POLITECNICO DI TORINO

DIPARTIMENTO DI AUTOMATICA E INFORMATICA - DAUIN
Corso di Laurea Magistrale in Ingegneria Informatica

TESI DI LAUREA MAGISTRALE

Scalabilità della Blockchain: simulazioni e analisi su Lightning Network

Relatore:
Dr. Antonio Vetrò

Laureando:
Giuseppe Di Bartolomeo

Correlatore:
Dr. Marco Conoscenti

Anno Accademico 2018-2019

SOMMARIO

Il grande entusiasmo che negli ultimi anni si è concentrato attorno a temi come *Bitcoin* e *Blockchain* si scontra con un limite di scalabilità: sulla Blockchain non è possibile registrare più di 7 transazioni al secondo. Questo problema rappresenta un freno concreto alle possibilità delle criptovalute di affermarsi come sistema di pagamento elettronico a livello mondiale.

Il contesto da cui nasce questo lavoro è l'esplorazione di soluzioni che possano mitigare il problema della scalabilità della Blockchain. In tal senso le payment channel network rappresentano in letteratura lo strumento più approfondito: esse consentono infatti di costruire reti di canali di pagamento dove è possibile eseguire pagamenti off-chain, i quali non hanno la necessità di essere memorizzati nella blockchain e pertanto non sono soggetti al limite delle 7 transazioni al secondo.

Lightning Network è la forma più nota e comunemente usata di payment channel network e si caratterizza per l'uso dei contratti HTLC per eseguire i pagamenti off-chain in maniera sicura. Lightning Network si trova tuttavia ancora nei suoi primi stadi di sviluppo ed è stato provato che presenta criticità che possono compromettere il suo funzionamento e che quindi meritano di essere investigate. Le principali criticità sono la ridotta capacità dei canali, la quale limita l'importo dei pagamenti che possono essere effettuati, e lo sbilanciamento dei canali (*unbalancing*), il quale rende pagamenti che superano un determinato importo impossibili da effettuare in una certa direzione di un canale che collega due nodi, a causa della scarsa balance (cioè la quantità di bitcoin) posseduta da una delle due parti nel canale di pagamento. Quello dell'unbalancing è un problema che, se non affrontato, può causare il fallimento dei pagamenti. Si tratta di una situazione abbastanza comune in Lightning Network, dove i balance dei nodi sono tipicamente bassi, e che quindi si manifesta tanto spesso quante più alte in media sono le cifre relative ai pagamenti che si intendono effettuare.

L'obiettivo del lavoro di ricerca è mettere a punto un'efficace strategia di ribilanciamento (*rebalancing*) che possa rappresentare un metodo per contrastare questo problema. Il metodo di ricerca adottato è consistito nel simulare diverse strategie di rebalancing utilizzando CLoTH, un simulatore in grado di riprodurre l'esecuzione di pagamenti in una rete che implementa i contratti HTLC e di restituire misure di performance come la probabilità di successo dei pagamenti o il tempo medio impiegato per portarli a compimento.

Le due serie di simulazioni che sono state condotte afferiscono rispettivamente a differenti versioni di rebalancing: attivo e passivo. Il rebalancing attivo si basa sull'eseguire un pagamento ad hoc con lo scopo di ribilanciare i canali, mentre il rebalancing passivo consiste nel regolare la policy sulle commissioni da corrispondere ad un nodo nel momento in cui esso fa da intermediario in un pagamento, con l'obiettivo di incentivare i pagamenti a passare dai nodi con più balance limitando quindi lo sbilanciamento dei canali.

Sono state provate varie strategie di rebalancing attivo e passivo: per quello attivo sono state apportate diverse modifiche al protocollo che esegue il pagamento di ribilanciamento; per quello passivo sono state provate varie funzioni che regolano le commissioni in base alla balance.

Dai risultati ottenuti emerge che la migliore implementazione trovata adotta il rebalancing attivo ed è in grado di ridurre di oltre la metà la probabilità di fallimento dei pagamenti per unbalancing. La particolarità di questa soluzione consiste nell'eseguire l'operazione di rebalancing *prima* del pagamento, qualora ci si accorga che il nodo che deve effettuarlo abbia un balance inferiore all'importo del pagamento. Occorre segnalare però che questa strategia comporta un lieve aumento del tempo medio di esecuzione dei pagamenti rispetto ad un'implementazione senza rebalancing.

L'importanza di tali risultati, che rappresentano il tangibile contributo offerto da questo lavoro, trovano più voce se analizzati in un contesto *service-provider*. Quello del service-provider è uno dei contesti in cui Lightning Network è più largamente utilizzata; qui i pagamenti sono indirizzati dalla maggior parte dei nodi della rete sempre verso gli stessi pochi nodi della rete (i service-providers) i quali vengono pagati come erogatori di qualche servizio. In uno scenario di questo tipo ha senso implementare strategie di ribilanciamento dato che lo sbilanciamento registrato nei canali è notevole. Esso è dovuto al fatto che i pagamenti vengono indirizzati sempre verso gli stessi nodi, e ciò porta i canali attraversati a sbilanciarsi più facilmente. Implementando la migliore soluzione di rebalancing nello scenario service-provider, la probabilità di successo dei pagamenti arriva a registrare un incremento del 32%.

La tesi assolve dunque il proprio compito, ossia porre le basi per migliorare il protocollo di Lightning Network e per rendere più sostenibile una sua adozione per i pagamenti elettronici tramite Bitcoin.

INDICE

ELENCO DELLE FIGURE

ELENCO DELLE TABELLE

1 INTRODUZIONE

Bitcoin si posiziona in prima fila tra le tecnologie che negli ultimi anni ha senza alcun dubbio fatto concentrare su di sé grande interesse e molteplici speranze. L'hype crescente nasce dalla possibilità che Bitcoin possa sostituire in un futuro non troppo lontano la moneta corrente che viene utilizzata ogni giorno; tutto questo grazie ad un sistema decentralizzato di facile accesso che consenta idealmente a chiunque sia in possesso di uno smartphone o qualsiasi dispositivo mobile di effettuare pagamenti, detenendo un proprio portafoglio virtuale, senza la necessità di intermediari [1], come ad esempio una banca, la cui presenza porterebbe a rendere le transazioni un po' più lente e un po' più costose, per via delle commissioni. La possibilità di creare su Internet un tale sistema di pagamento decentralizzato affonda le proprie radici sulla *Blockchain*, ossia un registro pubblico distribuito dove vengono annotate tutte le transazioni.

La distribuzione della Blockchain in più nodi della rete e il meccanismo di sincronizzazione porta tuttavia ad un limitato throughput delle transazioni [2], il che rappresenta un freno alle possibilità della criptovaluta di affermarsi come sistema di pagamento a livello mondiale.

La soluzione più esplorata per tentare di risolvere il problema della scalabilità è rappresentata dalle *payment channel network* [3, 4, 5, 6, 7, 8, 9], le quali attraverso pagamenti *off-blockchain* consentirebbero di effettuare pagamenti senza la necessità di aggiornare di volta in volta il registro pubblico; dunque tali transazioni non sarebbero soggette al limite del throughput della blockchain [10].

Lightning Network (LN) [11] è il payment channel network più comune e studiato per Bitcoin; questa rete conta oltre 10000 nodi, 35000 canali e una capacità complessiva di almeno 832 BTC [12]. Essa si basa su *Hashed Timelock Contract*, o HTLC, ossia un contratto stabilito in un payment channel che consente agli attori coinvolti in un pagamento di impegnare dei fondi per un segreto riscattabile entro un certo tempo tramite una sequenza di pagamenti off-chain [13].

HTLC risulta fondamentale in quanto garantisce il blocco di una transazione se anche solo un nodo tenta di imbrogliare: questo meccanismo consente dunque a due attori di avviare una transazione senza timore, pur non conoscendosi [10].

Come dimostrato in [10], LN presenta tuttavia delle criticità che minano le performance dei pagamenti in termini di probabilità di

successo: la ridotta capacità dei canali, la quale limita l'importo dei pagamenti che possono essere effettuati, e lo sbilanciamento dei canali (*channel unbalancing*), il quale rende pagamenti che superano un certo importo impossibili da effettuare in una certa direzione di un canale che collega due nodi, a causa della scarsa balance (cioè la quantità di bitcoin) posseduta da una delle due parti nel canale di pagamento. Quello dell'unbalancing è dunque un problema che, se non affrontato, può causare il fallimento dei pagamenti.

L'obiettivo di questo lavoro è mitigare gli effetti dell'unbalancing proponendo soluzioni di ribilanciamento (*channel rebalancing*) attivo e passivo, e in generale esplorare strade non battute finora per cercare di migliorare il protocollo LN. Il rebalancing attivo si basa sull'eseguire un pagamento con lo scopo di ribilanciare i canali, mentre il rebalancing passivo consiste nel regolare le commissioni da corrispondere ad un nodo nel momento in cui esso fa da intermediario in un pagamento, con l'obiettivo di spingere i pagamenti a passare dai nodi con più balance limitando quindi lo sbilanciamento dei canali.

Lo studio è stato condotto effettuando dunque delle simulazioni seguite da un'analisi statistica approfondita delle performance di Lightning Network dopo ogni modifica apportata; tali prestazioni sono determinate utilizzando CLoTH, un simulatore di LN che fornisce misure statistiche relative ai pagamenti, come ad esempio la loro probabilità di fallimento e il tempo per completarli.

Sono state provate varie strategie di rebalancing attivo e passivo: per quello attivo sono state apportate diverse modifiche al protocollo che esegue il pagamento di ribilanciamento; per quello passivo sono state provate varie funzioni che regolano le commissioni in base alla balance.

I risultati migliori sono stati raggiunti da una strategia di rebalancing attivo in grado di ridurre di oltre la metà la probabilità di fallimento dei pagamenti per unbalancing: essa ha la peculiarità di essere eseguita, se necessaria, *prima* del pagamento. Questa soluzione garantisce in uno scenario service-provider un miglioramento del 32% nelle performance. Un contesto service-provider prevede che i pagamenti siano indirizzati dalla maggior parte dei nodi della rete sempre verso gli stessi pochi nodi della rete.

La tesi si sviluppa nel seguente modo. I Capitoli 2 e 3 forniscono il background necessario per comprendere Bitcoin, Blockchain e Lightning Network. A ciò si aggiunge qualche dettaglio in più sul simulatore. Il Capitolo 4 presenta il gruppo di simulazioni effettuate discutendone i risultati, prendendo anche in esame lo scenario service-provider. Il Capitolo 5 consente di giungere alle conclusioni, le quali tracciano anche la possibile linea di studio da seguire in futuro.

2 BACKGROUND SU BITCOIN E BLOCKCHAIN

Questo Capitolo ha il compito di fornire le conoscenze necessarie riguardo la tecnologia in analisi in modo da costruire le fondamenta su cui poggerà la successiva discussione.

2.1 BITCOIN

Bitcoin è una collezione di concetti e tecnologie che formano le basi per un ecosistema di denaro digitale. Le unità di valuta chiamate bitcoin (BTC) vengono utilizzate per immagazzinare e trasferire valore tra i partecipanti del network Bitcoin, principalmente attraverso Internet. L'intero protocollo Bitcoin, disponibile come software open source, può essere utilizzato su un'ampia gamma di dispositivi digitali, inclusi laptop e smartphone.

Gli utenti possono trasferire bitcoin sulla rete per fare qualunque cosa sia possibile fare con le valute tradizionali, incluso comprare e vendere beni oppure mandare soldi a persone o organizzazioni. I bitcoin possono essere comprati, venduti e scambiati con altre valute su specifici mercati di cambio, e in questo senso essi rappresentano la forma di denaro perfetta per Internet perchè è veloce, sicura e senza frontiere.

A differenza delle monete tradizionali, i bitcoin sono completamente virtuali [14]. Il dispositivo di un utente su cui viene eseguito un software Bitcoin è chiamato *nodo bitcoin*, esso è connesso ad altri nodi bitcoin entrando a far parte della *bitcoin peer-to-peer network*.

Bitcoin fa uso di crittografia asimmetrica: ciò significa che ogni nodo è in possesso di una *chiave pubblica* e di una *chiave privata* [10]. La chiave pubblica rappresenta lo pseudonimo del nodo e pertanto, come con un indirizzo email, l'utente può condividerla in modo che altri utenti possano inviargli denaro. La chiave privata invece è da tenere ben salda e segreta in quanto permette ad ogni nodo di dimostrare di essere proprietari delle rispettive transazioni, sbloccando la valuta da spendere e trasferendola al nuovo ricevente [13].

Queste chiavi sono generalmente contenute in portafogli digitali chiamati *wallet*, nel dispositivo di ogni utente. Il possesso della chiave per sbloccare una transazione è il solo prerequisito per spendere bitcoin [14].

2.1.1 Le Transazioni

Una *transazione* consente a due parti di scambiare bitcoin [10], comunicando al network che il proprietario di un certo numero di bitcoin ha autorizzato il trasferimento di una parte di essi ad un altro proprietario [14].

Ogni transazione contiene uno o più *input*, ossia debiti verso l'account bitcoin mittente. L'altro lato della transazione presenta uno o più *output*, cioè crediti aggiunti all'account bitcoin destinatario del pagamento. Gli input e gli output (debiti e crediti) se sommati non totalizzano necessariamente lo stesso risultato. Al contrario, l'importo degli output risulta essere di poco inferiore a quello degli input e la differenza corrisponde alla *transaction fee*, ossia una commissione da riconoscere ad un nodo speciale che includerà tale transazione nella blockchain [14].

L'output di una transazione tuttavia non contiene solo la quantità di bitcoin trasferiti, ma anche uno script di blocco che codifica le condizioni necessarie per spenderli. D'altra parte il corrispondente input contiene uno script di sblocco che soddisfa tali condizioni [10].

Per esempio, la Figura 1 mostra una transazione in cui Alice trasferisce 1 bitcoin a Berto. L'input evidenziato fa riferimento all'output di una transazione precedente contenente 1 bitcoin mandato in passato ad Alice, oltre alla firma digitale di Alice, necessaria per poter spendere tale bitcoin. L'output è costituito da 1 bitcoin e da uno script di blocco, che contiene la chiave pubblica di Berto [10].

