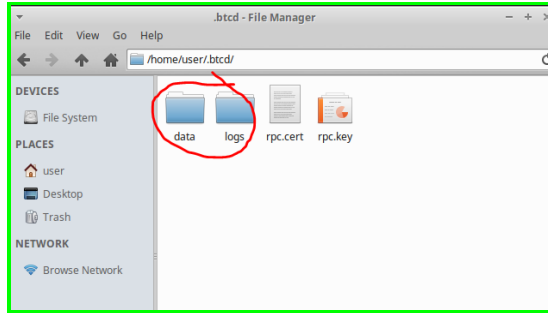


Okay so next time i go through the instruction:

-make sure the extra log file is deleted (hidden file)



-run the crossed out command instead of the other one

-have everyone mine blocks for funds before you actually create the channels

check to see if this is in the whitepaper that even if 1 person funds it, there has to be at least twice the amount in the channel

general instructions

steps of stuff you have to do

commands to copy and paste into terminals

the addresses and keys change each time you restart the network

copy and paste into files

keyboard instructions (to stop a process: ctrl + C or ctrl + z)

which directory to be in

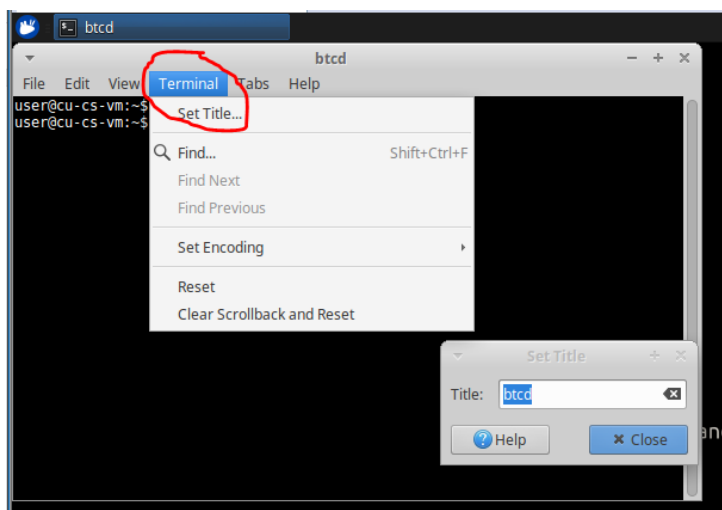
LN white paper application/ explanations

if you accidentally close the btcd , just close the whole window and enter

cd \$GOPATH

btcd --txindex --simnet --rpcuser=kek --rpcpass=kek

to name terminals



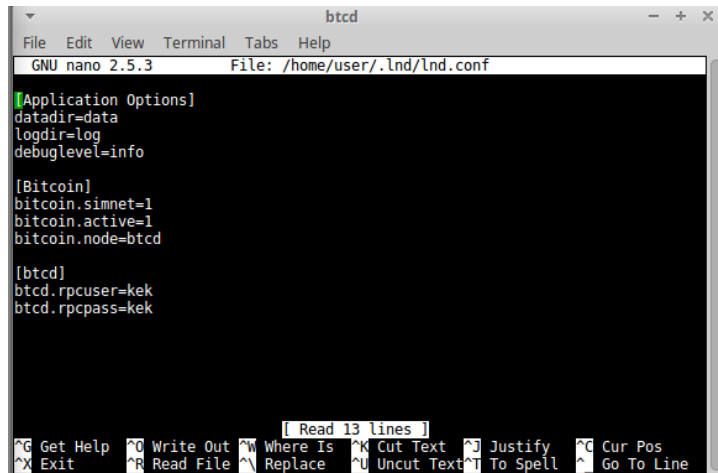
From <<https://dev.lightning.community/tutorial/01-lncli/index.html>>

STARTING HERE ONLY DO THIS THE FIRST TIME

add flags to the lnd.conf file. type this from any directory

```
nano ~/.lnd/lnd.conf
```

```
cd
```



```
GNU nano 2.5.3 File: /home/user/.lnd/lnd.conf

[Application Options]
datadir=data
logdir=log
debuglevel=info

[Bitcoin]
bitcoin.simnet=1
bitcoin.active=1
bitcoin.node=btcd

[btcd]
btcd.rpcuser=kek
btcd.rpcpass=kek

Read 13 lines
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line
```

to save the file then exit it

```
ctrl +O
```

```
enter
```

```
ctrl + X
```

create macaroon paths in bashrc file at the end

open the bashrc file from any terminal

```
sudo nano ~/.bashrc
```

and put this in the bashrc file at the end of the file:

```
#macaroon paths
```

```
alias InclI-alice="Incli --rpcserver=localhost:10001
```

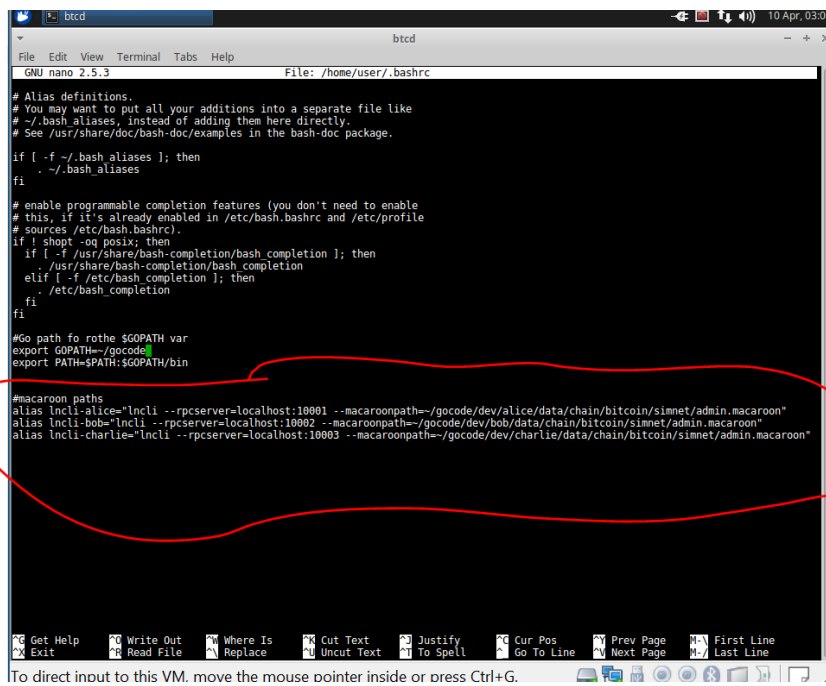
```
--macaroonpath=~/.gocode/dev/alice/data/chain/bitcoin/simnet/admin.macaroon"
```

```
alias InclI-bob="Incli --rpcserver=localhost:10002
```

```
--macaroonpath=~/.gocode/dev/bob/data/chain/bitcoin/simnet/admin.macaroon"
```

```
alias InclI-charlie="Incli --rpcserver=localhost:10003
```

```
--macaroonpath=~/.gocode/dev/charlie/data/chain/bitcoin/simnet/admin.macaroon"
```



```
File Edit View Terminal Tabs Help
GNU nano 2.5.3 File: /home/user/./bashrc

# Alias definitions.
# You may want to put all your additions into a separate file like
# ~/.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.

if [ -f ~/.bash_aliases ]; then
    . ~/.bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
    if [ -f /usr/share/bash-completion/bash_completion ]; then
        . /usr/share/bash-completion/bash_completion
    elif [ -f /etc/bash_completion ]; then
        . /etc/bash_completion
    fi
fi

# Go path for the GOPATH var
export GOPATH=/gocode
export PATH=$PATH:$GOPATH/bin

# macaroon paths
alias lncli-alice="lncli --rpcserver=localhost:10001 --macaroonpath=/gocode/dev/alice/data/chain/bitcoin/simnet/admin.macaroon"
alias lncli-bob="lncli --rpcserver=localhost:10002 --macaroonpath=/gocode/dev/bob/data/chain/bitcoin/simnet/admin.macaroon"
alias lncli-charlie="lncli --rpcserver=localhost:10003 --macaroonpath=/gocode/dev/charlie/data/chain/bitcoin/simnet/admin.macaroon"
```

to save the file then exit it

ctrl +O

enter

ctrl + X

TO EXECUTE THE BASHRC FILE:

source ~/.bashrc

END OF FIRST TIME INITIALIZATION OF FILES

TO START OR RESTART THE NETWORK

(assuming not using seed phrases, but starting with new wallets)

there will be 1 bitcoin daemon (btcd) terminal + 2 terminals per node opened, 1 as the lnd and 1 as the wallet

the btcd

cd \$GOPATH

btcd --txindex --simnet --rpcuser=kek --rpcpass=kek

```
btcd
File Edit View Terminal Tabs Help
user@cu-cs-vm:~$ cd $GOPATH
user@cu-cs-vm:~/gocode$ btcd --txindex --simnet --rpcuser=kek --rpcpass=kek
2019-04-10 05:13:18.658 [INF] BTCD: Version 0.12.0-beta
2019-04-10 05:13:18.672 [INF] BTCD: Loading block database from '/home/user/.btcd/data/simnet/blocks_ffldb'
2019-04-10 05:13:18.751 [INF] BCDB: Detected unclean shutdown - Repairing...
2019-04-10 05:13:18.754 [INF] BCDB: Database sync complete
2019-04-10 05:13:18.754 [INF] BTCD: Block database loaded
2019-04-10 05:13:18.764 [INF] INDX: Transaction index is enabled
2019-04-10 05:13:18.764 [INF] INDX: Committed filter index is enabled
2019-04-10 05:13:18.766 [INF] CHAN: Loading block index...
2019-04-10 05:13:18.767 [INF] CHAN: Chain state (height 0, hash 683e86bd5c6d110d91b94b97137ba6bfe02dbdb8e3dff722a669b5d69d77af6, totaltx 1, work 2)
2019-04-10 05:13:18.831 [INF] RPCS: RPC server listening on 127.0.0.1:18556
2019-04-10 05:13:18.835 [INF] AMGR: Loaded 0 addresses from file '/home/user/.btcd/data/simnet/peers.json'
2019-04-10 05:13:18.836 [INF] CMGR: Server listening on 0.0.0.0:18555
2019-04-10 05:13:18.836 [INF] CMGR: Server listening on [::]:18555
$ sudo apt install <package name>
```

START THE NODES

each one in their own terminal: (these can be tabs or new windows, doesn't matter)

for alice

|

(if you are starting the wallets over) you need to **delete the log and data files for each node**

check if you have old logs or data if you don't want to use seed phrases(want to start over with new wallets)

ls

```
btcd  alice lnd
File Edit View Terminal Tabs Help
user@cu-cs-vm:~/gocode/src/github.com/lightningnetwork/lnd$ cd $GOPATH/dev/alice
user@cu-cs-vm:~/gocode/dev/alice$ ls
data log
user@cu-cs-vm:~/gocode/dev/alice$
```

if you have data and log directories in the dev folder, delete both data and log

DO THIS CHECK FOR EVERY NODE from there dev/<node> location

rm -r data log

ls

```
CU-CS-VM-Fall-2018 [btcd] alice lnd bob lnd charlie lnd

alice lnd
user@cu-cs-vm:~/gocode/src/github.com/lightningnetwork/lnd$ cd $GOPATH/dev/alice
user@cu-cs-vm:~/gocode/dev/alice$ ls
data log
user@cu-cs-vm:~/gocode/dev/alice$ rm data
rm: cannot remove 'data': Is a directory
user@cu-cs-vm:~/gocode/dev/alice$ rm -r data
user@cu-cs-vm:~/gocode/dev/alice$ rm -r log
user@cu-cs-vm:~/gocode/dev/alice$ ls
user@cu-cs-vm:~/gocode/dev/alice$

bob lnd
user@cu-cs-vm:~/gocode/dev/alice$ cd $GOPATH/dev/bob
user@cu-cs-vm:~/gocode/dev/bob$ ls
data log
user@cu-cs-vm:~/gocode/dev/bob$ rm -r data log
user@cu-cs-vm:~/gocode/dev/bob$ ls
user@cu-cs-vm:~/gocode/dev/bob$

charlie lnd
user@cu-cs-vm:~/gocode/dev/bob$ cd $GOPATH/dev/charlie
user@cu-cs-vm:~/gocode/dev/charlie$ ls
data log
user@cu-cs-vm:~/gocode/dev/charlie$ rm -r data log
user@cu-cs-vm:~/gocode/dev/charlie$ ls
user@cu-cs-vm:~/gocode/dev/charlie$
```

all dev/<node> folders should be empty or else will get listening error on the ports if starting new wallets

