

# Xarxes: Pràctica 1, Programació d'aplicacions de xarxa

César Fernández, Paula Galucci, Enric Guitart, Carles Mateu

Febrer 2024

Finalitat de la pràctica:

- Aprendre a programar aplicacions de xarxes
- Entendre el model client-servidor
- Aprendre a programar i dissenyar un protocol de comunicacions

## Introducció

Es pretén desenvolupar un sistema per gestionar i controlar un conjunt de sensors i actuadors que poden ser instal·lats en diverses àrees d'un edifici. En concret, en aquesta tasca es centrarà en el desenvolupament del subsistema de comunicació encarregat de la transferència d'informació dels sensors i actuadors, així com del control de la seva operativitat. Per a la implementació d'aquest subsistema de comunicacions, s'optarà pel model client-servidor. Els clients seran els controladors dels sensors i actuadors, mentre que el servidor serà un equip central que albergarà el programari de gestió i control.

Els clients requeriran el desenvolupament d'un mòdul de comunicació amb el servidor que haurà de facilitar la identificació del controlador, mantenir una comunicació periòdica i permetre la transferència de dades en ambdues direccions.

Pel que fa al servidor, el mòdul que cal desenvolupar haurà de gestionar els clients autoritzats en el sistema, els clients actius, la comunicació bidireccional i l'emmagatzematge de dades i esdeveniments.

Les característiques i funcionalitats de cada un dels mòduls a implementar es detallen a les seccions següents.

## Client

El mòdul client s'hauria d'instal·lar en els controladors dels sensors i/o actuadors que han de formar part del sistema. Per la pràctica, el programari client s'executarà en un equip d'usuari simulant un controlador. Les dades de cada controlador a simular s'emmagatzemaran en un arxiu de configuració que es llegirà en iniciar-se el mòdul, això permetrà simular múltiples controladors en un sol equip d'usuari.

Genèricament el mòdul client ha de realitzar quatre tasques:

- Identificar-se en el servidor
- Mantenir una comunicació periòdica amb el servidor
- Enviar dades cap al servidor: valors mesurats per els sensors, esdeveniment produïts, etc.
- Rebre dades del servidor: valors pels actuadors, peticions d'informació, etc.

### INFORMACIÓ

**Sensor:** Dispositiu que monitoritza l'entorn captant informació d'esdeveniments o canvis.

### INFORMACIÓ

**Actuador:** Dispositiu que modifica l'estat d'algun element per variar les condicions de l'entorn.

### INFORMACIÓ

**Controlador:** Dispositiu que gestionen la informació rebuda dels sensors i la que cal enviar als actuadors.

Per simular la interacció amb els dispositius d'un controlador i l'enviament de dades cap al servidor, s'afegirà una cinquena tasca consistent en la espera de comandes per la consola del controlador.

## Dades de configuració

El programari client obtindrà de l'arxiu de configuració, emmagatzemat en el seu directori de treball, les dades d'identificació i les dades necessàries per la comunicació amb el servidor. Per defecte les dades es llegiran de l'arxiu `client.cfg`. Per llegir les dades de configuració d'un altre arxiu caldrà especificar l'opció `-c` en la crida del programari del client seguida del nom de l'arxiu de configuració (`-c <nom_arxiu>`). L'arxiu contindrà un paràmetre de configuració per línia i en cada línia s'especificarà el nom del paràmetre i el seu valor separats per el signe d'igualtat (=).

Els paràmetres que ha de contenir l'arxiu són:

- *Name*: Nom del controlador (alfanumèric de 8 dígitos)
- *Situation*: Codi per la situació del controlador amb el format `BxxLxxRxxAxx` (`B` → *Building*, `L` → *Level*, `R` → *Room*, `A` → *Area*), essent `xx` l'ordinal de cada element de situació. Els ordinals seran de dos dígitos, amb zeros a l'esquerra si escau.
- *Elements*: Dispositius del controlador. S'empra un codi alfanumèric de 7 dígitos per cada dispositiu (sensor o actuator) i els diferents dispositius del controlador se separen per punt i coma (;). El codi té el format `XXX-Y-Z`, on:
  - `XXX`: Magnitud que controla el dispositiu (`TEM`, `PRE`, `HUM`, `LUM`, `LEC`, `PRS`, `AUD`, `VID`, ..)
  - `Y`: Ordinal
  - `Z`: Tipus de dispositiu: `I` → entrada (actuator), `O` → sortida (sensor).
 Per cada dispositiu (element) caldrà definir una variable que emmagatzemi el seu valor. Un controlador pot tenir com a màxim 10 dispositius.
- *MAC*: Adreça MAC (**M**edia **A**ccess **C**ontrol) de la interfície de xarxa del controlador (12 caràcters hexadecimal).
- *Local-TCP*: Port TCP (**T**ransport **C**ontrol **P**rotocol) local per la transferència de dades amb el servidor.
- *Server*: Nom o adreça IP (**I**nternet **P**rotocol) del servidor.
- *Srv-UDP*: Port UDP (**U**ser **D**atagram **P**rotocol) del servidor.

