

4. laborategi-saioa: CSMA/CD Simulazioa

1. Praktikaren helburua

CSMA/CD protokoloa ikasteko tresna lagungarri bat ikasleari ematea da helburua. Protokolo horren oinarritzko ezaugarriak ondo ulertzeko, sare simulatu batean CSMA/CD protokoloaren inplementazioa egiten da.

Iraupena: 2 ordu eta 25 minutu

Kokapena: Irakasteko Laborategia: 4

2. Ikaslearen lana

C funtzio bat programatu behar da, CSMA/CD protokoloa egikaritzen duena. Sistemarekiko interfazea egiten duen funtzio-multzo bat ematen zaio ikasleari, lana errazteko.

Inplementazioaren arrazoiengatik derrigorrezkoa egiten da simuladorea `sisd00` makinan egikaritzea. Beraz, irakaskuntz-laborategiko makinatik *telnet* bat egin beharko dugu makina honetara, irakasleak emango dizun lan-kontua erabiliz.

Lanerako behar diren fitxategiak `"/users/alumnos/soft/acaz/ether/eusk"` katalogoan dituzue. Hauek zure laneko direktoria kopiatu beharko dituzu.

Ikaslearen katalogoan dagoen `'protokolo.c'` fitxategian gorde behar da programa (1. eranskinean ikus dezakezu honen edukia). Katalogo horretan dauden beste fitxategiak ondoko hauek dira:

`'configuracion'`: Sare simulatuaren konfigurazioa

`'nagusi.o'`: Sistemaren monitorea

`'laguntza.o'`: Ikasleari laguntzeko interfazeak

`'parametros.h'`: Sistemaren konstanteak

`'e1','e2', 'e3','s1','s2', 's3'`: Sarearen nodoei dagozkien sarrera eta irteerako fitxategiak.

OHARRA: Hauek konfigurazio konkretu bateko fitxategiak dira. Orokorrean, 3.2 eta 3.4 puntuetan azaltzen den bezala, `'configuracion'` fitxategian dagoen sarearen definizioaren arabera aldatzeko aukera izango dugu.

`'konp'`: Konpilatzeko komandoa

`'erak'`: Exekutagarria, erakustaldi batekin

'IRAKURRI': Zure lana errazteko oso komenigarria izango da hemen datozen azalpenak kontuan hartzea. Honen edukia dokumentu honen amaieran, 2. eranskinean, aurki dezakezu laborategira joan baino lehen landu ahal izateko.

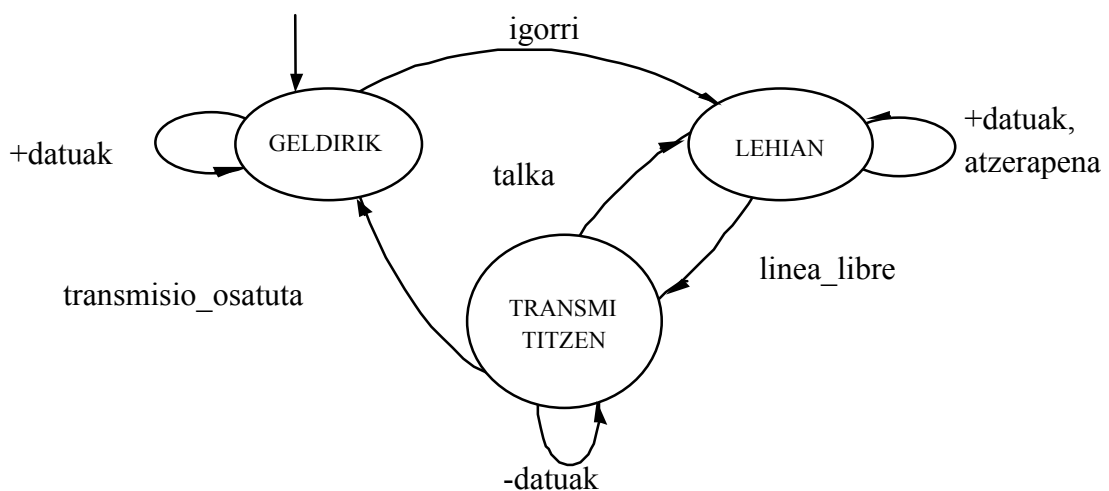
Lanari ekin baino lehen, ikasleak dagoeneko prest dagoen erakustaldia egikaritzea gomendatzen da. Horrela hobeto ulertuko da simuladorearen betebeharra eta, beraz, egin beharreko lana. Horretarako `'erak'` fitxategia deitu besterik ez da egin behar.