SET UP LISTENING PORTS (NODES)

each in their own, different terminal (this will be the 2nd terminal for each node)

for alice

in a new terminal (alice lnd)

```
cd $GOPATH/dev/alice
```

***use this one

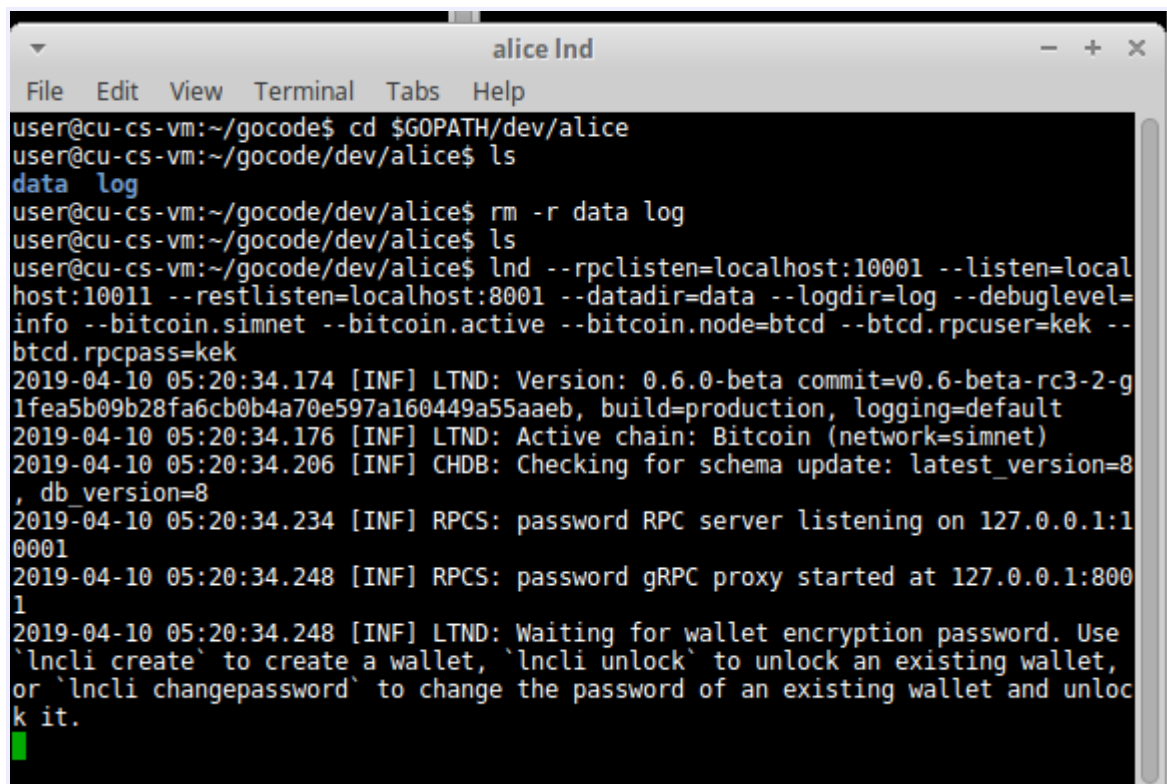
```
lnd --rpclisten=localhost:10001 --listen=localhost:10011 --restlisten=localhost:8001
```

that is not working for me

lncli-alice getinfo

***dont use this oneXXXXX

```
lnd --rpclisten=localhost:10001 --listen=localhost:10011 --restlisten=localhost:8001 --datadir=data  
--logdir=log --debuglevel=info --bitcoin.simnet --bitcoin.active --bitcoin.node=btcd --btcd.rpcuser=kek  
--btcd.rpcpass=kek
```



```
alice lnd
File Edit View Terminal Tabs Help
user@cu-cs-vm:~/gocode$ cd $GOPATH/dev/alice
user@cu-cs-vm:~/gocode/dev/alice$ ls
data log
user@cu-cs-vm:~/gocode/dev/alice$ rm -r data log
user@cu-cs-vm:~/gocode/dev/alice$ ls
user@cu-cs-vm:~/gocode/dev/alice$ lnd --rpclisten=localhost:10001 --listen=local
host:10011 --restlisten=localhost:8001 --datadir=data --logdir=log --debuglevel=
info --bitcoin.simnet --bitcoin.active --bitcoin.node=btcd --btcd.rpcuser=kek --
btcd.rpcpass=kek
2019-04-10 05:20:34.174 [INF] LTND: Version: 0.6.0-beta commit=v0.6-beta-rc3-2-g
1fea5b09b28fa6cb0b4a70e597a160449a55aaeb, build=production, logging=default
2019-04-10 05:20:34.176 [INF] LTND: Active chain: Bitcoin (network=simnet)
2019-04-10 05:20:34.206 [INF] CHDB: Checking for schema update: latest_version=8
, db_version=8
2019-04-10 05:20:34.234 [INF] RPCS: password RPC server listening on 127.0.0.1:1
0001
2019-04-10 05:20:34.248 [INF] RPCS: password gRPC proxy started at 127.0.0.1:800
1
2019-04-10 05:20:34.248 [INF] LTND: Waiting for wallet encryption password. Use
`lncli create` to create a wallet, `lncli unlock` to unlock an existing wallet,
or `lncli changepassword` to change the password of an existing wallet and unloc
k it.
```

leave it running like in the picture above, waiting for wallet encryption

SET UP WALLETS:

each one in their own terminal: (these can be tabs or new windows, doesn't matter)

I used password for the password for all of them

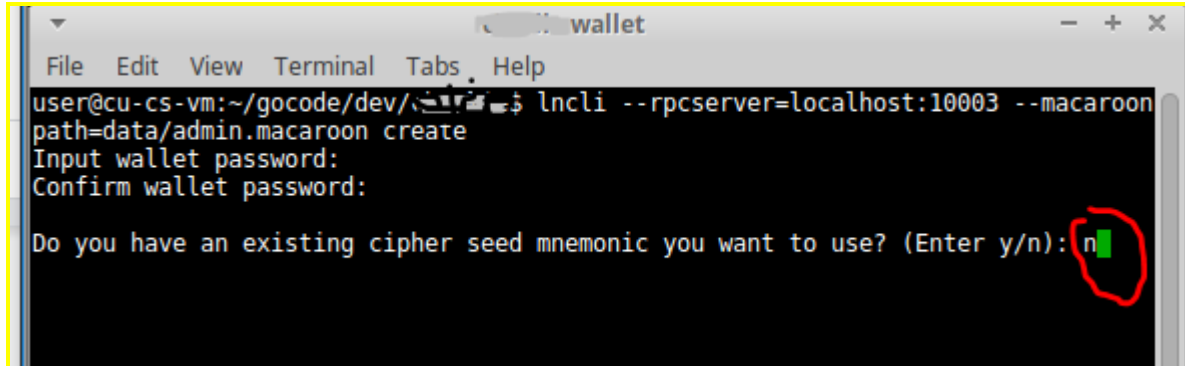
for alice

in a new terminal (alice wallet)

```
cd $GOPATH/dev/alice
```

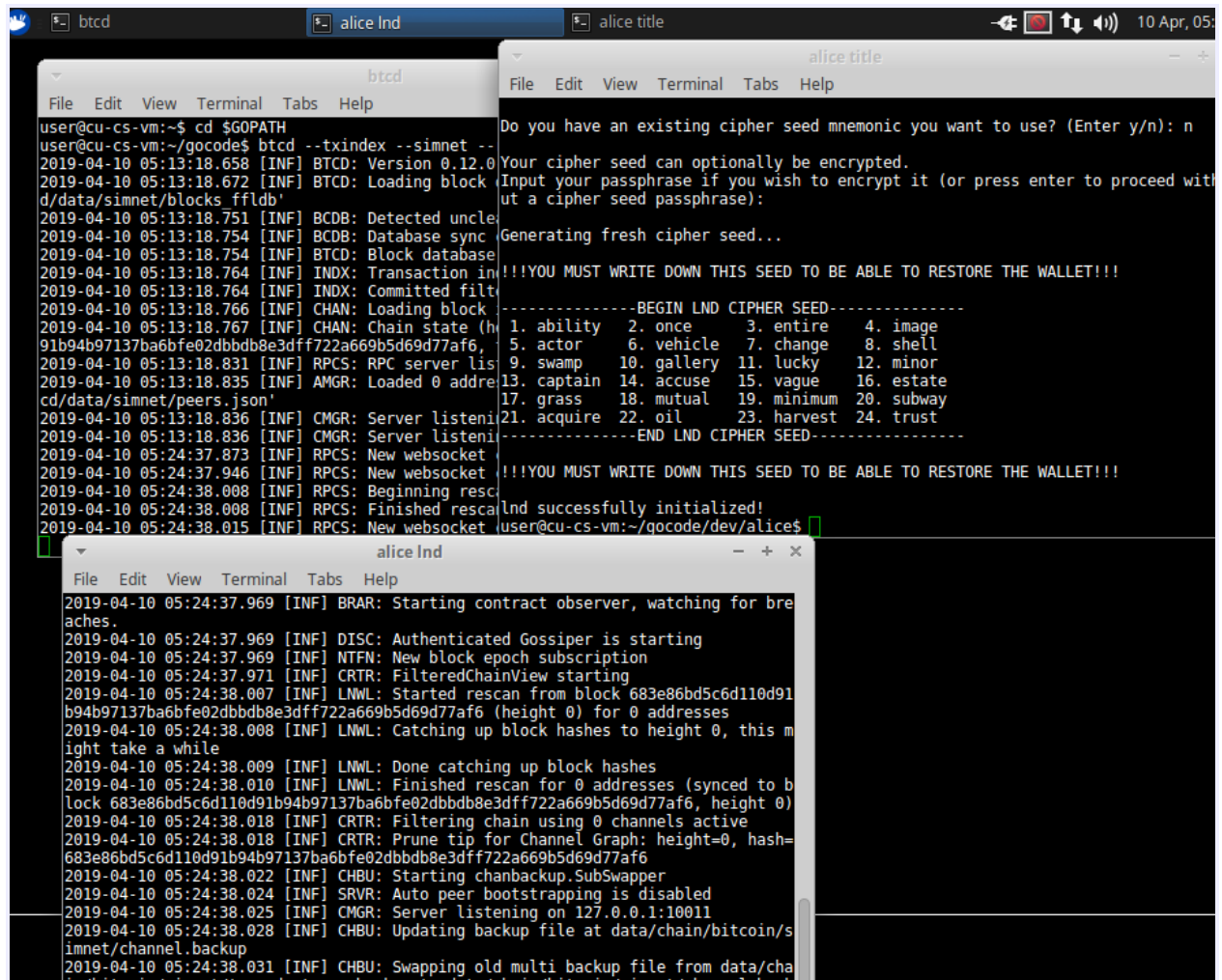
```
lncli --rpcserver=localhost:10001 --macaroonpath=data/admin.macaroon create
```

when it asks for a password you can use password, then say n, then enter



```
wallet
File Edit View Terminal Tabs Help
user@cu-cs-vm:~/gocode/dev/...$ lncli --rpcserver=localhost:10003 --macaroon
path=data/admin.macaroon create
Input wallet password:
Confirm wallet password:

Do you have an existing cipher seed mnemonic you want to use? (Enter y/n): n
```



```
btcd
File Edit View Terminal Tabs Help
user@cu-cs-vm:~$ cd $GOPATH
user@cu-cs-vm:~/gocode$ btcd --txindex --simnet --
2019-04-10 05:13:18.658 [INF] BTCN: Version 0.12.0
2019-04-10 05:13:18.672 [INF] BTCN: Loading block
d/data/simnet/blocks.ffldb'
2019-04-10 05:13:18.751 [INF] BCDB: Detected uncle
2019-04-10 05:13:18.754 [INF] BCDB: Database sync
2019-04-10 05:13:18.754 [INF] BTCN: Block database
2019-04-10 05:13:18.764 [INF] INDX: Transaction in
2019-04-10 05:13:18.764 [INF] INDX: Committed filt
2019-04-10 05:13:18.766 [INF] CHAN: Loading block
2019-04-10 05:13:18.767 [INF] CHAN: Chain state (h
91b94b97137ba6bfe02dbdb8e3dff722a669b5d69d77af6,
2019-04-10 05:13:18.831 [INF] RPCS: RPC server lis
2019-04-10 05:13:18.835 [INF] AMGR: Loaded 0 addre
cd/data/simnet/peers.json'
2019-04-10 05:13:18.836 [INF] CMGR: Server listeni
2019-04-10 05:13:18.836 [INF] CMGR: Server listeni
2019-04-10 05:24:37.873 [INF] RPCS: New websocket
2019-04-10 05:24:37.946 [INF] RPCS: New websocket
2019-04-10 05:24:38.008 [INF] RPCS: Beginning resc
2019-04-10 05:24:38.008 [INF] RPCS: Finished resc
2019-04-10 05:24:38.015 [INF] RPCS: New websocket
user@cu-cs-vm:~/gocode/dev/alice$

alice lnd
File Edit View Terminal Tabs Help
2019-04-10 05:24:37.969 [INF] BRAR: Starting contract observer, watching for bre
aches.
2019-04-10 05:24:37.969 [INF] DISC: Authenticated Gossiper is starting
2019-04-10 05:24:37.969 [INF] NTFN: New block epoch subscription
2019-04-10 05:24:37.971 [INF] CRTR: FilteredChainView starting
2019-04-10 05:24:38.007 [INF] LNWL: Started rescan from block 683e86bd5c6d110d91
b94b97137ba6bfe02dbdb8e3dff722a669b5d69d77af6 (height 0) for 0 addresses
2019-04-10 05:24:38.008 [INF] LNWL: Catching up block hashes to height 0, this m
ight take a while
2019-04-10 05:24:38.009 [INF] LNWL: Done catching up block hashes
2019-04-10 05:24:38.010 [INF] LNWL: Finished rescan for 0 addresses (synced to b
lock 683e86bd5c6d110d91b94b97137ba6bfe02dbdb8e3dff722a669b5d69d77af6, height 0)
2019-04-10 05:24:38.018 [INF] CRTR: Filtering chain using 0 channels active
2019-04-10 05:24:38.018 [INF] CRTR: Prune tip for Channel Graph: height=0, hash=
683e86bd5c6d110d91b94b97137ba6bfe02dbdb8e3dff722a669b5d69d77af6
2019-04-10 05:24:38.022 [INF] CHBU: Starting chanbackup.SubSwapper
2019-04-10 05:24:38.024 [INF] SRVR: Auto peer bootstrapping is disabled
2019-04-10 05:24:38.025 [INF] CMGR: Server listening on 127.0.0.1:10011
2019-04-10 05:24:38.028 [INF] CHBU: Updating backup file at data/chain/bitcoin/s
imnet/channel.backup
2019-04-10 05:24:38.031 [INF] CHBU: Swapping old multi backup file from data/cha
to this is the first of the new backup file. data/chain/bitcoin/simnet/channel.backup
```

```
lncli --rpcserver=localhost:10001
```

```
--macaroonpath=~/gocode/dev/alice/data/chain/bitcoin/simnet/admin.macaroon getinfo
```



```
user@cu-cs-vm:~/gocode/dev/alice$ lncli --rpcserver=localhost:10001 --macaroonpath ~/gocode/dev/alice/data/chain/bitcoin/simnet/admin.macaroon getinfo
{"version": "0.6.0-beta commit=v0.6-beta-rc3-2-g1fea5b09b28fa6cb0b470e597a160449a55aaeb",
"identity pubkey": "030149215988c6130a8ca857cbe4cd72087c8cc30130d96cda56c28d8d018cc96b",
"alias": "030149215988c6130a8c",
"num pending channels": 0,
"num active channels": 0,
"num inactive channels": 0,
"num peers": 0,
"block height": 0,
"block hash": "683e86bd5cd110d91b94b97137ba6bfe02dbdb8e3dff722a669b5d69d77af6",
"best header timestamp": 1401292357,
"synced to chain": false,
"testnet": false,
"chains": {
  "chain": "bitcoin",
  "network": "simnet"
},
"uris": null
}
```