## Identificar-se en el servidor

La primera tasca que ha de dur a terme un client del sistema és identificar-se en el servidor. A aquest procés es denominarà **subscripció** i fins que no finalitzi satisfactòriament el client no efectuarà cap altra tasca.

Per la comunicació d'aquesta tasca s'emprarà el protocol UDP amb la PDU (**P**rotocol **D**ata **U**nit) que es mostra en la figura 1.

tipus: unsigned char      char      char      char

Tipus paquet	Adreça MAC	Número aleatori	Dades
--------------	------------	-----------------	-------

Figura 1: Format PDU UDP

bytes:      1      13      9      80

Per la fase de registre s'han definit els tipus de paquets de la taula 1.

Valor	Mnemònic	Significat
0x00	SUBS_REQ	Petició de subscripció
0x01	SUBS_ACK	Acceptació de paquet de subscripció
0x02	SUBS_REJ	Rebuig de subscripció
0x03	SUBS_INFO	Paquet addicional de subscripció
0x04	INFO_ACK	Acceptació del paquet addicional de subscripció
0x05	SUBS_NACK	Error de subscripció

Taula 1: Tipus de paquet fase de subscripció

En la taula 2 s'indiquen els estats en els que pot estar un client del sistema.

Valor	Mnemònic	Significat
0xa0	DISCONNECTED	Controlador desconnectat
0xa1	NOT_SUBSCRIBED	Controlador connectat i no subscrit
0xa2	WAIT_ACK_SUBS	Espera confirmació primer paquet subscripció
0xa3	WAIT_INFO	Servidor esperant paquet SUBS_INFO
0xa4	WAIT_ACK_INFO	Espera confirmació segon paquet subscripció
0xa5	SUBSCRIBED	Controlador subscrit, sense enviar HELLO
0xa6	SEND_HELLO	Controlador enviant paquets de HELLO

Taula 2: Estats d'un client fase subscripció

### Procediment de subscripció

El client partirà de l'estat NOT\_SUBSCRIBED, haurà d'enviar una petició de subscripció [SUBS\_REQ] i esperar resposta del servidor. En enviar el primer paquet [SUBS\_REQ] el client passarà a estat WAIT\_ACK\_SUBS. Si no rep resposta del servidor en  $t$  segons tornarà a enviar la petició de subscripció. Per evitar saturar la xarxa, el temps  $t$  entre paquets s'anirà variant. Un cop enviats  $n$  paquets sense resposta s'esperarà  $u$  segons i s'iniciarà un nou procés de subscripció. Si passats  $o$  processos de subscripció no s'ha finalitzat satisfactòriament el procés de subscripció, el client finalitzarà indicant que no s'ha pogut contactar amb el servidor.

El procediment de subscripció sense resposta del servidor es mostra en la figura 2: L'interval d'enviament dels primers  $p$  paquets és  $t$ . Per cada paquet posterior a  $p$  l'interval d'enviament s'incrementarà en  $t$  segons fins arribar a  $q * t$  segons a partir dels quals l'interval d'enviament es mantindrà constant en aquest valor. Si després d'enviar  $n$  paquets no s'ha completat el procés de subscripció, s'espera  $u$  segons i s'inicia un nou procés de subscripció.

Per les proves del protocol s'han establert els següents valors dels temporitzadors i llindars:

$$t = 1, u = 2, n = 7, o = 3, p = 3, q = 3$$

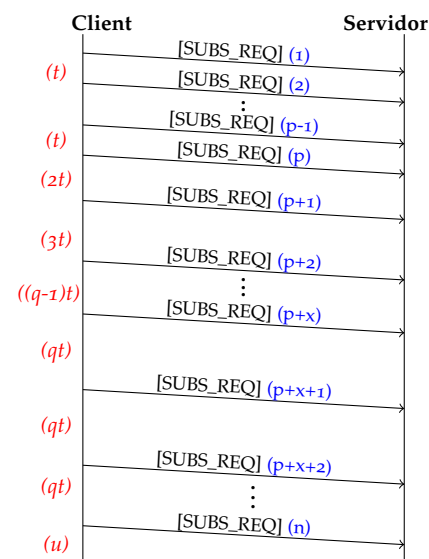


Figura 2: Temporitzacions procés de subscripció

Els valors d'una PDU tipus [SUBS\_REQ] han de ser:

- Adreça MAC: Adreça MAC de la interfície de xarxa del controlador (de l'arxiu configuració)
- Número aleatori: Zeros ("00000000")
- Dades: Nom del controlador i situació del controlador separats per coma (,) (de l'arxiu configuració)

