

Anàlisi i estudi de Lightning Network a Bitcoin

Miriam Houlail Sairouni

Resum– Actualment, la xarxa Lightning o Lightning Network està sent molt coneguda en el món de la informàtica i en el món de les criptomonedes, ja que aquesta xarxa té com a objectiu solucionar algunes de les mancances que té Bitcoin. Algunes de les mancances importants que pateix Bitcoin avui en dia és l'escalabilitat, els pagaments instantanis i els micropagaments. Per aquest motiu es farà un estudi i anàlisi de la xarxa Lightning a Bitcoin. Principalment el que s'ha fet és agafar un fitxer json de la xarxa Lightning implementada en LND per poder extreure les dades necessàries d'aquest i fer el graf. Quan ja s'ha tingut el graf implementat en Python, s'ha programat les funcions del graf per fer l'estudi estàtic de les dades. Una vegada que s'ha pogut treure resultat d'aquest estudi, s'ha volgut representar aquestes dades en una pàgina web mitjançant gràfics realitzats amb JavaScript. La pàgina web conté informació del graf com la densitat i altres propietats, informació dels nodes i de les arestes.

Paraules clau– BTC, xarxa Lightning, entrada, sortida, NetworkX, morris.js, HTML5, CSS, JavaScript, cadena de blocs, locktime, P2P, mili-satoshi, satoshi, CLTV, HTLC, Hashlock

Abstract– Nowadays the Lightning network is very well known in the computer world and in the world of crypto-currencies, since this network aims to solve some of the deficiencies that Bitcoin has. Some of the major deficiencies that Bitcoin suffers today are scalability, instant payments and micro-payments. For this reason, we will study and analyze the Lightning Network in Bitcoin. Mainly what has been done is to take a json file from the Lightning Network implemented in LND in order to extract the necessary data from it and make the graph. Once the implementation of the network in Python has been done, the network functions have been programmed to make a static study of the data. Once the results of this study have been obtained, we wanted to represent this data in a web page using graphics made with JavaScript. The web page contains graph information such as density and other properties, node and edges information.

Keywords– BTC, Lightning Network, input, output, NetworkX, morris.js, HTML5, CSS, JavaScript, blockchain, locktime, P2P, milli-satoshi, satoshi, CLTV, HTLC, Hashlock



1 INTRODUCTION

BITCOIN (BTC) és una moneda virtual descentralitzada i independent que ha revolucionat la indústria tecnològica amb la creació de la cadena de blocs. L'any 2009 es va registrar el primer bloc, creant així Bitcoin. Aquesta xarxa va evolucionar molt ràpidament arribant a tenir un valor de bilions de dòlars. Periòdicament es poden acceptar Bitcoins com a forma de pagament, perquè és una manera de pagar segura, ja que utilitza algorismes de

xifratge de clau pública.

Un altre tema que és molt important en el mercat de les criptomonedes és l'escalabilitat dels sistemes. Aquest problema de l'escalabilitat dels sistemes és el límit de processament de transaccions de la xarxa Bitcoin, que es deu a la mida del bloc actual d'un megabyte, que genera un coll d'ampolla per Bitcoin. Per tant, si es supera el límit del bloc, no es podria realitzar la transacció [1]. Moltes criptomonedes importants com Bitcoin o Ethereum pateixen d'aquest problema. La xarxa Lightning té com a finalitat resoldre el problema de l'escalabilitat a Bitcoin.

Per poder emmagatzemar aquestes criptomonedes, s'utilitza una cadena de blocs, blockchain, on es guarden els registres que estan enllaçats, i en alguns casos, els registres estan xifrats per protegir la seguretat i la integritat de la dada que ha sigut publicada. La xarxa Lightning o Lightning

- E-mail de contacte: miriam.houlail@e-campus.uab.cat
- Menció realitzada: Enginyeria de Tecnologies de la Informació
- Treball tutoritzat per: Guillermo Navarro Arribas (dEIC)
- Curs 2019/20

Network utilitza el protocol de contracte intel·ligent de cadena de blocs. El contracte intel·ligent és un acord que té la capacitat de complir-se de forma automàtica una vegada que les dues parts hagin acordat els terminis. Són un tipus de programari per portar a terme una tasca o conjunt de tasques determinades d'acord amb les instruccions introduïdes prèviament. El seu compliment no està subjecte a la interpretació de cap de les parts, és a dir, si una de les condicions succeeix, llavors la conseqüència es posarà en marxa de forma automàtica.

Lightning Network és una xarxa descentralitzada construïda sobre Bitcoin que permet realitzar pagaments instantanis a través d'una xarxa P2P. Lightning proposa que l'escriptura sobre la cadena de blocs s'utilitzi exclusivament per gestionar l'obertura i tancament de canals de pagaments bidireccionals. Els canals de pagaments és un mitjà de transacció fora de la cadena de blocs, en què dues persones es comprometen i es paguen entre elles, evitant haver d'esperar per confirmacions de la cadena de blocs subjacent.

La xarxa escala i accelera la cadena de blocs, resolent algunes limitacions que es troben actualment a Bitcoin. A les transaccions s'emmagatzema només les dades que són estrictament necessàries. Aquestes dades són les següents que es comentaran:

- TXID
- Entrades
- Sortides
- Mida
- Locktime
- Temps rebut
- Bloc inclòs

TXID és l'identificador únic de transacció amb una funció criptogràfica SHA256D utilitzada per identificar de forma exclusiva una transacció. L'entrada és la direcció des de la qual s'està emetent la transacció. Les sortides són els scriptPubKey o locking script de la quantitat transaccionada. La mida és quant ocupa en Bytes la transacció. El locktime és el temps o nombre de bloc en el qual es pot afegir la transacció a la cadena de blocs. Temps rebut és el temps en el qual es rep la transacció. El bloc inclòs és el bloc on es registra la transacció de Bitcoin.

El locking script o scriptPubKey és un script que determina la condició necessària perquè la sortida es pugui utilitzar com a entrada en una altra transacció. Per validar la transacció, el scriptSig de l'entrada ha de complir la condició necessària.

La xarxa Lightning a Bitcoin, com bé ja s'ha comentat, es basa en la tecnologia de cadena de blocs de Bitcoin, que realitza transaccions de micropagaments d'igual a igual fora de la cadena. Això trenca per complet el coll d'ampolla de la capacitat de la xarxa de Bitcoin actual amb el nombre de transaccions, i eliminarà perfectament el retard de propagació i millorarà la privacitat [2].

Avui en dia hi ha moltes aplicacions que tinguin una estreta relació amb Bitcoin, un exemple és l'aplicació anomenada Fold Kickbacks que serveix per realitzar pagaments a Bitcoin. Aquest programari és compatible

amb la xarxa Lightning de segona capa de Bitcoin, el que permet comprar targetes de regal per Bitcoin amb un reemborsament del 3%. Ara la gran empresa Airbnb que realitza intercanvis de cases, permet reservar amb Bitcoin i la xarxa Lightning a través de l'aplicació Fold. El programa també és compatible amb Uber, Starbucks i Amazon [3].

L'aplicació només funciona a països determinats com els Estats Units, Canadà, Austràlia, Irlanda, Mèxic i el Regne Unit. La firma espera desplegar serveis a Europa en un futur.

