Protocol LUMI Pràctica 2

Feng Lin i Marc Sánchez.

**1. Quins són els serveis que ha d’oferir la capa de LUMI? És a dir, per a què serveix, quines són les seves tasques? Recordeu que és una capa d’aplicació, a sobre de transport (UDP) i a sobre de xarxa (de fet, “interxarxa + xarxa”).**

Serveis :

Construïr el protocol LUMI

Interpretar el protocol LUMI

Comunicar-se amb el servidor LUMI

Reaccionar amb el servidor LUMI

Conversar amb un altre client LUMI

Comunicar-se mitjançant la interfície de sockets amb la capa de transport

El que ha de fer és de pont entre l’interfície d’usuari i la interfície de sockets. També ha de saber construïr i treballar amb el protocol LUMI que definirem a classe.

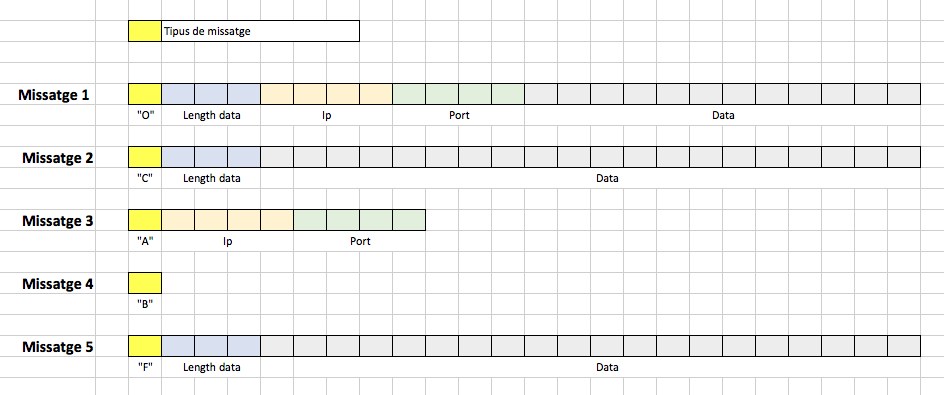
**2. Com serà la informació (“d’on es troba un usuari”) que mantindrà un “servidor” (és a dir, un node de LUMI)? L’adreça del socket de MI, la del socket de LUMI, ambdues, altres? Poseu un exemple de la informació que un servidor mantindria.**

Hi ha dues maneres viables de fer-ho, o bé el servidor es desa la ip i el port LUMI del client a una taula adreçada amb el nickname o bé el servidor es desa el l’adreça MI i el port MI del client i quan algú s’hi vol connectar li pregunta mitjançant el canal LUMI.

**3. Quin protocol de la capa de LUMI (PLUMI) proposeu? Cal que dissenyeu els noms dels missatges, el seu significat, format (camps i la seva longitud i valors possibles) i seqüències temporals, en els procediments de registre, localització i desregistre (“normal”). Recordeu que PLUMI va a sobre d’UDP (PLUMI/UDP/IP/x), que és no orientat a connexió i no fiable, i que en concret fa detecció més descart. Per tant, heu de tenir en compte en el disseny que els missatges PLUMI es poden perdre (o arribar duplicats). Què fareu si es perd un missatge PLUMI?**

El que es proposa son 3 tipus de missatge. Partint de la base de que el protocol LUMI desarà la direcció ip i port ip on escolta el client al que es vol connectar.

Veiem tot seguit el tipus de missatges en el dibuix

****

Tipus de missatge :

* **Missatge 1 :** Quan el client obre l’aplicació envia al servidor el seu nom, la ip i el port per fer-ho fa servir el tipus de missatge 1.
* **Missatge 2 :** El client vol connectar-se amb un altre client i li pregunta la direcció del nickname que envia.
* **Missatge 3 :** El servidor retorna a un client que li ha preguntat amb el missatge 2 la ip i el port on pot connectar-se amb l’altre client.
* **Missatge 4 :** El servidor pregunta a un client cada determinat temps si encara està viu. En cas que sí li contesta amb un missatge 1, en cas que no contesti passat un timeout el servidor elimina el registre de la taula.
* **Missatge 5 :** Un client notifica al servidor que l’aplicació es tenca i que pot borrar el registre de la taula.