En rebre una resposta del servidor, el client haurà d'avaluar el paquet rebut, l'estat en que es troba i actuar en conseqüència. El comportament del client segons el paquet rebut és el següent:

- [SUBS\_ACK]: Sol es pot rebre en l'estat WAIT\_ACK\_SUBS. En rebre aquest paquet el client haurà d'emmagatzemar les dades d'identificació del servidor (MAC, IP, Número aleatori) per poder comprovar que els successius paquets del servidor contenen aquestes dades. Seguidament s'enviarà un paquet [SUBS\_INFO], al port UDP indicat en el camp de dades del paquet [SUBS\_ACK] rebut. El contingut del paquet [SUBS\_INFO] serà:
  - Adreça MAC: Adreça MAC de la interfície de xarxa del controlador (de l'arxiu configuració)
  - Número aleatori: Valor enviat en el primer paquet del servidor (dades identificació servidor)
  - Dades: Port TCP per la transferència de dades amb el servidor i la llista de dispositius del controlador separats per punt i coma (;). El port TCP se separarà de la llista dispositius per coma (,)

El client passarà a l'estat WAIT\_ACK\_INFO

- [SUBS\_NACK]: El client passarà a l'estat NOT\_SUBSCRIBED i iniciarà l'enviament de paquets [SUBS\_REQ] sense iniciar un nou procés de subscripció.
- [SUBS\_REJ]: El client passarà a l'estat NOT\_SUBSCRIBED i s'iniciarà un nou procés de subscripció.
- [INFO\_ACK]: Sol es pot rebre en l'estat WAIT\_ACK\_INFO i les dades d'identificació del servidor han de ser correctes. El client passarà a l'estat SUBSCRIBED, considerarà finalitzada satisfactòriament la fase de subscripció i passarà a la següent tasca (fase). En la figura 3 es mostra el diagrama de temps d'un procés de subscripció finalitzat correctament.

La recepció d'un paquet diferent a l'esperat en un estat específic o bé amb les dades d'identificació del servidor incorrectes comportarà el pas del client a l'estat NOT\_SUBSCRIBED i l'inici d'un nou procés de subscripció.

### Mantenir comunicació periòdica

Per la comunicació periòdica amb el servidor s'emprarà el protocol UDP; la mateixa PDU i el mateix port del servidor que en la fase de subscripció. Per aquesta fase s'han definit 2 tipus de paquets que s'especifiquen en la taula 3.

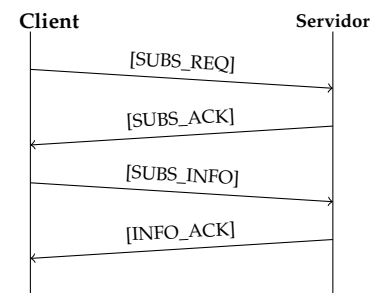


Figura 3: Procés de subscripció correcte

Valor	Mnemònic	Significat
0x10	HELLO	Enviament de HELLO
0x11	HELLO_REJ	Rebuig de recepció de HELLO

Taula 3: Tipus de paquet fase comunicació periòdica amb el servidor

### Procediment

Finalitzada satisfactòriament la fase de subscripció, estat SUBSCRIBED, el client enviarà al servidor un paquet [HELLO] cada  $v$  segons. Per la prova del protocol s'establirà  $v=2$ . El contingut d'aquest paquet ha de ser:

- Adreça MAC: Adreça MAC de la interfície de xarxa del controlador (de l'arxiu configuració)
- Número aleatori: Valor guardat en la identificació inicial del servidor
- Dades: Nom del controlador i situació del controlador separats per coma (,) (de l'arxiu configuració)

Tanmateix el client esperarà rebre del servidor un paquet [HELLO] cada  $v$  segons per constatar que hi ha comunicació entre les dues entitats. El diagrama de temps d'aquest procediment es mostra en la figura 4. Per cada paquet rebut haurà de comprovar la identitat del servidor (MAC, IP, Número aleatori) i que el camp de dades és el mateix que el que ha enviat en els seus paquets [HELLO]. Si hi ha discrepància entre les dades d'identificació rebudes i les emmagatzemades en el primer paquet, es considerarà que hi ha un problema de suplantació d'identitat. En aquest cas el client enviarà un paquet [HELLO\_REJ], passarà a l'estat NOT\_SUBSCRIBED i s'iniciarà un nou procés de subscripció.

En rebre el primer paquet [HELLO] del servidor el client passarà a l'estat SEND\_HELLO. Per la recepció d'aquest paquet s'estableix un temps màxim d'espera de  $r$  vegades l'interval d'enviament de [HELLO] (per les proves del protocol  $r=2$ ). Si no es rep aquest primer [HELLO] en el temps estipulat el client passarà a l'estat NOT\_SUBSCRIBED i s'iniciarà un nou procés de subscripció.