La motivació d'aquest treball de final de grau és fer un estudi i un anàlisi d'aquesta xarxa Lightning que proporciona i facilita les transaccions de Bitcoin. L'estudi no només se centrarà en les dades del graf en general, sinó que també es fa un estudi exhaustiu dels nodes i de les arestes del graf fet en Python. Bitcoin ha arribat a ser una criptomoneda molt important fins al moment. I per aquest motiu, és un tema molt interessant a estudiar. Per la realització d'aquest estudi s'ha fet servir un fitxer json facilitat pel tutor del treball de final de grau on es pot extreure molta informació rellevant per a la construcció del graf amb la llibreria de Python anomenada NetworkX. Les implementacions principals de la xarxa Lightning són les següents:

- LND
- C-Lightning
- Lit
- Eclair

D'aquestes implementacions principals de Lightning, la que ha tingut més èxit i que més recursos té disponibles per a desenvolupadors és LND. Per aquest motiu, és la implementació més utilitzada actualment. Ara a continuació, es mostrarà un exemple de la xarxa de canals de pagament o xarxa Lightning mitjançant la figura 1.

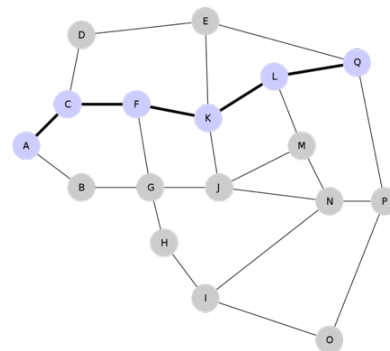


Fig. 1: Exemple de la xarxa Lightning

La implementació de la xarxa Lightning que s'ha utilitzat per fer l'estudi, ha sigut feta amb LND, que és una de les implementacions que té més informació, codi i documentació. En el fitxer està la implementació del graf amb els seus nodes i les seves arestes a partir de la informació que es treu de la xarxa Lightning. Tan aviat com es tingui la implementació del graf, s'han programat funcions a partir de la llibreria NetworkX de Python per estudiar grafos i

xarxes.

Durant la realització del projecte, s'han trobat algunes complicacions a l'hora d'implementar o programar alguna funció per fer l'estudi de la xarxa Lightning o durant la creació del graf, ja que anteriorment no es dominava gaire el llenguatge de programació Python. També s'han trobat algunes dificultats a l'hora de representar algunes de les dades extretes de les funcions. El problema de visualitzar la informació del projecte en gràfics, era que es treballava amb gran quantitat de dades.

El document queda estructurat de la següent manera, a la secció 2 s'explica l'estat de l'art on es posa en context el projecte, a la secció 3 es comenten els objectius de la realització d'aquest treball de final de grau durant la primera fase, a continuació, a la secció 4 es parla sobre la metodologia i planificació del projecte. A la secció 5 hi ha tot el desenvolupament inicial del projecte i el procés del desenvolupament final de projecte, a la secció 6 es comenta els resultats obtinguts i el perquè d'aquests resultats, la secció 7 es parla sobre l'experiència d'aquest projecte i quines dificultats s'han trobat. Finalment a la secció 8 es troben les conclusions del treball i per últim una secció d'agraïments i de referències bibliogràfiques.

2 ESTAT DE L'ART

En l'actualitat, es troben molts treballs o estudis de la xarxa Lightning. La majoria de gent que fa l'anàlisi de la Lightning Network, ho enfoquen d'una manera diferent. En el cas d'aquest projecte, no s'ha intentat millorar la xarxa Lightning, sinó que s'ha fet un estudi per saber com funciona aquesta xarxa. Normalment en aquests tipus de projectes, l'estudi se centra en els nodes de la xarxa Lightning. Ara a continuació, es comentaran alguns dels estudis que s'han realitzat de la xarxa Lightning.

Un dels estudis fets de Lightning, s'ha utilitzat la implementació Eclair. En aquest anàlisi, s'ha representat la ubicació geogràfica dels nodes en un mapamundi basant-se en la IP pública del node i els canals. Cada node té associat com a mínim 1 canal i com a màxim 1.162 canals [4].

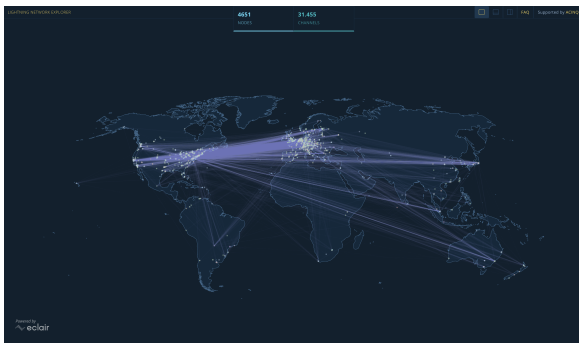


Fig. 2: Exemple de l'estudi Lightning Network

L'altre tipus d'estudi que es fa bastant és la representació de la informació de la xarxa Lightning, en taules on s'especifica el nombre de nodes, el nombre de canals, la capacitat de la xarxa i altre tipus d'estudis relacionats amb els nodes [5].

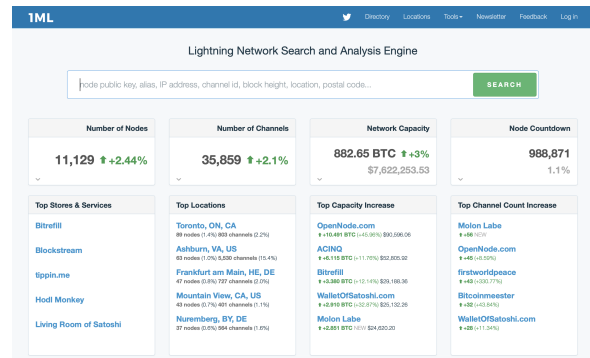


Fig. 3: Exemple de l'estudi Lightning Network

En canvi, en el cas d'aquest projecte, per poder realitzar aquest estudi, s'ha hagut d'aprendre a fer gràfics, a saber quins tipus de dades tenen els gràfics, quins tipus d'estudis es poden fer, ja que n'hi ha dos tipus d'anàlisi de dades:

- Anàlisi de dades estàtiques
- Anàlisi de dades dinàmiques

S'ha fet un estudi de dades estàtiques, això vol dir que a partir d'un fitxer json de la xarxa Lightning, s'ha fet l'anàlisi amb dades que tenen una mateixa mida i forma, que és constant durant l'execució del fitxer.

Les dades que s'han pogut treure de la programació de les funcions amb la llibreria de NetworkX, s'han volgut representar les dades a una pàgina web i saber què signifiquen aquestes dades. Sobretot s'ha investigat quins valors són per defecte a la xarxa Lightning.

3 OBJECTIUS

L'objectiu principal d'aquest Treball de Final de Grau és poder fer un anàlisi i un estudi de la xarxa Lightning a Bitcoin. I per poder fer aquest estudi de la xarxa, s'han de complir una sèrie d'objectius que s'han dut a terme durant el projecte, que seran comentats en format de llista:

- Estudi de què és i com funciona la xarxa Lightning
- Estudi del fitxer json de la xarxa Lightning
- Programació del graf a partir del fitxer json
- Programació de les funcions per treure dades de la xarxa amb la llibreria NetworkX
- Anàlisi de les dades estàtiques
- Programació de la pàgina web per representar l'estudi realitzat

Principalment el que es farà és agafar un fitxer en format json amb les implementacions de Lightning, que en aquest cas, són LND. A partir d'aquest fitxer que té la xarxa Lightning implementada, es construirà el graf en Python amb la llibreria NetworkX. Abans de procedir a la implementació del graf, s'ha d'haver fet una funció a Python per poder llegir el fitxer json. Quan s'ha llegit el fitxer json, es programa el graf a Python.