GET ALICE'S ADDRESS

lncli-alice newaddress np2wkh

```
user@cu-cs-vm:~/gocode/dev/alice$ lncli-alice newaddress np2wkh
{"address": "rbxr2nVJKZCNsvnmXuRfa19xNFd21q3aE"}
user@cu-cs-vm:~/gocode/dev/alice$
```

```
2019-04-10 05:24:37.969 [INF] BRAR: Starting contract observer, watching for breaches
2019-04-10 05:24:37.969 [INF] DTSC: Authenticated Gossiper is starting
2019-04-10 05:24:37.969 [INF] NTFN: New block epoch subscription
2019-04-10 05:24:37.971 [INF] CRTR: FilteredChainView starting
2019-04-10 05:24:38.007 [INF] LNMW: Started rescans from block 683e86bd5cd110d91b94b97137ba6bfe02dbdb8e3dff722a669b5d69d77af6 (height 0) for 0 addresses
2019-04-10 05:24:38.008 [INF] LNMW: Catching up block hashes to height 0, this might take a while
2019-04-10 05:24:38.009 [INF] LNMW: Done catching up block hashes
2019-04-10 05:24:38.010 [INF] LNMW: Finished rescans for 0 addresses (synced to block 683e86bd5cd110d91b94b97137ba6bfe02dbdb8e3dff722a669b5d69d77af6, height 0)
2019-04-10 05:24:38.018 [INF] CRTR: Filtering chain using 0 channels active
2019-04-10 05:24:38.018 [INF] CRTR: Prune tip for Channel Graph: height=0, hash=683e86bd5cd110d91b94b97137ba6bfe02dbdb8e3dff722a669b5d69d77af6
2019-04-10 05:24:38.022 [INF] CHBU: Starting chanbackup.SubSwapper
2019-04-10 05:24:38.024 [INF] SRVR: Auto peer bootstrapping is disabled
2019-04-10 05:24:38.025 [INF] CMGR: Server listening on 127.0.0.1:10011
2019-04-10 05:24:38.028 [INF] CHBU: Updating backup file at data/chain/bitcoin/simnet/channel.backup
2019-04-10 05:24:38.031 [INF] CHBU: Swapping old multi backup file from data/chain/bitcoin/simnet/temp-dont-use.backup to data/chain/bitcoin/simnet/channel.backup
2019-04-10 05:24:38.032 [INF] RPCS: [newaddress] addr=rbxr2nVJKZCNsvnmXuRfa19xNFd21q3aE
```


if this is first time ~~ever~~ opening wallets, do source bashrc for each Ind terminal, otherwise skip this step
source ~/.bashrc

SET UP THE PEERS (OTHER NODES) IN THE SIMNET

COMMAND LINE TERMINAL COMMANDS PER NODE		
after having made the btcd cd \$GOPATH btcd --txindex --simnet --rpcuser=kek --rpcpass=kek		
for alice alice Ind terminal in a new terminal (alice Ind) SET UP LISTENING PORT	for bob bob Ind terminal in a new terminal (bob Ind) SET UP LISTENING PORT	for charlie charlie Ind terminal in a new terminal (charlie Ind) SET UP LISTENING PORT
cd \$GOPATH/dev/alice ls rm -r data log ls Ind --rpclisten=localhost:10001 --listen=localhost:10011 --restlisten=localhost:8001 --datadir=data --logdir=log --debuglevel=info --bitcoin.simnet --bitcoin.active --bitcoin.node=btcd --btcd.rpcuser=kek --btcd.rpcpass=kek	cd \$GOPATH/dev/bob ls rm -r data log ls Ind --rpclisten=localhost:10002 --listen=localhost:10012 --restlisten=localhost:8002 --datadir=data --logdir=log --debuglevel=info --bitcoin.simnet --bitcoin.active --bitcoin.node=btcd --btcd.rpcuser=kek --btcd.rpcpass=kek	cd \$GOPATH/dev/charlie ls rm -r data log ls Ind --rpclisten=localhost:10003 --listen=localhost:10013 --restlisten=localhost:8003 --datadir=data --logdir=log --debuglevel=info --bitcoin.simnet --bitcoin.active --bitcoin.node=btcd --btcd.rpcuser=kek --btcd.rpcpass=kek
for alice alice wallet terminal in a new terminal (alice wallet) SET UP WALLET used password for the password	for bob bob wallet terminal in a new terminal (bob wallet) SET UP WALLET used password for the password	for charlie charlie wallet terminal in a new terminal (charlie wallet) SET UP WALLET used password for the password
cd \$GOPATH/dev/alice	cd \$GOPATH/dev/bob	cd \$GOPATH/dev/charlie

```
Incli --rpcserver=localhost:10001
--macaroonpath=data/admin.ma
caroon create
```

(if you have seed phrase enter it
here and press return on next 2
screens)

```
Incli --rpcserver=localhost:10001
--macaroonpath=~/.gocode/dev/a
lice/data/chain/bitcoin/simnet/a
dmin.macaroon getinfo
```

```
Incli-alice newaddress np2wkh
```

```
Incli --rpcserver=localhost:10002
--macaroonpath=data/admin.ma
caroon create
```

(if you have seed phrase enter it
here and press return on next 2
screens)

```
Incli --rpcserver=localhost:10002
--macaroonpath=~/.gocode/dev/
bob/data/chain/bitcoin/simnet/a
dmin.macaroon getinfo
```

```
Incli-bob newaddress np2wkh
```

```
Incli --rpcserver=localhost:10003
--macaroonpath=data/admin.ma
caroon create
```

(if you have seed phrase enter it
here and press return on next 2
screens)

```
Incli --rpcserver=localhost:10003
--macaroonpath=~/.gocode/dev/c
harlie/data/chain/bitcoin/simnet
/admin.macaroon getinfo
```

```
Incli-charlie newaddress np2wkh
```

```
CU-CS-VM-Fall-2018 x
[ btcld ] [ alice lnd ] [ alice wallet ] [ bob lnd ] [ bob wallet ] [ charlie lnd ] [ charlie wallet ] 10 Apr, 06:35

alice wallet
"version": "0.6.0-beta commit=v0.6-beta-rc3-2-glfea5b09b28fa6cb0b4a70e59
0449a55aaeb",
"identity_pubkey": "030149215988c6130a8ca857cbe4cd72087c8cc30130d96cda56
bd018cc96b",
"alias": "030149215988c6130a8c",
"num_pending_channels": 0,
"num_active_channels": 0,
"num_inactive_channels": 0,
"num_peers": 0,
"block_height": 0,
"block_hash": "683e86bd5c6d110d91b94b97137ba6bfe02dbdb8e3dff722a669b5d6
af6",
"best_header_timestamp": 1401292357,
"synced_to_chain": false,
"testnet": false,
"chains": [
  {
    "chain": "bitcoin",
    "network": "simnet"
  }
],
"uris": null
}

user@cu-cs-vm:~/gocode/dev/charlie$ incli-charlie newaddress np2wkh
{
  "address": "rVjFQK8wtRBeoFUXFrPdD5S1Mgt6ERZHoS"
}

user@cu-cs-vm:~/gocode/dev/charlie$

Some useful commands:
List directory contents
$ ls
Install Software
$ sudo apt install <package>
Remove Software
$ sudo apt remove <package>
Update Available Software List
$ sudo apt update
Upgrade Installed Software
$ sudo apt full-upgrade
```

TO GIVE ALICE MONEY RESTART BTCD

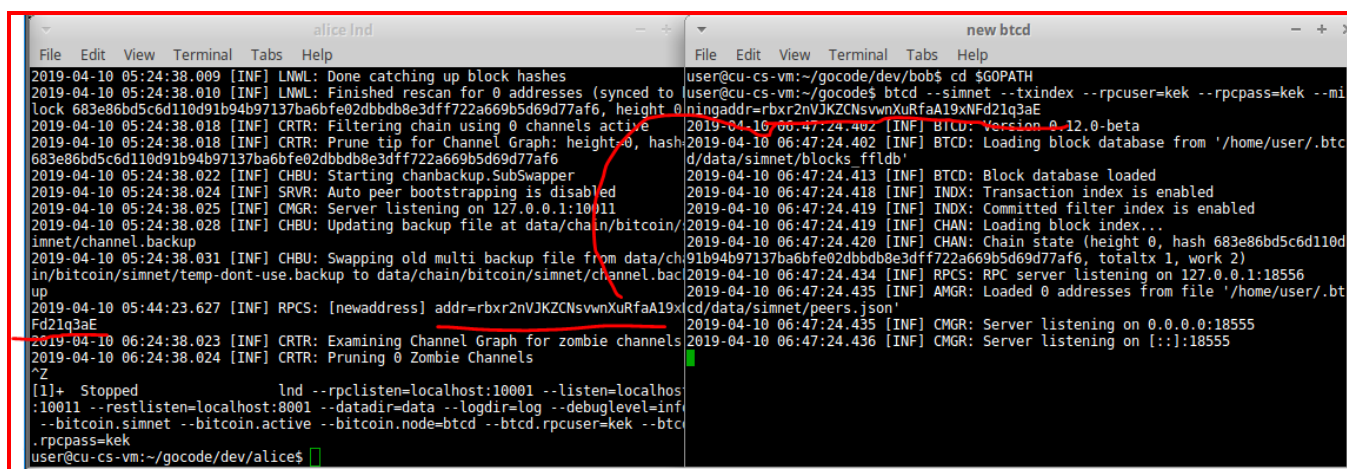
for all the lnd terminals, do `ctrl+z` or `ctrl+c`

close the btcd terminal window completely and open a new one
from new btcd terminal:

THE ADDRESSES WILL CHANGE EACH TIME YOU RESTART THE NETWORK IF YOU DONT USE THE SEED
PHRASE!! IF YOU USE THE SEED PHRASE, DON'T USE THE `newaddress np2wkh` COMMAND