3. Sistemaren deskribapena

3.1 Protokoloa eta egoera-makina

Ikasleak inplementatu behar duen protokoloa IEEE 802.3 estandarrean deskribatzen den transmisio-bidea atzitzeko era da. Estandarraren beste aldeak (tramaren formatua edo CRC kontrola, adibidez) ez dira landuko.

Ikasleak idatzi beharko duen C funtzioa izango da, hain zuzen ere, sarearen nodoek egikaritutako dutena, zerbait egin behar duten bakoitzean, informazioa transmititu edo jasotzeko garaian. Funtzio horren algoritmoa 1. irudian dagoen automatik adierazten duena da.

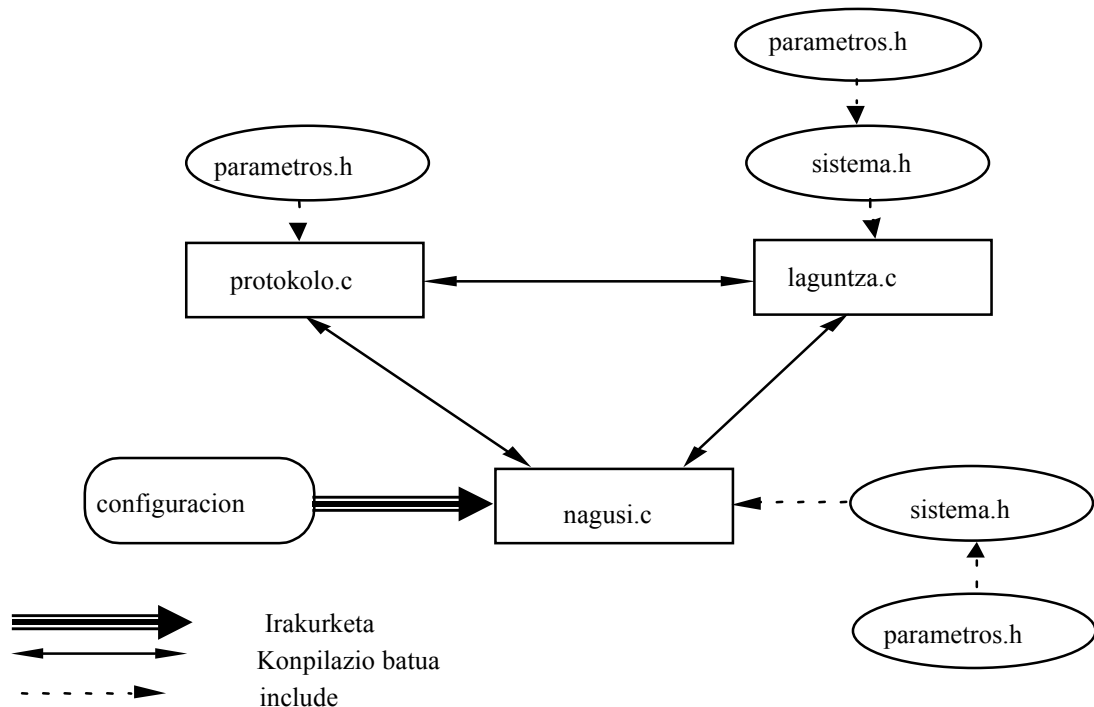


1. Irudia: Ikasleak inplementatu behar duen protokoloa adierazten duen egoera-makina.

GELDIRIK egoeran, nodoak ez du ezer igortzeko, baina linean zehar doana jasoko du. Datuak igortzeko eskaera dagoenean, LEHIAN egoerara pasatzen da, eta hor egongo da linea hutsik gelditu arte. Bitartean linean zehar doan trama guztiak jaso beharko dira. Linea askatzen denean, TRANSMITITZEN egoerara pasako da. Egoera honetan nodoak transmititzeko duen trama jarriko du linean, transmisioa bukatu arte edo talka antzeman arte. Lehenengo kasuan hasierako egoerara pasatzen da, GELDIRIKera alegia, eta bigarrenetan LEHIAN egoerara itzuliko da, bertan dagokion atzerapena jasanez. Transmisio-erroreei buruz ez dago ardurarik, goragoko mailen lana baita.

