# Lightning Network Parte 3

scritto da Alberto De Luigi | 30 Maggio 2016
< Torna alla Parte 2

## Come funziona il network

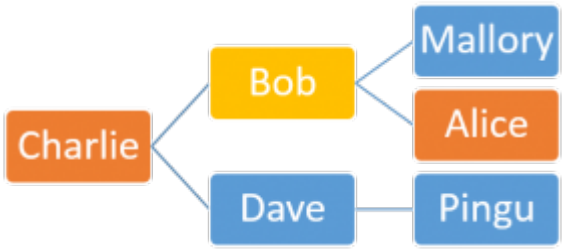
Gli utenti scambiano beni o servizi in cambio di bitcoin in modo continuativo attraverso una rete di utenti, dove ciascuna coppia di utenti è unita da un canale bilaterale, ma la rete nel suo insieme permette di scambiare bitcoin con chiunque senza caricare ulteriori dati sulla blockchain.

## La rete di canali

Ora ipotizziamo che a vendere lo smartphone ad Alice in realtà non è Bob, ma Charlie.

Charlie ha un channel aperto con Bob e Bob ha il channel aperto con Alice che abbiamo visto poc'anzi.

Charlie vende ad Alice. Il software di Charlie (o di Alice) cerca un percorso possibile fra i vari canali aperti. Trova Bob come intermediario. Ma gli step intermedi possono essere illimitati.




Alice pagherà Bob attraverso il channel aperto (esattamente come abbiamo appena visto), sapendo che Bob farà «fluire» i suoi bitcoin direttamente a Charlie. Alice è sicura che Bob sia l'intermediario giusto perché Bob può ricevere il pagamento solo ed esclusivamente se conosce una chiave segreta data da Charlie.

**Infatti Charlie crea una stringa casuale detta «pre-image». Crea l'hash della stringa, che chiamiamo «image» e lo invia ad Alice.** Senza la «pre-image» che possiede inizialmente Charlie, nessuno può ricevere il pagamento di Alice.

## Il processo in breve

Alice ha l'**Image** data da Charlie. Crea quindi una nuova **C**ommitment transaction nel canale aperto con Bob (ipotizziamo sia **C2**), che come prima, se trasmessa, permette sia a Bob che ad Alice di ri-trasmettere nei rispettivi wallet/output i fondi precedentemente allocati in **F**.

Con una differenza: nella nuova **C**ommitment, c'è anche un output di 1btc (il prezzo dello smartphone) che Bob può ricevere solo se conosce la **pre-image**. Altrimenti quell'output di 1btc torna disponibile ad Alice (n. blocchi dopo).

A sua volta, Bob può conoscere la **pre-image** solo se ha già creato una nuova **C**ommitment nel canale con Charlie, per cui si impegna a trasferire 1btc a Charlie.

Questo significa che Charlie ha aperto la «chiusa» con Bob, di conseguenza nel momento in cui Bob apre il la «chiusa» con Alice, inevitabilmente quell'output di 1btc fluisce come acqua direttamente da Alice a Charlie.

