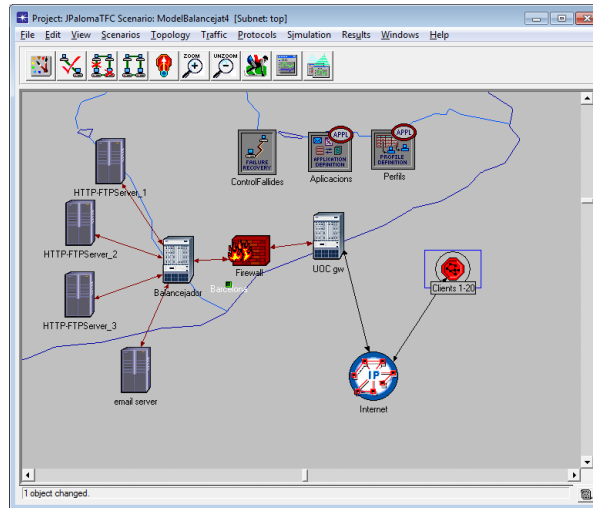


fig. 31 Connexió de nodes

Simulations using the IT Guru Academic Edition are limited to 20 nodes connected to two or more devices. Your simulation has exceeded this limit and has been stopped.

Sembla ser que el nostre software està limitat i no ens permet fer les proves que necessitàvem. Per poder continuar amb aquest projecte necessitaríem tenir el OPNET Modeler, del qual no en tinc llicència ni la UOC me la pot proporcionar.



## 8.- Estudi del balancejador amb el model WAN

En el treball d'Arquimbau va quedar patent que el seu model no era apte per a l'estudi del comportament dels balancejadors ja que els servidors sempre anaven massa lliures de càrrega.

En en el treball fet per en Lahuerta es demostrava que es podia treballar amb WANs de la mateixa manera que amb Workstations i que els servidors podien sobrecarregar la CPU. Però aquest model mai va ser provat amb un balancejador de càrrega. Segons deia en Lahuerta, els nodes LAN proporcionats per OPNET tenien un servidor intern que desvirtuava els resultats, això ho soluciona configurant els nodes de tal manera que enviïn el tràfic cap a un sol servidor. La pregunta que suscita aquesta configuració és:

**Afectarà això al funcionament del balancejador?** La teoria ens diu que si, ja que en aquest cas el balancejador no hauria d'intervenir les dades que li arriben donat que tenen com a destí un servidor en particular.

Tampoc tenim accés als dissenys elaborats per Lahuerta, per tant n'hauré de crear un de nou seguint les indicacions que va deixar en el seu treball.

El primer pas serà configurar-me un escenari simple en el que tindrem un node LAN amb 10 equips únicament, no me'n fan falta més per comprovar si el model d'en Lahuerta m'és útil per estudiar el comportament dels balancejadors.

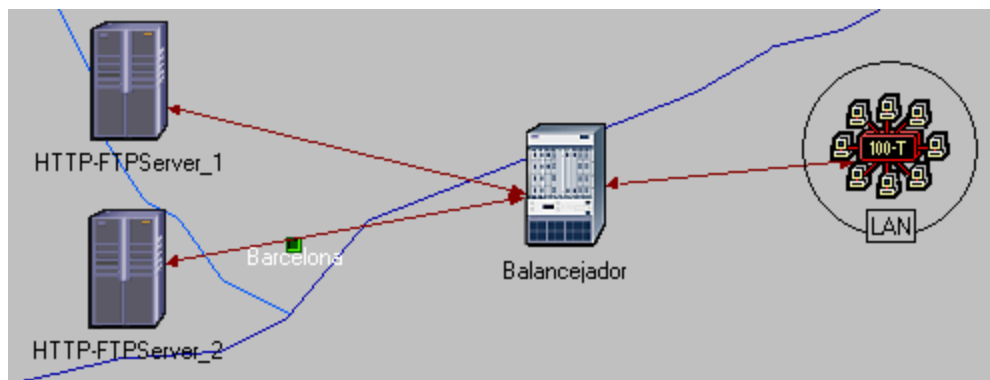


fig. 32 Model bàsic WAN

Els dos servidors tenen la mateixa configuració que els servidors que he fet servir en el disseny del model Arquimbau. La política de balanceig al balancejador és menyspreable per a l'objectiu buscat i a la LAN li he carregat el mateix perfil de tràfic HTTP que tenia al model Arquimbau. També deixo la utilitat "Control Fallides" que tornarà el Servidor 1 desde el segon 2000 al 3000 de la simulació.