Cada cop que un missatge plumi s’envia s’encén un timeout i si no es contesta dins d’aquest timeout es torna a enviar fins un màxim de 3 vegades.

**4. Un agent d’usuari de LUMI, com sabrà on està el seu node de LUMI? Ja que per enviar-li missatges ha de saber on està. Recordeu que el nom del vostre domini serà igual al nom DNS de la màquina on executareu el node de LUMI.**

Es demana per teclat o es posa dins del codi la ip i el port on el node de LUMI està escoltant, no hi ha altre manera que hardcodejar aquesta informació a cada un dels nodes. És el mateix procediment de configuració que configurar les direccions dns a un ordinador.

**5. Què passaria si l’aplicació de MI acabés l’execució de manera inesperada, perquè cau, es penja, etc., i no pogués arribar a fer el desregistre (“normal”)? Hi hauria alguna manera d’arreglar-ho (és a dir, alguna manera de fer un desregistre “anormal”)?**

Sí està contemplat anteriorment amb un missatge del servidor als diferents clients que té registrats, el que acabarà passant és que el servidor cada determinat temps preguntarà a cada un dels clients registrats mitjançant el socket que s’ha desat quan s’ha establert la connexió si encara és viu o no, si no és viu acabat un timeout definit anteriorment el servidor esborra (desregistra) el client. En cas contrari rebrà un missatge 1 i si ha canviat el nom dns actualitzarà la taula amb les dades que pertoquen.

**6. Quina interfície de la capa de LUMI voleu fer servir? És a dir, els noms de funcions, la descripció del que fan, què son els arguments i els valors de retorn (no confoneu les funcions de la interfície, que són “externes”, amb les “internes”). Quina relació tenen aquestes funcions amb les seqüències de missatges del protocol PLUMI? Quines d’aquestes funcions farà servir el “client” (és a dir, l’agent d’usuari de LUMI) i quines el “servidor” (és a dir, el node de LUMI)? I finalment, com integrareu l’agent d’usuari de LUMI a l’aplicació de MI?.**

**Funcions LUMI Client:**

Definim les funcions LUMI del client i del servidor espectants a saber i deduïr els paràmetres al llarc de la pràctica.

* LUMI\_registre() -> Notifica al servidor que s’ha iniciat una aplicació i dóna els paràmetres de localització i nom dns al servidor lumi.
* LUMI\_descobreix() -> Demana a un servidor la direcció ip i port per on s’ha de comunicar amb un altre client LUMI (mitjançant el protocol MI).
* LUMI\_finalitza() -> Notifica al servidor que ja ha finalitzat l’execució del programa.
* LUMI\_actualitza() -> Quan reb un

Totes les funcions anteriors estan dins de la capa de LUMI i per tant es dedueix que sempre utilitzaran els missatges definits anteriorment per realitzar les comunicacions amb el servidor.

**Funcions LUMI Servidor :**

* LUMI\_busy() -> El que fa és enviar un senyal a un client en concret per saber si està o no disponible per parlar amb altres clients LUMI.
* LUMI\_envia\_adreca() -> Envia a un client una direcció ip i un port encapsulats dins d’un paquet LUMI.

El protocol MI realitzat a la pràctica anterior és un protocol que s’utilitzarà per poder realitzar una conversa entre dos clients. Com que tant LUMI com MI son dues capes d’aplicació paral·leles una no integra l’altre sinó que estan al mateix nivell i les dues aniran dins de paquets de la capa de transport, en el cas de MI dins d’un TCP i en el cas de LUMI dins d’un UDP.

Aquest fet fa que sense cap tipus de problema les dues funcionalitats puguin conviure dins d’una mateixa aplicació utilitzant dos tipus de protocols diferents i dos protocols de transport diferents.