A continuació, es realitzaran funcions amb NetworkX, que

és el llenguatge Python. Es poden preparar les funcions per fer l'estudi i l'anàlisi del graf amb funcions que ja vénen en el paquet de NetworkX.

Aleshores, s'agafaran les funcions del paquet, el que es pretén amb això és poder posar l'atenció en l'anàlisi a realitzar, perquè serà l'objectiu final d'aquest treball. L'atenció final recau sobre les dades analitzades o estudiades. Aquest tipus de dades seran estàtiques, que finalment seran representades a una pàgina web.

4 METODOLOGIA I PLANIFICACIÓ

Durant tota la realització del projecte i compliment dels objectius establerts, s'ha estat utilitzant la metodologia Kanban. Kanban és una metodologia àgil que té com a objectiu gestionar de manera general com es van completant les tasques del projecte. S'ha utilitzat aquesta metodologia, perquè permet visualitzar el flux del projecte i visualitzar les fases per poder prioritzar les tasques. És tot un avantatge, ja que se sap en tot moment en quin punt del projecte s'ha arribat, com es va de temps i com es poden organitzar les següents tasques a realitzar. La metodologia Kanban s'utilitza en projectes grans i amb moltes persones, en aquest cas no ha sigut així, amb la finalitat de fer un projecte individual. Les úniques reunions que s'han dut a terme, han sigut amb el tutor per informar-lo com està anat el treball de final de grau, quins problemes s'han trobat i com s'han pogut solucionar aquests problemes.

La planificació prevista de les tasques del treball de final de grau estan fetes amb el programari Atlassian i s'ha fet el seguiment amb Jira i amb Trello. Per poder fer servir aquestes eines, no ha suposat ser molt complicat, ja que aquesta eina va ser utilitzada en una assignatura anomenada Enginyeria del Software, pel qual ja es tenia coneixement previ de com funcionava aquest programa.

Es mostra el diagrama de Gantt del treball de final de grau a la figura 4, on es poden veure totes les tasques del projecte, quan comencen i quan finalitzen. Entretant es realitzava el treball de final de grau, no han sorgit imprevistos que facin canviar la planificació inicial, perquè s'ha fet una bona programació del projecte.

Una de les tasques que han sigut planificades des d'un principi per realitzar durant l'última etapa del projecte, s'ha avançat, a causa del fet que la realització de la pàgina web necessitava molt de temps per la seva execució. S'havia de buscar una llibreria JavaScript que permetés fer gràfics a una pàgina web. Un altre tema molt important per haver avançat el treball de la pàgina és que s'havia de treballar amb moltes dades per la quantitat de nodes i arestes que tenia el graf de la xarxa Lightning.

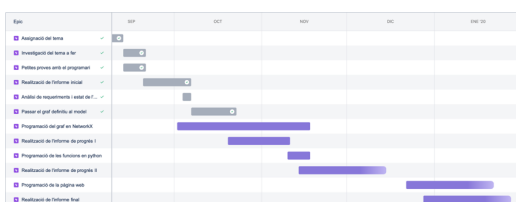


Fig. 4: Diagrama de Gantt del projecte

5 DESENVOLUPAMENT

Durant tot el procés del projecte, s'ha portat a terme un desenvolupament del treball de final de grau des d'un inici fins a la seva finalització.

El desenvolupament es divideix en dues subseccions que es comentaran a continuació:

- Desenvolupament del graf i de les seves funcions
- Desenvolupament de la pàgina web

Ara es començarà amb l'explicació de la primera subsecció del desenvolupament i la part més important, ja que a partir d'aquesta implementació, s'han pogut obtenir algunes de les dades del projecte.

5.1 Desenvolupament del graf i de les seves funcions per fer l'anàlisi de dades

Inicialment s'ha començat realitzant un estudi de què és la xarxa Lightning, com funciona aquesta xarxa, quins tipus d'implementacions principals té la xarxa, com es programa un graf en Python, què és i per què serveix la llibreria NetworkX. Es volia saber les coses bàsiques per poder començar a fer petites proves amb el programari. Les primeres proves realitzades, no s'han elaborat amb complexitat perquè no es tenia coneixement previ d'aquesta llibreria, ni tampoc de la realització de grafs en Python. NetworkX és una llibreria de Python que serveix per a la creació, manipulació i estudi de l'estructura, dinàmica i funcions de xarxes complexes.

Quan s'han fet diverses proves amb aquest llenguatge, es va estar treballant amb el fitxer json. Es volia utilitzar aquest json per poder fer el graf en NetworkX. Hi ha moltes formes de saber la informació del fitxer json. Una de les opcions per poder saber aquesta informació és escriure codi en un projecte, fer-ho des de la terminal o buscar per Internet un programari editor de codi en línia per executar el fitxer json. En aquest cas s'han utilitzat les tres maneres per saber el nombre de nodes i arestes, ja que es volia corroborar que tot coincidia.

Quan ja se sabia més o menys com funcionava NetworkX i Matplotlib, es va posar en marxa la programació del graf amb el qual s'ha realitzat l'estudi i l'anàlisi.

Matplotlib és una llibreria de programari per a generar gràfiques a partir de dades que estan guardades en llistes o vectors en el llenguatge de programació Python.

Al projecte es van importar les llibreries NetworkX i la llibreria Matplotlib. A més es va crear el multigraf vuit, per després, afegir-li els nodes i les arestes. Aquest graf creat no és un digraf, simplement és la representació abstracta d'un multigraf que es pretén implementar per fer un estudi de la xarxa Lightning.

Un digraf és un graf dirigit, és a dir, les arestes del graf tenen una direcció assignada, en canvi un multigraf és un graf que pot tenir arestes múltiples, és a dir, són arestes que relacionen els mateixos nodes. Les arestes no tenen una direcció assignada, per tant, és un graf no dirigit.

Per poder afegir els nodes, s'ha de realitzar una sèrie d'iteracions mitjançant una estructura que s'utilitza en format de llista d'aquests nodes a partir del fitxer json. Per poder fer el que es comenta, s'ha fet una funció que

s'anomena `construccioMultiGraf` on es du a terme la creació d'aquest graf.

Com ja s'ha comentat anteriorment, es crea una variable anomenada `G` on es guarda un multigraf buit. A continuació, es llegeix el fitxer de dades fent servir el mètode `load`. Aquest mètode llegeix la cadena de caràcters del fitxer per analitzar sintàcticament les dades del json. Quan ja es té la informació del fitxer llegida, el que es fa és guardar aquestes dades a la variable `data`.

Seguidament es fa la iteració de la variable `data`, per afegir els nodes. Els nodes s'afegeixen amb el mètode `add_node`, amb el qual es pot addicionar els identificadors, els àlies, les adreces i les últimes actualitzacions dels nodes. És molt important que cada node tingui aquest tipus d'informació, ja que després, es farà servir per tractar-la durant l'anàlisi.