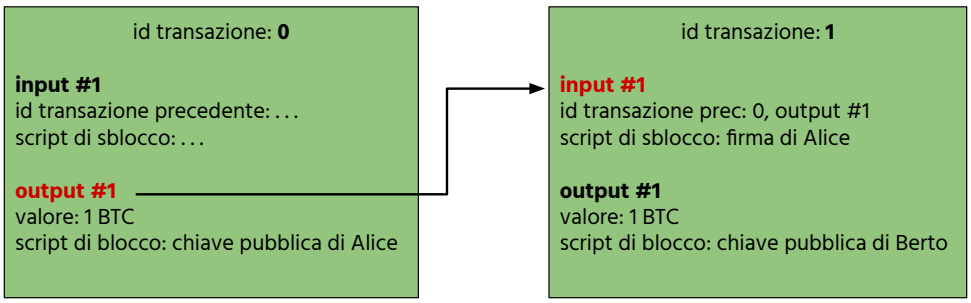

Figura 1: *Transazione Bitcoin*

Il nuovo proprietario Berto può quindi ora spendere questi bitcoin creando un'altra transazione che autorizza il trasferimento ad un nuovo proprietario, e così via, formando una catena di passaggi di proprietà [14].

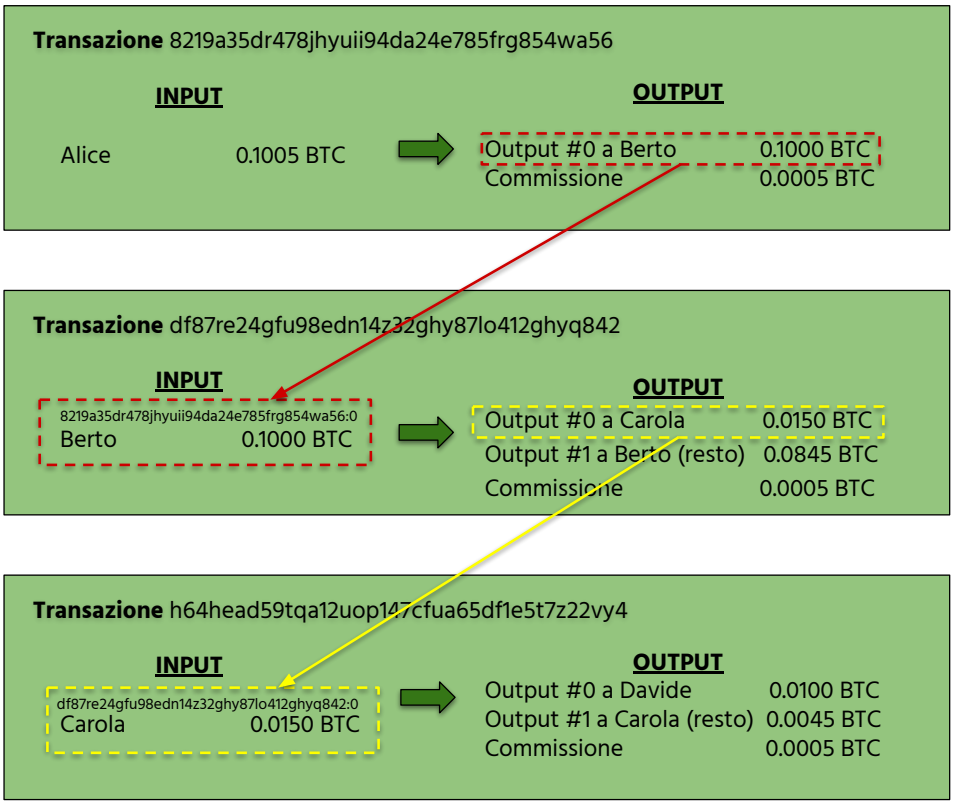

Figura 2: *Una catena di transazioni*

La Figura 2 mostra un esempio di quanto descritto, da cui si coglie la presenza di un terzo output oltre a quelli relativi alla somma pagata e alla commissione, ossia l'eventuale resto che ritorna nelle mani di chi ha generato la transazione.

Le transazioni formano in questo modo una catena, nella quale gli input dell'ultima transazione si riferiscono ad output di transazioni precedenti [14].

2.1.2 Tipologia di transazioni

É importante sottolineare che si possono effettuare diverse tipologie di transazioni più o meno elaborate con script di blocco corrispondenti più o meno complessi.

La forma più comune di transazione, visibile in Figura 3, è un semplice pagamento da un indirizzo ad un altro, il quale spesso include del resto che ritorna al proprietario originario [14].

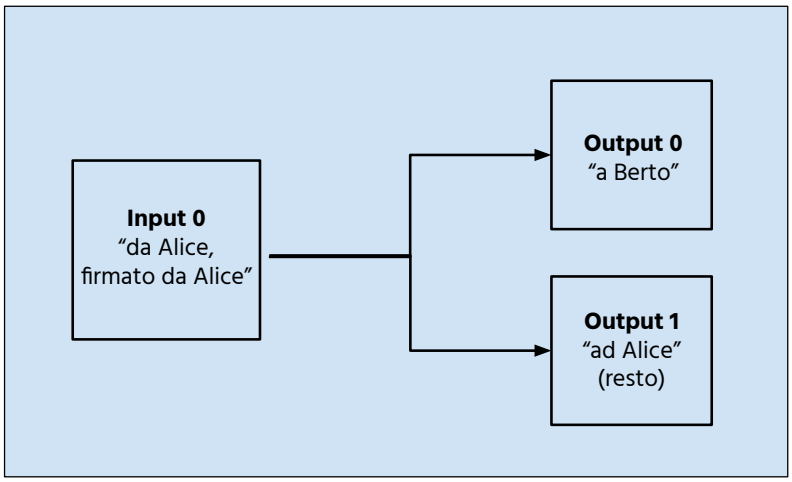

Figura 3: *Transazione comune*

In Figura 4 è rappresentata un'altra forma di transazione che aggrega più input in un solo output. Questa operazione corrisponde allo scambio di una pila di monete e banconote per una banconota singola di valore maggiore. Transazioni come queste sono talvolta generate da applicazioni wallet per far pulizia di transazioni di valore piccolo che sono state ricevute come resto di pagamenti precedenti [14].

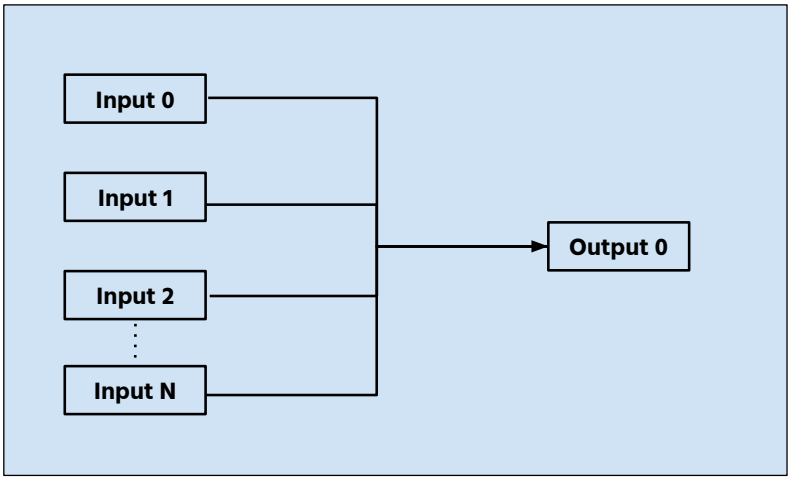

Figura 4: *Transazione aggregatrice*

Infine, la Figura 5 mostra una transazione che distribuisce un input a più di un output i quali rappresentano molteplici destinatari. Questo tipo di transazione è talvolta usato da esercizi commerciali per pagare gli stipendi ai vari dipendenti [14].

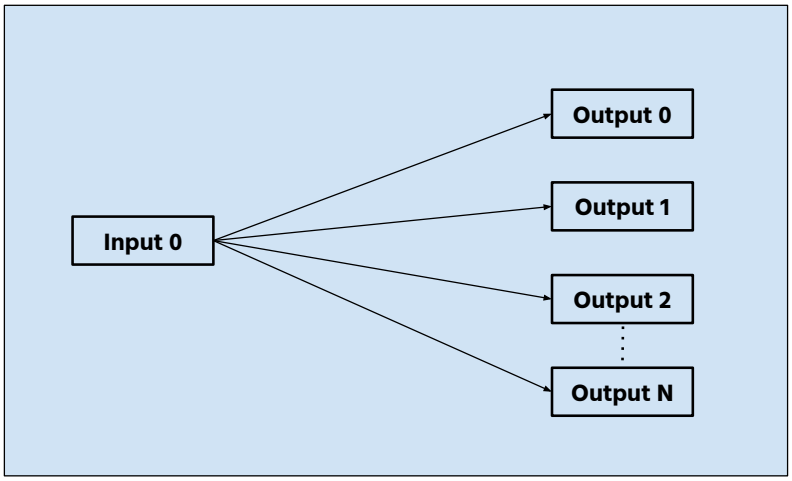

Figura 5: *Transazione che distribuisce fondi*

2.2 BLOCKCHAIN

Il problema principale che affligge la moneta digitale è il *double-spending*, cioè la possibilità di spendere la stessa moneta due volte. Infatti la moneta digitale può essere copiata senza problemi, così come qualsiasi oggetto digitale [10].

Ecash [15], introdotto da David Chaum nel 1983, fu il primo sistema di pagamento digitale che risolse il problema del double-spending. Egli inventò un sistema in cui ogni banconota digitale, identificata da un codice univoco, veniva tracciata in un registro: in questo modo diventava semplice controllare se una certa banconota era stata già spesa oppure no [10].

Tuttavia Ecash era un sistema centralizzato che prevedeva un'entità fidata, come una banca, con il compito di mantenere il registro e di fare i dovuti controlli al fine di scongiurare il double-spending [10].

Anche in Bitcoin il double-spending è risolto memorizzando le transazioni in un registro, chiamato *Blockchain*. A differenza però di Ecash, la blockchain non è gestita da un'entità centrale, bensì dagli stessi nodi Bitcoin. Per questo motivo Bitcoin è il primo sistema di pagamento che garantisce decentralizzazione: esso opera senza necessità di un'entità fidata [10].

Dunque la Blockchain è un registro dove vengono memorizzate tutte le transazioni Bitcoin. Il nome "Blockchain" deriva dalla sua struttura a catena di blocchi, dove ogni blocco racchiude un insieme di transazioni e possiede un hash che fa riferimento al blocco precedente. Questa struttura rende la blockchain resistente alle manomissioni: se la transazione di un blocco venisse modificata illecitamente, ad esempio provando a far risultare uno scambio di denaro in realtà mai

avvenuto, l'hash del blocco subirebbe una modifica che andrebbe a scombinare anche gli hash dei blocchi successivi nella catena [10].

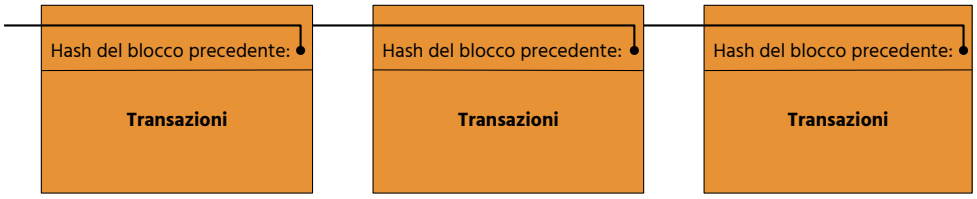

Figura 6: *La Blockchain*

La blockchain è distribuita: parecchi nodi bitcoin mantengono al loro interno una copia di essa.

Grazie ad un *protocollo di consenso distribuito* si garantisce che tutte le repliche di blockchain attraverso i vari nodi si mantengano consistenti memorizzando valori corretti, anche in presenza di guasti o nodi maligni [10].

2.2.1 Il Mining e il consenso

Il primo aspetto da considerare nel protocollo di consenso distribuito è che tutte le nuove transazioni vengono trasmesse in broadcast attraverso la rete Bitcoin peer-to-peer. A ciò si aggiunge che tali nuove transazioni vengono registrate nella blockchain da nodi detti *miner* tramite un particolare processo chiamato *mining* [10].

Ogni nodo Bitcoin può operare come miner, senza aver bisogno di particolari permessi, usando la potenza di calcolo del proprio computer ed entrando così in competizione con altri miner per trovare la soluzione di un complesso problema matematico che, una volta risolto, consenta la registrazione di un blocco di transazioni nella Blockchain. Mediamente ogni 10 minuti un miner riesce a validare le transazioni dei 10 minuti precedenti racchiudendole all'interno di un blocco, e per questo lavoro viene ricompensato con dei bitcoin nuovi di zecca: questo meccanismo decentralizza le funzioni di emissione di valuta che invece, con la moneta fisica, sono solitamente demandate ad una banca centrale [13].

Le transazioni che diventano parte di un blocco e aggiunte alla blockchain vengono considerate *confermate*, e ciò permette ai nuovi proprietari di spendere i bitcoin ricevuti nelle suddette transazioni [14].

Di fatto i miner provvedono alla potenza di calcolo per il network Bitcoin in cambio dell'opportunità di essere ricompensati [13].

In particolare essi ricevono due ricompense per il lavoro di mining:

- i nuovi bitcoin generati ad ogni nuovo blocco registrato nella blockchain;
- le commissioni di tutte le transazioni incluse nel blocco [13].

La soluzione al problema matematico da risolvere, chiamato *Proof of Work* (PoW), viene incluso nel nuovo blocco e rappresenta la prova che accerta l'adeguato sforzo computazionale impiegato dal miner [13].

Tale meccanismo, che prevede una gara per risolvere il PoW ed ottenere il diritto di registrare le transazioni nella blockchain con conseguente ottenimento di una ricompensa, è alla base del modello di sicurezza di bitcoin [13].