Quan el client passa a l'estat SEND\_HELLO obrirà el port TCP per la recepció de dades del servidor (*Local-TCP* de l'arxiu de configuració) i el mantindrà obert mentre es mantingui en aquest estat. En aquest estat també es podran introduir comandes per la consola del sistema.

Si el client deixar de rebre  $s$  [HELLO] consecutius, considerarà que s'ha perdut la comunicació amb el servidor, passarà a l'estat NOT\_SUBSCRIBED i s'iniciarà un nou procés de subscripció. Per les proves del protocol s'ha establert  $s=3$ .

Els paquets [HELLO] sol es poden rebre en els estats: SUBSCRIBED (per el primer paquet) i SEND\_HELLO.

Per últim si el client rep un paquet [HELLO\_REJ] passarà a l'estat NOT\_SUBSCRIBED i s'iniciarà un nou procés de subscripció.

El client es mantindrà en aquesta fase mentre es mantingui en l'estat SEND\_HELLO.

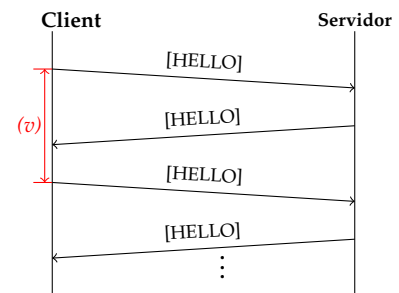


Figura 4: Manteniment de comunicació periòdic

## Enviar dades cap al servidor

Per enviar dades cap al servidor el client haurà d'obrir una comunicació TCP al port rebut del servidor en la fase de subscripció i enviar les dades corresponents.

En l'enviament d'informació al servidor s'emptra una PDU amb l'estructura descrita en la figura 5.

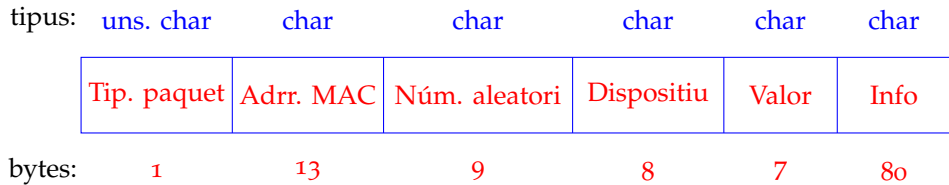


Figura 5: Format PDU TCP

Els tipus de paquet per la transferència de dades amb el servidor s'especifiquen en la taula 4.

Valor	Mnemònic	Significat
0x20	SEND_DATA	Enviament de dades des de el controlador
0x21	SET_DATA	Enviament de dades des de el servidor
0x22	GET_DATA	Petició de dades des de el servidor
0x23	DATA_ACK	Acceptació d'un paquet de dades
0x24	DATA_NACK	Error en un paquet de dades
0x25	DATA_REJ	Rebuig d'un paquet de dades

Taula 4: Tipus de paquet transferència de dades amb el servidor

L'enviament de dades cap al servidor s'activarà mitjançant una comanda en la consola del sistema (send <nom-dispositiu>). El client, un cop establerta la comunicació amb el servidor, haurà d'enviar un paquet [SEND\_DATA] amb els següents valors en la PDU:

- Adreça MAC: Adreça MAC de la interfície de xarxa del controlador (de l'arxiu configuració)
- Número aleatori: Valor enviat en el primer paquet del servidor (dades identificació servidor)
- Dispositiu: Nom del dispositiu especificat en la comanda send
- Valor: Valor associat al dispositiu
- Info: Valor buit ("")

Seguidament esperarà resposta del servidor durant  $w$  segons ( $w=3$  per les proves del protocol). Del servidor es poden rebre les següents respostes (tipus paquet):

- [DATA\_ACK]: Enviament d'informació acceptada, s'han emmagatzemat les dades en el servidor.
- [DATA\_NACK]: El servidor no ha pogut emmagatzemar les dades enviades o bé s'han rebut les dades errònies. Les dades no han estat acceptades.
- [DATA\_REJ]: La informació enviada al servidor ha estat rebutjada.

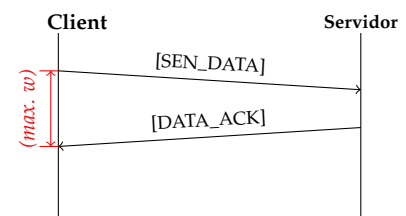


Figura 6: Enviament de dades al servidor

Si la resposta del servidor és [DATA\_REJ] o bé hi ha discrepàncies amb les dades d'identificació del servidor, el client passarà a l'estat NOT\_SUBSCRIBED i s'iniciarà un nou procés de subscripció.

