Lightning Network Parte 1

Vai alla Parte 2 >

Troppi dati vengono caricati sulla blockchain ogni anno.

Se le transazioni in bitcoin annuali arrivassero alla quantità di quelle di Visa, sarebbero necessari 50 terabyte di dati all'anno. Non è quindi sostenibile uno scenario in cui il bitcoin sostituisca la moneta fiat.

Ogni singolo utente dovrebbe scaricare una quantità di dati enorme, a meno di affidarsi ad alcune autorità che scarichino l'intera blockchain, facendo da intermediari e garantendo l'informazione su quali output sono già stati spesi.

La perdita di indipendenza dell'utente e l'accentramento del sistema contraddicono i presupposti ideologici su cui si fonda il bitcoin, e annullano i principali vantaggi rispetto ad altri sistemi monetari.

Il lightning network è un'estensione del Payment channel

Il software BitcoinJ ha implementato il «Payment channel»: un sistema sicuro di transazioni «off chain» (non trasmesse alla blockchain) che possano così ridurre il peso dei dati caricati sulla blockchain e il costo delle transazioni (le commissioni al miner).

Il «Payment channel» è un canale fra due utenti che hanno un rapporto continuativo di pagamenti (come un utente che paga la bolletta ogni mese alla compagnia telefonica). La transazione viene caricata sulla blockchain solo alla fine del «rapporto» (ogni tot pagamenti). È un valido strumento per la creazione di alcuni tipi di contratti.

Il sistema è già funzionante ma non risolve il problema di scalabilità. Un'estensione valida di questo sistema è il

Lightning Network.

Come funziona il Payment Channel

Ovvero come Bob e Alice possono scambiare beni e servizi in cambio di Bitcoin in modo sicuro, ma senza dover trasmettere sulla blockchain ogni singola transazione

La multisignature «funding» transaction

Poniamo che Alice abbia 10 bitcoin e voglia creare un channel payment con Bob.

Alice crea una transazione multisignature ("a firma congiunta") che chiamiamo F (funding transaction), la quale utilizza i 10 bitcoin del wallet di Alice come input e li spedisce come output a un "indirizzo" controllato sia da Alice che da Bob.

Dal punto di vista tecnico, la signature script per muovere i bitcoin da F presenta due public key hash, uno corrispondente alla coppia di chiavi pubblica/privata di Bob, uno alla coppia di chiavi pubblica/privata di Alice

Se F venisse trasmessa alla blockchain, né Bob né Alice potranno spostare autonomamente i bitcoin trasferiti in F come nuovo input per un'altra transazione, poichè per muovere l'output in F sono necessarie le chiavi private di entrambi. Per questo motivo, Alice non firma la transazione per spedire i propri bitcoin in F prima che Bob le abbia dato una garanzia che non la ricatterà con la minaccia di tenere bloccato il denaro in F.

Dal momento che, per ipotesi, sia Alice che Bob hanno interesse a creare un channel payment, Bob è disposto a darle questa garanzia. Viene infatti creata una nuova transazione che chiamiamo R, dove R sta per riscatto o refund transaction (ovvero la transazione che permette ad Alice di riscattare i

propri soldi). R è una transazione che utilizza i 10 bitcoin presenti in F come input e li rispedisce al wallet di Alice

Bisogna qui specificare che è possibile spedire i 10 bitcoin che sono in F anche se F non è ancora stata trasmessa alla blockchain. Infatti, per creare una transazione R collegata a F è sufficiente l'hash della transazione F, che è trasmesso a Bob senza che questi conosca altri dettagli di F. Bob firma il signature script che riporta il public key hash di R comunicato da Alice, immettendo come firma una chiave privata associata a un output ancora non speso: normalmente questo output si trova nella blockchain, ma nel caso di F si trova sul server in cui F è registrata.

La garanzia: una «refund» transaction

Perciò Bob firma la signature della transazione R con la chiave privata in suo possesso che sblocca l'output di F, facendolo divenire input per la transazione R. Bob trasmette R ad Alice.

Alice possiede l'altra chiave privata dell'output di F, può quindi in qualunque momento firmare il trasferimento di fondi e trasmettere R alla blockchain, ri-trasferendo così i 10 bitcoin al suo wallet.

Solo ed esclusivamente dopo che Bob firma R e la trasmette ad Alice, Alice firmerà F. Infatti ora Alice ha in mano le due chiavi private che permettono di spostare in qualunque momento l'output presente in F e spedirlo in R, tornando così al wallet di Alice. In questo momento sia Alice che Bob potrebbero trasmettere F alla blockchain, ma non è necessario farlo.

In questo modo è stato creato un "fondo" (la multisignature F) che funge da base per il channel payment (in gergo, viene chiamato "funding transaction«).

Le varie transazioni (i pagamenti nel channel) avvengono

utilizzando i bitcoin presenti in F.

Il vincolo temporale nLocktime

Alice può creare la transazione R in qualunque momento, ma potrà realmente «impossessarsi» dell'output in R (ovvero utilizzarlo come input per un'altra transazione) solo dopo che sono stati creati un tot di blocchi.

Infatti la transazione R è creata specificando un parametro che non permette ad Alice di usare l'output prima di un certo tempo. Nell'esempio grafico delle prossime slide, se il channel è creato lunedì, Alice non può «impossessarsi» dei bitcoin in R prima di venerdì.

Il periodi di tempo fra lunedì e venerdì è misurato attraverso il parametro "**nLocktime**", ovvero un numero di nuovi blocchi creati, ricordando che viene creato un blocco in media ogni 10 minuti.

All'interno di questo scarto temporale avvengono le transazioni del «Payment channel»

Nota preliminare (solo per i più pedanti – altrimenti saltare alla slide successiva): il vincolo temporale è solo per la REFUND

Nei prossimi grafici è mostrato un payment channel in cui ogni singola transazione (TX 1, 2 o 3) presenta un nLocktime e il relativo output è utilizzabile soltanto a una certa data (rispettivamente giovedì, mercoledì o martedì).

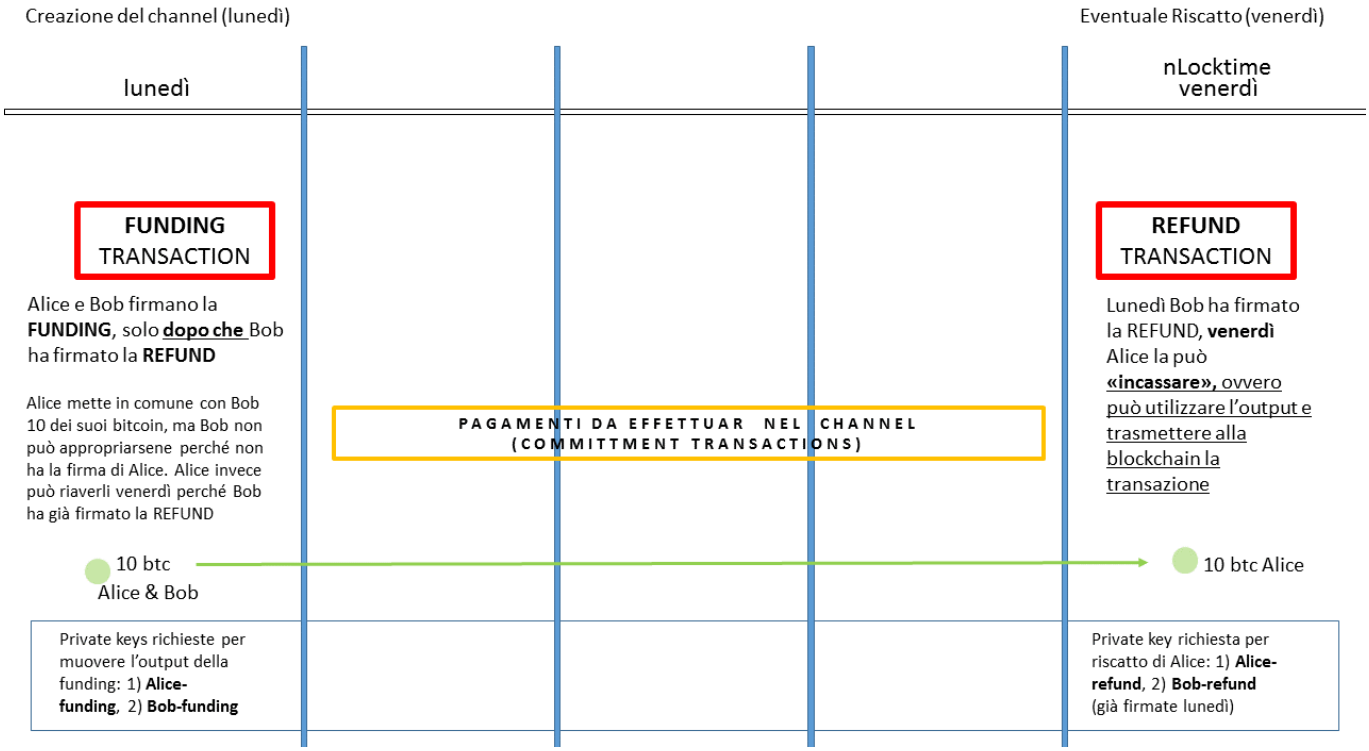
In realtà, nel payment channel implementato in BitcoinJ, solo la Refund ha un vincolo temporale specificato dal parametro nLocktime, poiché le transazioni TX sono salvate sul server, che cancella le transazioni vecchie, rimpiazzandole con le

nuove. Sia Bob che Alice potrebbero chiedere al server in qualunque momento di trasmettere l'ultima transazione avvenuta e incassare i soldi, chiudendo così il channel, ma non possono chiuderlo autonomamente (altrimenti potrebbero trasmettere, per esempio, TX1 anche se è già avvenuta TX2).

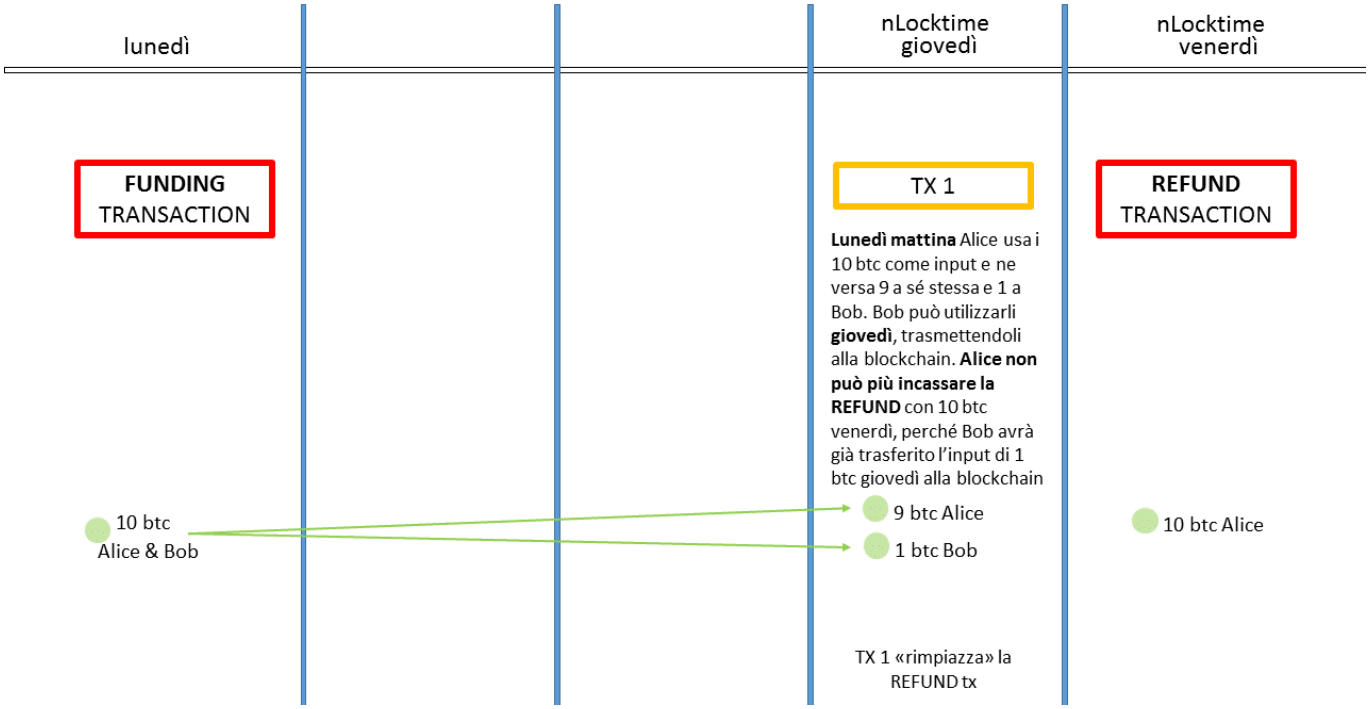
Il sistema non è perfetto perché richiede un'intermediazione: se il server viene distrutto e Bob tramsette F, Alice può recuperare il suo denaro trasmettendo R, che possiede salvata in locale sul suo PC; tuttavia nessuna transazione TX è recuperabile, anche se Bob ha già consegnato lo smartphone ad Alice. Bob può quindi essere «derubato» del suo smartphone in caso succedesse qualcosa all'intermediario BitcoinJ.