Il numero totale di bitcoin emessi e che un miner può conquistare per ogni blocco registrato decresce approssimativamente ogni quattro anni (o precisamente ogni 210.000 blocchi registrati). Si è partiti con 50 BTC per blocco nel Gennaio 2009 per poi passare a 25 BTC per blocco a Novembre 2012. Oggi per ogni blocco vengono garantiti 12.5 BTC. La ricompensa del mining continuerà a decrescere esponenzialmente fino all'anno 2140 circa, quando tutti i 21 milioni di bitcoin saranno stati emessi. Dopo il 2140 nessun nuovo bitcoin verrà emesso [13].

La probabilità di un miner di calcolare il PoW è direttamente proporzionale alla propria frazione di potenza computazionale all'interno della rete. Se l'hardware di un miner possiede l'1% della potenza computazionale complessiva della rete Bitcoin, ciò significa che questo nodo ha la probabilità di produrre un blocco ogni cento [10].

Il PoW è difficile da calcolare e periodicamente tale difficoltà viene incrementata in modo da mantenere i 10 minuti di intervallo tra un blocco registrato e il successivo, controbilanciando così l'aumento di capacità computazionale dei nodi. Tali 10 minuti di intervallo consentono a tutti i nodi della rete di disporre di un tempo sufficiente per aggiornare la loro copia di blockchain, mantenendola quindi consistente [10].

3 SCALABILITÀ E LIGHTNING NETWORK

Il motivo principale per cui il Bitcoin non è ancora diventato un sistema di pagamento largamente utilizzato in tutto il mondo è dovuto ad un problema di scalabilità. Questo capitolo, che ripercorre quanto affrontato nel secondo capitolo di [10], prende in esame tale questione passando poi ad una possibile soluzione, rappresentata da Lightning Network.

3.1 IL PROBLEMA DELLA SCALABILITÀ

Il Bitcoin non prevede che possano essere registrate più di 7 transazioni al secondo [16].

Un throughput del genere è molto basso rispetto alle migliaia di transazioni al secondo che ci si aspetta debba supportare un sistema di pagamento a livello mondiale [17]. Questo rappresenta un grande problema per una possibile espansione di Bitcoin.

Sono sostanzialmente due i parametri che mantengono basso il throughput di transazioni registrate nella blockchain:

- Latenza dei blocchi: come già specificato, essa prevede che nella blockchain venga aggiunto in media un blocco ogni 10 minuti, rispettando in questo modo il protocollo di consenso distribuito.
- Dimensione del blocco: la massima dimensione per ogni blocco, così come dettato nel protocollo Bitcoin, è minore di 4MB.

Il motivo per il quale viene imposto un limite alla dimensione del blocco è mantenere alto il grado di decentralizzazione. In [18] il grado di decentralizzazione di Bitcoin è misurato in base al numero di *nodi funzionanti* nella rete, ossia dei nodi che sono in grado di ricevere nuovi blocchi e validare nuove transazioni. Più alto è il numero di nodi funzionanti, più alto è il grado di decentralizzazione, perchè in questo modo le operazioni di validazione relative a blocchi e transazioni vengono addebitate a molti nodi uniformemente distribuiti, e non a pochi nodi centralizzati.

Un nodo funzionante deve possedere uno storage specifico, oltre a determinate caratteristiche di banda e computazionali:

- Le capacità di storage sono necessarie per salvare l'intera blockchain e l'insieme di output di transazioni non ancora spesi;

- Una buona banda è importante per ricevere e trasmettere transazioni e blocchi;
- Valide capacità computazionali sono essenziali per risolvere il PoW al fine di verificare le nuove transazioni.

La dimensione del blocco influenza la quantità di storage e le risorse di banda e computazionali necessarie per un nodo funzionante. Più grande è tale dimensione e più veloce sarebbe la crescita della blockchain, che a sua volta richiederebbe una maggior capacità di storage. Inoltre, più i blocchi sono grandi e più alta diventerebbe la banda necessaria per ricevere e trasmettere i nuovi blocchi sulla rete.

Per questo motivo, una dimensione maggiore del blocco produrrebbe come conseguenza un più basso livello di decentralizzazione della rete, poichè meno nodi deterrebbero sufficienti risorse per poter partecipare attivamente alla rete.

3.2 POSSIBILI SOLUZIONI

Nella letteratura sono state avanzate alcune alternative al classico protocollo Bitcoin al fine di rimediare al problema della scalabilità.

3.2.1 Ri-parametrizzazione

Tale soluzione consiste nel cambiare quei parametri che pongono un freno al throughput, in particolare riducendo la latenza dei blocchi e incrementando la loro dimensione.

Tuttavia, gli autori di [18] dimostrano che per garantire che il 90% dei nodi sia in grado di risultare funzionante, la dimensione dei blocchi non deve essere maggiore di 4 MB, mentre la latenza dei blocchi non deve scendere al di sotto dei 12 secondi. Questo comporta un throughput massimo di 27 transazioni al secondo.

Nello studio [19] si va oltre e viene mostrato tramite delle simulazioni che la ri-parametrizzazione può aumentare il throughput portandolo fino ad un massimo di 60 transazioni al secondo: superando questo limite si rischia di mettere a repentaglio la sicurezza del sistema.

Per questo motivo la ri-parametrizzazione non può essere considerata una valida soluzione al problema della scalabilità, tenendo conto che i risultati raggiunti con tale tecnica sono ancora insufficienti rispetto ai valori che si punta a conquistare.

3.2.2 Protocolli di consenso alternativi

Un'altra categoria di soluzioni al problema della scalabilità è rappresentata dall'utilizzo di protocolli di consenso alternativi, i quali possono migliorare o sostituire il Proof of Work.

Per esempio, *Bitcoin-NG* [20] rappresenta un miglioramento del protocollo Bitcoin; mantenendo inalterato il grado di decentralizzazione, Bitcoin-NG riduce la latenza dei blocchi incrementando il relativo throughput, fino ad un massimo definito dalle caratteristiche dei nodi e della rete.

Una scelta differente è data da *Proof of Stake* (PoS) [21] [22] che invece è una vera e propria alternativa al PoW. In PoS, la possibilità di un nodo di aggiungere un blocco alla blockchain è direttamente proporzionale alla quantità di bitcoin in suo possesso; non è quindi necessario risolvere nessun complesso problema matematico. Pertanto, visto che lo sforzo computazionale richiesto col PoW viene risparmiato, l'approccio PoS consente di aumentare il throughput delle transazioni. Le proprietà di sicurezza di PoS sono oggetto di ricerche non ancora concluse.

3.2.3 Payment Channel Network

Un approccio radicalmente diverso per risolvere il problema della scalabilità è rappresentato dalle Payment Channel Network, ossia reti di pagamento definite all'interno della rete Bitcoin. L'obiettivo di queste reti è quello di effettuare quante più transazioni possibile senza coinvolgere la blockchain; in tal modo il throughput di queste transazioni chiamate *off-blockchain* (o *off-chain*) non verrebbe limitato in quanto tali pagamenti non necessitano di essere memorizzati nella blockchain.

Questa soluzione si erge tra quelle più promettenti poichè non va a compromettere significativamente la sicurezza e la decentralizzazione della blockchain.

Le payment channel network possono rendere possibili pagamenti che, rispetto alle transazioni registrate nella blockchain, risultano essere:

- più economiche, visto che sono richieste commissioni inferiori rispetto alle transazioni *on-blockchain*;
- più veloci, dato che non necessitano di essere registrate nella blockchain;
- più riservate, poiché non sono visibili nel registro pubblico della blockchain.

3.3 PAYMENT CHANNEL NETWORK

Questa sezione fornisce i concetti fondamentali per consentire la comprensione delle payment channel network, necessaria ad esplorare una delle sue declinazioni più note, ossia Lightning Network.

Una payment channel network è una rete di canali di pagamento dove è possibile effettuare pagamenti off-blockchain, i quali non sono soggetti ad alcun limite nel throughput.

3.3.1 Payment channel

Il *payment channel*[1] è un canale bidirezionale alle cui estremità ci sono due nodi che possono scambiare pagamenti off-blockchain.

I due canali monodirezionali che insieme formano un payment channel prendono singolarmente il nome di *edge*.

Un semplice esempio mostrato in Figura 7 è quello di un payment channel tra i due nodi Alice e Berto. Questo canale è dunque costituito da due edge: uno che va da Alice a Berto e un altro che va da Berto ad Alice. Il primo passo che le due parti devono fare per aprire il payment channel è finanziarlo con parte dei loro bitcoin, per esempio sia Alice che Berto potrebbero allocare 0.5 BTC.

In questo caso, il payment channel ha una *capacità* complessiva di 1 BTC, data dalla somma dei bitcoin allocati dai due nodi, mentre per quanto riguarda il *balance*, ossia l'ammontare dei bitcoin che ogni nodo impegna in un canale, sia quello di Alice B_A che quello di Berto B_B contano 0.5 BTC.

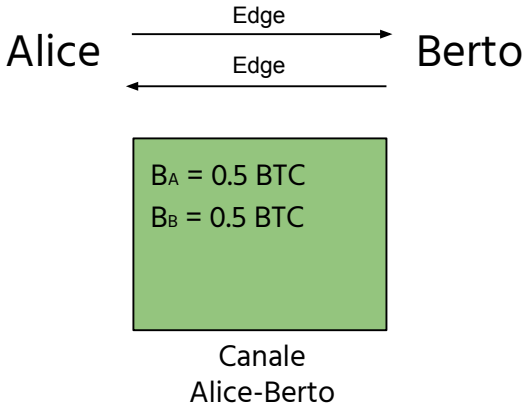

Figura 7: *Payment Channel*

Una volta che il canale è stato istituito, qualora Alice voglia pagare 0.1 BTC a Berto, anziché comunicare la transazione affinché venga

1 Da qui in avanti chiamato nel testo anche solo *canale*.

registrata nella blockchain, è sufficiente che le due parti aggiornino i rispettivi balance in modo da riflettere il trasferimento di bitcoin. Quindi, dopo il pagamento di 0.1 BTC da Alice a Berto, il balance della prima viene aggiornato a 0.4 BTC, mentre quello del secondo a 0.6 BTC.

3.3.2 Payment channel network

I payment channel da soli possono mettere in contatto solo due nodi. Per evitare però che ogni coppia di nodi sia costretta ad aprirne uno per scambiare pagamenti off-blockchain, più payment channel possono essere connessi insieme in modo da creare un *payment network*. Tale rete consente dunque a due parti, anche se non direttamente collegate, di scambiarsi denaro off-blockchain attraversando una catena di payment channel.

Un possibile esempio in questo caso è quello mostrato in Figura 8 di un payment network in cui Alice ha un canale con Berto, Berto con Carola e Carola con Davide. Anche se Alice e Davide non sono connessi direttamente tramite un payment channel, Alice può inviare un pagamento off-blockchain a Davide attraversando i canali intermedi, ossia il canale tra lei e Berto, il canale tra Berto e Carola, e infine il canale tra Carola e Davide.

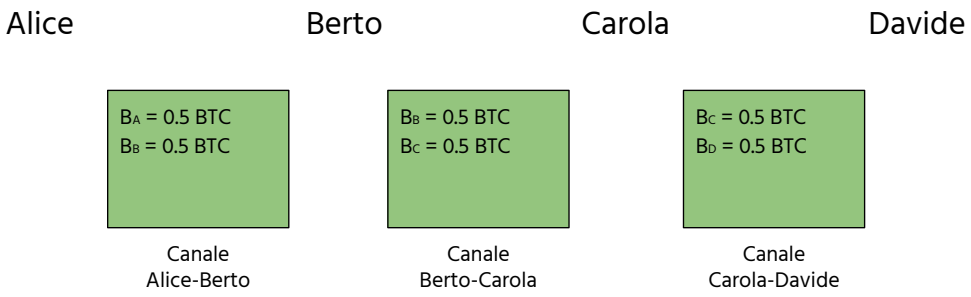

Figura 8: *Payment Network*

3.4 LIGHTNING NETWORK

Lightning Network è la forma più nota e comune di payment channel network, e il suo protocollo specifica come aprire e gestire i payment channel e come effettuare pagamenti off-chain in un payment network.

3.4.1 Apertura di un canale

Per poter istanziare un payment channel in Lightning Network, viene creata una transazione di apertura, detta *funding transaction*, tra Alice e Berto. Tale transazione ha come input i fondi allocati dalle due parti: ad esempio vengono allocati 0.5 BTC da Alice, i quali costituiranno il rispettivo balance iniziale nel canale; la stessa cosa viene fatta da Berto, che quindi avrà anche lui un balance iniziale di 0.5 BTC. L'output totale della funding transaction è 1 BTC, che costituisce la capacità complessiva del canale.

La funding transaction viene dunque inviata alla blockchain e, una volta confermata, il canale è considerato aperto.

Alice e Berto adesso possono creare una nuova tipologia di transazione, chiamata *commitment transaction*, la quale restituisce l'informazione relativa all'ammontare dei balance dei due nodi che si affacciano sul canale. Gli output di questa transazione sono due: uno restituisce 0.5 BTC ad Alice, l'altro 0.5 BTC a Berto.

É importante evidenziare come la commitment transaction non viene inviata alla blockchain (a meno che una delle due parti non voglia chiudere il canale).

3.4.2 Esecuzione di un pagamento

Per eseguire un pagamento sul canale, Alice e Berto creano una nuova commitment transaction off-chain con i balance aggiornati. Dunque quando Alice vuole pagare 0.1 BTC a Berto, questi creano un nuovo commitment transaction che assegna 0.4 BTC ad Alice e 0.6 BTC a Berto.

Questa operazione viene fatta completamente off-chain, quindi senza la necessità di interagire con la blockchain, e ciò viene ripetuto per ogni pagamento tra Alice e Berto, indipendentemente dalla direzione.

A differenza delle transazioni on-chain, in quelle off-chain non esiste un limite al throughput delle transazioni che non dipenda esclusivamente dalla velocità di rete di Alice e Berto.