`cd $GOPATH`

`btcd --simnet --txindex --rpcuser=kek --rpcpass=kek --miningaddr=rgBgmPryVzATHihNZTShnEiwAArLt8izpb`



```
alice lnd
2019-04-10 05:24:38.009 [INF] LNWL: Done catching up block hashes
2019-04-10 05:24:38.010 [INF] LNWL: Finished rescan for 0 addresses (synced to
lock 683e86bd5c6d110d91b94b97137ba6bfe02dbdbb8e3dff722a669b5d69d77af6, height 0
2019-04-10 05:24:38.018 [INF] CRTR: Filtering chain using 0 channels active
2019-04-10 05:24:38.018 [INF] CRTR: Prune tip for Channel Graph: height=0, hash=
683e86bd5c6d110d91b94b97137ba6bfe02dbdbb8e3dff722a669b5d69d77af6
2019-04-10 05:24:38.022 [INF] CHBU: Starting chanbackup.SubSwapper
2019-04-10 05:24:38.024 [INF] SRVR: Auto peer bootstrapping is disabled
2019-04-10 05:24:38.025 [INF] CMGR: Server listening on 127.0.0.1:10011
2019-04-10 05:24:38.028 [INF] CHBU: Updating backup file at data/chain/bitcoin/
simnet/channel.backup
2019-04-10 05:24:38.031 [INF] CHBU: Swapping old multi backup file from data/ch
in/bitcoin/simnet/temp-dont-use.backup to data/chain/bitcoin/simnet/channel.bac
up
2019-04-10 05:44:23.627 [INF] RPCS: [newaddress] addr=rbxr2nVJKZCNSvwnXuRfaA19x
Fd21q3aE
2019-04-10 06:24:38.023 [INF] CRTR: Examining Channel Graph for zombie channels
2019-04-10 06:24:38.024 [INF] CRTR: Pruning 0 Zombie Channels
^Z
[1]+  Stopped                  lnd --rpclisten=localhost:10001 --listen=localhos
:10011 --restlisten=localhost:8001 --datadir=data --logdir=log --debuglevel=inf
--bitcoin.simnet --bitcoin.active --bitcoin.node=btcd --btcd.rpcuser=kek --btc
.rpcpass=kek
user@cu-cs-vm:~/gocode/dev/alice$

new btcd
user@cu-cs-vm:~/gocode/dev/bob$ cd $GOPATH
user@cu-cs-vm:~/gocode$ btcd --simnet --txindex --rpcuser=kek --rpcpass=kek --mi
ningaddr=rbxr2nVJKZCNSvwnXuRfaA19xNfd21q3aE
2019-04-10 06:47:24.402 [INF] BTCD: Version 0.12.0-beta
2019-04-10 06:47:24.402 [INF] BTCD: Loading block database from '/home/user/.bt
d/data/simnet/blocks.ffldb'
2019-04-10 06:47:24.413 [INF] BTCD: Block database loaded
2019-04-10 06:47:24.418 [INF] INDx: Transaction index is enabled
2019-04-10 06:47:24.419 [INF] INDx: Committed filter index is enabled
2019-04-10 06:47:24.419 [INF] CHAN: Loading block index...
2019-04-10 06:47:24.420 [INF] CHAN: Chain state (height 0, hash 683e86bd5c6d110d
91b94b97137ba6bfe02dbdbb8e3dff722a669b5d69d77af6, totaltx 1, work 2)
2019-04-10 06:47:24.434 [INF] RPCS: RPC server listening on 127.0.0.1:18556
2019-04-10 06:47:24.435 [INF] AMGR: Loaded 0 addresses from file '/home/user/.bt
cd/data/simnet/peers.json'
2019-04-10 06:47:24.435 [INF] CMGR: Server listening on 0.0.0.0:18555
2019-04-10 06:47:24.436 [INF] CMGR: Server listening on [::]:18555
```

open a NEW window, alice miner:

mine 400 blocks, restart the new btcd window like so:

`btccctl --simnet --rpcuser=kek --rpcpass=kek generate 400`

(it will say error creating directory then the blocks will get mined like so:)

```
File Edit View Terminal Tabs Help
alice Ind x alice wallet x alice funds x
user@cu-cs-vm:~/gocode/dev/alice$ btcctl --simnet --rpcuser=kek --rpcpass=kek generate 400
Error creating a default config file: open /home/user/.btcd/btcd.conf: no such file or directory
[
  "476fe975e93770bc09464154360f36072258c96d2c21d940acde272c2bef2322",
  "36025103e6246b48119056de110a4335f2a25da961b54d90ac6b355ad605b136",
  "7d85d0a5b1002cd5118c99ca398cdf764054a0f162f4229c5ce5b3ee6e51edc1",
  "78d18e6587f77b8da1d38831897f67afa1a6960805142892fa7d1181928f1981",
  "11da315e83ec52916beef9ee6e6869b13766ab19e8976e2f22b1cb24bc97b4b",
  "604f53ce16626bce90074bec60d5406bfcabc23c72cf4a0a8ad01ee0287e511f0",
  "6132008415639a5b258742ec277919270020a1a22c269cdad55f63ea80d1b7f0",
  "11b8160d1e05604ada570e4e652f1d93114725c13f25de684bd3593012fb5e62",
  "74efc84e971d5d3c35efcfc041d882a01d26ec94e6be0971231bd819b176e6bb",
  "26e0adb322e6167830ee24d90edf4da4b05f5778a935dca2bed656b901a2fc73"
]
```

check segwit from alice's wallet

```
btcctl --simnet --rpcuser=kek --rpcpass=kek getblockchaininfo | grep -A 1 segwit
```

```
alice wallet
File Edit View Terminal Tabs Help
alice Ind x alice wallet x alice funds x

  "block_hash": "683e86bd5c6d110d91b94b97137ba6bfe02dbbdb8e3dff722a669b5d6
d77af6",
  "best_header_timestamp": 1401292357,
  "synced_to_chain": false,
  "testnet": false,
  "chains": [
    {
      "chain": "bitcoin",
      "network": "simnet"
    }
  ],
  "uris": null

ser@cu-cs-vm:~/gocode/dev/alice$ lncli-alice newaddress np2wkh

  "address": "rgBgmPryVzATHihNZTShnEiwAArLt8izpb"

ser@cu-cs-vm:~/gocode/dev/alice$ btcctl --simnet --rpcuser=kek --rpcpass=kek ge
blockchaininfo | grep -A 1 segwit
error creating a default config file: open /home/user/.btcd/btcd.conf: no such f
le or directory
  "segwit": {
    "status": "active",
ser@cu-cs-vm:~/gocode/dev/alice$
```

check alice's wallet balance from alice wallet

lncli-alice walletbalance

```
alice wallet
File Edit View Terminal Tabs Help
alice-lnd x alice wallet x alice funds x

    "chain": "bitcoin",
    "network": "simnet"
  },
  "uris": null
}
user@cu-cs-vm:~/gocode/dev/alice$ lncli-alice newaddress np2wkh
{
  "address": "rgBgmPryVzATHihNZTShnEiwAARLt8izpb"
}
user@cu-cs-vm:~/gocode/dev/alice$ btcctl --simnet --rpcuser=kek --rpcpass=kek getblockchaininfo | grep -A 1 segwit
Error creating a default config file: open /home/user/.btcd/btcd.conf: no such file or directory
  "segwit": {
    "status": "active",
user@cu-cs-vm:~/gocode/dev/alice$
user@cu-cs-vm:~/gocode/dev/alice$ lncli-alice walletbalance
{
  "total_balance": "15050000000000",
  "confirmed_balance": "15050000000000",
  "unconfirmed_balance": "0"
}
user@cu-cs-vm:~/gocode/dev/alice$
```

give charlie some money

close the new btcd terminal completely and open a new one, the new new btcd

```
new new btcd
File Edit View Terminal Tabs Help
user@cu-cs-vm:~$ cd $GOPATH
user@cu-cs-vm:~/gocode$
```

from new btcd terminal:

THE ADDRESSES WILL CHANGE EACH TIME YOU RESTART THE NETWORK!!

`cd $GOPATH`

`btcd --txindex --simnet --rpcuser=kek --rpcpass=kek --miningaddr=` put charlie's address here like you did
alice from step above

The screenshot shows a terminal window with two panes. The left pane shows the output of the `btcd` command, which includes the version (0.12.0-beta), block database loading, and the generation of a fresh cipher seed. The right pane shows the wallet initialization process, including the generation of a fresh cipher seed and the display of a 24-word seed phrase. The seed phrase is: 1. able, 2. antenna, 3. razor, 4. civil, 5. bamboo, 6. vanish, 7. color, 8. enrich, 9. brief, 10. rare, 11. wash, 12. violin, 13. team, 14. virtual, 15. jealous, 16. weird, 17. jar, 18. scare, 19. unlock, 20. near, 21. own, 22. swarm, 23. powder, 24. harsh. The terminal also shows the wallet being successfully initialized and the new address being generated: `"address": "rWiruPHYk87td2os1vJMjwVwjdGZsTTuy"`.

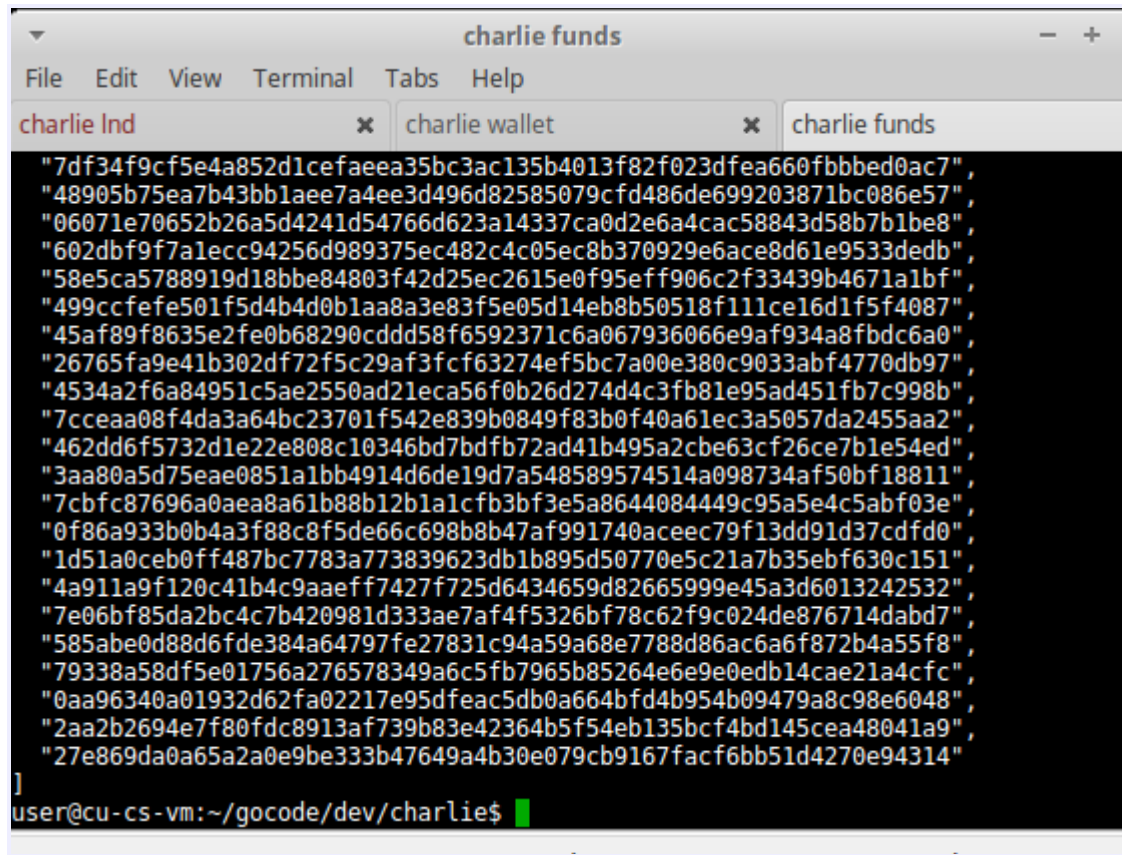
mine blocks for charlie

create new terminal: charlie funds terminal

The screenshot shows a terminal window titled "charlie funds". The terminal displays the command `btctl --simnet --rpcuser=kek --rpcpass=kek generate 100` being entered. The terminal also shows the output of the command, which is a list of 100 block hashes.