È più utile pensare a ogni transazione del channel con un vincolo temporale come mostrato nei grafici. Questo non richiede un'intermediazione e ci avvicina all'idea che sta dietro al Lightning Network.

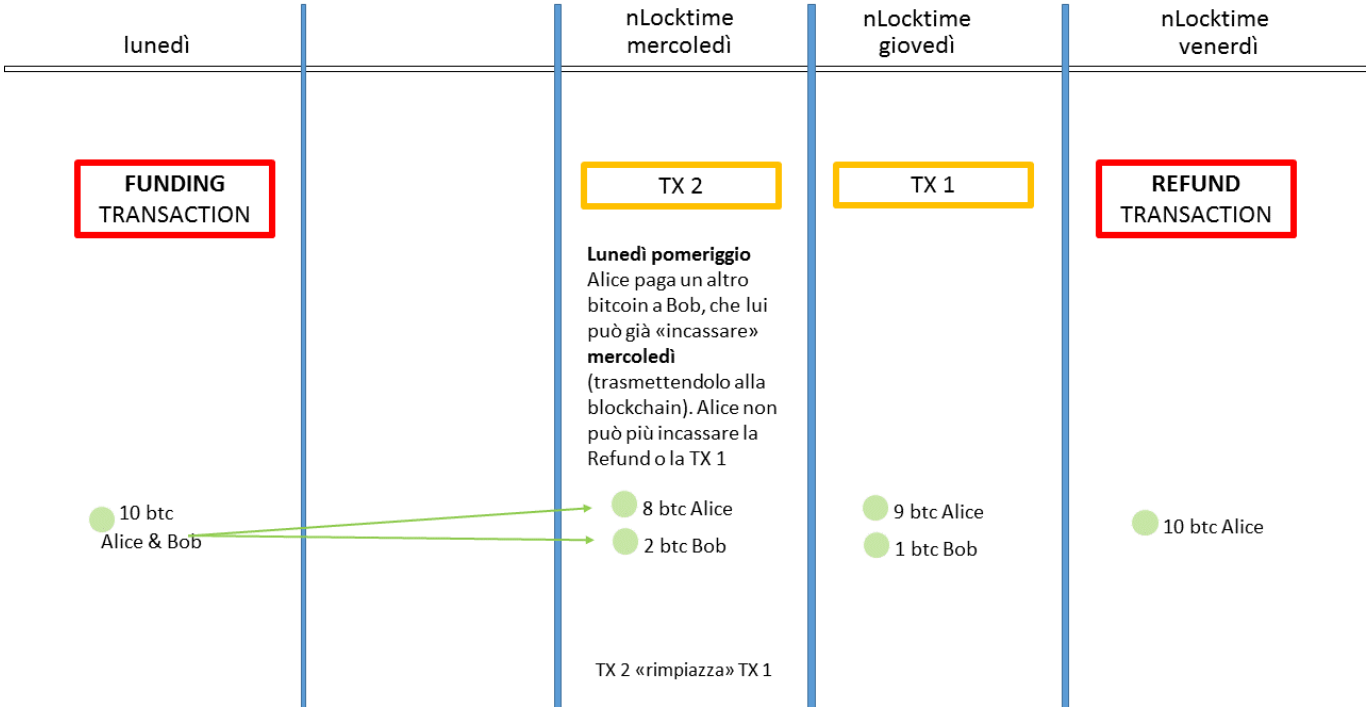
Nota: il termine «commitment transaction» è preso in prestito dalla terminologia Lightning Network.


Creazione del channel (lunedì)
Eventuale Riscatto (venerdì)
lunedì
nLocktime
venerdì
FUNDING
TRANSACTION
Alice e Bob firmano la **FUNDING**, solo **dopo che** Bob ha firmato la **REFUND**
Alice mette in comune con Bob 10 dei suoi bitcoin, ma Bob non può appropriarsene perché non ha la firma di Alice. Alice invece può riaverli venerdì perché Bob ha già firmato la REFUND
PAGAMENTI DA EFFETTUAR NEL CHANNEL
(COMMITTMENT TRANSACTIONS)
REFUND
TRANSACTION
Lunedì Bob ha firmato la REFUND, **venerdì** Alice la può **«incassare»,** ovvero può utilizzare l'output e trasmettere alla blockchain la transazione
10 btc
Alice & Bob
10 btc Alice
Private keys richieste per muovere l'output della funding: 1) **Alice-funding**, 2) **Bob-funding**
Private key richiesta per riscatto di Alice: 1) **Alice-refund**, 2) **Bob-refund** (già firmate lunedì)

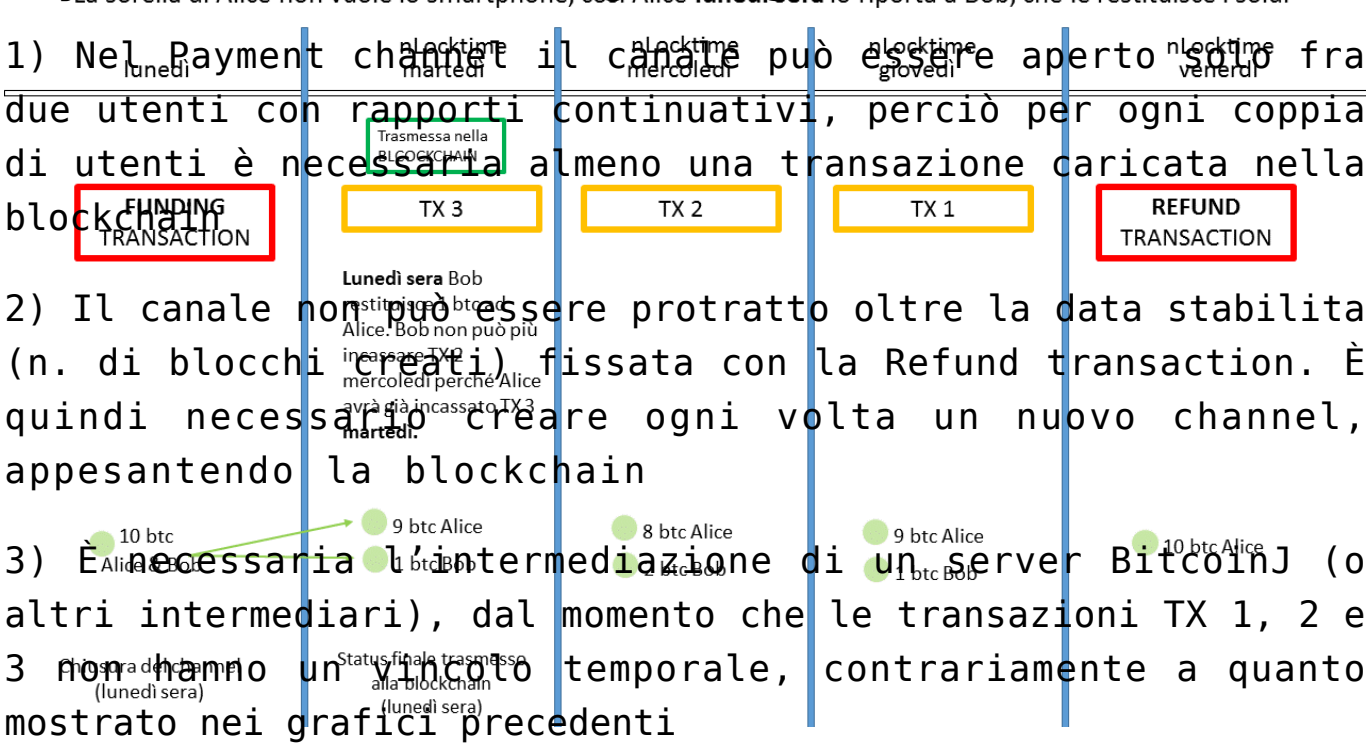
Lunedì mattina Alice va al negozio di Bob e compra uno smartphone da 1 btc



Lunedì pomeriggio Alice torna da Bob, le è piaciuto lo smartphone e ne compra uno anche per sua sorella



Payment channel vs. Lightning Network



1) Nel Payment channel il canale può essere aperto solo fra due utenti con rapporti continuativi, perciò per ogni coppia di utenti è necessaria almeno una transazione caricata nella blockchain

2) Il canale non può essere protratto oltre la data stabilita (n. di blocchi creati) fissata con la Refund transaction. È quindi necessario creare ogni volta un nuovo channel, appesantendo la blockchain

3) È necessaria l'intermediazione di un server BitcoinJ (o altri intermediari), dal momento che le transazioni TX 1, 2 e 3 non hanno un vincolo temporale, contrariamente a quanto mostrato nei grafici precedenti

vs

1) Nel Lightning network il canale è aperto a tempo indeterminato

2) Non è necessaria l'intermediazione di un server BitcoinJ (o altri)

3) Il channel è sostituito da una rete di utenti (potenzialmente, tutti gli utilizzatori di bitcoin)

Vai alla Parte 2 >