Posteriorment es procedeix a programar les arestes del multigraf. Per poder acomplir aquest fet, s'ha de realitzar una sèrie d'iteracions mitjançant una estructura que s'utilitza en format de llista com els nodes. A fi que es té la informació necessària del fitxer json, per a la construcció del graf amb els seus nodes i arestes, ja no és tan complicat com fer-ho manualment. La informació utilitzada a l'hora d'afegir les arestes al multigraf buit, han sigut els dos identificadors de les arestes, la capacitat, les últimes actualitzacions, les polítiques de cada un dels identificadors, els identificadors dels canals i el punt de trobada de les arestes. Amb les arestes s'aprofiten les dades que s'insereixen per analitzar-les i estudiar-les.

Els identificadors són per referenciar a les arestes d'una forma única sense repetir. La capacitat d'un canal és la quantitat d'informació que pot ser transmesa sobre canals de comunicacions. Per tant cada aresta, té un pes per determinar la capacitat del canal. Les últimes actualitzacions són un procés per millorar les arestes o realitzar qualsevol canvi. Les polítiques són les restriccions que es posa a les arestes. En aquest cas hi ha dues polítiques, ja que les arestes tenen dos extrems, llavors a cada extrem s'aplica una de les polítiques. Els identificadors de canals són referències per nomenar els canals, ja que es treballen amb molts canals, per tant, hi ha un gran interès que els canals tinguin identificadors per saber per on passar quan es vol realitzar una transacció. El punt de trobada és on s'acabaran trobant les arestes quan es realitzi la transacció. Per últim, quan ja s'ha acabat de realitzar la funció `construccioMultiGraf`, es mostra per pantalla el multigraf fet utilitzant un `print`, que ensenya l'execució del mètode. El graf creat no es veu definit, ja que es treballen amb 4.571 nodes i 33.022 arestes, per tant, la imatge del graf no es veu clarament. En aquest cas no s'ha intentat utilitzar una llibreria externa o programa extern per visualitzar millor el graf, perquè no interessa per l'anàlisi de les dades, simplement el que importa en aquest treball és tenir el graf per després, fer la programació de les funcions que donaran les dades que seran analitzades. A continuació, es visualitza el graf que s'ha programat a la figura 5.



Fig. 5: Programació del graf

Al finalitzar la programació del graf, s'ha posat l'enfocament a investigar quin tipus d'anàlisi fer del projecte, quines funcions implementar per tenir les dades. L'anàlisi de les dades del graf són molt importants perquè és el gran objectiu d'aquest treball de final de grau. Primerament s'ha tingut l'interès de treure les dades del multigraf en general, com per exemple la capacitat mitjana del graf, si el graf és o no és dirigit, el grau mitjà que té el graf, la densitat d'aquest, la centralitat del grau i l'histograma de graus del graf. Totes aquestes primeres funcions calculades han sigut fetes en `NetworkX` a `PyCharm`, s'han fet a partir de la construcció del graf. Moltes d'aquestes funcions per fer l'anàlisi, han sigut agafades de la mateixa llibreria de `NetworkX` així com:

- La densitat
- El grau mitjà
- La centralitat del grau
- El graf és dirigit o no dirigit

Es reutilitza la funció en la qual es crea el graf, els seus nodes i les arestes per fer el càlcul de la capacitat mitjana i de totes les altres propietats del graf.

Ara a continuació, s'explicarà què vol dir la capacitat d'un graf, la densitat, el grau mitjà, la centralitat del grau, què vol dir si el graf és o no és dirigit.

La capacitat d'un graf és la capacitat mitjana que té un canal de pagament definit. La densitat d'un graf és quan el nombre d'arestes del graf és proper al nombre màxim d'arestes. En aquest cas existeixen dues fórmules per calcular la densitat d'un graf, i com el graf és no dirigit, es fa servir aquesta fórmula $d = \frac{2*arestes}{nodes*(nodes-1)}$. El grau d'un node és el nombre d'arestes incidents al node. La centralitat d'un graf és una mesura possible d'un node en el graf, que determina la importància relativa dins d'aquest. El graf, com ja s'ha mencionat anteriorment, és no dirigit.

Per a la programació de la informació dels nodes, s'ha enfocat en saber per exemple les adreces IP dels nodes, i quants d'ells són adreces IP amb el protocol Onion o TOR. La clau pública dels nodes són úniques, pel qual es tenen 4.571 claus públiques diferents i a més, tots els nodes utilitzen el protocol TCP, a part d'alguns nodes que utilitzen el protocol Onion, ja que aquest protocol només accepta connexions TCP. Cal afegir que la xarxa Lightning només fa servir TCP.

Es tenen 4.571 nodes i 3.766 adreces al multigraf. La quantitat d'adreces IP que es tenen són 2.840, de les quals 926 són adreces que utilitzen el protocol Onion. De totes les adreces IP que s'han obtingut, hi ha de dos tipus, les que són d'IPv4 i les que són IPv6. En aquest cas hi ha 2.738 adreces IPv4 i 102 adreces IPv6. Dels nodes, només s'ha pogut fer aquest tipus d'anàlisi, ja que no es podia fer l'estudi de l'àlies perquè és una dada molt irrellevant. Les dades més interessants a estudiar han sigut les que s'han comentat anteriorment i saber d'on provenen les IP, de quin país provenen.

Per a la programació de la informació de les arestes, s'han tingut més dades amb les quals s'han de treballar, ja que les arestes tenen moltes propietats. Les dades en les quals s'ha posat l'enfocament té a veure amb la política que tenen les arestes. Cada aresta té dues polítiques anomenades `node1_policy` i `node2_policy`. Aquestes polítiques tenen dades com `time_lock_delta`, `min_htlc`, `fee_base_msat`, `max_htlc_msat`, `disabled`, `fee_base_milli_msat` i `last_update` de l'aresta. Es té un total de 33.022 arestes, però no totes les arestes tenen `node1_policy` o `node2_policy` amb valors, hi ha algunes polítiques que estan igualades al valor nul, és a dir, són nul·les, pel qual no tenen propietats com `time_lock_delta` ni `last_update` de l'aresta. Llavors en total es tenen 64.847 polítiques amb configuracions. Ara s'explicaran algunes de les configuracions en les quals s'ha enfocat per fer l'estudi i l'anàlisi.

Les arestes tenen dues polítiques perquè les arestes tenen dos extrems, per tant, cada extrem té la seva política i cada política té les seves configuracions. La configuració anomenada `time_lock_delta` és la diferència mínima en el nombre de blocs entre el CLTV d'un HTLC entrant i el HTLC reenviat. El CLTV (CheckLockTimeVerify) permet als usuaris crear una transacció Bitcoin en el qual les sortides de la transacció queden activades en un futur i no en el moment de realitzar la transacció. Per determinar aquest temps en un futur, els bitcoins transferits queden bloquejats en el temps amb una data específica o fins que s'hagin minat certa quantitat de blocs. L'HTLC (Hash Time Locked Contracts) és una classe de pagaments que fan servir hashblocks i timelocks per exigir que el receptor d'un pagament reconegui haver rebut el pagament abans de la data límit en generar un comprovant de pagament criptogràfic o el receptor perdrà la capacitat de reclamar el pagament, tornant el pagament a l'emissor. Els HTLC es fan servir per fer pagaments multicanal. El Hashlock és un tipus de càrrega que restringeix la despesa d'una sortida fins que es revela públicament la preimatge d'un hash. Una vegada obert públicament un hashlock, també es pot obrir un altre hashlock assegurat amb la mateixa clau [6]. Un Timelock és un tipus de contracte intel·ligent primitiu que restringeix la despesa d'alguns Bitcoins fins a un temps futur o una altura de bloc especificada.