`cd ~/gocode/dev/charlie`


```
btccctl --simnet --rpcuser=kek --rpcpass=kek generate 100
```



```
charlie funds
File Edit View Terminal Tabs Help
charlie Ind x charlie wallet x charlie funds
"7df34f9cf5e4a852d1cefaeea35bc3ac135b4013f82f023dfea660fbbbed0ac7",
"48905b75ea7b43bb1aee7a4ee3d496d82585079cfd486de699203871bc086e57",
"06071e70652b26a5d4241d54766d623a14337ca0d2e6a4cac58843d58b7b1be8",
"602dbf9f7a1ecc94256d989375ec482c4c05ec8b370929e6ace8d61e9533dedb",
"58e5ca5788919d18bbe84803f42d25ec2615e0f95eff906c2f33439b4671a1bf",
"499ccfefe501f5d4b4d0b1aa8a3e83f5e05d14eb8b50518f111ce16d1f5f4087",
"45af89f8635e2fe0b68290cddd58f6592371c6a067936066e9af934a8fbdca0",
"26765fa9e41b302df72f5c29af3fcf63274ef5bc7a00e380c9033abf4770db97",
"4534a2f6a84951c5ae2550ad21eca56f0b26d274d4c3fb81e95ad451fb7c998b",
"7cceaa08f4da3a64bc23701f542e839b0849f83b0f40a61ec3a5057da2455aa2",
"462dd6f5732d1e22e808c10346bd7bdfb72ad41b495a2cbe63cf26ce7b1e54ed",
"3aa80a5d75eae0851a1bb4914d6de19d7a548589574514a098734af50bf18811",
"7cbfc87696a0aea8a61b88b12b1a1cfb3bf3e5a8644084449c95a5e4c5abf03e",
"0f86a933b0b4a3f88c8f5de66c698b8b47af991740aceec79f13dd91d37cdfd0",
"1d51a0ceb0ff487bc7783a773839623db1b895d50770e5c21a7b35ebf630c151",
"4a911a9f120c41b4c9aaeff7427f725d6434659d82665999e45a3d6013242532",
"7e06bf85da2bc4c7b420981d333ae7af4f5326bf78c62f9c024de876714dabd7",
"585abe0d88d6fde384a64797fe27831c94a59a68e7788d86ac6a6f872b4a55f8",
"79338a58df5e01756a276578349a6c5fb7965b85264e6e9e0edb14cae21a4cfc",
"0aa96340a01932d62fa02217e95dfeac5db0a664bfd4b954b09479a8c98e6048",
"2aa2b2694e7f80fdc8913af739b83e42364b5f54eb135bcf4bd145cea48041a9",
"27e869da0a65a2a0e9be333b47649a4b30e079cb9167facf6bb51d4270e94314"
]
user@cu-cs-vm:~/gocode/dev/charlie$
```

check his balance from charlie wallet

Incli-charlie walletbalance

```
charlie wallet
File Edit View Terminal Tabs Help
charlie lnd x charlie wallet x charlie funds x
{"block_hash": "683e86bd5c6d110d91b94b97137ba6bfe02dbbdb8e3dff722a669b5d6
9d77af6",
  "best_header_timestamp": 1401292357,
  "synced_to_chain": false,
  "testnet": false,
  "chains": [
    {
      "chain": "bitcoin",
      "network": "simnet"
    }
  ],
  "uris": null
}
user@cu-cs-vm:~/gocode/dev/charlie$ lncli-charlie newaddress np2wkh
{
  "address": "rXpku7DsaQ7aYijXoVbPNdB5rjiouex6GT"
}
user@cu-cs-vm:~/gocode/dev/charlie$ lncli-charlie walletbalance
{
  "total_balance": "5000000000",
  "confirmed_balance": "5000000000",
  "unconfirmed_balance": "0"
}
user@cu-cs-vm:~/gocode/dev/charlie$
```

CONNECTING THE PEER TO PEER NETWORK

Alice and Charlie now have money

Alice will connect to Charlie by going through Bob as a channel

Alice -----> Bob -----> Charlie

get Bob's identity pubkey
from bob wallet

cd ~/gocode/dev/bob

Incli-bob getinfo

```
bob wallet
File Edit View Terminal Tabs Help
user@cu-cs-vm:~/gocode/dev/bob$ lncli-bob getinfo
{
  "version": "0.6.0-beta commit=v0.6-beta-rc3-2-g1fea5b09b28fa6cb0b4a70e597a160449a55aaeb",
  "identity_pubkey": "031ff4f5a0d50eb83e7c404b3b6db6c31797d5af1c49017f42f172c643db7da97b",
  "alias": "031ff4f5a0d50eb83e7c",
  "num_pending_channels": 0,
  "num_active_channels": 0,
  "num_inactive_channels": 0,
  "num_peers": 0,
  "block_height": 100,
  "block_hash": "5394556889fbf1dc7e8cb9f797a8b4377b2bb8651372c82143e93210acd70b90",
  "best_header_timestamp": 1554914585,
  "syncd_to_chain": true,
  "testnet": false,
  "chains": [
    {
      "chain": "bitcoin",
      "network": "simnet"
    }
  ],
  "uris": null
}
user@cu-cs-vm:~/gocode/dev/bob$
$ sudo apt update
Upgrade Installed Software
$ sudo apt full-upgrade
```

from alice wallet connect Alice -----> Bob
(the pubkey will change each time you restart the network)
from alice wallet

Lncli-alice connect

038205ebb13bf0e8f242c53c0e851e83c5019beacfaef8cc4d9b02b3f7c567d75d@localhost:10012

```
bob wallet
File Edit View Terminal Tabs Help
cs-vm:~/gocode/dev/bob$ lncli-bob getinfo
{
  "version": "0.6.0-beta commit=v0.6-beta-rc3-2-g1fea5b09b28fa6cb0b4a70e597a160449a55aaeb",
  "identity_pubkey": "031ff4f5a0d50eb83e7c404b3b6db6c31797d5af1c49017f42f172c643db7da97b",
  "alias": "031ff4f5a0d50eb83e7c",
  "num_pending_channels": 0,
  "num_active_channels": 0,
  "num_inactive_channels": 0,
  "num_peers": 0,
  "block_height": 100,
  "block_hash": "5394556889fbf1dc7e8cb9f797a8b4377b2bb8651372c82143e93210acd70b90",
  "best_header_timestamp": 1554914585,
  "syncd_to_chain": true,
  "testnet": false,
  "chains": [
    {
      "chain": "bitcoin",
      "network": "simnet"
    }
  ],
  "uris": null
}
cs-vm:~/gocode/dev/bob$

alice wallet
File Edit View Terminal Tabs Help
user@cu-cs-vm:~/gocode/dev/alice$ btcctl --simnet --rpcuser=kek --rpcpass=kek getblockchaininfo | grep -A 1 segwit
Error creating a default config file: open /home/user/.btcd/btcd.conf: no such file or directory
{
  "segwit": {
    "status": "active",
    "total_balance": "150500000000",
    "confirmed_balance": "150500000000",
    "unconfirmed_balance": "0"
  }
}
user@cu-cs-vm:~/gocode/dev/alice$ lncli-alice walletbalance
{
  "total_balance": "150500000000",
  "confirmed_balance": "150500000000",
  "unconfirmed_balance": "0"
}
user@cu-cs-vm:~/gocode/dev/alice$ lncli-alice connect 031ff4f5a0d50eb83e7c404b3b6db6c31797d5af1c49017f42f172c643db7da97b@localhost:10012
{
  "total_balance": "150500000000",
  "confirmed_balance": "150500000000",
  "unconfirmed_balance": "0"
}
user@cu-cs-vm:~/gocode/dev/alice$
```

check that Bob and Alice are connected as peers
from alice wallet

Incli-alice listpeers

```
cs-vm:~/gocode/dev/bob$ lncli-bob getinfo
{"version": "0.6.0-beta commit=v0.6-beta-rc3-2-g1fea5b09",
"identity_pubkey": "031ff4f5a0d50eb83e7c404b3b6db6c31797",
"alias": "031ff4f5a0d50eb83e7c",
"num_pending_channels": 0,
"num_active_channels": 0,
"num_inactive_channels": 0,
"num_peers": 0,
"block_height": 100,
"block_hash": "5394556889fbf1dc7e8cb9f797a8b4377b2bb865143db7da97b",
"best_header_timestamp": 1554914585,
"synced_to_chain": true,
"testnet": false,
"chains": [
  {
    "chain": "bitcoin",
    "network": "simnet"
  }
],
"uris": null
}
cs-vm:~/gocode/dev/bob$

user@cu-cs-vm:~/gocode/dev/alice$ lncli-alice connect 031ff4f5a0d50eb83e7c404b3b6db6c31797d5af1c49017f42f172c643db7da97b@localhost:10012
{
  "pub_key": "031ff4f5a0d50eb83e7c404b3b6db6c31797d5af1c49017f42f172c643db7da97b",
  "address": "127.0.0.1:10012",
  "bytes_sent": "329",
  "bytes_recv": "329",
  "sat_sent": "0",
  "sat_recv": "0",
  "inbound": false,
  "ping_time": "3792",
  "sync_type": "ACTIVE_SYNC"
}
user@cu-cs-vm:~/gocode/dev/alice$ lncli-alice listpeers
{
  "peers": [
    {
      "pub_key": "031ff4f5a0d50eb83e7c404b3b6db6c31797d5af1c49017f42f172c643db7da97b",
      "address": "127.0.0.1:10012",
      "bytes_sent": "329",
      "bytes_recv": "329",
      "sat_sent": "0",
      "sat_recv": "0",
      "inbound": false,
      "ping_time": "3792",
      "sync_type": "ACTIVE_SYNC"
    }
  ]
}
user@cu-cs-vm:~/gocode/dev/alice$
```

```
alice Ind x alice wallet x alice funds
user@cu-cs-vm:~/gocode/dev/alice$ lncli-alice openchannel --node key=038205ebb13bf0e8f242c53c0e851e83c5019beacfaef8cc4d9b02b3f7c567d75d --local_amt=1000000
{
  "funding_txid": "b55c1f04bb5beccd863c6e8c53e0d7281867a98a8acc1d9bbb1e904f1a287757"
}
user@cu-cs-vm:~/gocode/dev/alice$ lncli-alice listpeers
{
  "peers": [
    {
      "pub_key": "038205ebb13bf0e8f242c53c0e851e83c5019beacfaef8cc4d9b02b3f7c567d75d",
      "address": "127.0.0.1:10012",
      "bytes_sent": "2077",
      "bytes_recv": "1983",
      "sat_sent": "0",
      "sat_recv": "0",
      "inbound": false,
      "ping_time": "2454",
      "sync_type": "ACTIVE_SYNC"
    }
  ]
}
user@cu-cs-vm:~/gocode/dev/alice$
```

from bob wallet

Incli-bob listpeers

```
user@cu-cs-vm:~/gocode/dev/bob$ lncli-bob listpeers
{
  "peers": [
    {
      "pub_key": "03d4dd41255f1ccb04079aa38259d7a160449a55aeb",
      "address": "127.0.0.1:56626",
      "bytes_sent": "407",
      "bytes_rcv": "407",
      "sat_sent": "0",
      "sat_rcv": "0",
      "inbound": true,
      "ping_time": "1686",
      "sync_type": "ACTIVE_SYNC"
    }
  ]
}
user@cu-cs-vm:~/gocode/dev/bob$

user@cu-cs-vm:~/gocode/dev/alice$ lncli-alice getinfo
{
  "version": "0.6.0-beta commit=v0.6-beta-rc3-2-g1fea5b09b28fa6cb0b4a70e597a160449a55aeb",
  "identity_pubkey": "03d4dd41255f1ccb04079aa38259d6aa13dd58670e63224acc7697376f33445ed8",
  "alias": "03d4dd41255f1ccb0407",
  "num_pending_channels": 0,
  "num_active_channels": 0,
  "num_inactive_channels": 0,
  "num_peers": 1,
  "block_height": 100,
  "block_hash": "5394556889fbf1dc7e8cb9f797a8b4377b2bb8651372c82143e93210acd70b90",
  "best_header_timestamp": 1554914585,
  "syncd_to_chain": true,
  "testnet": false,
  "chains": [
    {
      "chain": "bitcoin",
      "network": "simnet"
    }
  ],
  "uris": null
}
user@cu-cs-vm:~/gocode/dev/alice$
```

in the same way, connect Bob -----> Charlie

from charlie's wallet get his info

lncli-charlie getinfo

from bob wallet

lncli-bob connect

0376d11f24a3f581ad050fee5e51138edfdb8fb2bcf82fe4f97f994d75a8d87bd6@localhost:10013

check that they are connected

from charlie wallet

lncli-charlie listpeers

from bobwallet

lncli-bob listpeers

SENDING PAYMENTS

open a channel from Alice to Bob (alice is funding it)