3.4.3 Chiusura di un canale

Quando le due parti di un payment channel vogliono chiudere il canale, essi trasmettono alla blockchain il commitment transaction più recente, che restituisce i balance dei rispettivi proprietari. Una volta che tale transazione viene confermata, il canale può venir chiuso.

Questa operazione può essere svolta anche se una delle due parti non risponde da tempo o risulta non essere collaborativa. La controparte in tal caso può anche da sola inoltrare alla blockchain la com-

mitment transaction più recente e in questo modo recuperare i propri fondi.

3.4.4 Sanzioni

Dopo che Alice ha pagato 0.1 BTC a Berto e questi hanno creato una nuova transazione di commit per aggiornare i rispettivi balance, Alice potrebbe essere tentata ad ingannare Berto inviando alla blockchain la transazione di commit precedente al pagamento, che le attribuirebbe più bitcoin rispetto a quelli riconosciuti dopo l'ultima transazione.

Per questo motivo il protocollo di Lightning Network consente di punire tali comportamenti andando innanzitutto a revocare le vecchie transazioni di commit quando ne viene creata una più nuova, e punendo il nodo che tenta di inviare alla blockchain una transazione revocata.

Qualora una delle due parti tenti di imbrogliare l'altra in tal modo, quest'ultima potrà far propri tutti i fondi del canale, compresi quelli relativi al balance della parte disonesta, che quindi perderà tutti i propri averi.

Il rischio di incorrere in questa sanzione rappresenta una garanzia per gli onesti: il nodo di un canale non ha bisogno di fidarsi della sua controparte. Anche se Alice è disonesta e tenta di ingannare Berto, quest'ultimo potrà sempre recuperare i propri fondi.

Le pena, tuttavia, non è automatica: è necessario che Berto rimanga costantemente all'erta e monitori la blockchain per accertarsi che Alice non invii una transazione di commit revocata.

3.5 HTLC

In Lightning Network l'instradamento dei pagamenti in un percorso, detto *route*, attraverso più payment channel, avviene mediante un contratto chiamato *Hashed Timelock Contract* (HTLC). HTLC garantisce tutte le parti che risultano coinvolte all'interno della route: queste infatti non potranno mai perdere i loro bitcoin, anche se gli altri nodi nel percorso si comportano in maniera disonesta o scorretta.

HTLC implementa un meccanismo di pagamenti off-chain condizionali all'interno dei payment channel. Quando Alice mette in piedi un HTLC con Berto per un valore di 0.1 BTC, ciò significa che Alice pagherà a Berto 0.1 BTC solo se quest'ultimo le invierà un particolare valore R (detto *preimage*) entro un certo timeout; altrimenti il pagamento non avrà luogo.

Come già spiegato in precedenza, la transazione di commit nel ca-

nale tra Alice e Berto, dove sia il balance di Alice che quello di Berto ammonta a 0.5 BTC, possiede due output: uno conferisce 0.5 BTC ad Alice e l'altro 0.5 BTC a Berto.

Nel momento in cui viene stabilito un HTLC di 0.1 BTC tra Alice e Berto, viene creata una nuova transazione di commit costituita stavolta da tre output: un output che assegna 0.4 BTC ad Alice; un output che attribuisce 0.5 BTC a Berto; un output HTLC che contiene 0.1 BTC. Lo script di blocco dell'output HTLC è costituito da due parti:

- Una parte contiene l'hash di *R*. Questo fa in modo che Berto debba necessariamente conoscere *R* per poter correttamente calcolare il suo hash e quindi riscattare la somma associata all'HTLC;
- L'altra parte contiene un timelock, il quale implementa un timeout (ad esempio un giorno[2]) alla cui scadenza, se Berto non avrà ancora trasmesso *R*, Alice potrà riprendere possesso dei fondi temporaneamente bloccati nell'HTLC.

Con tale meccanismo, se Berto mostra *R* entro un giorno, l'HTLC è soddisfatto e quindi il pagamento può essere effettuato, pertanto il valore associato all'HTLC viene trasferito al balance di Berto. A questo punto viene creata una nuova transazione di commit, dove l'output di HTLC viene eliminato, un output associa 0.4 BTC ad Alice e un altro output assegna 0.6 BTC a Berto.

Al contrario, qualora Berto non dovesse presentare *R* entro un giorno l'HTLC fallisce, quindi il pagamento viene annullato e il valore associato all'HTLC viene trasferito al balance di Alice. Dunque viene creata una nuova transazione di commit dove l'output di HTLC viene eliminato, un output attribuisce 0.5 BTC ad Alice e l'altro output 0.5 BTC a Berto.

Si sottolinea che anche i pagamenti tramite HTLC vengono effettuati totalmente off-chain; allo stesso modo sono off-chain le transazioni di commit prodotte quando un HTLC viene creato, fallisce o termina con successo.

A ciò si aggiunge che anche i pagamenti mediante HTLC sono protetti contro eventuali disonesti. Se una parte trasmette alla blockchain una transazione di commit revocata, la controparte può impossessarsi di tutti i fondi del canale, compresi quelli associati all'HTLC.

2 In realtà il timelock viene espresso come numero di blocchi: dopo un certo numero di blocchi l'output di una transazione può essere speso.

3.5.1 Pagamenti multihop tramite HTLC

Figura 9: *Pagamento multihop tramite HTLC*

La Figura 9 mostra il pagamento di 0.1 BTC attraverso più canali tra Alice e Davide. Per poter far questo, viene stabilito un HTLC in ogni canale attraversato dal pagamento, e tutti questi HTLC richiederanno lo stesso preimage R per essere soddisfatti.

Innanzitutto, Davide genera R e consegna tramite Internet ad Alice l'hash di R. In seguito, un HTLC con l'hash di R viene stabilito in tutti i canali coinvolti.

In particolare vengono stabiliti i seguenti HTLC:

- un HTLC di 0.1 BTC nel canale tra Alice e Berto con un timelock di 3 giorni;
- un HTLC di 0.1 BTC nel canale tra Berto e Carola con un timelock di 2 giorni;
- un HTLC di 0.1 BTC nel canale tra Carola e Davide con un timelock di 1 giorno.

In questo modo sono realizzati gli HTLC tra Davide e Alice. Davide mostra R a Carola entro 1 giorno e riceve i 0.1 BTC dall'HTLC con Carola. Carola mostra R a Berto entro 2 giorni e riceve i 0.1 BTC dall'HTLC con Berto. Berto mostra R ad Alice entro 3 giorni e riceve i 0.1 BTC dall'HTLC con Alice. Alla fine di questo procedimento risulterà che 0.1 BTC sono stati trasferiti da Alice a Davide.

É importante sottolineare che fra Davide ed Alice gli HTLC hanno visto incrementare i propri timelock. Ciò garantisce che ogni parte coinvolta abbia tempo a sufficienza per conoscere R e ottenere i fondi spettanti.

Nel caso in cui R non venisse mai rivelato, gli HTLC fallirebbero e, dopo la scadenza dei timelock, i fondi associati a ciascun HTLC tornerebbero ai rispettivi proprietari in ciascun canale. Se Davide non rivela R entro un giorno, ossia entro la scadenza del proprio timelock, Carola riprenderà possesso dei fondi temporaneamente da lei allocati per l'HTLC. Questa operazione può essere svolta in collaborazione con Davide, creando una nuova transazione di commit off-chain; tuttavia, se Davide si dovesse rivelare non collaborativo, Carola potrebbe comunque in maniera indipendente inviare la transazione di commit più recente alla blockchain e dunque chiudere il canale. A questo punto Carola propagherebbe indietro l'informazione sull'avvenuto fallimento, affinchè anche gli HTLC degli altri canali coinvolti possano terminare alla stessa maniera.

Occorre evidenziare che se R non viene rivelato, i fondi negli HTLC rimangono bloccati fino alla scadenza dei rispettivi timelock.

Un ultimo dettaglio da segnalare è relativo alle commissioni. Quando Alice vuole trasferire 0.1 BTC a Davide, lei aggiunge a tale somma una piccola quantità di bitcoin da usare come ricompensa per tutti i nodi intermedi (Berto e Carola) per la loro collaborazione all'inoltro del pagamento attraverso i propri canali.

Infine in Lightning Network il routing, ossia la ricerca del percorso migliore per poter effettuare un pagamento tra due nodi non direttamente collegati, viene demandato al mittente del pagamento: si parla quindi di *source routing*.

3.6 IL PROBLEMA DELLO SBILANCIAMENTO

Nell'esempio in Figura 9 ogni nodo si trova ad inviare un pagamento di una cifra inferiore rispetto al balance a disposizione nel canale da cui si intende effettuare il pagamento. Talvolta questa condizione però non viene rispettata, specie quando i canali sono sbilanciati.

Un canale si dice *sbilanciato* nel momento in cui si fa consistente la differenza tra il balance di un nodo e quello della controparte: considerato il canale tra Alice e Berto, esso risulta sbilanciato verso Berto se ad esempio il balance di Alice ammonta a 0.1 BTC mentre quello di Berto a 0.9 BTC.

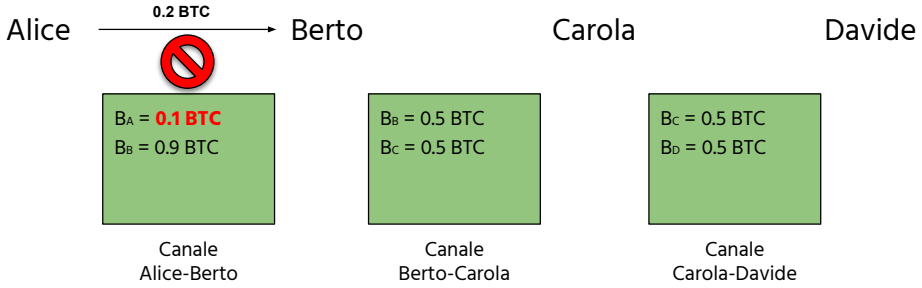

Figura 10: *Canale sbilanciato*

In una situazione del genere, illustrata in Figura 10, se Alice intende effettuare un pagamento di 0.2 BTC verso Davide, non potrà farlo per mancanza di fondi sufficienti. Tale condizione potrebbe interessare il nodo che intende effettuare il pagamento (Alice) oppure un nodo che fa da intermediario nel pagamento (Berto o Carola).

Questo problema, chiamato sbilanciamento del canale (*channel unbalancing*), se non affrontato decreta inevitabilmente il fallimento del pagamento. Si tratta di una situazione abbastanza comune in LN, dove i balance dei vari nodi sono tipicamente bassi, e che quindi si manifesta quanto più alte in media sono le cifre relative ai pagamenti che si intendono effettuare.

Obiettivo del prossimo capitolo è proprio quello di sperimentare più soluzioni al fine di mitigare tale fenomeno: esse vanno tutte sotto il cappello della strategia chiamata ribilanciamento del canale (*channel rebalancing*).

3.7 IL SIMULATORE CLOTH

Nelle prossime sezioni, per poter effettuare le analisi di performance del protocollo Lightning Network, è stato utilizzato il simulatore

CLoTH descritto nel terzo capitolo di [10], il quale simula l'esecuzione di pagamenti in una payment channel network basata su HTLC.

CLoTH è scritto in C e riceve in input la definizione di una payment channel network e una lista di pagamenti che vengono eseguiti durante la simulazione. Il flusso di esecuzione del simulatore si divide in tre fasi, come mostrato in Figura 11:

- Nella fase di pre-processing CLoTH riceve in ingresso dei file utili al processamento ottenuti da un generatore di rete e di pagamenti;
- Nella fase di simulazione vengono eseguiti i pagamenti in LN. I dettagli relativi a rete e pagamenti sono riportati nei file ricevuti in input nella fase precedente;
- Nella fase di post-processing CLoTH restituisce le misure di performance legate ai pagamenti simulati, le quali vengono aggregate in maniera opportuna in un apposito file. Esso raccoglie statistiche relative ai pagamenti (ad esempio la probabilità di successo/fallimento dei pagamenti e il tempo medio di completamento di un pagamento).

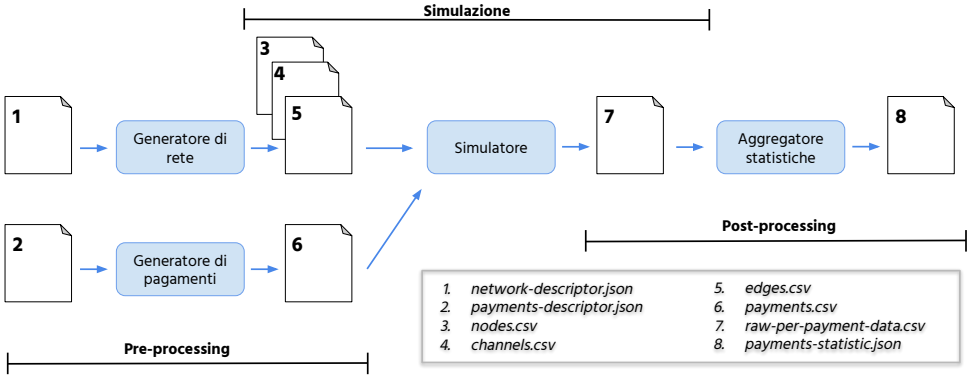

Figura 11: *Flusso di esecuzione di CLoTH*

3.7.1 Strutture dati

La payment channel network e i pagamenti sono rappresentati nel simulatore dalle strutture dati presenti nella Figura 12. Un `canale` connette due `nodi` (ciascuno identificato da un ID) e ha una certa capacità. Poichè il canale è bidirezionale, questo può essere attraversato in entrambe le direzioni, dal nodo1 al nodo2 oppure dal nodo2 al nodo1. Un canale quindi possiede due edge, ciascuno dei quali rappresenta una direzione del canale. Un `edge` contiene: l'ID del canale

a cui l'edge appartiene; il balance disponibile nella direzione indicata dall'edge; feeBase e feeProp, ossia la commissione base e proporzionale richieste affichè un pagamento possa attraversare l'edge (feeBase è costante, feeProp varia in base all'importo del pagamento); il timelock relativo all'HTLC impostato nella direzione dell'edge; l'importo minimo consentito per i pagamenti inoltrati nella direzione dell'edge. Un `pagamento` è descritto da mittente, destinatario, importo e istante di tempo in cui viene avviato nel simulatore.