La configuració anomenada `min_htlc` és el mínim temps en el qual es farà un pagament utilitzant hashblocks i timelocks per exigir que el receptor d'un pagament hagi rebut el pagament abans d'un termini. La propietat `max_htlc_msat` és el màxim temps en el qual es farà un pagament utilitzant hashblocks i timelocks per exigir que el receptor d'un pagament hagi rebut el pagament de les tarifes d'encaminament pagades abans d'un termini. La propietat `fee_base_msat` és la tarifa base cobrada independentment del nombre de

mili-satoshis enviats. Aquesta tarifa base és cobrada pel propietari del canal, ja que permet l'utilització d'aquest canal. Un satoshi és la unitat més petita de Bitcoin, però a la xarxa Lightning es poden transaccionar amb unitats encara més petites mentre els canals estan oberts. L'import s'arrodoneix fins al satoshi més proper quan es tanca el canal i es transmet a la cadena de blocs per adherir-se al límit de bitcoins. Llavors un mili-satoshi és una mil·lèsima d'un satoshi. La configuració anomenada `fee_base_milli_msat` és el tipus de comissió efectiu en mili-satoshis. La precisió d'aquest valor puja fins a 6 dígits decimals. La configuració `disabled` és quan un canal està desactivat, pel qual no es pot realitzar una transacció per aquell canal. I per últim les arestes tenen una configuració anomenada `last_update` on es posa quan va ser l'última vegada que s'ha actualitzat aquella aresta.

I fins aquí s'ha fet el desenvolupament del graf i de les funcions d'aquest graf per poder realitzar l'estudi de les dades. Aquest estudi ha sigut llarg perquè es treballava amb gran quantitat de nodes i gran quantitat d'arestes, on cada aresta tenia dues polítiques amb les seves configuracions.

S'ha finalitzat l'explicació del primer desenvolupament, ara es passarà a explicar el segon tipus de desenvolupament fet durant aquest projecte que consisteix en la realització de la pàgina web.

5.2 Desenvolupament de la pàgina web

Per poder realitzar aquest desenvolupament, prèviament s'ha d'haver fet el desenvolupament del graf i les funcions del graf per fer l'estudi de les dades. Abans de començar a fer la pàgina web, s'ha fet un estudi previ de quines dades analitzades es volien mostrar, com es volien mostrar aquestes dades, com seria l'estructura de la pàgina web i com organitzar el conjunt d'informació que es visualitzarà. Quan ja s'ha pensat sobre l'estructuració de la pàgina web, es va començar a realitzar l'anàlisi de les dades. La majoria d'anàlisi sobre les arestes del graf, s'han fet funcions en Python, però per treballar amb els valors d'algunes configuracions, s'ha fet manualment a partir del fitxer json, ja que no es podia realitzar d'una altra manera o no s'ha trobat una altra manera de fer-ho.

S'ha realitzat un prototipus per poder portar a terme el desenvolupament de la pàgina web i tenir una idea precedent de com començar amb l'última part del treball de final de grau. La iniciativa principal de com seria la pàgina web, era dividir la informació de la xarxa Lightning en 3 parts. La primera part seria dedicada a la informació del graf. La segona part de la informació que s'anava a mostrar seria dels nodes. I l'última part i no la menys important, es volia mostrar la informació de les arestes. L'especulació de la primera idea que es tenia per la realització d'aquesta implementació, no ha variat durant el desenvolupament de la pàgina web, per tant, s'ha mantingut la idea inicial que es va tenir. Ara a continuació, es mostrarà un dels prototipus fets a la figura 6.



Fig. 6: Prototipus disseny web

Quan ja s'havia realitzat el prototipus, es va començar a pensar en el llenguatge que s'utilitzaria per fer la pàgina web i com fer els gràfics a partir de l'anàlisi. La pàgina web és estàtica, només és una pàgina web informativa, i quan es volen veure els valors de l'anàlisi, les dades són variants. El llenguatge que s'ha fet servir pel desenvolupament, ha sigut HTML5, CSS i JavaScript. S'ha utilitzat JavaScript per poder fer els gràfics en format web amb la llibreria morris.js. Morris.js és un programari proporcionat per poder realitzar gràfics en pàgines web. Els tipus de gràfics que es poden fer són els següents:

- Diagrama de barres
- Diagrama de sectors
- Gràfic de línia
- Gràfic comparatiu

Per representar les dades del graf com la capacitat mitjana, la densitat del graf, el grau mitjà i les mitjanes de les configuracions, s'ha fet en petites taules. Cada dada s'ha fet amb una petita taula que té el nom del càlcul o funció realitzada, i el resultat d'aquest càlcul o funció.

Per representar algunes dades dels nodes i dades de les arestes, s'han fet servir gràfics de línia. Les dades que s'han visualitzat amb aquest tipus de gràfic han sigut les que es comentaran seguidament:

- Histograma de graus
- time_lock_delta
- fee_base_msat
- fee_rate_milli_msat

Per a la representació d'algunes de les dades restants de les configuracions de les arestes que es comentaran seguidament, s'ha utilitzat els gràfics de barres o diagrama de barres:

- min_htlc
- max_htlc_msat

En el cas d'interpretar les dades de l'histograma de graus, s'ha posat a l'eix de les x els graus que té la xarxa, que en aquest cas, hi ha la mateixa quantitat de graus que nodes. I

a l'eix de les y s'ha posat el valor que té cada grau del graf. Els graus que estaven igualats a 0, no s'han ficat tots al gràfic, perquè es feia molt repetitiu i no es podien apreciar els altres valors.

Per mostrar cada time_lock_delta de les arestes, s'ha treballat amb 64.847 valors. Per tant es va començar buscant el valor per defecte, que en aquest cas és de 144. Posteriorment s'han anat cercant altres tipus de valors, fins a completar el total de 64.875 valors. A l'eix de les x s'ha posat el time_lock_delta i a l'eix de les y s'ha posat la quantitat de valors dels time_lock_delta. Els valors obtinguts s'han hagut de posar un a un al gràfic, ja que no es permetia ficar les dades a un editor de text i a partir d'aquest editor de text, representar les dades amb la llibreria de JavaScript. No s'ha volgut classificar valors en un determinat rang, ja que molts dels valors que s'han cercat, es repeten, llavors no era necessari fer l'agrupació.

La representació de min_htlc, max_htlc_msat, fee_base_msat i fee_rate_milli_msat s'ha fet el mateix sistema que time_lock_delta. S'ha treballat amb 64.847 valors i es va començar buscant els valors que més es repeteixen a la xarxa Lightning. En iniciar a trobar aquests valors, s'han anat apuntant a un editor de text. Quan ja es tenien tots els valors, es va començar a fer el gràfic en morris.js. A l'eix de les x s'ha posat el min_htlc, max_htlc_msat, fee_base_msat o fee_rate_milli_msat, i a l'eix de les y s'ha posat els valors emmagatzemats a l'editor de text.