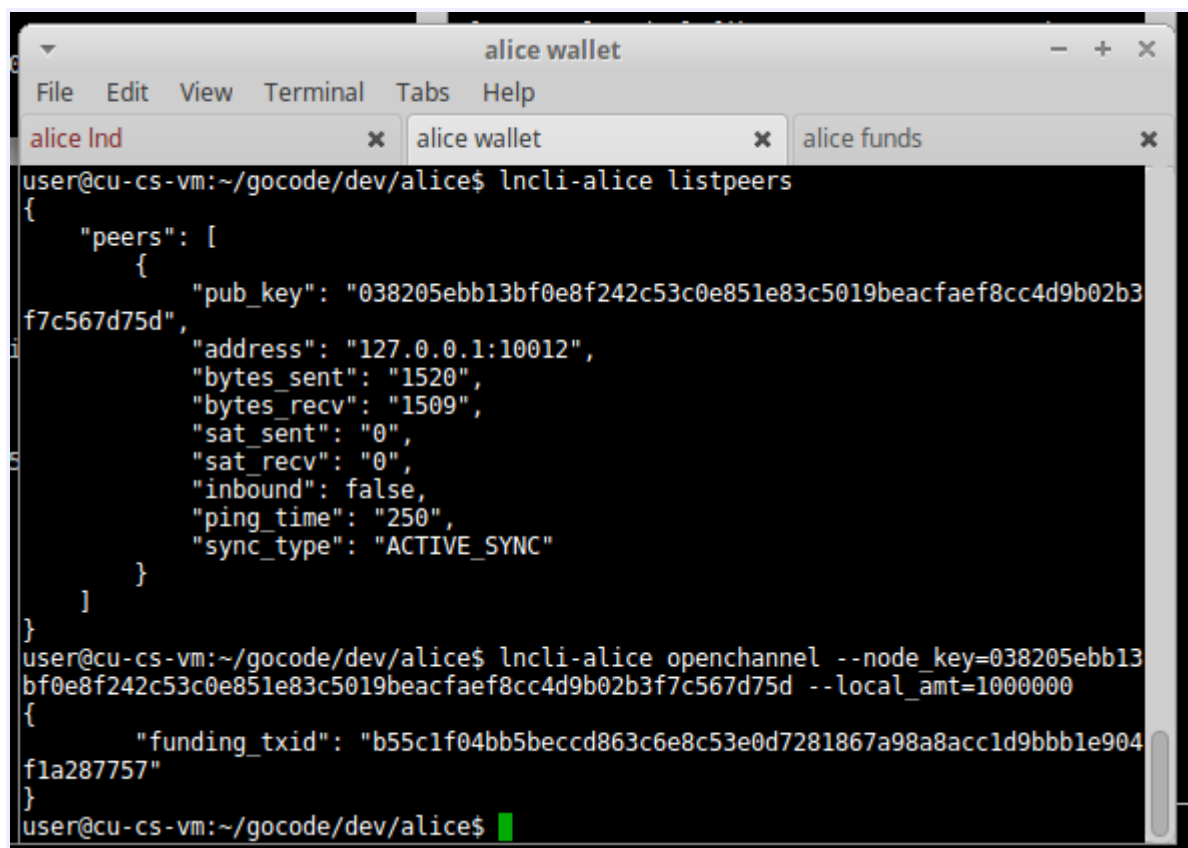
from alice wallet

lncli-alice openchannel

--node_key=038205ebb13bf0e8f242c53c0e851e83c5019beacfaef8cc4d9b02b3f7c567d75d

--local_amt=1000000

^^^use bobs pubkey here



```
alice wallet
File Edit View Terminal Tabs Help
alice lnd x alice wallet x alice funds x
user@cu-cs-vm:~/gocode/dev/alice$ lncli-alice listpeers
{
  "peers": [
    {
      "pub_key": "038205ebb13bf0e8f242c53c0e851e83c5019beacfaef8cc4d9b02b3f7c567d75d",
      "address": "127.0.0.1:10012",
      "bytes_sent": "1520",
      "bytes_recv": "1509",
      "sat_sent": "0",
      "sat_recv": "0",
      "inbound": false,
      "ping_time": "250",
      "sync_type": "ACTIVE_SYNC"
    }
  ]
}
user@cu-cs-vm:~/gocode/dev/alice$ lncli-alice openchannel --node_key=038205ebb13bf0e8f242c53c0e851e83c5019beacfaef8cc4d9b02b3f7c567d75d --local_amt=1000000
{
  "funding_txid": "b55c1f04bb5beccd863c6e8c53e0d7281867a98a8acc1d9bbb1e904f1a287757"
}
user@cu-cs-vm:~/gocode/dev/alice$
```

six blocks have to be mined for it to be valid

from anyone's funding channel

```
btccctl --simnet --rpcuser=kek --rpcpass=kek generate 6
```

make sure they are connected in a channel

from alice wallet

```
lncli-alice listchannels
```

:(

to set specific requests such as how many blocks before mining

```
user@bcu-cs-vm:~/gocode/dev/alices$ lncli openchannel --help
NAME:
  lncli openchannel - Open a channel to a node or an existing peer.
USAGE:
  lncli openchannel [command options] node-key local-amt push-amt
CATEGORY:
  Channels
DESCRIPTION:
  Attempt to open a new channel to an existing peer with the key node-key
  optionally blocking until the channel is 'open'.

  One can also connect to a node before opening a new channel to it by
  setting its host:port via the --connect argument. For this to work,
  the node key must be provided, rather than the peer.id. This is optional.

  The channel will be initialized with local-amt satoshis local and push-amt
  satoshis for the remote node. Note that specifying push-amt means you give that
  amount to the remote node as part of the channel opening. Once the channel is open,
  a channelPoint (txid:vout) of the funding output is returned.

  One can manually set the fee to be used for the funding transaction via either
  the --conf_target or --sat_per_byte arguments. This is optional.
OPTIONS:
  --node_key value      the identity public key of the target node/peer serialized in compressed format
  --connect value       (optional) the host:port of the target node
  --local_amt value     the number of satoshis the wallet should commit to the channel (default: 0)
  --push_amt value      the number of satoshis to give the remote side as part of the initial commitment state, this is equivalent to first opening a channel and sending the remote party funds, but done all in one step (default: 0)
  --block               block and wait until the channel is fully open
  --conf_target value   (optional) the number of blocks that the transaction "should" confirm in, will be used for fee estimation (default: 0)
  --sat_per_byte value  (optional) a manual fee expressed in sat/byte that should be used when crafting the transaction (default: 0)
  --private              make the channel private, such that it won't be announced to the greater network, and nodes other than the two channel endpoints must be explicitly told about it to be able to route through it
  --min_htlc_msat value (optional) the minimum value we will require for incoming HTLCs on the channel (default: 0)
  --remote_csv_delay value (optional) the number of blocks we will require our channel counterparty to wait before accessing its funds in case of unilateral close. If this is not set, we will scale the value according to the channel size (default: 0)
  --min_conf value      (optional) the minimum number of confirmations each one of your outputs used for the funding transaction must satisfy (default: 1)
user@bcu-cs-vm:~/gocode/dev/alices$
```

OPTIONS:

--node_key value the identity public key of the target node/peer serialized in compressed format

--connect value (optional) the host:port of the target node

--local_amt value the number of satoshis the wallet should commit to the channel (default: 0)

`--push_amt` value the number of satoshis to give the remote side as part of the initial commitment state, this is equivalent to first opening a channel and sending the remote party funds, but done all in one step (default: 0)

`--block` block and wait until the channel is fully open

`--conf_target` value (optional) the number of blocks that the transaction **should** confirm in, will be used for fee estimation (default: 0)

`--sat_per_byte` value (optional) a manual fee expressed in sat/byte that should be used when crafting the transaction (default: 0)

`--private` make the channel private, such that it won't be announced to the greater network, and nodes other than the two channel endpoints must be explicitly told about it to be able to route through it

`--min_htlc_msat` value (optional) the minimum value we will require for incoming HTLCs on the channel (default: 0)

`--remote_csv_delay` value (optional) the number of blocks we will require our channel counterparty to wait before accessing its funds in case of unilateral close. If this is not set, we will scale the value according to the channel size (default: 0)

`--min_confs` value (optional) the minimum number of confirmations each one of your outputs used for the funding transaction must satisfy (default: 1)

bob sends an invoice

single hop 2-party Unidirectional multiSig Payment Channel explanation of outcomes	multi hop 2-Party Bidirectional multiSig Payment Channel explanation of outcomes	
--	--	--

Alice -----> Bob

Bob <-----> Charlie

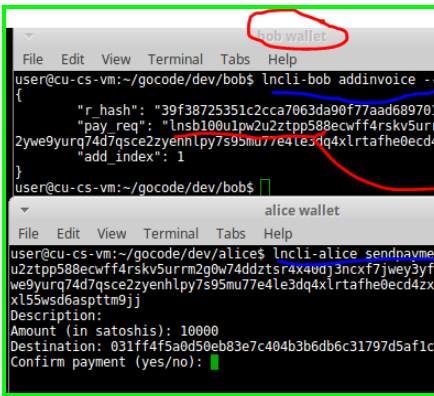
from bob wallet

Incli-bob addinvoice --amt=10000

from alice wallet

Incli-alice sendpayment

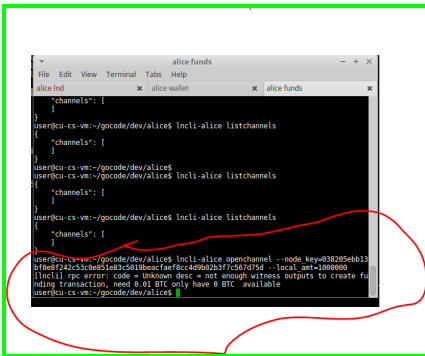
--pay_req=038205ebb13bf0e8f24
2c53c0e851e83c5019beacfaef8cc4
d9b02b3f7c567d75d



```
user@cu-cs-vm:~/gocode/dev/bob$ incli-bob addinvoice --
{
  "r_hash": "39f38725351c2cca7063da90f77aad689701
  "pay_req": "lnsb100u1pw2u2ztp588ecwff4rskv5urr
  Zywe9yurg74d7qsce2zyehnlpy7s9smu77e4te3q4xlrtafhe0ecd4
  "add_index": 1
}
user@cu-cs-vm:~/gocode/dev/bob$

alice wallet
user@cu-cs-vm:~/gocode/dev/alice$ incli-alice sendpayme
u2ztp588ecwff4rskv5urrm2g0w74ddzts74x4uqj3ncxf7jwey3yfi
we9yurg74d7qsce2zyehnlpy7s9smu77e4le3dq4xlrtafhe0ecd4zx
xl55wsd6aspttm9jj
Description:
Amount (in satoshis): 10000
Destination: 031ff4f5a0d50eb83e7c404b3b6db6c31797d5af1c
Confirm payment (yes/no):
```

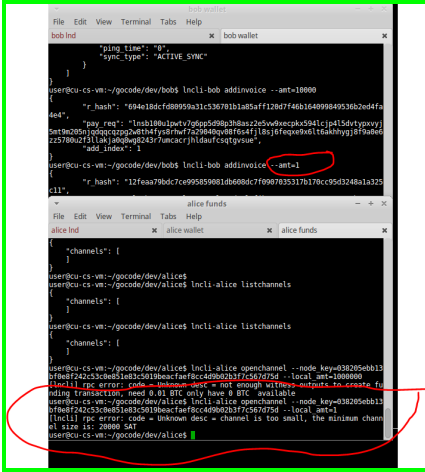
if alice tries to open a channel
without having enough bitcoin:



```
alice Ind
alice wallet
alice funds
{
  "channels": [
  ]
}
user@cu-cs-vm:~/gocode/dev/alice$ incli-alice listchannels
{
  "channels": [
  ]
}
user@cu-cs-vm:~/gocode/dev/alice$ incli-alice listchannels
{
  "channels": [
  ]
}
user@cu-cs-vm:~/gocode/dev/alice$ incli-alice listchannels
{
  "channels": [
  ]
}
user@cu-cs-vm:~/gocode/dev/alice$ incli-alice openchannel --node_key=038205ebb13
ff0ef243cb5e051e83c5019beacfaef8cc4d9b02b3f7c567d75d --local_amt=1000000
lncli rpc error: code = unknown desc = not enough witness outputs to create fulling transaction, need 0.01 BTC only have 0 BTC available
user@cu-cs-vm:~/gocode/dev/alice$
```

not enough witness outputs to
create funding transaction, need

if they try to open a channel that's
got too little transaction amount



```
File Edit View Terminal Tabs Help
bob@bob: ~$ cat bobwallet
{
  "ping_time": "0",
  "sync_type": "ACTIVE_SYNC"
}
}

user@cs-vn: ~/gocode/dev/bob$ lncli-bob addinvoice --amt=10000
{
  "r_hash": "694e18dcfd0959a3c336701ba5aff12b774631640998495362ed4fa
  4e4",
  "pay_req": "lnsb10h1pvtv/gppp5d9p3h0as2z5v0b0ecpk594lcp415dvtypvvy/
  2etth0000/gocodedev/bob14f9d1huf723b0b0q08f0e4f18a18f0e9e01etabkyyaj187ab0e
  2z5780u2f51lakj0b0ag02437umacarjbiduafcsqtgvxue",
  "add_index": 1
}

user@cs-vn: ~/gocode/dev/bob$ lncli-bob addinvoice --amt=1
{
  "r_hash": "12f6aa796d7ce995089081db600d7109078353170170c95d3240a1a325
  c11",
  "add_index": 1
}

alice@alice: ~$ cat alicefunds
{
  "channels": [
  ]
}

user@cs-vn: ~/gocode/dev/alice$ lncli-alice listchannels
{
  "channels": [
  ]
}

user@cs-vn: ~/gocode/dev/alice$ lncli-alice listchannels
{
  "channels": [
  ]
}

user@cs-vn: ~/gocode/dev/alice$ lncli-alice openchannel --node-key=0302050bb13
  f8e8f242c3c0e51e83c501b0acfaefccdd902b37f567d75d --local_amt=1000000
[lncli] rpc error: code = unknown desc = not enough witness outputs to create fu
  nding transaction, need 0.01 BTC only have 0 BTC available




















user@cs-vn: ~/gocode/dev/alice$ lncli-alice openchannel --node-key=0302050bb13
  f8e8f242c3c0e51e83c501b0acfaefccdd902b37f567d75d --local_amt=1
[lncli] rpc error: code = unknown desc = channel is too small, the minimum chan
  nel size is: 20000 sat

user@cs-vn: ~/gocode/dev/alice$
```

channel is too small,

EXPLANATION OF OUTCOMES

Key Definitions for LN multiSig Transactions

Payment Channels	Unidirectional Channel 		Bidirectional Channel 			
Chains	Off-Chain (Strictly Lightning Network) Not recorded on main-chain 	On-Chain (Strictly Bitcoin Blockchain) Recorded on main-chain 	Side-Chain for (2 to n... signatures) for both cooperative and uncooperative parties (variations of transactions between Lightning Network channels and Main Bitcoin Blockchain)  			
Users and Nodes	Public Lightning Network Node 	Users on Public Lightning Network Nodes (current Lightning Network architecture)   		Users on Private Lightning Network Nodes (proposed solution Nodes)   		
Technologies	Time-lock nLockTime 	Hash  SHA-256	Address 	Merkle Root 	Key Pairs 	Payments/ Transactions  Bitcoin
Terminology	Payment channel == address == Smart Contract 2 of 2 multiSig= both parties must consent to creating spends by signing					

LN Unidirectional multiSig

For debtor to pay collector:

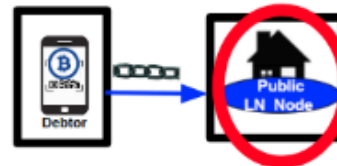
Step 1.
Create a refund transaction.