3.2 Datu eta fitxategien antolaketa

Simuladorearen egitura ondoko hau da:



2. Irudia: Simuladorearen fitxategi-egitura

Irudian sistemaren hiru zatiak agertzen dira: 'nagusi.c' fitxategian dagoen monitorea (edo kudeatzailea), 'protokolo.c' fitxategian ikasleak programatu behar duen atzipen-protokoloa, eta programazio horretarako behar diren laguntzak, 'laguntza.c' fitxategian. 'parametro.h' eta 'sistema.h' fitxategiek sistemari dagozkion konstante, datu-egiturak eta aldagaien definizioak dituzte. Konfigurazioaren fitxategian sarearen definizioa dago. Monitorea behin bakarrik irakurriko du fitxategi hau, simulazioaren hasieran. Komenigarria da fitxategian dagoen sarearen deskribapena aldatzea, programatutako funtzioa sare ezberdinetan probatzeko.

Goiko maila (ikasleak inplementatuko dituen MAC-zerbitzuen erabiltzailea dena) simulatzeko fitxategi bikoteak erabiltzen dira. Badakigunez, trama bat bidali behar denean, MAC mailak jasoko du goiko mailatik. Trama hau helburuko nodoan jasoko da eta MAC mailak goiko mailari pasako dio. Hau guztia simulatzeko, nodo bakoitzari bi fitxategi esleituko dizkiogu; 'ei' eta 'si', hain zuzen ere, non *i* nodoaren identifikadorea den. 'ei' moduko fitxategietan nodoari dagozkion bidaltzeko tramak egongo dira. 'si' fitxategietan, berriz, dagokion nodoak jasotako tramak aurkitu beharko genituzke simulazioaren amaieran. Tramen egitura ez da IEEE 802.3 estandarrean definitzen dena, askoz sinpleagoa baizik. Edozein testu (guk geuk asmatutakoa) eduki dezakete, dokumentazio honen azken atalean azaltzen diren mugak kontutan izanik.

3.3 Sistemaren monitorea

‘nagusi.c’ fitxategian dagoen funtzioa da. Emandako konfigurazioaren sarea eraikitzen du, denboran zehar nodoen jarduera antolatzen du, eta seinalea hedatzen du linean. Bere lanaren muina eta simulazio osoarena ondoko begizta da:

```
do
{
    for (nodoa=0; nodoa<nodo_kop; nodoa++)
        csmc_cd(nodoa, NODO_ZERR.egoera);
    linea_eguneratu();
}
while (!denak_bukatu_dute());
```

Begizta honetan sarearen nodo guztiek egikaritzen duten ‘csmc/cd’ funtzioa ikasleak programatu behar duena da. Begiztan dagoen beste oinarrizko funtzioa ‘actualizar_linea’ da. Honek hedatzen du seinalea linean, eta talkak suertatuz gero, jakinaraziko du.

3.4 Sistemaren parametrizazioa

Simulatu behar den sarearen definizioa ez dago finkatuta. Definizio hori ‘configuración’ fitxategian egiten da, ondoko egitura jarraituz:

l --> Linearen luzera, metrotan.

n --> Sarearen nodo kopurua.

a1 --> Lehenengo nodok sortzen duen informazio-kopurua segunduko (Mb/s-etan).

b1 --> Nodo honi dagokion sarrerako fitxategiaren izena.

c1 --> " " " irteerako " " "

a2 --> 2. nodok sortzen duen informazio-kopurua segunduko (Mb/s-etan).

b2 --> Nodo honi dagokion sarrerako fitxategiaren izena.

c2 --> " " " irteerako " " "

. . . (a, b eta c-ko lerroak n aldiz agertu beharko dira)

an

bn

cn

Fitxategi honen edukia ikasleak adieraziko du, sare ezberdinak definitu ditzan bere simulazioetarako. Adibidez, nodo-kopurua eta hauen informazio-tasa aldatuz, oso sare lanpetutan CSMA/CD protokoloaren eragozpenak egiaztatu daitezke.