En el cas de la representació de les IP dels nodes, s'ha fet servir un gràfic de sectors. D'entrada a l'anterior subsecció s'han obtingut totes les IP dels nodes. Algunes d'aquestes IP eren de tipus IPv4, IPv6 o adreces IP Onion. S'han agafat totes les IP que s'han obtingut i s'ha fet l'anàlisi de l'origen de cada IP que s'ha adquirit. Després, s'ha fet un diagrama de sectors amb els països de les IP. Per fer això, s'ha calculat el percentatge d'IP de cada país.

Per últim, s'ha fet una demostració de les mitjanes de les configuracions de les arestes com time_lock_delta, min_htlc, max_htlc_msat, disabled, no disabled, fee_base_msat i fee_rate_milli_msat de cada política. S'han fet dues taules per mostrar tota aquesta informació a la pàgina web. Les dues taules tenen la mateixa estructura, és a dir, tenen les 7 files de les configuracions i dues columnes. A la primera columna es representa el nom de les configuracions i a la segona columna es mostra el valor de les mitjanes calculades.

S'ha volgut representar les dades com ja s'ha comentat amb anterioritat, perquè es volia visualitzar la informació de forma organitzada i que fos visualment fàcil d'entendre per no trobar problemes de comprensibilitat.

6 RESULTATS

En finalitzar la part de desenvolupament del graf i de les funcions del graf, s'ha procedit a implementar la pàgina web que ha requerit l'anàlisi i l'estudi de les dades de la xarxa Lightning a Bitcoin. En aquesta secció s'avançaran els resultats que s'han obtingut durant l'evolució del treball de final de grau.

Els resultats es divideixen en dues subseccions diferenciades, amb la finalitat de mostrar què s'ha obtingut a ca-

da subsecció durant la realització del projecte, perquè no tot s'ha pogut tenir mentrestant s'implementava la primera part, sinó que s'han anat adquirint tenint necessàriament els dos desenvolupaments del projecte. Les subseccions són les següents:

- Resultats obtinguts durant la implementació del graf i de les seves funcions
- Resultats obtinguts durant la implementació de la pàgina web

Seguidament es començarà explicant la primera subsecció dels resultats.

6.1 Resultats obtinguts durant la implementació del graf i de les seves funcions

Durant la fase del desenvolupament del graf i de les seves funcions en Python, s'han tingut les dades que es comentaran a continuació, en format de llistat:

- La capacitat mitjana
- El grau mitjà
- La densitat
- El graf és o no és dirigit
- El nombre d'adreces IP
- El nombre d'IP Onion
- El nombre d'IPv4 i IPv6
- Les mitjanes de les configuracions de les arestes

La capacitat mitjana del graf dona 3.500,33 BTC o shatoshis arrodonint el valor. Aquest resultat és la mitjana d'un canal de comunicacions definit del graf. Això vol dir que és la mitjana dels valors que tenen cada un dels canals de comunicacions definit del graf. Al graf tenim 33.022 arestes, de les quals algunes d'elles tenen un canal de comunicacions ja definit, per tant quan es calcula la mitjana d'aquests canals de comunicacions, dona un valor de 3.500,33.

El grau mitjà del graf dona un valor de 14,4485. El grau mitjà és el grau que tenen els nodes, per tant, cada node té el seu grau. N'hi ha molts nodes que tenen grau de valor 0 o 1, llavors si es calcula la mitjana, surt un valor bastant coherent, que en aquest cas és de 14,4485. Aquesta coherència es deu al fet que es treballa amb un valor màxim de 1.116 que no es torna a repetir i valor mínim de 0 que es repeteix moltes vegades, llavors quan es calcula la mitjana dels nodes, surt un nombre de 14,45 pel que ja s'ha mencionat.

La densitat del graf és de 0,0032 arrodonint decimals, això és possible, ja que per fer el càlcul de la densitat, s'ha de tenir en compte el nombre de nodes i d'arestes. Com ja s'ha comentat, el nombre de nodes és de 4.571 i el nombre d'arestes és de 33.022, això fa que es faci servir la fórmula de graf no dirigit i doni el resultat de 0,0032.

El graf és no dirigit, per aquest fet les arestes del graf no tenen una direcció associada, llavors les arestes són

bidireccionals.

Els nodes tenen associades unes adreces. No tots els nodes en tenen. Llavors per fer el càlcul del nombre d'adreces que té la xarxa Lightning, es llegeixen a partir del fitxer json. A saber, surten 3.766 adreces tenint 4.571 nodes. S'ha de tenir el coneixement que 1.088 adreces són nul·les, tenen un valor nul, per aquest motiu surten només 3.766 adreces disponibles, de les quals 2.840 adreces són IP i 926 adreces Onion. D'aquestes adreces IP, 2.738 són adreces IPv4 i 102 són adreces IPv6. De les adreces que hi ha que utilitzen Onion, només es pot saber l'origen d'aquesta adreça algunes d'elles, en aquest cas són 165 adreces Onion totals. Les adreces Onion són adreces IP anònimes accessibles per l'intermediari de la xarxa TOR.

Seguidament es mostrarà un gràfic de sectors on es posa el percentatge de l'origen de les adreces Onion, d'on provenen aquestes adreces Onion mitjançant la figura 7.

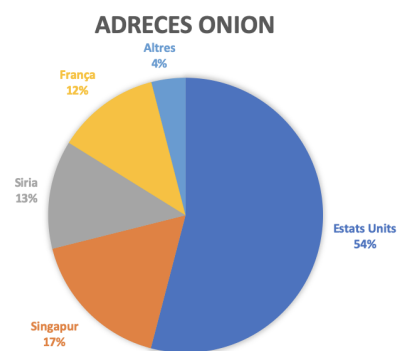


Fig. 7: Gràfic de sectors Onion

Les mitjanes que s'han calculat de les configuracions de les arestes són `time_lock_delta`, `min_htlc`, `fee_base_msat`, `fee_base_milli_msat`, `disabled` i `max_htlc_msat` de cada política, que en aquest cas són dues polítiques de les arestes. Pel càlcul d'aquestes configuracions, el que s'ha fet és agafar la informació de les arestes del fitxer json, s'ha anat llegint una a una les configuracions i s'ha posat un comptador al bucle de les dues polítiques. Més tard s'ha dividit la configuració entre el comptador i dona la mitjana. En el cas de les configuracions `disabled` i `no disabled`, hi ha 25.580 configuracions de tipus `disabled` i 6.842 `no disabled` de la política 1, `node1_policy`. En total, entre les dues polítiques tenim 51.120 valors `disabled` i un total de 13.727 valors `no disabled`.

Ara a continuació, es facilitarà una taula de les mitjanes de la política 1, `node1_policy`, de les arestes que té 32.422 valors.