One or more of the parties create a 2 of 2 multiSig address together on the main-chain where both put in funds. Collector puts in the amount of Bitcoin that will be used to refund, i.e. make change for, the Debtor (can be 0 Bitcoin). Collector signs 1 of 2 multiSig, and sets a Time-lock. Collector's deposit is the safety measure for Debtor. In case Collector is uncooperative, Debtor can sign the 2 of 2 multiSig at the end of the timelock and get all his bitcoin back by keeping Collector's deposit. The debtor puts sufficient funds into the address.



Step 2.
Debtor makes payment(s).

Off-chain, Debtor can send Collector all the payments they want to the address they created before the timelock expires. The only valid balance is the most recent one. There are no main-chain fees. The payments are subtracted from the Debtor's deposit balance in the smart contract.



Step 3.
Channel closes by either signed 2 of 2 multiSig or expired Time-lock.

FULL COOPERATION BY ALL
When Collector signs the 2 of 2 multiSig for any of Debtor's payments, the channel closes and the transaction is committed to the main-chain as a UTXO from the multiSig address and the channel closes.



UNCOOPERATIVE COLLECTOR
If Collector fails to sign the 2 of 2 multiSig by Time-lock, the Debtor keeps the entire Bitcoin amount in their payment channel address and the Collector forfeits their Collector's deposit and the channel closes.



UNCOOPERATIVE DEBTOR
If Debtor fails to broadcast payment updates by the Time-Lock expiration, Collector keeps the entire Bitcoin amount in their payment channel address and the Collector forfeits their refund and the channel closes.

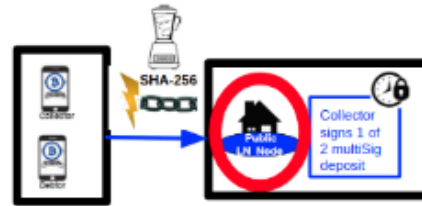


LN Unidirectional multiSig

For debtor to pay collector:

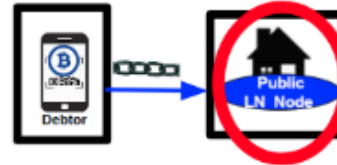
Step 1.
Create a refund transaction.

One or more of the parties create a 2 of 2 multiSig address together on the main-chain where both put in funds. Collector puts in the amount of Bitcoin that will be used to refund, i.e. make change for, the Debtor (can be 0 Bitcoin). Collector signs 1 of 2 multiSig, and sets a Time-lock. Collector's deposit is the safety measure for Debtor. In case Collector is uncooperative, Debtor can sign the 2 of 2 multiSig at the end of the timelock and get all his bitcoin back by keeping Collector's deposit. The debtor puts sufficient funds into the address.



Step 2.
Debtor makes payment(s).

Off-chain, Debtor can send Collector all the payments they want to the address they created before the timelock expires. The only valid balance is the most recent one. There are no main-chain fees. The payments are subtracted from the Debtor's deposit balance in the smart contract.



Step 3.
Channel closes by either signed 2 of 2 multiSig or expired Time-lock.

FULL COOPERATION BY ALL
When Collector signs the 2 of 2 multiSig for any of Debtor's payments, the channel closes and the transaction is committed to the main-chain as a UTXO from the multiSig address and the channel closes.



UNCOOPERATIVE COLLECTOR
If Collector fails to sign the 2 of 2 multiSig by Time-lock, the Debtor keeps the entire Bitcoin amount in their payment channel address and the Collector forfeits their Collector's deposit and the channel closes.



UNCOOPERATIVE DEBTOR
If Debtor fails to broadcast payment updates by the Time-Lock expiration, Collector keeps the entire Bitcoin amount in their payment channel address and the Collector forfeits their refund and the channel closes.



example of uncooperative debtor, expired timelock:

```

bob wallet
File Edit View Terminal Tabs Help
user@cu-cs-vm:~/gocode/dev/bob$ lncli-bob addinvoice --amt=10000
{
  "r_hash": "39f38725351c2cca7063da90f77aad689701d4d57b6519e0c9f49d924
  "pay_req": "lnsb100u1pw2u2ztp588ecwff4rskv5urrm2g0w74ddztsr4x40dj3n
2ywe9yurq74d7qsce2zyenhlp7s95mu77e4le3dq4xlrtafhe0ecd4zxt455djuujsjqhghzrd
  "add_index": 1
}
user@cu-cs-vm:~/gocode/dev/bob$

alice wallet
File Edit View Terminal Tabs Help
user@cu-cs-vm:~/gocode/dev/alice$ lncli-alice sendpayment --pay_req=lnsb100u1pw2
u2ztp588ecwff4rskv5urrm2g0w74ddztsr4x40dj3ncxf7jwey3yfmh8sdqqcqzpgtf3n2u50wf62y
we9yurq74d7qsce2zyenhlp7s95mu77e4le3dq4xlrtafhe0ecd4zxt455djuujsjqhghzrdpwky6v
xl55wsd6aspttm9jj
Description:
Amount (in satoshis): 10000
Destination: 031ff4f5a0d50eb83e7c404b3b6db6c31797d5af1c49017f42f172c643db7da97b
Confirm payment (yes/no): yes
[lncli] rpc error: code = Unknown desc = caveat "time-before 2019-04-10T17:39:35
.348802546Z" not satisfied: macaroon has expired
user@cu-cs-vm:~/gocode/dev/alice$

```

LN Bidirectional multiSig between 2 parties

LN Multidirectional multiSig as Micropayment Channels Using Cryptographic Signatures

(Section 3.3.5 of LN Whitepaper)

For debtor to pay collector and collector to update change as each spending transaction occurs (similar to an open tab on an escrow):

Step 1.Create an address that requires **both party's signatures**Script:
SIGHASH_NOINPUT

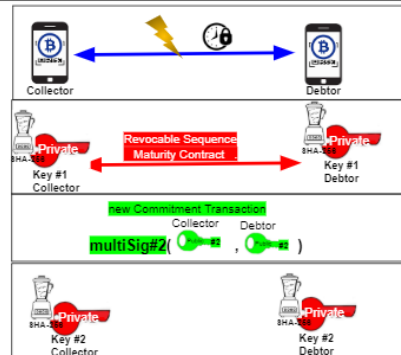
- Both parties agree to create a Commitment Transaction
- Together they create 1 multiSig
- Each party puts in their Public Key of their asymmetric key pair
- The funds are like an escrow to fund the channel and can't be spent
- The collector must match the debtor's UTxO amount (if the debtor skips out, the main-chain is still correctly balanced by the collector's escrow)

Commitment Transaction #1 = multiSigCollector's **Public Key #1**, Debtor's **Public Key #1****Step 2.**

A new Commitment Transaction is created on the side-chain each time the debtor's tab changes

Both parties must sign the same **Revocable Sequence Maturity Contract (RSMC)** with their **private keys**Script:
OP_RELATIVECHECKLOCKTIMEVERIFY

- The tab changes so the UTxO changes
- A new UTxO requires a new Commitment Transaction
- For security reasons (to avoid replay attacks based on a merkle trees property) they have to use new key pairs
- Using their **private keys** both parties sign the same **RSMC**
- New key pairs are generated for each party by adding **+1 to each party's public key**
- The current and verifiable key pairs are the **new keys: Public Keys #2**
Commitment Transaction #2 = multiSigCollector's **Public Key #2**, Debtor's **Public Key #2**
- New private keys** are generated as a result of the new public keys (a feature of asymmetric encryption).
- Both this new tab balance and the previous tab balance (the UTxO's) are broadcast on the sidechain channel only (the Lightning network)



Off-chain, the tab can change infinite number of times before the timelock expires.

There are no main-chain fees for any of these changes while on the sidechain.

The updates happen only on the sidechain.

IF BOTH PARTIES ARE HONEST

- Repeat Step 2 each time there is an update to the tab, i.e. a change in the UTxO amounts
- When both parties want to close the tab (close the sidechain channel) go to Step 3

IF THERE IS A PROBLEM

- The channel is closed and published on the mainchain
- The penalty: the bad party loses ALL their funds to the victim party
- The victim has to wait some set number of main-chain published blocks to get those funds

PUBLISHING THE WRONG BALANCE- BAD ACTOR

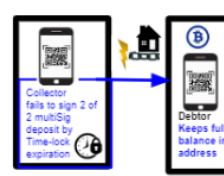
- If anything older than the current and immediate previous UTxO is broadcast, it is considered cheating

**NOT PUBLISHING AN UPDATE- UNCOOPERATIVE DEBTOR**

- If a party doesn't broadcast the latest Commitment Transaction by the timelock expiration, it is considered cheating

**COLLECTOR FAILS TO SIGN- UNCOOPERATIVE COLLECTOR**

- If the collector doesn't sign the 2 of 2 MultiSig to close out by the timelock, they are considered in default



Step 3.

Close the channel.

Script:

REDEEMING THE FUNDS

COOPERATIVE COUNTERPARTIES

- Either party can redeem the funds

- 1) A Commitment Transaction is published by either party
- 2) The counterparty can spend their funds off the mainchain immediately, thus acknowledging this was the correct final ledger
- 3) The original party must wait out the timelock, and can then broadcast a Revocable Delivery Transaction (RDT)
- 4) The original party then waits another round of the same timelock, and the channel is then closed and the funds are redeemable by the original party

***if the original party attempts to broadcast early, the 2nd round timelock is restarted from that point*

IF THEY CHANGE THEIR MINDS ABOUT THE FINAL BALANCES

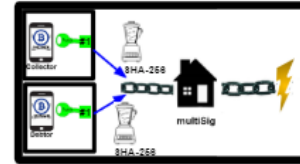
- A Breach Remedy Transaction (BRT) is signed mutually by both parties
- The BRT overrides the BRT
- The same penalty as before applies if there is an attempt to cheat

PUBLISHING THE WRONG BALANCE- BAD ACTOR

- If anything older than the current and immediate previous UTXO is broadcast, it is considered cheating



- To receive payment, the victim must broadcast the correct BRT within the timelock cycle
- If the victim fails to publish the correct BRT within the timelock cycle, the bad actor is able to take all the funds because the channel is open



n-Party Bidirectional multiSig Payment Channel explanation of outcomes

[link back to](#) **SENDING PAYMENTS**

ERRORS WITH THE CODE

ALICE had funds before, then she signed back in with her seed phrase and had lost all her funds

```
alice wallet
File Edit View Terminal Tabs Help
alice lnd x alice wallet x alice funds x

    "inbound": false,
    "ping_time": "0",
    "sync_type": "ACTIVE_SYNC"
  }
]
}
}
user@cu-cs-vm:~/gocode/dev/alice$ lncli-alice openchannel --node_key=038205ebb13bf0e8f242c53c0e851e83c5019beacfaef8cc4d9b02b3f7c567d75d --local_amt=1000000
{
  "funding_txid": "8a7abe3daca05be79fb4bce5c1a498b43cfba9a3918ca87f71381d085834546aa8"
}
user@cu-cs-vm:~/gocode/dev/alice$ lncli-alice listchannels
{
  "channels": [
  ]
}
user@cu-cs-vm:~/gocode/dev/alice$ lncli-alice walletbalance
{
  "total_balance": "0",
  "confirmed_balance": "0",
  "unconfirmed_balance": "0"
}
user@cu-cs-vm:~/gocode/dev/alice$
```