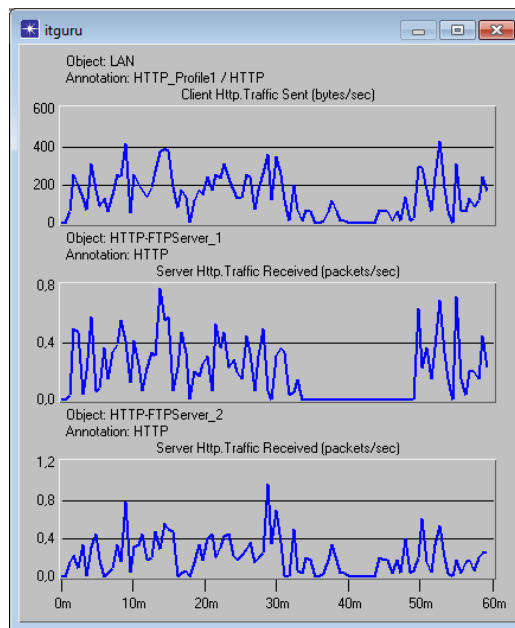


fig. 33 Simulació model WAN

La configuració és correcta, la LAN formada per 10 equips està enviant dades HTTP a través del balancejador (gràfica superior) i aquest les balanceja entre el primer i segon servidor (gràfiques inferiors). Veiem que durant els segons 2000 i 3000 el servidor 1 no rep tràfic i aquest és absorbit pel segon servidor.

Configurarem ara el node LAN tal i com indica Lahuerta. Explicaré les passes perquè al seu treball no queda gaire clar.

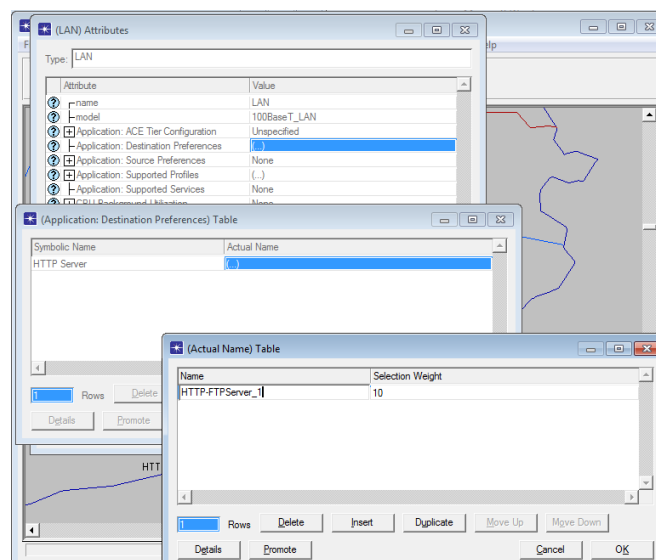
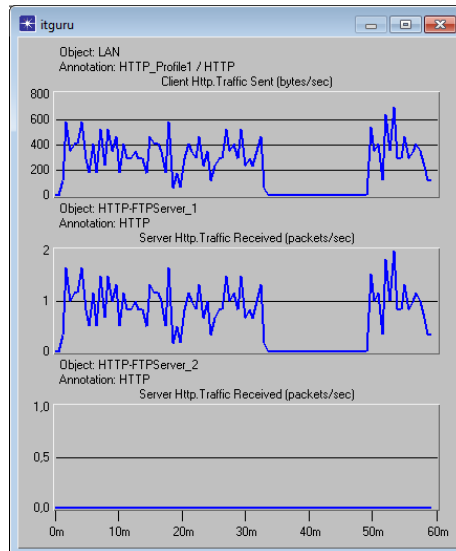


fig. 34 Configuració node LAN

Per a configurar el Servidor destí s'han d'editar els atributs del node LAN i modificar el valor del camp "Application: Destination preferences", al camp

Symbolic Name escollim "HTTP Server" i a Actual Name haurem d'escriure el nom del servidor manualment.

Com es pot veure a la imatge, he configurat la LAN amb el servidor 1 com a preferent. Ara en un principi la LAN es comporta com si hi haguessin deu subxarxes amb una màquina client penjada a un router.



*fig. 35 Simulació model WAN*

El que havia suposat es confirma, si observem les gràfiques de la figura superior podem veure com el node LAN genera tràfic (gràfica superior). Analitzant les dues gràfiques inferiors, les quals indiquen el tràfic rebut al servidor 1 i 2 respectivament, veiem que l'únic que rep tràfic és el servidor 1, mentres que el servidor 2 no rep tràfic. Per tant, concloem que el balancejador no està realitzant el balanceig de tràfic degut a que tot el tràfic definit al node LAN està enviant les dades al primer servidor.