Adibide moduan, emandako ‘configuración’ fitxategiaren edukia ondokoa da:

2500
3
1
e1
s1
1
e2
s2
1
e3
s3

3.5 Interfazeak

‘laguntza.c’ fitxategian dauden funtzioek egiten dute interfazearen lana. Ondoko hiru multzotan sailkatu daitezke:

- Linearekiko interfazea.
- Goiko mailarekiko interfazea
- Sistemarekiko interfazea

Linearekiko interfazea

Bi funtzio daude: bat lineatik igortzeko eta, bestea, lineatik datorrena jasotzeko:

```
char escuchar(int nodo);
```

‘nodo’-z identifikatutako nodoa kokatuta dagoen linearen puntuan dagoen karakterea itzultzen du.

```
void transmitir(int nodo);
```

‘nodo’-ri dagokion igortzeko-bufferrean dagoen hurrengo karakterea hartu eta linean idazten du.

Goiko mailarekiko interfazea

Saretik heldu den informazioa, goiko mailen bidez, erabiltzeari helaraztea eta bidaltzeko dagoen informazioa jasotzea bideratzen dute.

```
void guardar(int nodo, char x);
```

‘x’ karakterea ‘nodo’-ri dagokion jasotze-bufferrean gordetzen du. Trama oso bat noiz jaso den antzematen du eta gordetzen du ‘nodo’-ri dagokion irteerako fitxategian.

```
int emitir(int nodo);
```

Goiko mailari galdetzen dio ea transmititzeko datuak dauden. Baiezkotan, ‘nodo’-ri dagokion sarrerako fitxategitik irakurriko du trama bat eta ‘nodo’-an dagoen igortze-buferrean gordeko du. Kasu honetan egiazko emaitza itzultzen du. Ezezkotan, faltsua itzuliko du.

OHARRA: Bi interfaze hauetako funtzioetan ikus daitezkeenez, simulatutako datu-transmisioa karakterez-karaktere egiten da.

Sistemarekiko interfazea

Funtzio hauen bidez sistemak eskaintzen dituen zerbitzuak atzigarri izango dituzte nodoek, hau da, ikaslearen programan beharko diren zenbait ekintza inplementatzen dituzte. Ondoko zazpi hauek dira:

```
void cambiar_estado(int nodo, int egoera);
```

‘nodo’ nodoa ‘egoera’ egoerara aldatzen du. Egoera posibleak hiru dira (‘parametros.h’-n definituta daude): GELDIRIK, LEHIAN edo TRANSMITITZEN.

```
int transmision_completa(int nodo);
```

‘nodo’-ri dagokion igortze-bufferra hutsik dagoenean, egiazkoa itzultzen du. Horrela trama horren igorpena ongi bukatu dela (talkarik gabe) adierazten du. Bestela faltsua itzultzen du.

```
void vaciar_buffer(int nodo);
```

‘nodo’-ri dagokion jasotze-bufferra berriro hasieratzen du. Talka bat antzematen denean, jasotze-buferrean dagoena baztertzeko erabiliko dugu.

```
int no_bloqueado(int nodo);
```

Funtzio honez bidez, talka bat suertatu eta gero nodo batek itxoin behar duen atzerapena simulatuko dugu. Egiazko balioa ('1') itzuliko du nodoak itxoin behar ez badu transmititu baino lehen. Bestela faltsua ('0') ematen du, eta nodoari dagokion atzerapen-balioari bat kentzen dio.

```
void retardo_aleatorio(int nodo);
```

Ausazko zenbaki oso bat sortzen du, algoritmo bitarra esponentziala jarraituz. Zenbaki hori funtzioa egikaritu duen nodoari esleituko zaio. Nodo honek eguneratuko du balio hau 'no_bloqueado' funtzioari deitzen zaion bakoitzean.