alice and bob open a channel when alice tries to pay it doesn't go through and bob is not notified

```
bob wallet
File Edit View Terminal Tabs Help
bob lnd x bob wallet x

  "pay_req": "lnsb100u1pwtv7g6pp5d98p3h8as22e5vw9xecpkx5941cjp4l5dvtypxvyj5mt9m205njddqczpg2w8th4fys8rhwf7a29040qv08f6s4fjl8sj6feqxe9x6lt6akhhygj8f9a0e6zz5780u2f3llakja0q8wg8243r7umcacrjhldaufcsqtgvsue",
  "add_index": 1
}
user@cu-cs-vm:~/gocode/dev/bob$ lncli-bob addinvoice --amt=1
{
  "r_hash": "12feaa79bdc7ce995859081db608dc7f0907035317b170cc95d3248a1a325c11",
  "pay_req": "lnsb10n1pwtvlqpp5zt1257dacl8fjkzpqmwvzku0uyswq6nz7chpny46vjg5x3jtsqsdqczpg7jm24urv74frkfp253ldeh7vwudnf3rkl9d2ye8q6lr33zp54xy36tfdl8u3mw14era7aarlmercazlgj82hq8zmmr4c2p83z6rggypq2np",
  "add_index": 2
}
user@cu-cs-vm:~/gocode/dev/bob$ lncli-bob addinvoice --amt=10000
{
  "r_hash": "effc41686f46cab6cf698fb54b93d0ac37c31e12dc52d6f2ddc643a6f629b729",
  "pay_req": "lnsb100u1pwtv7g6pp5d98p3h8as22e5vw9xecpkx5941cjp4l5dvtypxvyj5mt9m205njddqczpg2w8th4fys8rhwf7a29040qv08f6s4fjl8sj6feqxe9x6lt6akhhygj8f9a0e6zz5780u2f3llakja0q8wg8243r7umcacrjhldaufcsqtgvsue",
  "add_index": 3
}
user@cu-cs-vm:~/gocode/dev/bob$

alice wallet
File Edit View Terminal Tabs Help
alice lnd x alice wallet x alice funds x

user@cu-cs-vm:~/gocode/dev/alice$ lncli-alice sendpayment --pay_req=lnsb100u1pwtv7g6pp5d98p3h8as22e5vw9xecpkx5941cjp4l5dvtypxvyj5mt9m205njddqczpg2w8th4fys8rhwf7a29040qv08f6s4fjl8sj6feqxe9x6lt6akhhygj8f9a0e6zz5780u2f3llakja0q8wg8243r7umcacrjhldaufcsqtgvsue
[lncli] rpc error: code = Unknown desc = string not all lowercase or all uppercase
user@cu-cs-vm:~/gocode/dev/alice$ lncli-alice sendpayment --pay_req=lnsb100u1pwtv7g6pp5d98p3h8as22e5vw9xecpkx5941cjp4l5dvtypxvyj5mt9m205njddqczpg2w8th4fys8rhwf7a29040qv08f6s4fjl8sj6feqxe9x6lt6akhhygj8f9a0e6zz5780u2f3llakja0q8wg8243r7umcacrjhldaufcsqtgvsue
Description:
Amount (in satoshis): 10000
Destination: 038205ebb13bf0e8f242c53c0e851e83c5019beacfaef8cc4d9b02b3f7c567d75d
Confirm payment (yes/no): y
Confirm payment (yes/no): y
Confirm payment (yes/no): yes
{
  "payment_error": "unable to find a path to destination",
  "payment_preimage": "",
  "payment_route": null
}
[lncli] unable to find a path to destination
user@cu-cs-vm:~/gocode/dev/alice$
```



```
cd $GOPATH
```

```
btcd --txindex --simnet --rpcuser=kek --rpcpass=kek
```

I think To include watchtower flags start up btcd with reindex flag but idk how to do that ,or use this command or something similar

```
getmempoolentry "txid"
```

<https://medium.com/chlunetwork/bitcoin-lightning-network-tutorial-how-to-integrate-lightning-to-a-crypto-project-f48743c8644a>

```
cd $GOPATH
```

```
btcd --txindex --simnet --reindex --rpcuser=kek --rpcpass=kek
```

From

<<https://medium.com/chlunetwork/bitcoin-lightning-network-tutorial-how-to-integrate-lightning-to-a-crypto-project-f48743c8644a>>

<pre>cd \$GOPATH/dev/alice rm -r data log lnd --rpclisten=localhost:10001 --listen=localhost:10011 --restlisten=localhost:8001 --datadir=data --logdir=log --debuglevel=info --bitcoin.simnet --bitcoin.active --bitcoin.node=btcd --btcd.rpcuser=kek --btcd.rpcpass=kek NEW WINDOW lncli --rpcserver=localhost:10001 --macaroonpath=data/admin.macaroon create lncli --rpcserver=localhost:10001 --macaroonpath=~/.gocode/dev/alice/data/chain/bitcoin/simnet/admin.macaroon create lncli-alice newaddress np2wkh rhV6X4fnR7HpTpyEeZunhX2t1HRFgYvt32</pre>	<pre>cd \$GOPATH/dev/bob rm -r data log lnd --rpclisten=localhost:10002 --listen=localhost:10012 --restlisten=localhost:8002 --datadir=data --logdir=log --debuglevel=info --bitcoin.simnet --bitcoin.active --bitcoin.node=btcd --btcd.rpcuser=kek --btcd.rpcpass=kek NEW WINDOW lncli --rpcserver=localhost:10002 --macaroonpath=data/admin.macaroon create # This is the alias in the ~/.bashrc file lncli --rpcserver=localhost:10002 --macaroonpath=~/.gocode/dev/bob/data/chain/bitcoin/simnet/admin.macaroon getinfo #then to control bob we type lncli-bob \$command lncli-bob newaddress np2wkh rpcte1cKpvyoGZPU35bbq8ZjPTWsBCmufv</pre>	<pre>cd \$GOPATH/dev/charlie rm -r data log lnd --rpclisten=localhost:10003 --listen=localhost:10013 --restlisten=localhost:8003 --datadir=data --logdir=log --debuglevel=info --bitcoin.simnet --bitcoin.active --bitcoin.node=btcd --btcd.rpcuser=kek --btcd.rpcpass=kek NEW WINDOW lncli --rpcserver=localhost:10003 --macaroonpath=data/admin.macaroon create lncli-charlie getinfo lncli --rpcserver=localhost:10003 --macaroonpath=~/.gocode/dev/charlie/data/chain/bitcoin/simnet/admin.macaroon getinfo lncli-charlie newaddress np2wkh rYGLpFTiJ4vTKYW1PWUsUExWZ9yMCJz8Qh</pre>
--	--	--

TO GIVE them MONEY RESTART BTCD

MONEY for Alice		
<p>Btcd- close it down completely</p> <p>Open a new window</p> <p>cd \$GOPATH</p> <p>btcd --simnet --txindex --rpcuser=kek --rpcpass=kek --miningaddr=rhV6X4fnR7HpTpyEeZunhX2t1HRFgYvt 32</p>		
<p>Open a new tab in alice- alice miner</p> <p>mine 400 blocks, in her miner window like so:</p> <p>btctl --simnet --rpcuser=kek --rpcpass=kek generate 400</p> <p>check segwit from alice's wallet</p> <p>btctl --simnet --rpcuser=kek --rpcpass=kek getblockchaininfo grep -A 1 segwit</p> <p>check alice's wallet balance from alice wallet</p> <p>Incli-alice walletbalance</p>		

MONEY for Charlie		
<p>Btcd- close it down completely</p> <p>Open a new window</p> <p>cd \$GOPATH</p> <p>btcd --simnet --txindex --rpcuser=kek --rpcpass=kek --miningaddr=rYGLpFTiJ4vTKYW1PWUsUExWZ9yMCJz8Qh</p>		

Open a new tab in charlie- charlie miner
mine 400 blocks, restart her window like so:

```
btctl --simnet --rpcuser=kek --rpcpass=kek generate 400
```

check segwit from charlie's wallet

```
btctl --simnet --rpcuser=kek --rpcpass=kek getblockchaininfo | grep -A 1 segwit
```

check charlie's wallet balance from alice wallet

```
Incli-charlie walletbalance
```

MONEY FOR BOB

```
btcd --txindex --simnet --rpcuser=kek --rpcpass=kek --miningaddr=raL2cCCUBEMVx71uwmsXojcCtgM59XezPG
```

CONNECTING THE PEER TO PEER NETWORK

Open a channel from alice to bob

User	In wallet you are connecting to	In wallet you are connecting from			
alice wallet	Incli-bob getinfo	Incli-alice connect8e62fca34e52963d7cd663fb4fbd192f1bcdd7091654a3a0182c7bca30ae7bf2@localhost:10012			
bob wallet					

Check connection

User	In wallet you are connecting to	In wallet you are connecting from		
bob		Incli-bob listpeers		
alice	Incli-alice listpeers			

Open a channel from bob to charlie

User	In wallet you are connecting to	In wallet you are connecting from			
bob wallet	Incli-charlie getinfo	Incli-bob connect 0263009c90eca398e2664525baf8b43c5f31655ddafb43a3dd50842b4c136f23d6@localhost:10013			

Check connection (bob should have 2 peers)

User		In wallet you are connecting from		
charlie		Incli-charlie listpeers		

Check connection

User	In wallet you are connecting to	In wallet you are connecting from		
bob		Incli-bob listpeers		
charlie	Incli-charlie listpeers			

Check connection (bob should have 2 peers)

User		In wallet you are connecting from		
bob wallet		Incli-bob listpeers		

SENDING PAYMENTS

open a channel from Alice to Bob (alice is funding it)

Alice is paying Bob

```
btcd --testnet --rpcuser=kek --rpcpass=kek
```

From <<https://dev.lightning.community/guides/installation/>>

Open a channel

User	Who will be in the channel	Person who is funding	Person you are paying	Mine 6 blocks	
alice	Alice and bob	alice	bob		
Alice wallet		<pre>Incli-alice openchannel --node_key=028e62fca34e5 2963d7cd663fb4fbd192f1bc dd7091654a3a0182c7bca30 ae7bf2 --local_amt=1000000</pre> <p>This is bob's pubkey</p>			
Anyone's miner window				<pre>btcctl --simnet --rpcuser=kek --rpcpass=kek generate 6</pre>	
Alice wallet		<pre>Incli-alice listchannels</pre>			

charlie Opens a channel to bob

User	Who will be in the channel	Person who is funding	Person you are paying	Mine 6 blocks	
alice	Alice and bob	alice	bob		
Alice wallet		<pre>Incli-charlie openchannel --node_key=035fdf4a 10104a695fdddbdab0 f67c0c07b447af05c62 d6ff84d27922435312 7b1 --local_amt=1000000</pre> <p>This is bob's pubkey</p>			
Anyone's miner window				<pre>btctl --simnet --rpcuse r=kek --rpcpas s=kek generate 6</pre>	
Alice wallet		<pre>Incli-alice listchannels</pre>			

Bob makes an invoice

Make an invoice

User					
Bob wallet	Incli-bob addinvoice --amt=10000				

Mine 6 blocks so the invoice posts

User					
Any miner terminal	btccctl --simnet --rpcuser=kek --rpcpass=kek generate 6				

Alice pays Bob on that channel

Alice sends payment

User				
Alice wallet	Incli-alice sendpayment --pay_req=lnsb100u1pwvykuxpp5payec6w2a2vrh5qe8pr5mlxrz443pr0cndgqey443zreymksg9nsdqccqzpgnjxy8yyyfq45sknzklg5258kasrjdppeuvwyrq9eelcy3q6cmqp4cxnmfaqy085l0upse637yjahguj5ks5qcme4zcz838g8xt7jl8qpr2ala			

Alice closes the channel with Bob

Incli-alice listchannels

Incli-alice closechannel

--funding_txid=bb6ec0889ef37312bb9ca2741c34178bd94113ccd126f05ebb587d6ee3eb21c4

Incli-alice listchannels

THE MISSING PICS ARE IN THE OneNote FILE, WE CAN ADD THEM LATER IF NEED BE

1st Fraud Attempt

Alice and Bob make a few transactions and Alice tries to send an earlier indexed transaction to cheat Bob

Step 1-- Alice gets Bob's pubkey

035fdf4a10104a695fdddbdab0f67c0c07b447af05c62d6ff84d279224353127b1

Step 2-- She uses his info to fund and open a channel with Bob

Incli-alice openchannel

--node_key=035fdf4a10104a695fdddbdab0f67c0c07b447af05c62d6ff84d279224353127b1


```
--local_amt=1000000
```

Verify Alice and Bob have a channel between each other

Bob makes an invoice, note the "add_index" field , it's keeping track of which transaction has happened between them

Alice pays Bob For THIS #2 invoice on that channel

```
Incli-alice sendpayment
```

```
--pay_req=lnsb100u1pwvykuxpp5payec6w2a2vrh5qe8pr5mlxrz443pr0cndgqey443zreymksg9nsdqqcqzpgnjxy8y  
yyfq45sknzklg5258kasrjdppeuvwyrq9eelcy3q6cmqp4cxnmfaqy085l0upse637yjahguj5ks5qcme4zcz838g8xt7jl8qp  
rr2ala
```

Proof it's already paid:

Alice's walletbalance at this point:

Bob's walletbalance at this point:

Bob runs ANOTHER invoice, invoice #3

```
lnsb100u1pwvyhg8pp5ga7e2en45asw6fg5s23ckp4zapvtqe35lftg3d49xv2r9p77melqdqqcqzpgvz9j4n45zgs0j443u  
ca99t0dhdrv3pmjvq5h3kd8m7w0m3vt8jze9uu7sgvr22kwv6ges9fmk5x4px7jy43zlnjuc5mjgmvtg6p5cp7r6tnt
```

Alice tries to resend an old payment to Bob For invoice #3 to fool him, on that same channel

```
Incli-alice sendpayment
```

```
--pay_req=lnsb100u1pwvykuxpp5payec6w2a2vrh5qe8pr5mlxrz443pr0cndgqey443zreymksg9nsdqqcqzpgnjxy8y  
yyfq45sknzklg5258kasrjdppeuvwyrq9eelcy3q6cmqp4cxnmfaqy085l0upse637yjahguj5ks5qcme4zcz838g8xt7jl8qp  
rr2ala
```

Alice wallet balance at this point:

```
Incli-alice listchannels  
Incli-alice closechannel  
--funding_txid=bb6ec0889ef37312bb9ca2741c34178bd94113ccd126f05ebb587d6ee3eb21c4  
Incli-alice listchannels
```

She is not able to close without paying and is punished by having to give Bob all the \$ in the channel

Bob's new balance: