

The Lightning Network

Laboratorio

APÉNDICE

A1. **The Lightning Polar** es una herramienta para pruebas de entornos locales de Lightning Network sobre la red de prueba regtest. Te permite interactuar y diseñar redes para agilizar el desarrollo sobre esta tecnología, evitando la tediosa instalación y configuración de los nodos para tu red. Proporciona:

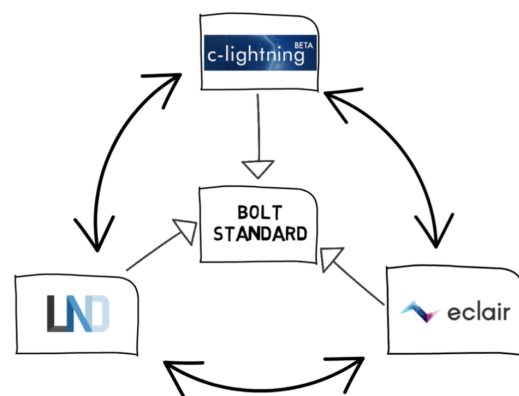
- Modelado de arquitectura
- Interacción y operativa en los nodos y canales
- Creación de nodos de Bitcoin en la red de prueba

A2. **Estándares e implementaciones de The Lightning Network.** Existe una especificación establecida en por estándares bajo B.O.L.T. (Basis of Lightning Technology). En estos estándares se definen los protocolos y especificaciones para cada implementación de Lightning Network.

<https://github.com/lightning/bolts>

¿Qué implementaciones existen ¹?

- LND (golang) - La más utilizada, y la que utilizaremos.
- c-lightning (c++)
- eclair (erlang)



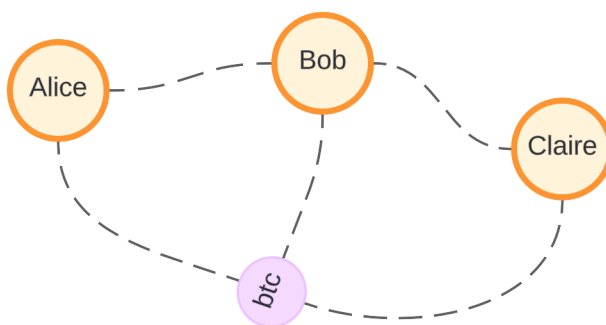
¹ LND (golang) <https://github.com/lightningnetwork/lnd>
c-lightning (c) <https://github.com/ElementsProject/lightning>
eclair (erlang) <https://github.com/ACINQ/eclair>

CUESTIONES

C1. Preparación del entorno. Para la preparación del entorno en local necesitaremos de las siguientes herramientas: Polar y Docker. Es necesario disponer de un contexto de contenedores Docker instalado previamente configurado en tu máquina o la herramienta Polar no funcionará. Descarga e instala Lightning Polar en tu sistema <https://lightningpolar.com/>



C2. Nuestra primera red. Crea una red propia con tres nodos Alice, Bob y Claire (o Carol) conectados a un nodo Bitcoin.



1. Una vez instalada la herramienta Polar procede a crear una red “Create a Lightning Network”. Necesitaríamos una red con tres nodos con la implementación LND (ver detalles de implementación en la sección de apéndice).

* Network Name

Lab 1 - Lightning Network Overview

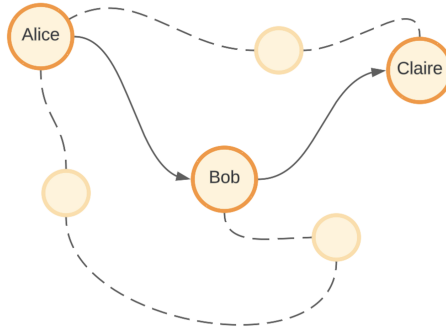
How many Managed Nodes?

| * LND | * Core Lightning | * Eclair | * Bitcoin Core |
|-------|------------------|----------|----------------|
| 3 | 0 | 0 | 1 |

Create Network

2. Una vez hayamos creado la red iniciamos la red. Pueden tardar varios minutos en inicializarse si la herramienta necesitara descargar la versión de la imagen Docker para el nodo de tipo LND ([v0.17.4 beta](https://github.com/lightningpolar/polar/blob/master/Dockerfile)).

C3. Topología de fondos. Necesitamos fondos en los nodos y construir un escenario para hacer una prueba de pago del nodo Alice a Claire. Para ello necesitamos fondos y comunicación transitiva entre los nodos involucrados.



1. Cada nodo está vinculado a una wallet en Bitcoin (on-chain), teniendo dos tipos de balances: on-chain y off-chain. Nuestro objetivo es obtener primero fondos on-chain para los tres nodos como caso de uso. Necesitaremos activar en el nodo de Bitcoin el autominado cada 30 segundos (En un entorno testnet o mainnet los mineros trabajan bloques con una frecuencia de ~10 minutos)



2. Nos dirigimos a cada nodo, en el caso del diagrama con Alice, comprobamos este balance y nos vamos a la pestaña *Actions*. Desde esta pestaña la herramienta nos permite realizar operaciones vía comando o por una llamada RPC.

3. Depositamos 0.05 BTC (5000000 Sats) para cada nodo y esperamos el próximo ciclo de minado para comprobar los balances de cada nodo.

alice **5.0M sats**

Info Connect **Actions**

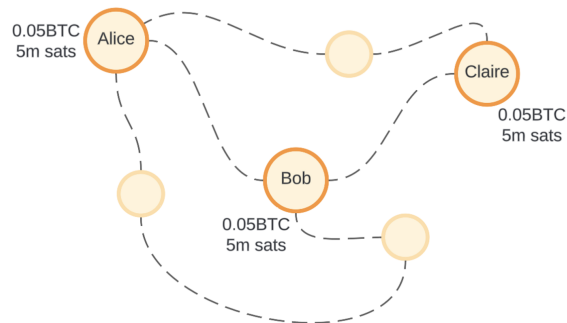
Deposit Funds

5,000,000

⬇ Deposit

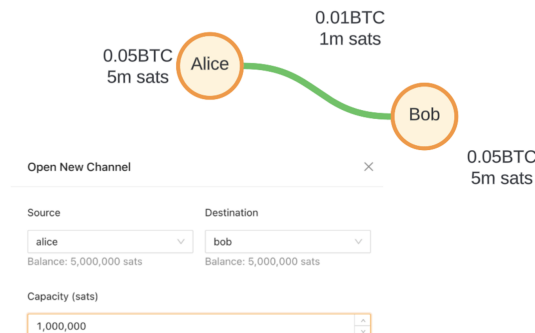
Nota: El deposito de fondos es una tarea menos complicada gracias a la herramienta Polar que abstrae a un solo un paso. Generalmente se necesitan de los siguientes operaciones:

- Generar una address para recibir los fondos on-chain: *Incli newaddress p2wkh*
- Enviar fondos a dicha dirección y verificamos el saldo con: *Incli walletbalance*

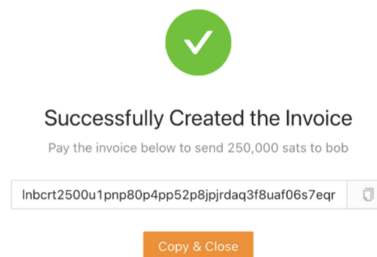


C4. Primer caso de uso: Bob es un negocio centralizado conectado a la red The Lightning Network. Alice realiza una visita al negocio y necesita realizar un primer micropago de uno de los servicios que ofrece Bob.

1. Necesitamos abrir un canal entre Alice y Bob. Bob tiene liquidez, y Alice piensa que no va a gastar más de 0.01BTC en la red, así que abre un canal directo con Bob. Ahora Alice puede hacer negocios con Bob y viceversa, siempre que haya liquidez.



2. Bob le pasa un recibo a Alice con la cantidad de 0.0025 BTC (250.000 satoshis). Este paso lo denominamos: "Create Invoice". Es tan fácil desde Polar, con ir al nodo de Bob en la pestaña Actions y pinchar en la subsección el botón "Create Invoice". Nota: este paso se puede hacer desde la consola del nodo con el comando Incl: `Incli addinvoice --amt 250000`



3. Este invoice genera un código en base58 con el prefijo Inbcrt llamado BOLT11 (<https://github.com/lightning/bolts/blob/master/11-payment-encoding.md>). Es una solicitud codificada que un receptor genera para solicitar un pago específico de un remitente que si lo deconstruimos contiene todos los detalles necesarios para el pago. Se puede generar como QR para permitir pagos con wallets adaptadas.

Nota: La herramienta online <https://lightningdecoder.com/> nos permite deconstruir el código completo y averiguar la cantidad; y algunas particularidades que se pueden añadir al pago, como la caducidad o las fees.

Lightning Decoder

Decode Lightning Network Requests
BOLT11, LNURL, and Lightning Address

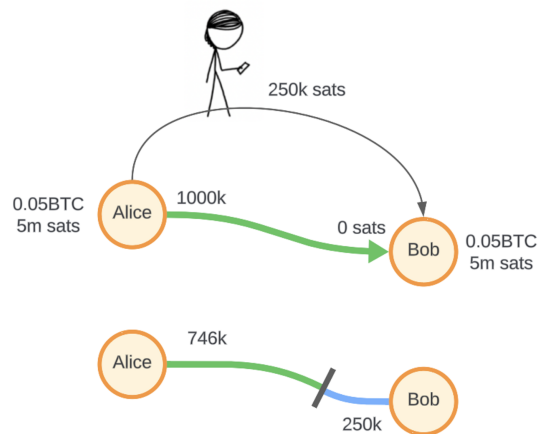
 `lnbcrt2500u1pnp80p4pp52p8jpjrdag3f8uaf06s7eqrhcl19s2892e43zcm3akk3m7p9wsdqccqz5vqsp54aunuxzqt0zgqv2u0a2ce8lfh4gjftwqn5c94sqdjnce8d05f3zq9qyysqynlaph48mj87vm6atuxf7vkg5h5s7vvuf7krrpj3hc6x6f4qpswst0q787c0a4wsxe0ehs0y953s9wd8gnmn033ax4u76jvh4j4pqlqq4dqted` ✕