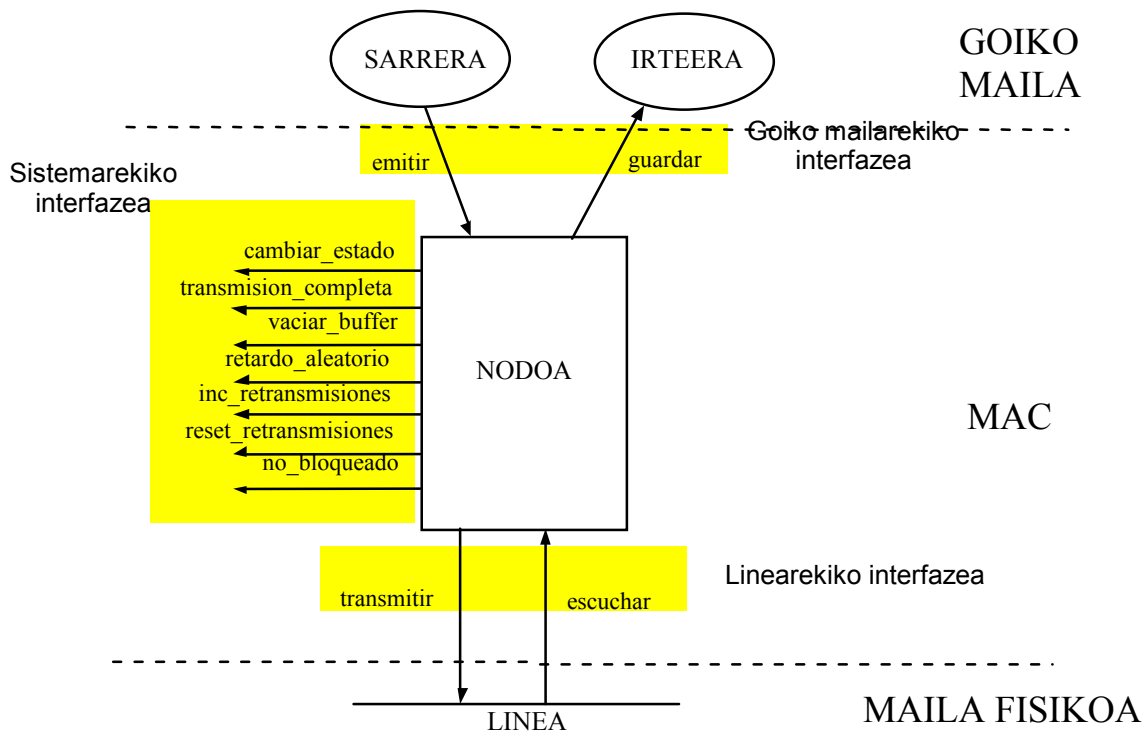
```
void inc_retransmisiones(int nodo);
```

'nodo'-ak egin dituen trama baten birtransmisio kopuruari 1 gehitzen dio.

```
void reset_retransmisiones(int nodo);
```

Trama baten birtransmisio kopuruari zero balioa esleitzen dio.

3. irudiak funtzio hauek eta bere erabilgarritasuna laburtzen ditu.



3. Irudia: Interfazeak eta laguntzeko funtzioak.

4. Egikaripena

Gogoratu gomendagarria dela, lanari ekin baino lehen, dagoeneko prest dagoen erakustaldia egikaritzea. Horretarako 'erak' fitxategia deitu besterik ez da egin behar. Sistemak galdetuko dio ikasleari ea nolako egikaripena nahi duen, ondoko bi aukera hauen artean hauta dezan:

- 1) Kontrolik gabeko simulazioa, non nodoen jardura aurreikustezina den. Aukera honen erabilgarritasuna bukatuta dagoen sistema baten definizioa frogatzea da, egoera ez normalak antzematearren.
- 2) Simulazio kontrolatuta. Honetan zenbaki bat eskatuko zaio erabiltzaileari eta honen arabera simuladoreak portaera jakin bat izango du. Berriz, portaera bera aztertu nahi baldin badugu, zenbaki bera sartu beharko dugu. Normalean bigarren aukera honekin egiten da lan, antzemandako portaera berezien zuzenketa egiaztatzeke balio duen aukera bakarra baita.

Behin simulazio mota bat aukeratuta, pantailan agertuko dira linea eta berari lotuta dauden nodoak, 0 zenbakitik aurrera (ezkerretatik eskuinetara) identifikatuta. Nodo bakoitzari lotzen zaio letra bat, zer egoeratan dagoen adieraziz (R=GELDIRIK, T=TRANSMITITZEN, C=LEHIAN). Sarearen errepresentazio honetan adieraziko dira nodoek igorritako tramak, haien hedapena linean zehar eta sortzen diren talkak.

Behin simulazioa martxan jarrita, nodo bat egoeraz aldatzen denean simulazioa gelditzen da. Horrela ikasleak jarrai ditzake sarearen gertaerak, pausuz-pausu. *Return* sakatuz jarraitzen du simulazioak.