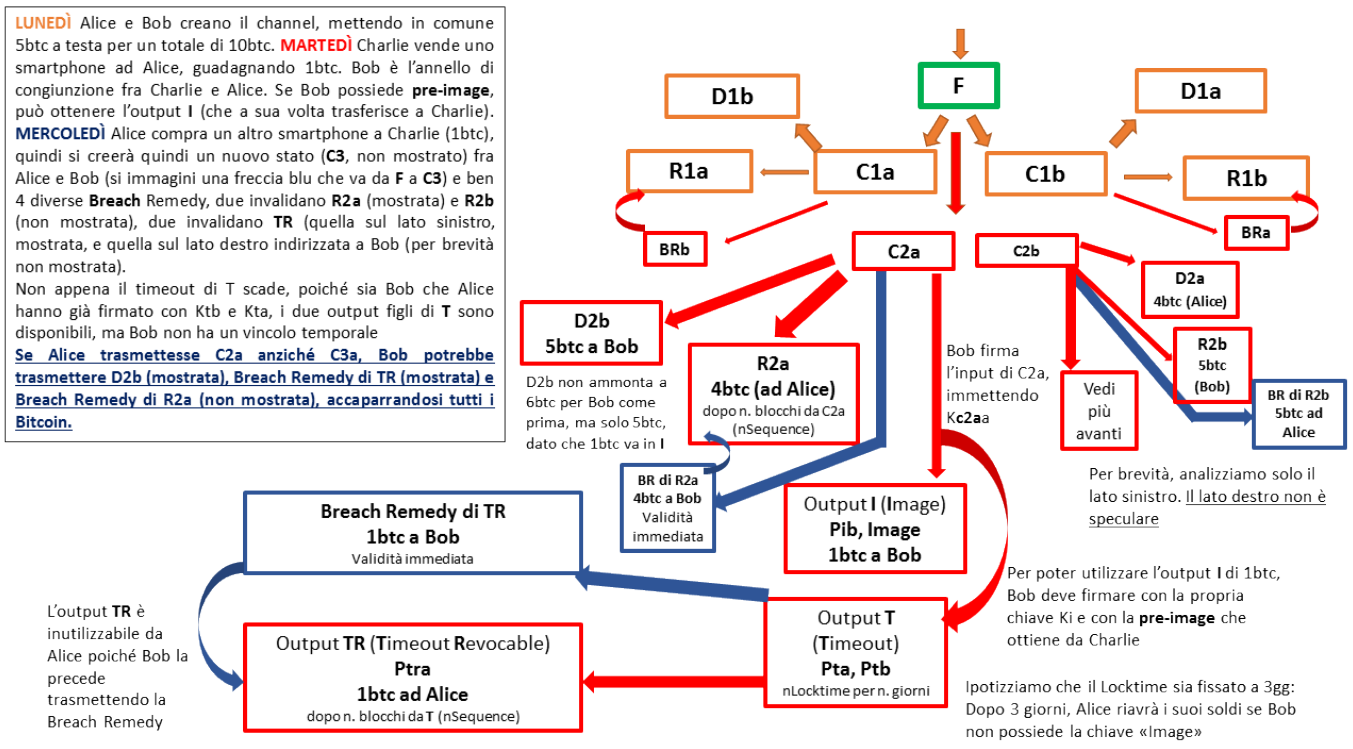
## Più tecnicamente:

Alice possiede l'**image**

Alice utilizza l'**image** per creare il signature script di un nuovo output di **C2**, rendendo disponibile tale output solo con le seguenti «firme» (altrimenti torna ad Alice): la firma di Bob (Kib nel prossimo grafico) e la **pre-image**.

Bob utilizza l'**image** data da Alice per creare il sigscript dell'output di una nuova **C** con Charlie, disponibile solo con la firma di Charlie e la **pre-image**.

Se è vero che Charlie ha venduto lo smartphone ad Alice, comunicandole l'**image**, allora Charlie ha effettivamente la **pre-image** corrispondente a quell'**image**. La usa per ricevere il pagamento di Bob, aprendo la «chiusa» a valle. La **pre-image** diventa così disponibile anche a Bob, che la usa per firmare l'output di **C2**, aprendo la «chiusa» a monte. I bitcoin «scorrono» da Alice a Charlie (o meglio, *la possibilità di trasmettere sulla blockchain la transazione che porta quei bitcoin nel proprio output* «scorre» da Alice a Charlie).

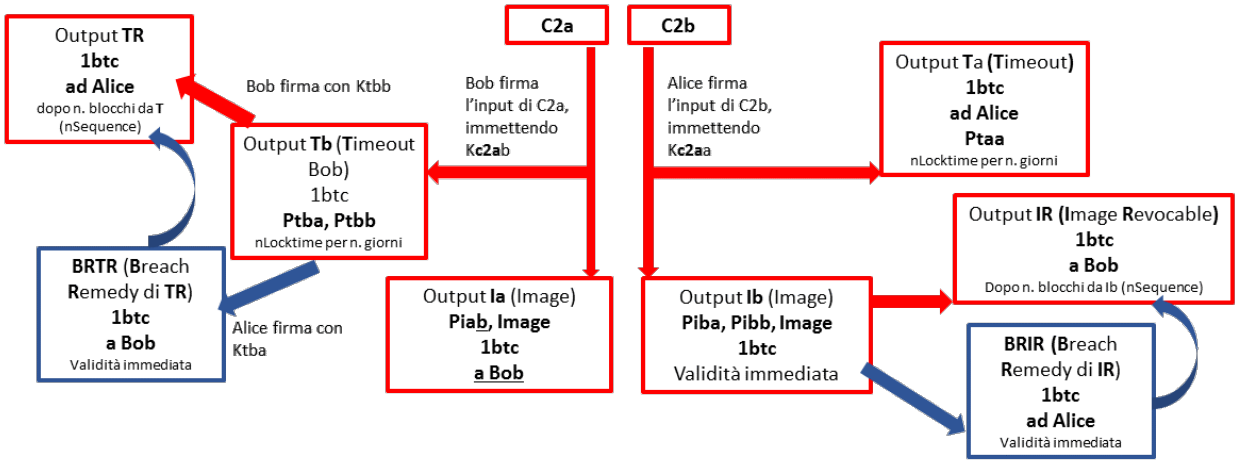



I due rami non sono del tutto speculari. Se Alice trasmette C2a, Bob può avere i bitcoin in **Ia** immediatamente se conosce **pre-image**. Se non la conosce, Alice trasmette **Tb** e di conseguenza la figlia **TR** (Bob ha già firmato in partenza con Ktbb) e così avrà indietro 1btc. In un secondo momento (mercoledì) **TR** si può invalidare grazie a **BRTR.**

Se Bob trasmette **C2b** e conosce la **pre-image**, avrà 1 btc perché trasmette immediatamente **Ib** e la figlia **IR**, da cui dopo n. giorni otterrà 1btc. Se non conosce **pre-image**, non può trasmettere **Ib** e Alice si riapproprierà di 1btc grazie a **Ta**

dopo n. giorni di Locktime. Anche **IR**, infine, è revocabile in un secondo momento grazie a **BRIR.**

Né Bob né Alice hanno convenienza a trasmettere per primi C2 perché in quel caso dovranno attendere n. giorni (n. blocchi) prima di ottenere l'ouptut spendibile.




## Ma come l'output «passa» a Charlie?

Bob ha firmato con Charlie una nuova **C**ommitment esattamente come ha fatto con Alice, e ha creato una scriptsignature per l'output **I** destinato a Charlie che ha la stessa **image** che Alice ha usato per creare la scriptsignature dell'output **Ia** per Bob (si ricordi che l'**image** è stata inizialmente trasmessa da Charlie ad Alice).

Una volta che sono state create tutte le transazioni, se Charlie firma con **pre-image**, 1btc passa da Bob a Charlie e Bob usa **pre-image** per firmare **Ia** e **Ib** con Alice, così che otterrà da lei 1btc.

Se Charlie non firma con **pre-image** e quindi Bob non può usare **pre-image** con Alice, quest'ultima recupera 1btc con **Tb->TR** oppure con **Ta** (in base a chi trasmette per primo **C2**) e Bob in modo analogo recupera 1btc nei confronti di Charlie

# Problemi:

Funding troppo piccola, Congelamento de fondi nel Channel, Rischio di «ritardo», Free riding

## Problemi motivazionali: congelamento e rischio di «ritardo»

C’è un «problema» di scarto temporale, per cui ogni attore coinvolto vede i propri bitcoin momentaneamente «congelati» nel channel.

Se un attore vuole chiudere un channel, quindi portando i propri bitcoin dalla funding transaction multisignature al proprio wallet, non può farlo istantaneamente. Infatti il primo che trasmette ha sempre uno scarto temporale prima di poter utilizzare l’output come nuovo input.

Per fare un esempio, Charlie chiude il Channel con Bob dopo aver acquistato, tramite Bob, uno smartphone da Alice. Bob vuole ora recuperare 1btc da Alice e trasmettere sulla blockchain. Non può farlo immediatamente a meno che Alice trasamette **C2**, ma Alice potrebbe non trasmettere per prima. Quindi Bob è costretto a trasmettere **C2** e ad attendere il vincolo temporale dell’output **IR**.