## 9.- Proposició del model WAN – workstation

Fins el moment hem vist que no podem estudiar el comportament del balancejador utilitzant un model de workstation. Això és degut a una limitació en el software que s'està fent servir.

Per una altra banda també hem comprovat que el sistema WAN proporcionat per en Lahuerta tampoc és vàlid per estudiar el comportament dels balancejadors a OPNET. Això és degut a que la configuració requerida dins dels nodes WAN impedeix que el balancejador tracti els paquets de dades i faci que la seva funció de repartiment per els servidors quedi anul·lada.

Per tant, teoritzem que el millor model per a poder estudiar el comportament dels balancejadors serà fer una barreja del model workstation i el model WAN. És a dir, el que haurem de fer és usar els nodes workstations explicats al punt 7.9.1 d'aquest treball com a clients generadors de tràfic juntament amb la interconnexió que Lahuerta va establir en el seu treball.

### 9.1.- Problema d'interconnexió al model workstation

Abans de que en Lahuerta dissenyés el model WAN, es va estar fent servir un node que OPNET proporciona per defecte anomenat ip\_cloud32. Aquest node presentava una sèrie de problemes:

- Aquest node disposa de 32 entrades sèrie per les quals entren i surten els paquets de dades. Aquests paquets són tractats per el node mitjançant una cua FIFO (first input first output). Evidentment el fet de tractar els paquets amb aquest sistema provoca un retard en el tractament de les dades i per tant desvirtua els resultats de la simulació.
- Es feien servir línies de connexió del tipus PPP DS1 per connectar els routers clients a internet, aquestes línies donen una velocitat màxima de transferència de dades de 1.5Mbps. Aquest connexionat limita molt la capacitat del sistema i es converteix en un coll d'ampolla per a les dades dirigides cap als servidors.

## 9.2.- Solució a l' interconnexió del model workstation

Per a solucionar aquest problema, Lahuerta dissenyà un nou node amb connexions ethernet que solucionava els problemes de connexionat plantejats en el punt anterior.

Per a construir el node hem de pitjar "Topology -> Device Creator -> Cloud -> Multiprotocol".

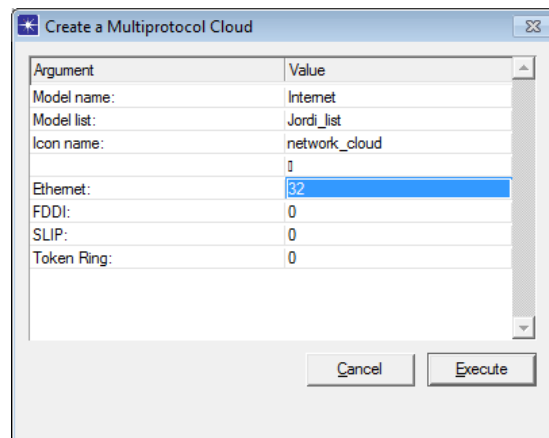


fig. 36 Device creator

Per crear el nou node hem de completar una sèrie de paràmetres:

- **Model Name:** És el nom que li donarem al node. En el nostre cas es dirà Internet.
- **Model List:** En quin llistat de la paleta d'objectes apareixerà el node. En el nostre cas a la llista Jordi\_List.
- **Icon name:** Nom de la icona. En el nostre cas deixem el que ve per defecte.
- **Ethernet:** Nombre de ports Ethernet. En el nostre cas el configurarem amb 32 ports.
- **FDDI, SLIP, Token ring:** Nombre de prots d'aquest tipus. En el nostre cas no serà necessari activar-ne cap.

Proseguirem creant els nodes encarregats de generar el tràfic cap els servidors. S'ha de tenir en compte que ara els equips hauran de ser connectats amb els nodes mitjançant cables Ethernet. Això és així perquè busquem salvar el coll d'ampolla que ens generava fer servir cables DS1.

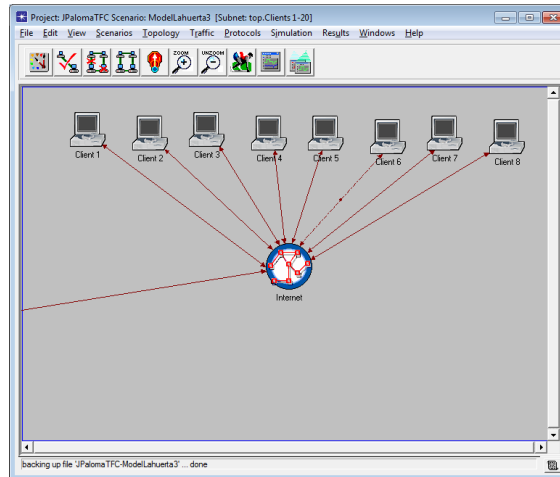


fig. 37 Node model WAN

Per unir el node Internet i els clients es poden fer servir cables de Gigaethernet, d'aquesta manera ens assegurem que no tindrem problemes de capacitat d'ample de banda.