Pantailan agertzen diren karaktereak nodoek bidaltzen dituzten tramenak dira. Gainera, beste karaktere bereziak ere agertzen dira. Nodoei dagozkien fitxategietan egon behar ez diren karaktere horiek, ondokoak dira:

'.'	-->	linea librea
'@'	-->	talka (<i>jamming</i> seinalea)
'#'	-->	tramaren betegarria
'~'	-->	sarrerako fitxategien lerroko amaiera-markaren ordezkia
'\0'	-->	trama-bukaeraren marka

Tramak duten testua irakurtzeko, kontuan hartu behar da seinalearen hedapenaren noratza: eskuinetik ezkerreara ez dago arazorik, baina kontrako norantzan hizkiak idazteko erabiltzen dugun kontrako norantzan ere agertuko dira.

Egikaripenaren bukaeran, nodo bakoitzari dagokion irteerako fitxategian nodoak lineatik jaso dituen tramak agertu beharko dira. **Komenigarria da egikaritu baino lehen fitxategi horiek garbitzea;** bestela aurreko simulazioetan zehar jasotakoa pilatzen da fitxategietan, azken simulazioaren emaitza aztergaitz bihurtuz.

OHARRA: Egikaripen batzuetan, simulazioan zehar, *Return* tekla sakatu eta gero ematen du ez duela aurrera jarraitzen edo ordenadoreak ez duela erantzuten. Ez larritu. Ez dago arazorik. CSMA-CD protokoloak azartzen duenez, talkak dauden bitartean, berriro transmititzen saiatzeko itxaron behar den denbora gero eta handiagoa izango da. Beraz, noizbait hau handia izan daiteke, eta itxarotea tokatuko zaigu.

Ikasleak bere funtzioa programatu duenean, 'konp' komandoa egikaritu beharko du. Honek 'protokolo.c' fitxategia hartuko du eta sistemarekin batera konpilatuko du. Errorerik ez badago, has daiteke simulazioa, konpiladoreak sortutako 'main' fitxategi exekutagarriaren bidez ('main' horretan grabatzen da konpilazioaren emaitza).

1. Eranskina: “protokolo.c” fitxategiaren edukina

```
#include "parametros.h"
void csma_cd(int nodoa, int egoera)
{
    char x;
    if (egoera==GELDIRIK)
    {
    }
    else if (egoera==LEHIAN)
    {
    }
    else if (egoera==TRANSMITITZEN)
    {
    }
}
```

2. Eranskina: “IRAKURRI” fitxategiaren edukina

Programatzerakoan ondo etorriko zaizue ondokoak kontutan hartzea:

- 0) Ez egin inongo begiztarik 'csma_cd' funtzioan. Pentsa ezazue simuladoreak, nodo guztien aritzea une berdinean egiten dela antzemateko, denbora une txikietan zatitzen du, eta horrelako une txiki bakoitzean deitzen du zuen csma_cd funtzioa. Unetxo horien balioa karakter bat transmititzeko denbora denez, zuen funtzioa egikaritzen den bakoitzean karakter bakarra bidal edo irakur daiteke.
- 1) Horregatik, protokoloaren egikaripen batek egoera bakar batean zehar iraundera du, eta bukaeran, behar bada, egoeraz aldatuko du.
- 2) Sistemak kudeatzen du igortzeko bufferra, eta hustu egiten du trama bat inongo arazorik gabe (talkarik gabe) transmititzen denean. Beraz, zuek ez zarete horretaz arduratu behar.
- 3) Antzekoa gertatzen da jasotzekoarekin: guk karaktereka jasoko dugu, eta sistema arduratuko da noiz jaso dugun trama osoa antzemateaz eta trama hori nodoari dagokion irteera-fitxategian grabatzeaz.
- 4) Linean(pantailan) agertzen diren '!' karaktereak tramen hasierako eta bukaerako sinkronizaziorako markak dira. Sistemak sartzen ditu, sare-txartelarena eginez.
- 5) Linea libre dagoen ala talka dagoen jakin arren, linean dagoen karakterea irakurri behar da. '@' baldin bada, talka dago; '-' bada, libre dugu.
- 6) Programa guztiz zuzena noiz den jakiteko biderik ez dago. Egikaripen batzuk egitea eta nodoen joera aztertzea, ea zuzena den ikusteko, besterik ez dago. Komenigarria da kontrolik gabeko egikaripen batzuk egitea (0 aukera), eta konfigurazio-aldaketaren bat egitea