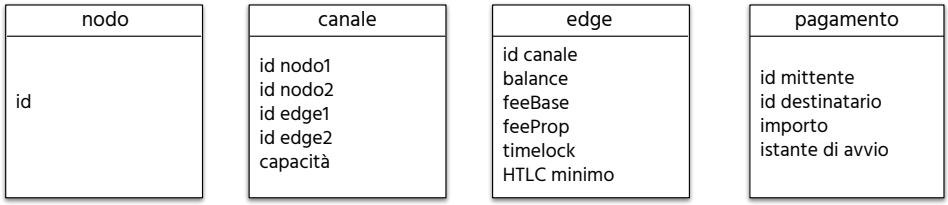

Figura 12: *Strutture dati del simulatore CLoTH*

3.7.2 Input

Nella fase di pre-processing CLoTH riceve in ingresso i seguenti file di input:

- *nodes.csv*: descrizione dei nodi della rete, a ciascuno dei quali viene assegnato un ID univoco;
- *channels.csv*: il file contiene la lista dei canali identificati da un ID. Ad ogni canale sono associati due edge e due nodi. Ciascun canale possiede una certa capacità e una latenza, ossia il tempo necessario per attraversarlo;
- *edges.csv*: ogni edge è identificato da un ID ed è associato ad un certo canale. Ogni edge possiede un balance e una policy impostata dal nodo da cui parte;
- *payments.csv*: anche qui ciascun pagamento viene associato ad un ID. Ogni pagamento ha un nodo mittente, un nodo destinatario, l'importo e l'istante di tempo in cui deve essere schedulato. I pagamenti vengono generati usando variabili casuali ottenute a partire da parametri di input:
  - r_π: numero medio di pagamenti al secondo. In particolare, il tempo di inter-arrivo dei pagamenti è modellato come una distribuzione esponenziale negativa;

  - σ_a: variabile che regola l'importo dei pagamenti. È la larghezza della distribuzione gaussiana la cui coda viene utilizzata per scegliere gli ordini di grandezza degli importi di pagamento. Maggiore è questa larghezza, maggiori saranno gli importi di pagamento.

3.7.3 Parametri di output

Al termine di ogni simulazione CLoTH restituisce risultati dai quali vengono estratte delle statistiche che rientrano nel file *payments-statistic.json*. I valori riportati al suo interno sono i seguenti:

- P_s: probabilità che i pagamenti vadano a buon fine;
- P_{fb}: probabilità che i pagamenti falliscano per unbalancing;
- P_{fp}: probabilità che i pagamenti falliscano per assenza di percorsi tra nodo mittente e nodo destinatario del pagamento;
- P_{ft}: probabilità che i pagamenti falliscano a causa della scadenza del timeout (60 sec);
- *T (ms)*: tempo medio impiegato dai pagamenti per andare a buon fine;
- N_{att}: numero medio dei tentativi che fanno i pagamenti per andare a buon fine;
- *R (hops)*: lunghezza media dei percorsi tra nodo mittente e nodo destinatario nei pagamenti che vanno a buon fine.

4 REBALANCE ATTIVO E PASSIVO: SIMULAZIONI E RISULTATI

Questo capitolo presenta le simulazioni che sono state condotte su Lightning Network, mediante il simulatore CLoTH, con l'obiettivo di studiare le performance della rete andando ad apportare diverse modifiche al protocollo adottato.

Le variazioni del protocollo mettono in piedi una strategia di rebalancing, atta a mitigare gli effetti negativi dell'unbalancing. Questo metodo può essere utilizzato seguendo due strade alternative: rebalancing attivo e passivo.

Il Capitolo si divide in questa maniera: la prima sezione (4.1) delinea gli obiettivi da raggiungere e le modalità con cui vengono avviate le simulazioni, riportando anche i risultati di quelle relative allo scenario senza rebalancing; la seconda (4.2) e la terza (4.3) sezione affrontano le strategie di rebalancing attivo e passivo con le rispettive modifiche apportate, descrivendo le simulazioni effettuate e commentando i risultati; nella quarta sezione (4.4) vengono commentati i risultati ottenuti dalla combinazione tra le due migliori modifiche apportate rispettivamente al protocollo di rebalancing attivo e passivo; la quinta e ultima sezione (4.5) si occupa di descrivere lo scenario service-provider commentando le prestazioni registrate dalle migliori modifiche del rebalancing attivo e passivo, implementate stavolta separatamente.

4.1 OBIETTIVO E DESIGN

4.1.1 Obiettivo

L'obiettivo di questo capitolo è trovare la soluzione di rebalancing che permetta al meglio di ridurre la probabilità di fallimento dei pagamenti per unbalancing. Una conseguenza è l'incremento della probabilità di successo dei pagamenti.

4.1.2 Metodo

Per raggiungere l'obiettivo verranno mostrati e discussi i risultati delle simulazioni eseguite su CLoTH implementando più varianti della tecnica di rebalancing, sia nella sua versione attiva che in quella passiva, mettendo a confronto le prestazioni registrate con quelle garantite nel caso in cui il rebalancing sia assente.

Le prestazioni misurate riguardano i pagamenti generati ad ogni simulazione. Tra le statistiche restituite da CLoTH ci sono le seguenti:

- P_s: probabilità che i pagamenti vadano a buon fine;
- P_{fb}: probabilità che i pagamenti falliscano a causa dell'unbalancing;
- P_{fp}: probabilità che i pagamenti falliscano per assenza di un percorso che colleghi il mittente del pagamento con il destinatario.

Dunque lo scopo del rebalancing è massimizzare la probabilità di successo P_s cercando di ridurre al minimo la quota P_{fb} relativa ai pagamenti falliti a causa di canali sbilanciati. P_{fp} non rientra nel target del rebalancing, pertanto eventuali sue variazioni rispetto alla situazione senza rebalancing verranno mostrate solo se particolarmente rilevanti.

4.1.3 Design

I parametri di input delle simulazioni sono:

- r_π: rate dei pagamenti, ossia quanti pagamenti vengono eseguiti al secondo;
- σ_a: variabile che regola l'importo dei pagamenti.

Per quanto riguarda r_π ogni simulazione esegue 100 pagamenti al secondo. La scelta di questo rate deriva dal lavoro svolto in [10], dove si dimostra che le performance generali non variano sensibilmente passando da 10 a 100 pagamenti al secondo. Dunque la scelta di eseguire 100 pagamenti al secondo permette di avvicinarsi ai valori riscontrabili nella realtà senza però pesare troppo in termini di tempi di esecuzione per ogni simulazione. Dunque il rate dei pagamenti che viene adottato è fisso. 100 pagamenti al secondo inoltre è la giusta via di mezzo tra 10 pagamenti al secondo, rate che LN intende superare, e 1000 pagamenti al secondo, rate di un sistema di pagamento ben consolidato che LN aspira a diventare in futuro.

Per ciascuna versione implementata del protocollo del simulatore verrà chiamato un ciclo di simulazioni, dove l'unica variabile indipendente considerata dunque sarà σ_a: essa rappresenta la probabilità che

i pagamenti avvengano con cifre più o meno elevate. Quindi più aumenta il valore di σ_a e più aumenta l'ammontare medio dei singoli pagamenti che vengono simulati ad ogni iterazione di CLoTH.

La Tabella 1 mostra come gli importi più o meno elevati associati ai singoli pagamenti vengono distribuiti su un intervallo di σ_a che va da 1 a 10. In particolare la prima colonna riporta diversi ordini di grandezza associati ad importi di pagamento espressi in Satoshi[3]. Il valore più basso 10^0 viene scelto per produrre solo piccoli pagamenti. Il valore più alto 10^5 viene scelto per produrre pagamenti con importi più elevati, ma comunque mai superiori a 0.1 BTC, poichè al momento Lightning Network non supporta pagamenti con cifre più grandi. In ogni colonna, in corrispondenza di ciascun valore di σ_a, sono riportate delle percentuali che indicano in che modo i pagamenti simulati per quel valore di σ_a si distribuiscono tra i vari ordini di grandezza dei rispettivi importi. Ad esempio, per σ_a=1, il 97.44% dei pagamenti simulati ha un importo con un ordine di grandezza pari a 10^0.

Si evince che all'aumentare di σ_a dovrebbe diminuire P_s per via della crescita di P_{fb}.

Verrà dunque scelto un intervallo di valori interi per σ_a e per ciascuno di essi il simulatore eseguirà 100.000 pagamenti restituendo di volta in volta le statistiche sulle performance al variare di σ_a.

Nel primo ciclo di simulazioni, ossia quello eseguito sulla versione del simulatore senza alcuna tecnica di rebalancing implementata, l'intervallo scelto per σ_a va da 1 a 10. Nei cicli successivi il più basso valore di σ_a adottato corrisponderà a quello più piccolo che nel primo ciclo avrà fatto registrare $P_{fb} \geqslant 1\%$, soglia di fallimento considerata minima per poter valutare il problema dell'unbalancing.

Ordine di grandezza (Satoshi)	σ_a									
	1	2	3	4	5	6	7	8	9	10
10^0	97.44%	65.58%	42.93%	31.42%	26.64%	22.74%	21.12%	20.13%	19.59%	18.75%
10^1	2.55%	25.41%	26.97%	24.92%	21.86%	21.16%	19.98%	19.61%	18.63%	17.91%
10^2	0.01%	7.05%	16.15%	18.32%	18.50%	17.81%	17.58%	17.61%	17.76%	17.40%
10^3	0.00%	1.63%	8.61%	12.29%	15.19%	15.58%	16.08%	15.54%	15.84%	16.64%
10^4	0.00%	0.28%	3.82%	8.01%	10.16%	12.90%	13.50%	14.21%	14.69%	14.78%
10^5	0.00%	0.05%	1.52%	5.04%	7.65%	9.81%	11.74%	12.90%	13.49%	14.45%

Tabella 1: *Distribuzione dei pagamenti con un certo ordine di grandezza per ogni valore di* σ_a

4.1.4 No Rebalancing

Di seguito vengono mostrati i grafici relativi alle prestazioni registrate dal simulatore nel caso in cui non venga adottata nessuna strategia di rebalancing.

3 1 Satoshi = 10^{-8} BTC.

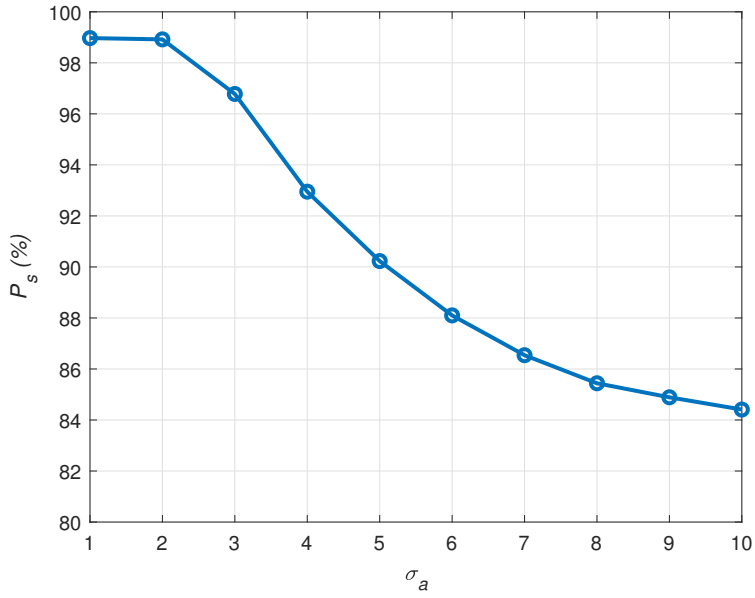

Figura 13: *No rebalancing:* P_s

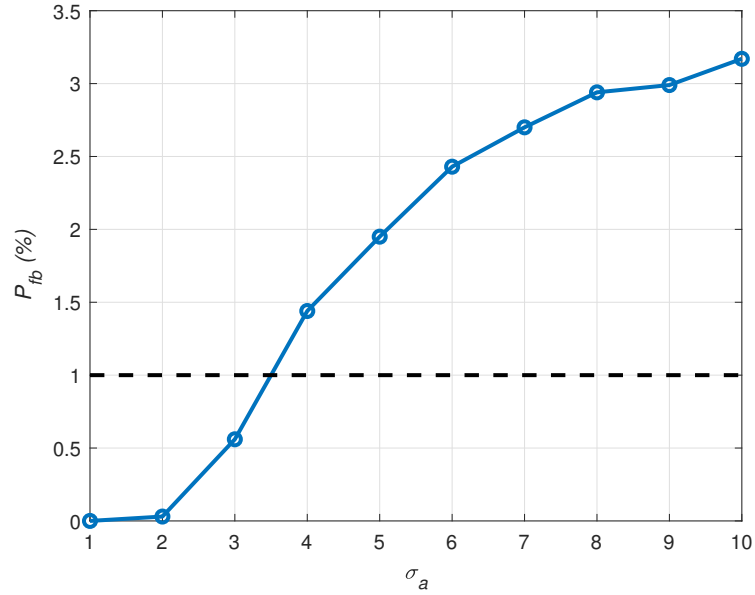

Figura 14: *No rebalancing:* P_{fb}

Le Figure 13 - 14 - 15 mostrano le performance garantite al variare di σ_a. Si nota che la probabilità di successo dei pagamenti P_s va al di sotto del 90% per σ_a=6. Quando σ_a=1, la probabilità di successo dei pagamenti è pari al 98.97% e l'unica causa di fallimento è dovuta all'assenza di un percorso che colleghi mittente e destinatario del pagamento. All'aumentare di σ_a, la percentuale P_{fp} di fallimento per assenza di percorsi aumenta e, a partire da σ_a=2, una parte minore dei pagamenti fallisce per unbalancing, con la percentuale P_{fb} che su-

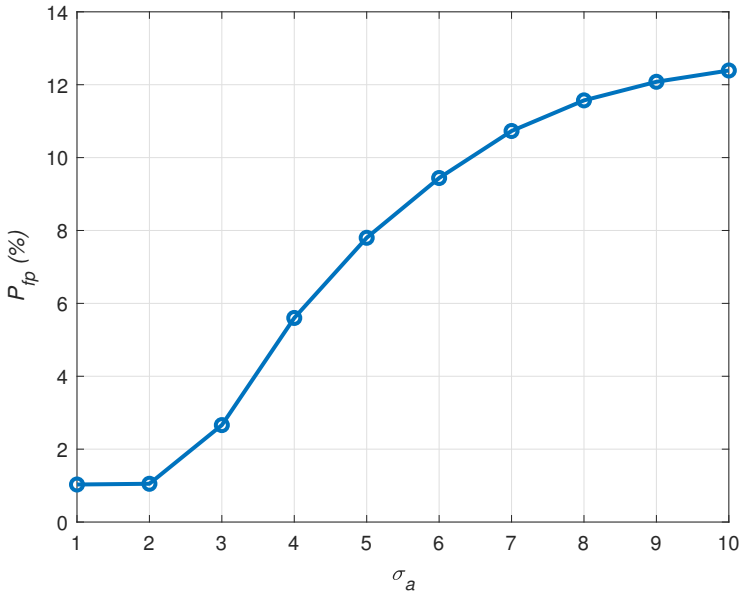

Figura 15: *No rebalancing:* P_{fp}

pera la soglia dell'1% in corrispondenza di σ_a=4; oltrepassata questa percentuale, il problema dell'unbalancing non è più trascurabile.