En aquests nodes podem connectar fins a 31 equips (recordem que hem creat un node amb 32 connexions ethernet), hem de comptar que una de les 32 connexions disponibles la farem servir per al connexionat amb la resta de nodes. La topologia quedaria així:

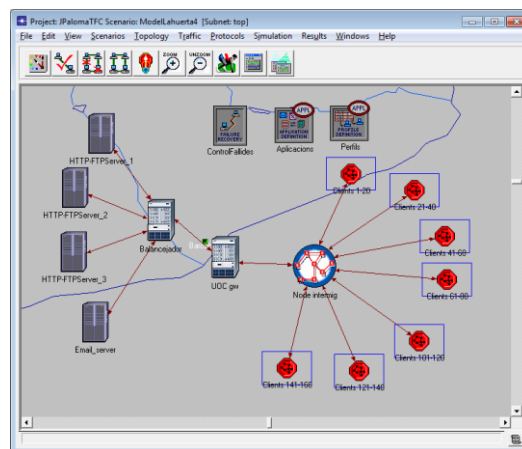


fig. 38 Exemple amb 120 clients

En la figura superior veiem un node internet amb una sèrie de nodes de 20 equips client connectats. Evidentment si fem un copia i enganxa d'aquests nodes podem aconseguir una xarxa de varis milers d'equips de forma ràpida i senzilla. Davant la impossibilitat de simular un escenari com aquest degut a limitacions de software, deixaré la proposta per a futurs projectes.

## 10.- Conclusions i línies de recerca.

L'objectiu principal del projecte era el d'investigar el comportament dels balancejadors dins del campus virtual basant-nos en els treballs realitzats en projectes anteriors per Arquimbau i Lahuerta.

Per arribar a l'objectiu necessitava recrear l'escenari ideat per Arquimbau implantant una sèrie de canvis proposats per Lahuerta. Durant el projecte m'he trobat amb impediments tècnics que no m'han permès finalitzar el treball, però si extreure alguna conclusió pel que fa a l'estudi dels balancejadors.

Al llarg del projecte Castelldefels sempre hi han hagut dos tipus de models confrontats. El model workstation, que defensa que la millor manera per a representar la topologia amb la que s'enfronta el Campus Virtual és recrear-lo amb subxarxes d'un equip penjant d'un router, i el model WAN que defensa que el model es pot representar de manera similar a la primera fent servir nodes LAN de varis equips.

Al final del meu projecte s'ha demostrat que mitjançant el model WAN, tot i ser similar al workstation, no serveix com a model perquè fa que els balancejadors no siguin capaços de balancejar el tràfic. Per tant per a representar el Campus Virtual haurem de crear un model basat en workstations, sempre i quan es vulguin tenir balancejadors dins del model.

Per a poder realitzar aquest model i simular-lo serà necessari disposar del software OPNET Modeler, ja que amb la llicència universitària no és possible representar-lo.

Queda pendent doncs estudiar el comportament dels balancejadors fent servir subxarxes d'un equip amb els canvis proposat per en Lahuerta a la part WAN de la connexió.

## 11.- Bibliografia

**Juan Francisco Lahuerta Martínez (2007):** Modelado y simulación del sistema informático que da soporte al Campus Digital – WAN

**Hugo Freire Gil (2007):** Modelatge i simulació del sistema informàtic que dona suport al Campus Digital- Escalabilitat del model i balancejadors de càrrega.

**Miguel Àngel Arquimbau Jaques (2007):** Modelatge i simulació del sistema informàtic que dona suport al Campus Digital – Balancejadors de càrrega

**Javier Martin Garcia-Asenjo (2011):** The Castelldefels Project 2011: Estado del arte de los TFC sobre el Campus Virtual de la UOC.

**2006 OPNET Technologies, Inc. (2005) :** OPNET Modeler Product documentation, release 11.5.

**Martínez Bazán, Norbert (2006):** Modelatge i simulació del sistema informàtic que dona suport al Campus Digital. Projecte de Final de Carrera, Enginyeria Informàtica, Universitat Oberta de Catalunya.

**Universitat Politècnica de Catalunya (2004):** OPNET: Manual de usuario. Departament d'Enginyeria Telemàtica, Secció d'Enginyeria Telemàtica de l'EPSEVG.