La recepció d'un paquet de resposta [DATA\_NACK], la manca de resposta del servidor o la discrepància entre les dades del dispositiu enviades i rebudes (Nom o valor), es considerarà que les dades no han estat acceptades. En aquest cas caldria reenviar-les fins rebre confirmació correcta de la recepció. En la pràctica no caldrà implementar aquest reenviament.

Després de rebre una resposta o bé després de la temporització per manca de resposta, el client tancarà la comunicació TCP amb el servidor. El diagrama de temps de l'enviament de dades al servidor és el mostrat en la figura 6.

### Rebre dades del servidor

En rebre una connexió TCP, per el port obert un cop el client passa a l'estat SEND\_HELLO, s'esperarà un paquet del servidor durant  $w$  segons. La PDU i els tipus de paquets per aquesta tasca són els mateixos que per l'enviament de dades al servidor.

Del servidor es poden rebre dos tipus de paquets:

- [SET\_DATA]: Establir el valor d'un dispositiu d'entrada (actuador). Caldrà comprovar que el dispositiu pertany al controlador i que es tracta d'un dispositiu d'entrada. Si tot és correcte s'assignarà el valor rebut a la variable associada al dispositiu. El client respondrà amb un paquet [DATA\_ACK] que contindrà les seves dades d'identificació (MAC i Número aleatori) i els valors del dispositiu rebuts (Nom i Valor).
- [GET\_DATA]: Obtenir el valor d'un dispositiu. S'haurà de comprovar que el dispositiu pertany al controlador i en cas afirmatiu es respondrà amb un paquet [DATA\_ACK] que contindrà les seves dades d'identificació (MAC i Número aleatori) el nom del dispositiu i el valor de la variable associada.

Si en el paquet rebut del servidor hi ha discrepàncies en les dades d'identificació del servidor, s'enviarà un paquet [DATA\_REJ] amb les dades d'identificació del client, les dades rebudes del dispositiu (Nom i Valor) i en el camp Info el motiu del rebuig. El client passarà a l'estat NOT\_SUBSCRIBED i s'iniciarà un nou procés de subscripció.

Cas d'haver discrepàncies amb la identificació del dispositiu (tipus o nom), s'enviarà un paquet [DATA\_NACK] amb les dades d'identificació del client, les dades rebudes del dispositiu (Nom i Valor) i en el camp Info el motiu del error.

Després d'haver enviat el paquet de resposta al servidor, o si no s'ha rebut cap paquet del servidor en el temps estipulat, es finalitzarà la comunicació amb el servidor. La figura 7 mostra el diagrama de temps per la recepció de dades del servidor ([GET\_DATA]).



Figura 7: Rebre dades del servidor

### Espera de comandes

Un cop el client estigui en l'estat SEND\_HELLO es podran introduir les següents comandes per la consola del sistema:

- **stat:** Visualitza les dades d'identificació del controlador (MAC, Nom i Situació), tots els dispositius del controlador i el valor actual de cadascun d'ells.
- **set <nom-dispositiu> <valor>:** Simula la lectura del valor d'un sensor o bé l'estat d'un actuador. Canvia el valor associat al dispositiu <nom-dispositiu> a <valor>. En una implementació real la lectura dels valors dels dispositius es faria periòdicament o bé per esdeveniments.
- **send <nom-dispositiu>:** Envia el valor associat al dispositiu <nom-dispositiu> al servidor. S'emprarà el procediment descrit en la secció: **Enviar dades cap al servidor**. Igual que s'ha comentat en la comanda anterior, en una implementació real l'enviament de dades dels valors dels dispositius es faria periòdicament o bé per esdeveniments.
- **quit:** Finalització del client. Es tancaran tots els ports de comunicacions oberts i es finalitzaran tots els processos.

## Servidor

Per la seva part, el servidor genèricament ha d'efectuar quatre tasques:

- Atendre les peticions de subscripció dels clients.
- Gestionar el manteniment de comunicació amb els equips subscrits en el sistema.
- Esperar connexions TCP dels clients i tractar la informació que envien.
- Enviar informació i peticions d'informació als clients mitjançant TCP.

Seguint el mateix esquema que en el client, en el servidor hi haurà una cinquena tasca que permetrà introduir comandes per la consola del sistema per simular l'enviament de dades i de peticions d'informació cap als clients.

El servidor haurà d'atendre múltiples clients alhora (servidor concurrent).

Igual com s'ha descrit en el client, el servidor ha de llegir d'un arxiu el seus paràmetres de configuració. Aquest arxiu ha de tenir la mateixa estructura que l'esmentada per el client i per defecte s'anomenarà: `server.cfg`. Els paràmetres que ha de contenir aquest arxiu seran:

- **Name:** Nom del servidor (codi alfanumèric de 8 dígitos).
- **MAC:** Adreça MAC de la interfície de xarxa del servidor (12 caràcters hexadecimal).
- **UDP-port:** Port UDP per on espera rebre els paquets dels clients en la fase de subscripció i els del manteniment de la comunicació periòdica.
- **TCP-port:** Port TCP per on espera rebre les connexions dels clients per l'enviament de dades.