Da qui segue che le successive simulazioni verranno eseguite con σ_a che varia da 4 a 10.

La spiegazione della crescita di P_{fp} all'aumentare di σ_a è che i pagamenti con un importo crescente hanno maggiore difficoltà nel trovare un percorso con capacità di canale sufficienti, e ciò porta anche all'unbalancing, dato che pagamenti con un importo elevato tendono a sbilanciare più rapidamente i canali. In corrispondenza del valore massimo di σ_a, circa il 12% della probabilità di fallimento è dovuto ad assenza di percorsi, mentre poco più del 3% è causato dall'unbalancing.

Il numero medio delle volte in cui ciascun pagamento viene tentato affinchè vada a buon fine è 1.69[4] (A.1).

É interessante analizzare questo dato poichè un tentativo di pagamento può fallire non solo per assenza di un percorso ma anche per unbalancing. Viene quindi logico pensare che adottando una valida strategia di rebalancing questo valore possa subire un decremento.

4 Per ricavare tale valore, in questa e nelle analisi delle successive implementazioni, viene considerata la simulazione dove σ_a=10, essendo questa la situazione considerata più critica, data la quantità media maggiore di *amount* da trasferire in ogni pagamento.

4.2 REBALANCING ATTIVO

Il rebalancing attivo consiste nell'eseguire un pagamento con lo scopo di ribilanciare i canali. In particolare se un nodo in un canale ha un balance basso e in un altro ha invece un balance alto, esso può effettuare un pagamento che va dal secondo canale al primo in modo da riequilibrarli [10]. Si consideri l'esempio in Figura 16.

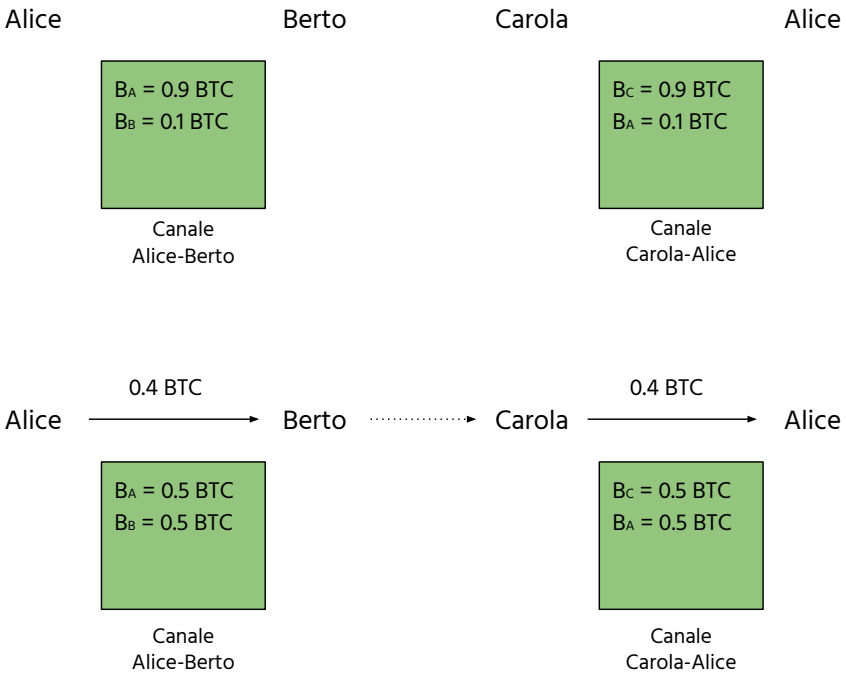

Figura 16: *Pagamento di rebalancing nel rebalancing attivo*

Nel canale con Carola, Alice possiede un balance basso (10% della capacità totale del canale), mentre nel canale con Berto lei possiede un balance elevato (90% della capacità totale del canale). Quindi Alice esegue un *pagamento di rebalancing* che sposta 0.4 BTC dal canale con Berto al canale con Carola, e in questo modo il canale con Carola risulterà bilanciato, dato che sia Alice che Carola avranno in questo modo un balance che corrisponde al 50% della capacità totale del canale.

La nomenclatura in uno scenario di rebalancing attivo è la seguente:

- Il *nodo che esegue il rebalancing* (Alice);
- L'*edge sbilanciato* (edge dove Alice ha un balance basso);
- L'*edge candidato* (edge dove Alice ha un balance alto).

Considerando per praticità la configurazione dell'esempio, il rebalancing attivo non va a buon fine nei casi seguenti:

- Alice non possiede nessun edge candidato che possa essere utilizzato per il rebalancing. Questo edge, se presente, deve soddisfare due condizioni:
  - il balance di Alice nell'edge candidato deve essere più grande della metà della capacità del rispettivo canale;
  - il balance di Alice nell'edge candidato deve bastare per trasferire la quantità di bitcoin necessaria (0.4 BTC nell'esempio fatto) per l'operazione di rebalancing;
- Il pagamento di rebalancing fallisce perchè non viene trovata nessuna strada tra Berto e Carola;
- Il pagamento di rebalancing fallisce perchè non c'è balance a sufficienza in uno o più degli edge[5] intermedi presenti nel percorso che collega Berto e Carola (ci sono uno o più edge con balance inferiore a 0.4 BTC).

Per semplicità si assuma che i pagamenti di rebalancing vengano eseguiti istantaneamente e senza alcuna commissione richiesta.

4.2.1 Implementazione iniziale

La prima forma di rebalancing attivo che è stata presa in considerazione nei test e su cui sono state effettuate le successive modifiche è quella messa a punto in [10]. In questa implementazione l'operazione di rebalancing viene gestita da una funzione apposita chiamata `rebalance()` di cui viene fornito di seguito l'algoritmo:

A. Calcolo dell'ammontare *amount* del pagamento di rebalancing

B. Per ogni edge relativo al nodo che esegue il rebalancing
   a. calcolo della frazione *fraction* del balance dell'edge rispetto alla capacità del rispettivo canale
   b. se $fraction > 0.5$
      i. se il balance è almeno uguale ad *amount*
         1 l'edge attuale è quello da cui far partire il pagamento di rebalancing
         2 Interruzione del ciclo

C. Se non viene trovato nessun edge candidato

5 Il *balance dell'edge* è il balance del nodo da cui parte l'edge considerato. Ad esempio, nel canale tra Alice e Berto, il balance dell'edge che va da Berto ad Alice corrisponde al balance che Berto possiede nel canale con Alice.

   a. Fallimento del rebalancing

D. Calcolo del percorso *hops* per effettuare il pagamento di rebalancing

E. Se *hops* non viene trovato

   a. Fallimento del rebalancing

F. Controllo dei balance presenti nel percorso trovato

G. Se almeno un balance intermedio non ha fondi a sufficienza per trasferire *amount*

   a. Fallimento del rebalancing

H. Esecuzione del pagamento di rebalancing

La funzione `rebalance()` viene chiamata ogni volta che un nodo invia o inoltra un qualsiasi pagamento da uno dei suoi edge, qualora il rispettivo balance dopo la transazione vada al di sotto del 20% della capacità totale del canale corrispondente. L'*amount* trasferito con il pagamento di rebalancing corrisponde alla quantità necessaria affinchè il balance dell'edge sbilanciato raggiunga il 50% della capacità del rispettivo canale.

Di seguito vengono mostrate le prestazioni di questa prima versione di rebalancing attivo.

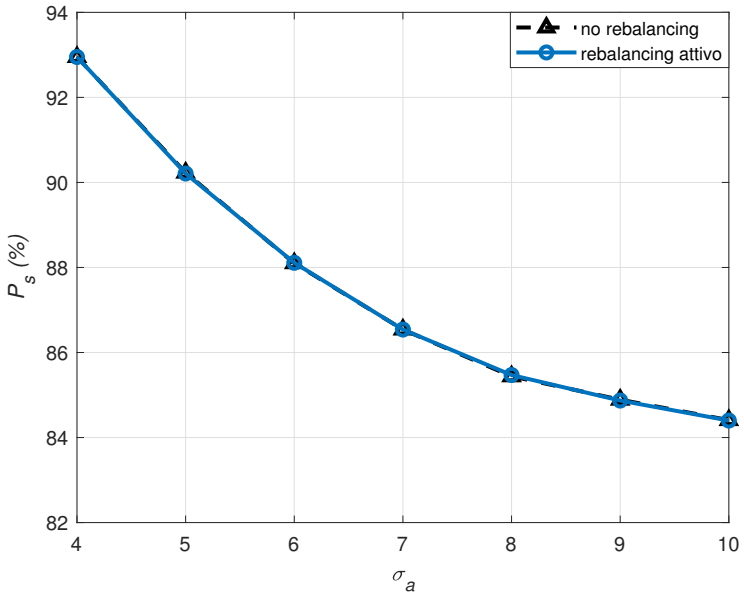

Figura 17: *Rebalancing Attivo - Setup iniziale:* P_s

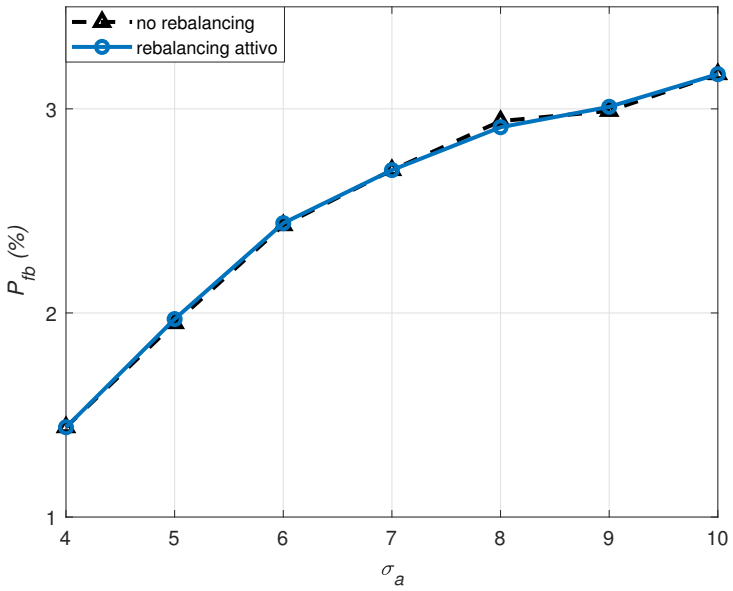

Figura 18: *Rebalancing Attivo - Setup iniziale: P_{fb}*

Per il rebalancing attivo risulta interessante analizzare anche le statistiche relative ai tentativi di rebalancing effettuati. In particolare, al variare di σ_a, vengono mostrati:

- *N° Tentativi*: numero di tentativi di rebalancing;
- *Successo* (%): percentuale dei tentativi andati a buon fine;
- *No Channel* (%): percentuale dei tentativi falliti per assenza di un edge candidato da cui far partire il pagamento di rebalancing;
- *No Route* (%): percentuale dei tentativi falliti per assenza di un percorso da far attraversare al pagamento di rebalancing;
- *No Balance* (%): percentuale dei tentativi falliti per insufficienza di balance nel percorso.

σ_a	N° Tentativi	Successo (%)	No channel (%)	No route (%)	No balance (%)
4	35769	16.47	24.45	0.34	58.74
5	45188	17.38	27.34	0.31	54.97
6	61172	19.59	25.98	0.67	53.76
7	53430	18.29	30.91	0.52	50.28
8	58677	17.65	30.15	0.50	51.70
9	70272	17.65	26.97	0.79	54.59
10	73185	17.21	28.54	0.28	53.96

Tabella 2: *Rebalancing Attivo - Setup iniziale: statistiche*

Le prestazioni garantite da questa prima implementazione di rebalancing attivo sono di fatto le stesse registrate in uno scenario di no rebalancing.

Riguardo i tentativi di rebalancing, questi hanno un trend crescente all'aumentare di σ_a, con un incremento altalenante nella percentuale di successo. Questa crescita avviene perchè importi crescenti nei pagamenti tendono a far sbilanciare più facilmente i canali. Ciò rende necessari maggiori tentativi di rebalancing che vengono ostacolati prevalentemente dall'insufficienza di balance.

Il numero medio di tentativi che vengono fatti per ciascun pagamento è 1.57 (A.2): si riscontra dunque una riduzione del 7.10% rispetto al no rebalancing[6].

L'obiettivo delle successive simulazioni sarà quello di minimizzare la percentuale di fallimento dei pagamenti per unbalancing tramite l'incremento della percentuale di successo dei tentativi di rebalancing.