Nom de la configuració	Mitjana
<code>time_lock_delta</code>	100,89698352970206
<code>min_htlc</code>	1.710,7255258774906
<code>max_htlc_msat</code>	2.510.776.847,8744373
<code>fee_base_msat</code>	1.021,5378138301154
<code>fee_rate_milli_msat</code>	65.086,55665905866
<code>disabled</code>	0,7889704521621121
<code>no disabled</code>	0.21102954783788785

TAULA 1: TAULA MITJANES CONFIGURACIONS

El resultat de la mitjana de la configuració `time_lock_delta` dona un valor de 100,9 arrodonit. Sembla ser un resultat coherent, ja que com ja s'ha comentat amb anterioritat, el valor que es repeteix més i és un valor que ve per defecte, és el 144. I el valor més gran trobat és de 1.000, però no es repeteix tant com el valor per defecte. També hi ha valors petits, per tant, quan es calcula la mitjana surt un resultat tènac.

La mitjana de `min_htlc` dona 1.710,73, perquè no hi ha molta variació de valors diferents com a les altres configuracions, tot i que hi ha valors majors a la configuració `min_htlc` que no pas a `time_lock_delta`. El valor que més es repeteix a aquesta configuració és 1.000, pel que en calcular la mitjana surt un valor gairebé pròxim a aquell valor.

Les configuracions `max_htlc_msat`, `fee_base_msat` i `fee_rate_milli_msat` es treballa amb molta més variació que a les altres dues configuracions anteriors. En aquests casos, hi ha valors molt grans que no es repeteixen tant, però en fer la mitjana, surten resultats majors. Les mitjanes obtingudes tenen sentit pel qual s'ha comentat.

Els resultats de `disabled` i `no_disabled` donen 0,79 i 0,21 arrodonint. La mitjana de `disabled` i `no_disabled`, semblen ser bons, perquè hi ha més quantitat de `disabled`, que no pas de `no_disabled`. Llavors s'han obtingut valors coherents durant el desenvolupament de la primera part de projecte.

Com ja s'ha finalitzat l'explicació de la primera subsecció de la part de resultats assolits, ara es passarà a explicar la segona part de resultats del desenvolupament portat a terme.

6.2 Resultats obtinguts durant la implementació de la pàgina web

Durant la fase del desenvolupament de la pàgina web, s'han tret dades diferents de la primera part del desenvolupament mentre es feia la pàgina web. Les dades que s'han obtingut durant aquesta fase del projecte, seran comentades a continuació:

- Histograma de graus
- `time_lock_delta`
- `min_htlc`
- `max_htlc_msat`
- `fee_base_msat`
- `fee_rate_milli_msat`

Mentrestant la primera part del desenvolupament, s'han tret les mitjanes de les configuracions de les polítiques de les arestes i el resultat de l'histograma de graus dels nodes, però no s'ha pogut treure els valors que tenia cada una de les configuracions. Llavors entretant durant la fase del desenvolupament web, s'ha dedicat a treure els valors de cada configuració i cada valor de grau.

Per treure els resultats de l'histograma de graus no ha sigut molt feixuc com les configuracions de les arestes, ja que molts dels valors dels graus era 0 o 1, i a més només es treballaven amb 4.571 valors, per tant, només

es tenia l'interès d'enfocar-se en els altres valors que eren diferents de 0 i 1. Ara es mostrarà un exemple de la part de l'histograma de graus quan es realitza el gràfic per la representació de les dades analitzades durant l'estudi:

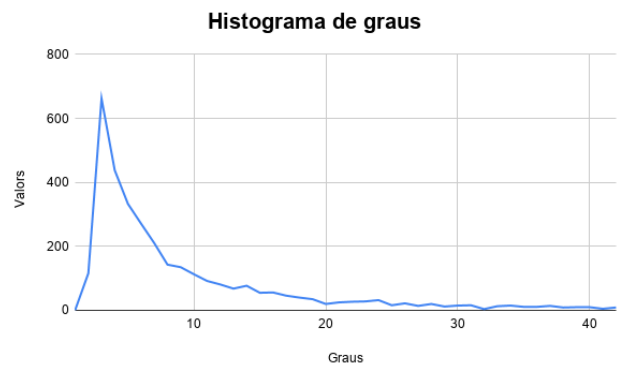


Fig. 8: Gràfic de línia de l'histograma de graus

Els resultats de `time_lock_delta`, `min_htlc`, `max_htlc_msat`, `fee_base_msat` i `fee_rate_milli_msat` s'han tret de la mateixa manera. En el cas que la configuració tingués valors per defecte, s'anava a buscar aquell valor, si no, es buscaven valors que eres típics d'aquestes configuracions. Es va fer una a una, fins a completar el nombre de configuracions de les dues polítiques, que en aquest cas són 64.847 valors.

Es mostrarà un exemple d'alguns dels valors trobats a `min_htlc` en una taula. Aquesta taula a la pàgina web, ha sigut representada en format de gràfic.

Valor	Quantitat de valors
0	3.247
1	5.527
18	208
50	14
99	1
450	12
500	9
1.000	55.367

TAULA 2: TAULA VALORS MIN_HTLC

Com es pot veure, s'ha mostrat només una part de la taula dels valors de `min_htlc`, que a partir de les dades, s'ha pogut realitzar els gràfics que es volien per visualitzar la informació analitzada. El mateix s'ha fet amb les altres configuracions com `time_lock_delta`, `max_htlc_msat`, `fee_base_msat` i `fee_rate_milli_msat`. Malgrat que les configuracions `max_htlc_msat`, `fee_base_msat` i `fee_rate_milli_msat`, l'anàlisi de les dades ha estat molt més pesat, a causa de tenir valors molt variats i que es repeteixen poc.

Fins aquí ha arribat l'explicació dels resultats i el perquè de cada resultat obtingut. S'ha explicat la primera part de la implementació que s'han obtingut determinats resultats i després, s'ha explicat la segona part del projecte, que s'han assolit els objectius i resultats que finalment han sigut representats a una pàgina web.

7 PROBLEMES SORGITS

L'experiència que s'ha adquirit durant el procés del desenvolupament de l'anàlisi i estudi de la xarxa Lightning a Bitcoin ha sigut positiu, ja que he après moltes coses que no les coneixia abans. També fer aquests tipus de projecte al grau, fa que la gent aprengui a treballar de forma independent. A més, la investigació quan es fa un projecte, és un punt clau, perquè és com s'aprèn i es millora en tema coneixement.

Durant el desenvolupament del treball de final de grau, han aparegut alguns problemes. Els problemes que s'han trobat entretant que es feien les fases de la implementació, han sigut diversos.

El gran problema que s'ha trobat, ha sigut a l'hora de fer el graf en NetworkX a partir del fitxer json, perquè no es tenia el coneixement per fer-ho, és a dir, no s'havia treballat mai amb NetworkX, ni amb grafs a Python. El graf fet en LND, era molt complex perquè hi havia molts nodes i arestes a tenir present, pel que va suposar un desavantatge.

Les dificultats que s'ha trobat quan s'estava implementant les funcionalitats del multigraf, és la quantitat de treball, perquè s'havia de treballar amb moltes arestes. En aquest cas, no s'havia de treballar amb 33.022 arestes, sinó que s'havia de treballar amb 64.847 polítiques d'arestes. Això es deu perquè cada aresta té dues polítiques, pel qual les dades estan multiplicades per dos. Com bé s'ha comentat anteriorment, s'ha hagut de treballar amb 64.847 polítiques d'arestes perquè hi ha algunes arestes que són nul·les, pel qual no tenen dades a ser estudiades.