Inoltre, Bob potrebbe non controllare la blockchain abbastanza spesso, rischiando che Charlie si prenda il suo bitcoin da Bob, mentre Alice si tenga per sé il dovuto, profittando della distrazione di Bob che non trasmette in tempo.

Se Bob vuole assicurarsi dal rischio, dovrebbe utilizzare un software (magari non pienamente affidabile) o un intermediario per trasmettere le proprie transazioni. È costretto quindi a un costo. Ma non sembra sensato che Bob sostenga un costo solo per fare da intermediario fra Charlie e Alice.

È quindi necessario che in caso di pagamenti non unidirezionali ogni nodo intermediario si faccia pagare una

commissione? È più probabile che alcuni utenti (o «banche») agiranno come grandi intermediari, presso i quali la maggior parte degli utenti apriranno un canale, comunicando con gli altri utenti sempre attraverso un unico intermediario, come un nodo della rete molto trafficato.

Bisogna dire tuttavia che c'è un caso in cui i client LN possano funzionare anche senza un third party: il caso di pagamenti unidirezionali. Se infatti il channel viene utilizzato solo per fare pagamenti, e non per ricevere denaro, trasmettere onchain transazioni passate significa solo scontare qualche pagamento all'utente!

## Funding troppo piccola per il pagamento? Necessità di intermediazione

Spesso gli utenti potrebbero incontrare problemi a trovare il «percorso» di intermediari giusto per fare transazioni off-chain tramite il Lightning Network, a causa di funding troppo piccole.

Ad esempio, Charlie e Alice trovano Bob come unico intermediario sul Lightning Network, ma quest'ultimo ha troppo poco a disposizione sulla funding in condivisione con Alice per poter sostenere lo scambio fra i due, che sono così costretti ad aprire una funding fra loro.

È probabile che gli utenti in futuro tengano parti consistenti dei propri bitcoin presso intermediari molto grandi («banche») per poter sostenere ogni tipo di pagamento off-chain.

## Liquidità ed estensione del Network

Il Network può avere o non avere «liquidità». Se ce l'ha, è perché grossi intermediari assicurano che le funding siano abbastanza consistenti da poter coprire la maggior parte delle più comuni transazioni.

Il problema del congelamento dei bitcoin nelle funding

transactions non è un grave problema se:

1 – Nel network c’è buona liquidità

2 – Il network è molto esteso

Infatti la maggior parte degli scambi avverrebbe off-chain e gli utenti non avranno bisogno di «recuperare» sulla blockchain quanto precedentemente condiviso nella funding multisignature.

Se tuttavia il Network è poco esteso o non c’è liqudità, gli utenti si troveranno spesso a dover fare molte transazioni on-chain, perciò saranno meno propensi ad utilizzare le funding e quindi in generale il sistema Lightning Network.

## L’utilizzo di «banche» come intermediari

Sicuramente, l’utilizzo di massa del Lightning Netowork risolve il problema di scalabilità della blockchain. C’è dunque un beneficio collettivo di tutti gli utenti. Ma gli utenti dal punto di vista individuale hanno un beneficio a utilizzare il Network?

L’utente che utilizza il Network, per evitare dei rischi «di ritardo» potrebbe voler pagare un intermediario che controlli eventuali tentativi di frode. E in ogni caso, perché il Network sia esteso e con sufficiente liquidità, sarà probabilmente necessario rivolgersi a grandi intermediari che vorranno delle commissioni.

Di contro, senza l’utilizzo del Network, oltre a pagare i miner per la creazione dei blocchi, potrebbe essere in futuro necessario pagare un «controllore» che ci avverta che gli output non sono già stati spesi (ma questo solo quando la blockchain sarà troppo grande perché i comuni utenti possano scaricarla autonomamente).

Sono i vantaggi del Network percepiti dai singoli individui maggiori dei costi?

Se al beneficio «pubblico» (soluzione del problema di scalabilità) non corrisponde un beneficio «privato» adeguato, sorge un problema di «free riding» e il Network potrebbe avere difficoltà a funzionare.

< Torna alla Parte 2