4.2.2 Simulazioni

Le simulazioni a seguire sono relative a modifiche apportate al protocollo che esegue il pagamento di ribilanciamento. Per ogni modifica sono state avviate delle simulazioni con un rate r_π pari a 100 pagamenti al secondo e con σ_a che varia da 4 a 10.

MODIFICA 1 Nell'implementazione di partenza il rebalance fallisce, una volta individuato un edge candidato, se non si trova un percorso che collega l'edge candidato con l'edge sbilanciato. Si giunge allo stesso esito se qualche edge intermedio nel percorso non possiede balance a sufficienza. In questa implementazione, se si verificano tali condizioni, il rebalance non fallisce ma si passa ad analizzare altri possibili edge candidati finchè questi non finiscono.

6 Anche nelle successive implementazioni di rebalancing attivo, salvo diverse indicazioni, questo valore verrà confrontato rispetto al caso di no rebalancing.

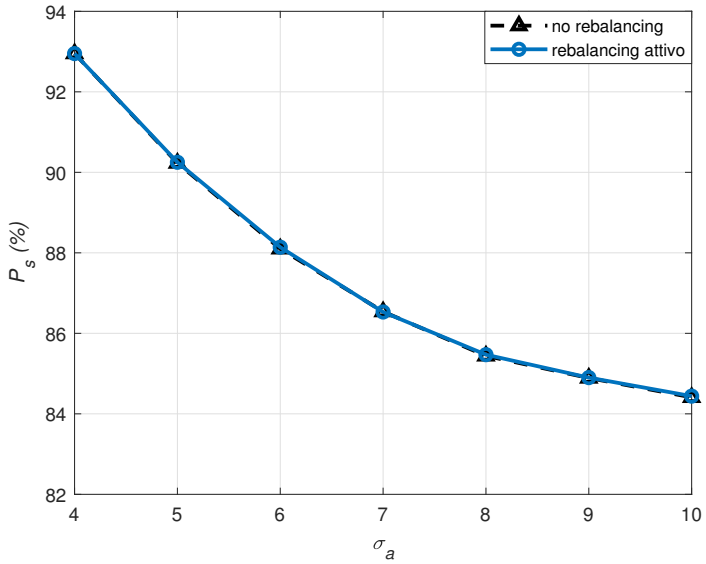

Figura 19: *Rebalancing Attivo - Modifica 1: P_s*

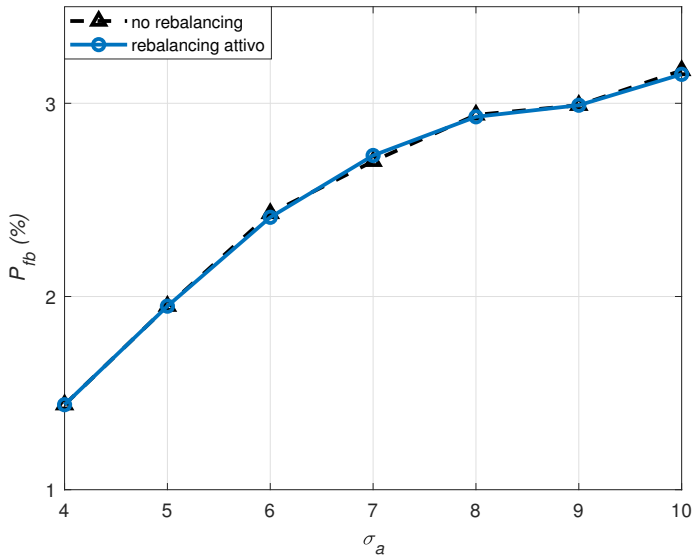

Figura 20: *Rebalancing Attivo - Modifica 1: P_{fb}*

σ_a	N° Tentativi	Successo (%)	No channel (%)	No route (%)	No balance (%)
4	23046	45.21	32.89	0.28	21.61
5	28864	46.64	34.87	0.36	18.14
6	35595	47.74	34.71	0.24	17.31
7	37465	47.05	35.74	0.44	16.77
8	39936	47.84	36.09	0.19	15.87
9	39459	48.11	36.06	1.00	14.83
10	43408	46.14	37.82	0.22	15.82

Tabella 3: *Rebalancing Attivo - Modifica 1: statistiche*

Confrontando questa implementazione di rebalancing attivo con la soluzione senza rebalancing, si riscontra per P_s un miglioramento dello 0.02%, con un calo di P_{fb} dello 0.10%[7].

Riguardo i tentativi di rebalancing, rispetto all'implementazione iniziale, viene registrato un netto miglioramento, visto che per σ_a=10 c'è un passaggio dal 17.21% al 46.14% nella percentuale di successo e una riduzione dal 53.96% al 15.82% nella percentuale di fallimento per no balance. L'aspetto più importante da segnalare è che l'assenza di un edge candidato porta al fallimento di più di un terzo dei tentativi di rebalancing attivo[8].

I tentativi per pagamento sono in media 1.16, ossia il 31.36% in meno rispetto al no rebalancing (A.3). Un miglioramento di questo dato significa che i canali risultano più bilanciati, e che pertanto necessitano di un minor numero di tentativi per ogni pagamento.

Questo leggero incremento di performance è dato proprio dal nuovo approccio adottato nella MODIFICA 1 che insiste in fase di rebalancing nel cercare l'edge candidato che abbia un percorso garantito verso l'edge sbilanciato.

MODIFICA 2 Questa implementazione mantiene le variazioni relative alla MODIFICA 1. Nella MODIFICA 2 viene fatto in modo che non ci si accontenti del primo edge candidato trovato ma di cercare l'edge candidato migliore, ossia col più alto balance. Questa tecnica serve ad evitare che, dopo il rebalancing, quello che era un edge bilanciato diventi un nuovo edge sbilanciato.

7 Tutti i miglioramenti/peggioramenti di performance sono relativi e considerati usando come termine di paragone le prestazioni garantite nello scenario di no rebalancing. Essi inoltre sono calcolati facendo una media al variare di σ_a da 4 a 10.

8 I confronti sulle statistiche relative ai tentativi di rebalancing sono fatti considerando σ_a=10.

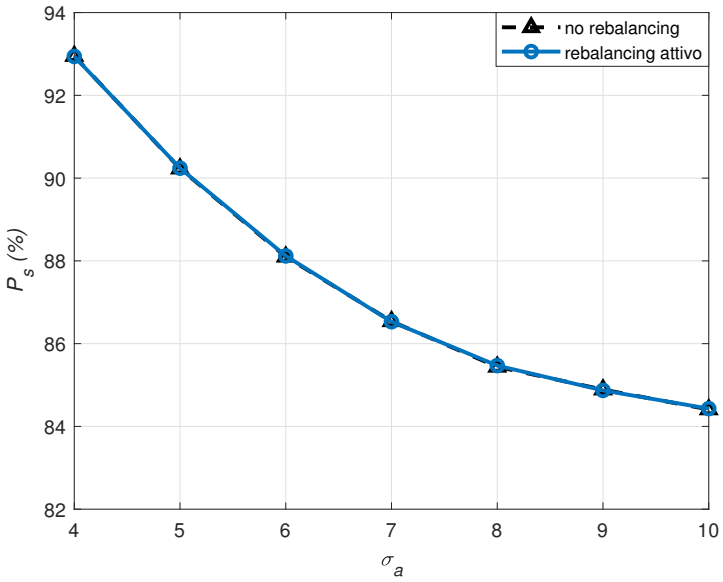

Figura 21: *Rebalancing Attivo - Modifica 2: P_s*

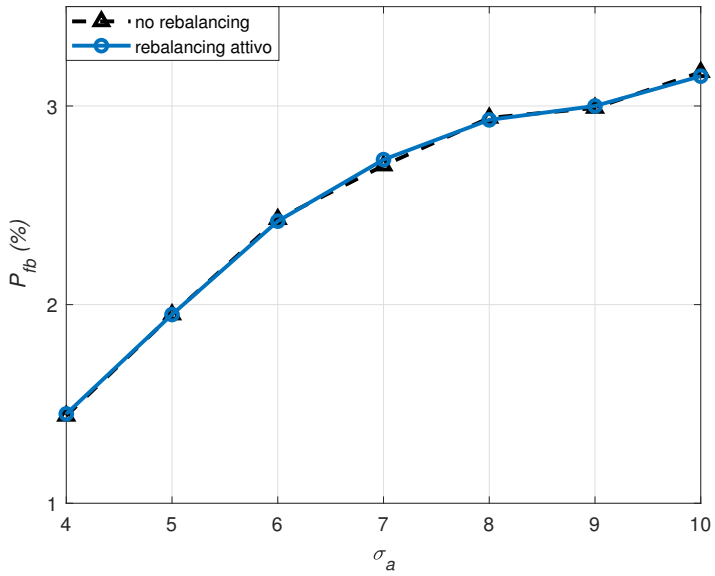

Figura 22: *Rebalancing Attivo - Modifica 2: P_{fb}*

σ_a	N° Tentativi	Successo (%)	No channel (%)	No route (%)	No balance (%)
4	24961	40.62	35.56	0.28	23.54
5	29517	46.61	34.34	0.35	18.70
6	37545	43.27	34.90	0.26	21.56
7	37773	45.82	35.97	0.40	17.81
8	42810	44.89	36.88	0.24	17.98
9	41172	46.13	36.77	0.98	16.12
10	43335	45.50	36.53	0.25	17.72

Tabella 4: *Rebalancing Attivo - Modifica 2: statistiche*

Le performance registrate in questa modifica non suggeriscono miglioramenti ulteriori rispetto a quelli ottenuti con la MODIFICA 1.

Si nota tuttavia rispetto alla MODIFICA 1 un leggero peggioramento relativo dell'1.39% nell'esito dei tentativi di rebalancing a causa di un aumento del 12% della percentuale di fallimento per insufficienza di balance.

Quest'ultimo peggioramento è in perfetta antitesi con lo scopo di questa implementazione. Esso può essere spiegato supponendo che, data la topologia della rete su cui sono state avviate le simulazioni, la scelta dei nodi con più alto balance come quelli da cui far partire i pagamenti di rebalancing vada a penalizzare altri pagamenti di rebalancing che si trovano ad attraversare gli stessi nodi.

Il numero medio di tentativi per pagamento è 1.16, esattamente alla pari con il valore riscontrato nella MODIFICA 1 (A.4).

MODIFICA 3 Qui viene rimossa la MODIFICA 1 in modo tale da apprezzare solo gli effetti della MODIFICA 2.

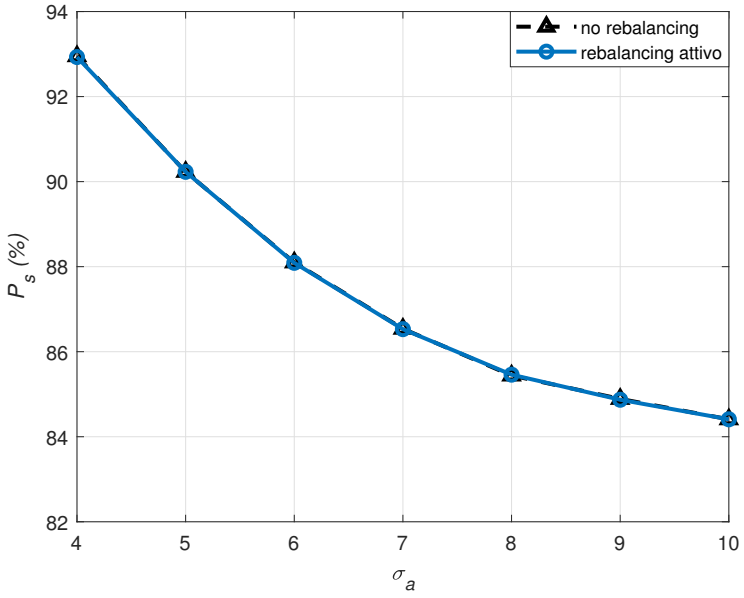

Figura 23: *Rebalancing Attivo - Modifica 3: P_s*

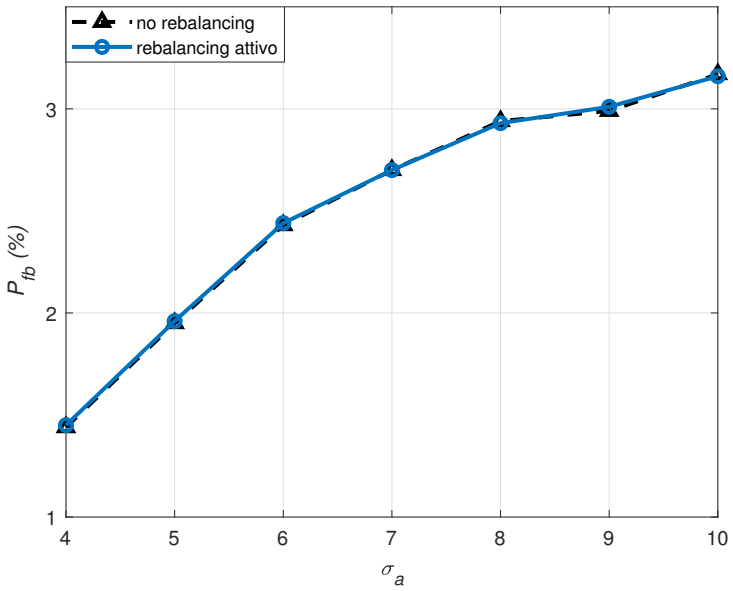

Figura 24: *Rebalancing Attivo - Modifica 3: P_{fb}*

σ_a	N° Tentativi	Successo (%)	No channel (%)	No route (%)	No balance (%)
4	38096	18.35	23.86	0.22	57.57
5	41225	19.59	28.54	0.38	51.49
6	47684	20.17	29.93	0.98	48.92
7	49878	22.30	32.69	0.62	44.39
8	56218	22.14	29.73	0.33	47.80
9	53269	21.68	30.40	0.83	47.09
10	59763	22.10	30.55	0.38	46.97

Tabella 5: *Rebalancing Attivo - Modifica 3: statistiche*

Anche stavolta le percentuali di successo e fallimento dei pagamenti simulati non mutano considerevolmente.