Els controladors autoritzats en el sistema estaran emmagatzemats en un arxiu que el servidor llegirà en iniciar-se. Per defecte aquest arxiu s'anomenarà `controlers.dat` i en cada línia contindrà la informació d'un controlador: nom del controlador i adreça MAC separats per coma (,). Juntament amb aquesta informació el servidor haurà de mantenir en memòria una estructura de dades que li permeti controlar l'estat dels clients, les adreces IP associades, la situació, el port TCP, etc.

En iniciar-se el servidor haurà d'obrir el dos canals de comunicació, UDP i TCP, i esperar les comunicacions dels clients. Per cada client, i cada protocol, haurà de destinar un procés que l'atengui.

Es descriuen a continuació cadascuna de les quatre tasques que ha d'efectuar el servidor:

### Atendre peticions de subscripció

En cada petició de subscripció el servidor haurà de comprovar que qui l'efectua és un controlador autoritzat en el sistema (MAC i nom) i que li envia les dades correctes (Número aleatori tot a zeros i situació del controlador). Si hi ha discrepància amb aquestes dades, el servidor respondrà amb un paquet [SUBS\_REJ] i el client es passarà a l'estat DISCONNECTED finalitzant el procés de subscripció. El contingut d'aquest paquet ha de ser:

- Adreça MAC: Adreça MAC de la interfície de xarxa del servidor (arxiu configuració)
- Número aleatori: Valor zeros ("oooooooo")
- Dades: Motiu del rebuig

Si totes les dades d'identificació del client són correctes i aquest es troba en l'estat DISCONNECTED, el servidor generarà un nombre aleatori per la comunicació i obrirà un nou port UDP per continuar el procés de subscripció amb el client i li enviarà un paquet [SUBS\_ACK] amb el contingut:

- Adreça MAC: Adreça MAC de la interfície de xarxa del servidor (arxiu configuració)
- Número aleatori: Valor del nombre aleatori generat
- Dades: Nou port UDP obert per continuar amb la subscripció

Seguidament el client es passarà a l'estat WAIT\_INFO i esperarà rebre un paquet [SUBS\_INFO], per el nou port UDP obert, durant  $s$  vegades el temps inicial d'enviaments de peticions de subscripció (per les proves del protocol  $s=2$ ). Si no es rep el paquet en el temps esperat es passarà el client a l'estat DISCONNECTED i finalitzarà el procés de subscripció en el servidor.

La recepció d'un paquet [SUBS\_INFO] correcte, tant en les dades d'identificació del client com en les dades addicionals, comportarà l'enviament d'un paquet [INFO\_ACK], el canvi de l'estat del client a SUBSCRIBED, l'emmagatzemament de les dades rebudes i la finalització del procés de subscripció. El contingut del paquet [INFO\_ACK] ha de ser:

- Adreça MAC: Adreça MAC de la interfície de xarxa del servidor (arxiu configuració)
- Número aleatori: Valor del nombre aleatori generat
- Dades: Port TCP del servidor per l'enviament d'informació dels controladors

Si el paquet [SUBS\_INF0] és incorrecte s'enviarà un paquet [SUBS\_REJ] igual al descrit en l'inici d'aquesta secció, el client canviarà a l'estat DISCONNECTED i finalitzarà el procés de subscripció en el servidor.

## Gestionar el manteniment de comunicació

La gestió de la comunicació periòdica amb els clients té dues parts:

1. Respondre als paquets [HELLO]: El servidor respondrà a cada paquet [HELLO] correcte (dades d'identificació, dades addicionals i estat del client) amb un altre paquet [HELLO] que contindrà la seva MAC, el número aleatori assignat al client i les dades rebudes del client. Amb aquesta operació el client podrà constatar que el servidor es troba operatiu. En rebre el primer paquet [HELLO] d'un client que està en l'estat SUBSCRIBED es passarà el client a l'estat SEND\_HELLO. Si el paquet [HELLO] rebut és incorrecte s'enviarà un paquet [HELLO\_REJ] i es passarà el client a l'estat DISCONNECTED.
2. Comprovar que els clients subscrits estan operatius: S'haurà de comprovar que tots els clients que han superat la fase de subscripció mantinguin la comunicació periòdica. En aquest sentit, si es deixen de rebre  $x$  paquets [HELLO] consecutius d'un client ( $x=3$  per les proves del protocol), el client es passarà a l'estat DISCONNECTED.

## Espera connexions TCP dels clients

Per cada connexió TCP que rebi el servidor haurà de crear un procés per atendre-la (servidor concurrent). En aquesta connexió sol pot rebre un tipus de paquet: [SEND\_DATA]. El servidor haurà de comprovar que es tracta d'un client autoritzat en el sistema, que les dades d'identificació són correctes, que es troba en l'estat SEND\_HELLO i que l'element referenciat en el paquet pertany al controlador. Si tot és correcte, el servidor emmagatzemarà les dades a disc i respondrà al client amb un paquet [DATA\_ACK] amb les seves dades d'identificació i les dades de l'element i valor rebudes del client. Un cop enviat el paquet [DATA\_ACK] es finalitzarà la comunicació amb el client.

Les dades rebudes del client s'emmagatzemaran en el directori de treball del servidor en un arxiu històric de dades. El nom d'aquest arxiu estarà format per el nom i la situació del controlador, separats per un guionet (-), i amb l'extensió .data. Per cada dada rebuda es generarà una línia de dades en l'arxiu i cada línia ha de contenir:

- La data de recepció (DD-MM-YY)
- L'hora de recepció (HH:MM:SS)
- El tipus de paquet que ha generat la dada

- El nom de l'element
- El valor rebut de l'element

En cada línia els components se separaran per una coma (,)

Si el servidor no pot emmagatzemar les dades rebudes del client s'enviarà un paquet [DATA\_NACK] amb les dades d'identificació del servidor, l'element i el valor rebut i en el camp info el motiu de la fallida d'emmagatzemament. Aquest mateix paquet s'enviarà si hi ha errors en les dades rebudes (element i/o valor), indicant en el camp info el motiu de l'error.

Cas que el paquet [SEND\_DATA] sigui erroni en la identificació del controlador, s'enviarà un paquet [DATA\_REJ] amb les seves dades d'identificació i les dades de l'element i valor rebudes del client, el client es passarà a DISCONNECTED i es finalitzarà la comunicació amb el client.

Per les comunicacions TCP en el servidor s'ha establert la mateixa temporització per la recepció de paquets que en el client: *w*.

### Enviar informació i peticions d'informació

Des del servidor es podrà enviar informació als actuadors dels controladors i també es podrà sol·licitar informació als controladors dels valors dels sensors i de l'estat dels actuadors. Aquestes operacions es duran a terme mitjançant una connexió TCP al port especificat per el client en la fase de subscripció. La PDU per aquesta comunicació es la descrita en la figura 5. Per l'enviament d'informació s'emprarà el paquet [SET\_DATA] i per la sol·licitud d'informació el paquet [GET\_DATA]. Ambdós paquets s'hauran d'omplir amb l'adreça MAC del servidor, el nombre aleatori assignat al client amb el que es vol comunicar, el nom de l'element i el valor assignat en el cas de [SET\_DATA] o bé valor buit ("" ) en el cas de [GET\_DATA].

Un cop enviat el paquet corresponent el servidor esperarà resposta del client durant *w* segons. L'acció del servidor dependrà del tipus de paquet rebut:

- [SEND\_ACK]: El paquet contindrà les dades sol·licitades (o les dades enviades) i caldrà emmagatzemar-les tal com s'ha especificat en l'apartat anterior. Un cop emmagatzemades es tancarà la comunicació amb el client.
- [SEND\_NACK]: Es finalitzarà la comunicació amb el client considerant l'operació fallida.
- [SEND\_REJ]: El client passarà a l'estat DISCONNECTED i es finalitzarà la comunicació amb el client. L'operació també es considerarà fallida.
- Sense resposta: Es finalitzarà la comunicació amb el client considerant l'operació fallida.

En el supòsit que el servidor no pugui contactar amb el client per el port TCP es passarà el client a l'estat DISCONNECTED i es finalitzarà la comunicació.

Les operacions d'enviament i petició d'informació es duran a terme mitjançant les comandes en la consola del sistema que es descriuen en el següent apartat.

## Espera de comandes

La consola de sistema del servidor ha de permetre les següents comandes:

- `list`: Visualitza la llista dels controladors autoritzats en el sistema, el seu estat i, per els controladors subscrits, la informació associada: IP, número aleatori, situació i elements
- `set <nom-controlador> <nom-dispositiu> <valor>`: Simula l'enviament d'informació a un element d'entrada (actuador) del controlador. Es procedirà segons el descrit en la secció **Enviar informació i peticions d'informació**.
- `get <nom-controlador> <nom-dispositiu>`: Simula la petició d'informació a un element del controlador. Es procedirà segons el descrit en la secció **Enviar informació i peticions d'informació**.
- `quit`: Finalitza el servidor. Es tancaran tots els ports de comunicacions oberts i es finalitzaran tots els processos.

## Implementació

La implementació de la pràctica s'haurà de fer en dos llenguatges:

- **Client**: Python (3.x [superior a 3.7])
  - Només es poden emprar llibreries estàndard
  - De les llibreries de comunicacions s'han d'emprar `socket` i `select`
- **Servidor**: Ansi C (C99).
  - Sol es poden emprar llibreries estàndard
  - S'emprarà l'eina `make` per compilar
  - S'ha de compilar amb `gcc` i les opcions `-ansi -pedantic -Wall`

Excepcionalment es permetrà, previa consulta al professor, i assumint que no seran llenguatges treballats a les pràctiques ni al material de classe les implementacions amb:

- **Client**: Go Lang (1.x [superior a 1.15])
  - Només es poden emprar llibreries estàndard
  - De les llibreries de comunicacions s'han d'emprar el `net` package
- **Servidor**: Rust (>1.59.0).
  - Només es poden emprar llibreries estàndard
  - S'ha d'emprar el mòdul: `std::net`

Tant el client com el servidor han de permetre els següents paràmetres d'entrada:

- [-d]: Fer *debug*. S'ha de presentar per consola un missatge per cada esdeveniment significatiu. Aquests missatges han de permetre fer un seguiment del funcionament de la implementació i del protocol.
- [-c <arxiu>]: Arxiu de dades de configuració del programari. Especifica el nom de l'arxiu d'on es llegiran les dades necessàries per la comunicació entre client i servidor. Si no s'especifica aquest paràmetre el nom per defecte serà especificat en la documentació.

El servidor ha de permetre un paràmetre addicional:

- [-u <arxiu>]: Arxiu de controladors autoritzats. Especifica el nom de l'arxiu que conté el l'adreça MAC i el nom dels controladors autoritzats en el sistema. Per defecte el seu valor ha de ser: `controlers.dat`.

En el codi del client s'ha d'emprar *shebang* (`#!`) que apunti a una versió 3 de python.

```
#!/usr/bin/env python
```

Independentment de l'opció de *debug* esmentada anteriorment, tant el client com el servidor han de mostrar per consola un missatge amb els canvis d'estat dels clients.

Per facilitar el desenvolupament de la pràctica, es proposa dividir-la en dues parts i cada part dividir-la en cinc fases progressives:

#### 1. Implementació del client

- Subscripció
- Manteniment de la comunicació
- Introducció de comandes
- Enviament de dades al servidor
- Recepció de dades del servidor

#### 2. Implementació del servidor

- Subscripció
- Manteniment de la comunicació
- Introducció de comandes
- Recepció de dades dels clients
- Enviament de dades als clients

## Lliurament

### Documentació

La pràctica és individual, cada alumne haurà de lliurar un únic arxiu amb les següents característiques:

- *Nom:* **XARXES-P1-nom** (On *nom* son els dos cognoms de l'autor separats per guionet (-))
- *Format:* ZIP, TGZ o TBZ.
- *Contingut:*
  - Codi font degudament comentat pel seu seguiment.
  - Arxius necessaris pel correcte funcionament del codi (configuracions clients, servidor, etc).
  - Un informe en format PDF que inclogui:
    1. L'estructura, a nivell esquemàtic (diagrama de blocs), del client i del servidor.
    2. L'estratègia emprada per el manteniment de la comunicació, tant en el client com en el servidor.
    3. Un diagrama d'estats del protocol implementat sobre UDP (subscripció i manteniment de la comunicació).
    4. Totes aquelles consideracions que cregueu oportunes.

Aquest document ha de complir les següents condicions:

- \* Una bona estructuració.
  - \* Un format dels elements de text correctes.
  - \* Un contingut clar i una redacció correcta (no s'acceptaran errades ortogràfiques ni abreviatures o símbols per substituir paraules).
- S'ha de lliurar al campus virtual (apartat Activitats) i no s'acceptarà per cap altre mitjà.



#### IMPORTANT

**La documentació que no compleixi tots aquests requisits no serà avaluada.**

### Termini

El termini per rebre la documentació relacionada amb aquesta pràctica finalitza el dia **21 d'abril de 2024**.

## Avaluació

Per a que una pràctica pugui ser avaluada haurà de complir els següents requisits:

- El codi del servidor no ha de donar cap tipus d'error en el procés de compilació ni en la execució.
- El codi del client s'ha d'executar sense errors.
- S'executarà el servidor i dos clients en la mateixa màquina i, com a mínim, han d'arribar a la fase de subscripció de la comunicació (els client enviant periòdicament HELLO i el servidor responent a cadascun d'ells). En el servidor han de funcionar les comandes `list` i `quit` i en el client `stat` i `quit`.
- Complir tots els requeriments de l'apartat *Documentació* de la secció **Lliurament**

La avaluació de la pràctica es realitzara en un sistema Linux (RockyLinux 9.x).