També un gran problema és que a Internet hi havia molta documentació de les funcions de la llibreria NetworkX, però moltes d'aquestes funcions no eren suficients. Hi havia algunes funcions que van ser molt idònies, però hi havia d'altres que no servien per a aquest estudi. A més, a Internet no hi ha molta informació de la xarxa Lightning, i si n'hi ha o és molt fàcil d'entendre i no s'aprofundeix molt o és complexa i no s'entén gaire, llavors quan es volia buscar informació a vegades cercaven a la pàgina IEEE [7].

8 CONCLUSIONS

Al llarg del desenvolupament del projecte, quan s'ha estat fent l'anàlisi i estudi de la xarxa Lightning a Bitcoin, s'ha pogut veure que s'han anat completant tots els objectius proposats des d'un principi. En aquest apartat es volen exposar les conclusions que s'han aconseguit realitzant aquest projecte.

Aquest projecte que s'ha estat realitzant per aprendre sobre el funcionament de la xarxa Lightning, no s'ha fet per millorar la xarxa Lightning, simplement s'ha realitzat l'estudi per entendre el seu funcionament. Per fer l'estudi, s'ha hagut de seguir un procés d'investigació, de proves, de desenvolupament i d'anar aprenent a poc a poc mentre es feia algunes de les fases.

La part del projecte que més temps ha portat en acabar-la ha sigut els dos tipus d'implementacions i l'anàlisi de les dades. Els desenvolupaments han durat molt perquè s'havia d'aprendre com funciona Python per fer grafs i com funciona la llibreria NetworkX per a la realització de funcions que finalment acabaran donant uns resultats. A més,

quan s'estava fent la implementació de la pàgina web, s'havia d'aprendre sobre les tecnologies web i la llibreria de JavaScript morris.js, per tant s'ha reforçat el coneixement d'aquest estudi.

Finalment, quan s'ha acabat de fer l'estudi de la xarxa Lightning a Bitcoin, es considera que s'han assolit els objectius plantejats des d'un primer moment del projecte. Per aquest motiu, es dona per conclòs aquest treball de final de grau.

AGRAÏMENTS

En primer lloc vull d'agrair al meu tutor Guillermo Navarro Arribas per l'esforç, la paciència i les ganes que ha tingut per ajudar-me en moments crítics del treball i per les tutories que m'ha estat proporcionant.

En segon lloc vull agrair a tots els meus amics i familiars que m'han estat animant i escoltant sense saber sobre el tema quan he estat en moments de frustració perquè no sabia com avançar.

I per últim m'agradaria agrair als meus companys de la universitat que m'han estat escoltant i m'han intentat aconsellar durant aquesta etapa del grau.

REFERÈNCIES

- [1] A. J. P. Oliveros, «¿Lightning Network consiguió la estabilidad de Bitcoin?», CRIPTO TENDENCIA, 08-oct-2019. [En línia]. Disponible a: <https://criptotendencia.com/2019/10/08/lightning-network-consiguio-la-estabilidad-de-bitcoin/>. [Consulta: 04-feb-2020].
- [2] «Lightning Network - Overview», Crunchbase. [En línia]. Disponible a: <https://www.crunchbase.com/organization/lightning-network>. [Consulta: 01-feb-2020].
- [3] H. Partz, «Airbnb Now Bookable With Bitcoin and Lightning Network via Fold App», Cointelegraph, 11-nov-2019. [En línia]. Disponible a: <https://cointelegraph.com/news/airbnb-now-bookable-with-bitcoin-and-lightning-network-via-fold-app>. [Consulta: 01-feb-2020].
- [4] «Lightning Network Explorer». [En línia]. Disponible a: <https://explorer.acinq.co/>. [Consulta: 01-feb-2020].
- [5] «1ML - Lightning Network Search and Analysis Engine - Bitcoin mainnet». [En línia]. Disponible a: <https://1ml.com/>. [Consulta: 01-feb-2020].
- [6] «Hashlock - BitcoinWiki». [En línia]. Disponible a: <https://en.bitcoinwiki.org/wiki/Hashlock>. [Consulta: 01-feb-2020].
- [7] «IEEE - The world's largest technical professional organization dedicated to advancing technology for the benefit of humanity.» [En línia]. Disponible a: <https://www.ieee.org/>. [Consulta: 01-feb-2020].

APÈNDIX

A continuació, en aquesta secció es presentaran algunes imatges de la pàgina web que s'ha realitzat per mostrar l'estudi i l'anàlisi de Lightning Network a Bitcoin. Es mostraran exemples dels gràfics fets per visualitzar les dades extretes de l'estudi.

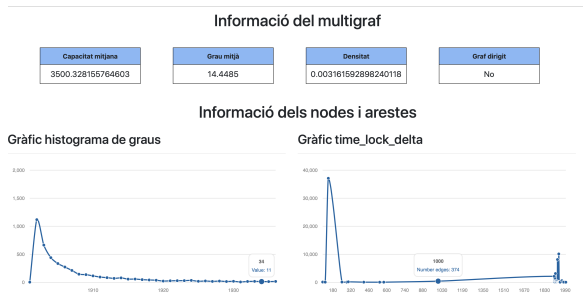


Fig. 9: Pàgina web secció multigraf

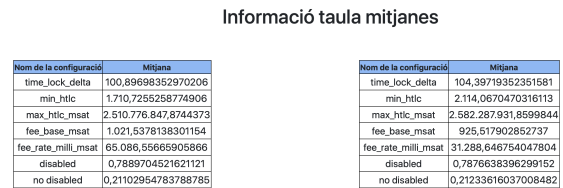


Fig. 13: Pàgina web taula configuracions

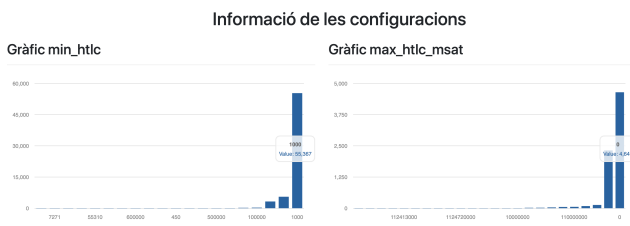


Fig. 10: Pàgina web secció configuracions

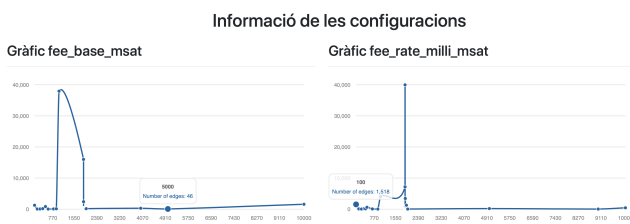


Fig. 11: Pàgina web secció configuracions

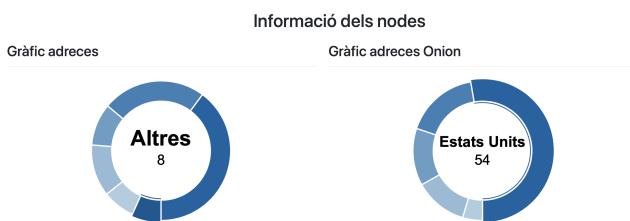


Fig. 12: Pàgina web secció nodes