| | |
|------------------------|---|
| Chain | <code>regtest</code> |
| Amount (Millisatoshis) | <code>250000000</code> |
| Payee Pub Key | <code>0327ff119fe31139b6b7e7ec8a69e7ff144ee535868a335e8d6faf9270253971c2</code> |
| Invoice | <code>lnbcrt2500u1pnp80p4pp52p8jpjrdag3f8uaf06s7eqrhcl19s2892e43zcm3akk3m7p9wsdqccqz5vqsp54aunuxzqt0zgqv2u0a2ce8lfh4gjftwqn5c94sqdjnce8d05f3zq9qyysqynlaph48mj87vm6atuxf7vkg5h5s7vvuf7krrpj3hc6x6f4qpswst0q787c0a4wsxe0ehs0y953s9wd8gnmn033ax4u76jvh4j4pqlqq4dqted</code> |
| Prefix | <code>lnbcrt2500u</code> |
| Recovery Flag | <code>0</code> |
| Amount (Satoshis) | <code>250000</code> |

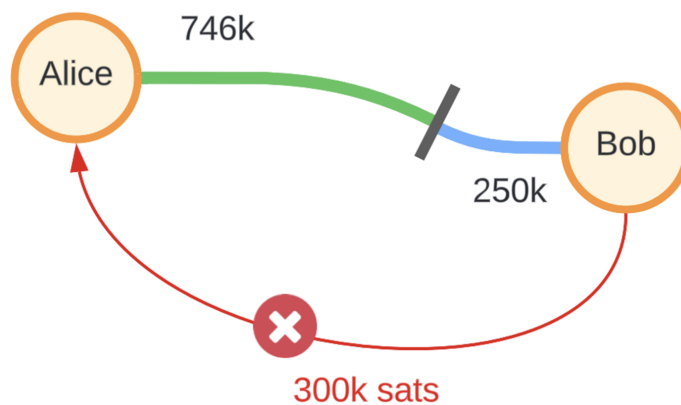
4. Con este código podemos realizar el pago desde cualquier nodo. Para nuestro caso haremos el pago desde Alice. El pago se realizará de una forma directa al existir un canal directo entre Alice y Bob. El comando que representa un pago de un invoice sería por `Incli payinvoice lnbcrt25...9d`



5. Una vez realizado el pago el balance de Alice sería de aproximadamente 750k satoshis (depende del ratio de fee de canal) y de 250k de liquidez en el lado de Bob. Ya que inicialmente fue Alice quien puso los 1M de sats de liquidez para el canal.

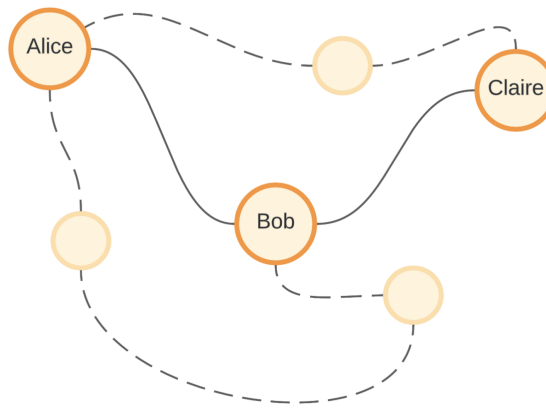


C5. ¿Qué ocurriría si Bob tuviera que devolverle los 250k de satoshis junto con una comisión a Alice, por ejemplo un total de 300k satoshis debido a un error? ¿Qué pasos deberíamos de dar? ¿Qué ocurriría durante el pago de vuelta?



C6. Flujos más realistas. Alice y Claire se conocen personalmente, y deciden ajustar cuentas. Claire le envía un recibo a Alice, y ésta realiza el pago, Alice no tiene un canal abierto con Claire, pero confía que existe un camino hacia ella. Os pedimos los siguientes pasos:

1. Configura la red de manera que utilicen a Bob como intermediario (véase diagrama de abajo). Bob es un negocio y tiene canal abierto con muchos de sus clientes. Requisito inicial: Bob tenía inicialmente un canal abierto con Claire de 2M sats.
2. Realiza los siguientes micropagos: Alice (200k) → Claire, Alice (150k) → Claire, Claire (150k) → Alice, Claire (500k) → Alice. Analiza el estado de los canales y los caminos recorridos en el grafo. Anota los errores si no se pudo realizar la operación, y deduce. Anímate y usa los comandos `lncli` desde el interfaz de consola de cada nodo.



REFERENCIAS ÚTILES

¿Cómo configurar una red en local con Polar? <https://github.com/lightninglabs/docs.lightning.engineering/blob/master/lapps/guides/polar-lapps/local-cluster-setup-with-polar.md>

Primeros pasos con LND. <https://docs.lightning.engineering/lightning-network-tools/lnd/first-steps-with-lnd>