In questa soluzione tuttavia emerge l'importanza della MODIFICA 1 che, in sua assenza, vede dimezzare la percentuale di successo dei tentativi di rebalancing rispetto all'implementazione precedente, e ciò è dovuto soprattutto all'aumento netto dei tentativi falliti per no balance, la cui percentuale passa da 17.72% a 46.97%.

A suggellare quanto affermato interviene anche il dato relativo ai tentativi di pagamento che sale a 1.40, con un rialzo del 21% rispetto all'implementazione precedente (A.5).

Quest'ultimo dato rappresenta il sintomo che i canali risultano mediamente più sbilanciati, e ciò è perfettamente in linea con l'aumento dei fallimenti per no balance dei tentativi di rebalancing.

MODIFICA 4 Questa implementazione è costruita su quella relativa alla MODIFICA 3. Fino a questo momento la funzione `rebalance()` è stata chiamata nel momento in cui un nodo, dopo aver inviato o

inoltrato un pagamento, avesse visto scendere il proprio balance al di sotto del 20% della capacità del canale. In questa variante del protocollo si è portato questa soglia al 30%. L'obiettivo è limitare l'*amount* relativo al pagamento di rebalancing sufficiente a portare il balance dell'edge sbilanciato al 50% della capacità del rispettivo canale. In questo modo è possibile migliorare la percentuale di successo nel rebalancing grazie ad un abbassamento dei fallimenti causati da no balance.

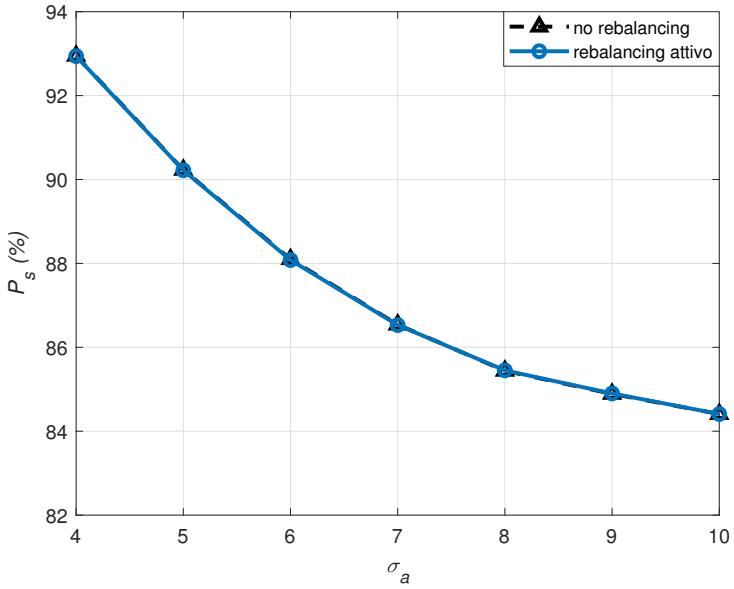

Figura 25: *Rebalancing Attivo - Modifica 4:* P_s

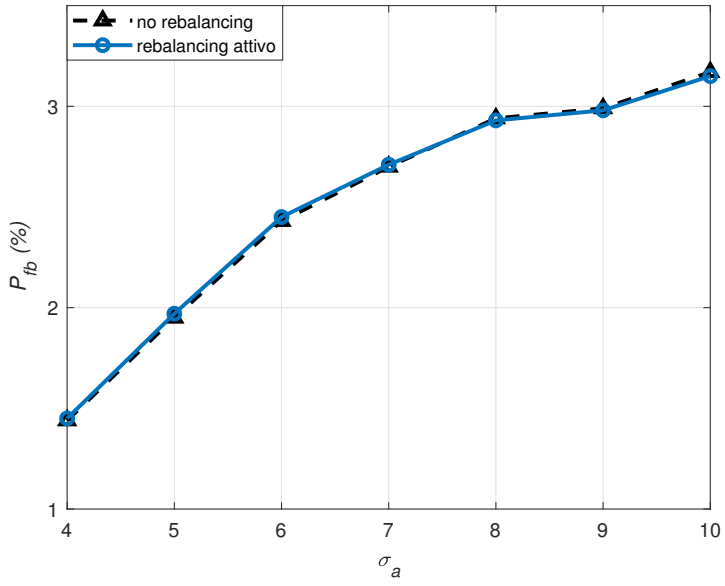

Figura 26: *Rebalancing Attivo - Modifica 4:* P_{fb}

σ_a	N° Tentativi	Successo (%)	No channel (%)	No route (%)	No balance (%)
4	62480	29.83	26.89	0.65	42.64
5	70458	29.64	29.57	0.49	40.31
6	80373	30.20	30.38	0.45	38.97
7	83262	31.77	31.79	0.66	35.78
8	94853	33.47	31.39	0.56	34.58
9	87260	32.82	30.49	0.73	35.96
10	93175	32.83	33.13	0.43	33.60

Tabella 6: *Rebalancing Attivo - Modifica 4: statistiche*

A fronte di prestazioni dei pagamenti inalterate, questa implementazione rispetta le aspettative con un aumento della percentuale di successo dei tentativi di rebalancing rispetto alla modifica precedente che passa da 22.10% al 32.83%, grazie all'abbassamento della percentuale di fallimenti causati da no balance che scende da 46.97% a 33.60%.

Aumentando infatti la soglia di balance sotto la quale si attiva il processo di rebalancing, diminuisce l'*amount* necessario per il pagamento di rebalancing. Ciò aumenta la probabilità che gli edge intermedi abbiano un balance uguale o superiore a questo *amount*, e quindi sufficiente a gestire il pagamento di rebalancing.

La percentuale di successo dei tentativi di rebalancing rimane comunque al di sotto del 46.14% della MODIFICA 1. Infatti nell'implementazione corrente non si prevede di analizzare altri eventuali edge candidati qualora quello inizialmente individuato non consenta di effettuare il rebalancing; questo meccanismo invece è previsto nella MODIFICA 1.

Il numero medio di tentativi per pagamento scende a 1.36, con un decremento del 19.53% rispetto al caso no rebalancing (A.6).

MODIFICA 5 Questa variante si basa sull'implementazione iniziale del rebalancing attivo. Essa eredita dalle modifiche precedenti solo la variazione di soglia eseguita nella MODIFICA 4.

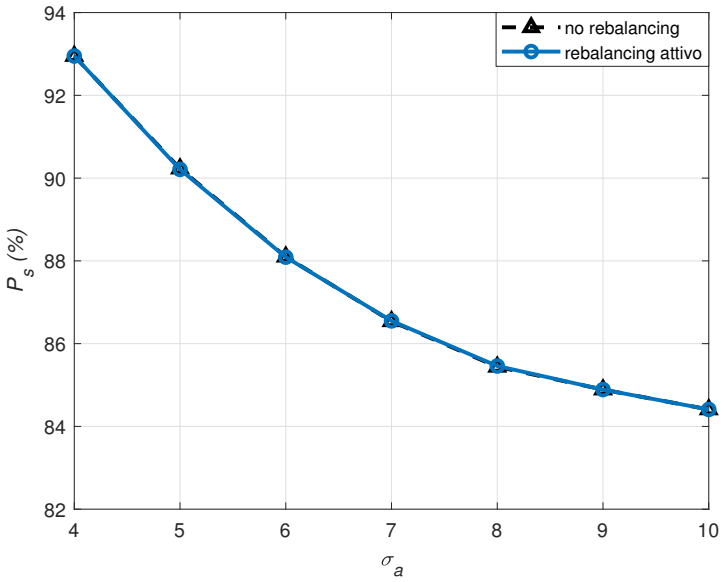

Figura 27: *Rebalancing Attivo - Modifica 5: P_s*

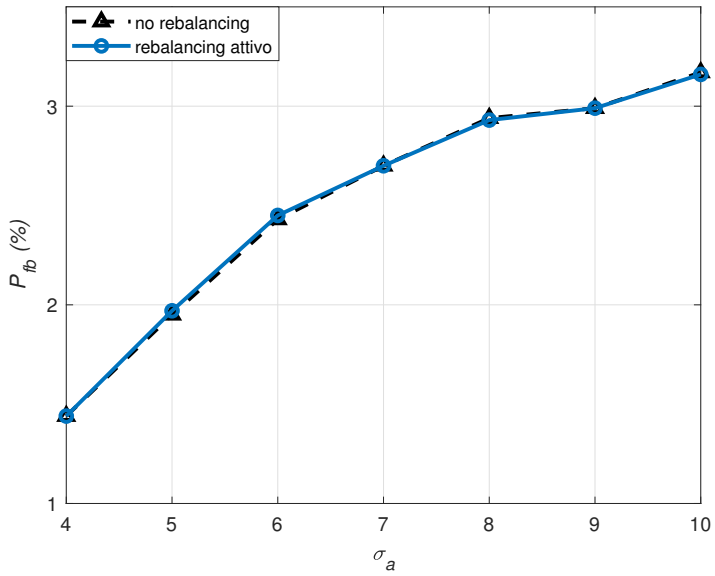

Figura 28: *Rebalancing Attivo - Modifica 5: P_{fb}*

σ_a	N° Tentativi	Successo (%)	No channel (%)	No route (%)	No balance (%)
4	58707	26.20	27.47	0.38	45.96
5	71254	25.78	30.18	0.38	43.66
6	86150	26.41	30.19	0.38	43.02
7	83903	27.79	31.79	0.52	39.91
8	90863	27.54	32.82	0.39	39.25
9	98709	28.09	28.67	0.60	42.65
10	110198	26.65	28.88	0.32	44.15

Tabella 7: *Rebalancing Attivo - Modifica 5: statistiche*

In questa soluzione il dato più importante che emerge è l'aumento del numero di tentativi di rebalancing, che per σ_a=10 aumenta di quasi 20000 unità rispetto all'implementazione precedente.

Ciò si giustifica con la rimozione della MODIFICA 2, che di fatto ha la missione di limitare il sorgere di canali sbilanciati che rendano quindi necessari i tentativi di rebalancing.

Fa da eco l'aumento del numero di tentativi per pagamento che si porta a 1.45 (A.7).

MODIFICA 6 Questa modifica è costruita sull'implementazione della MODIFICA 5. Fino a questo momento nella funzione `rebalance()`, una volta definito il percorso tra edge candidato ed edge sbilanciato, si procede con un controllo per verificare se tale percorso attraversa edge con balance a sufficienza; in caso negativo il rebalance fallisce.

Si è modificato questo comportamento tenendo traccia in maniera appropriata degli edge con scarso balance. Dunque viene calcolato un nuovo percorso che escluda il passaggio attraverso tali edge. La `rebalance()` in questo modo fallisce solo nel momento in cui anche l'ultimo percorso che collega l'edge candidato con l'edge sbilanciato si attesta essere inappropriato a causa di scarso balance in qualche edge intermedio.

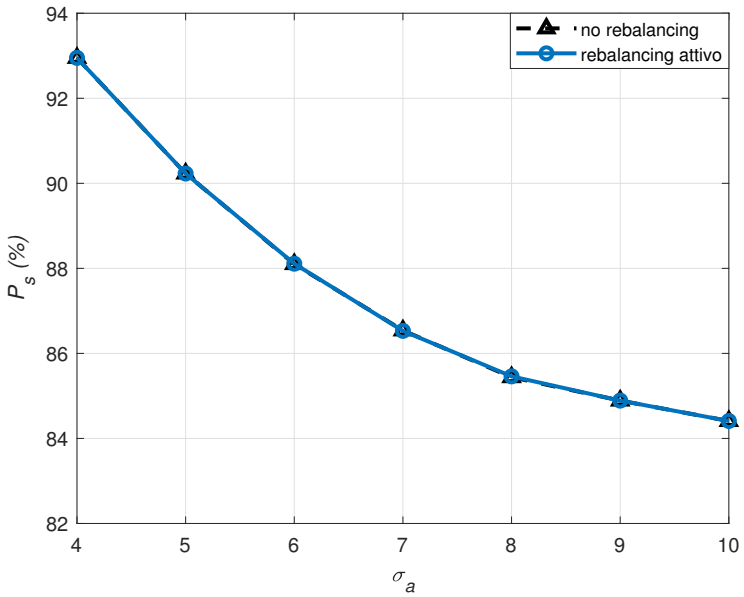

Figura 29: *Rebalancing Attivo - Modifica 6: P_s*

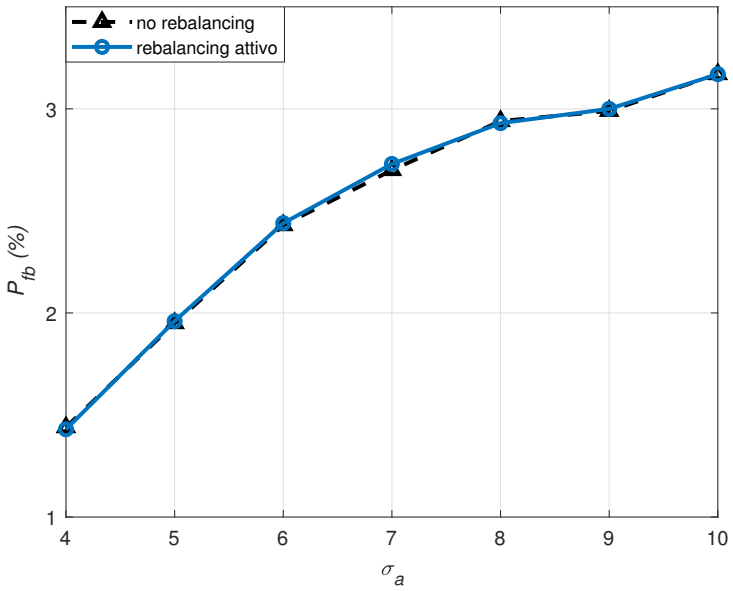

Figura 30: *Rebalancing Attivo - Modifica 6: P_{fb}*

σ_a	N° Tentativi	Successo (%)	No channel (%)	No route (%)	No balance (%)
4	48066	49.01	31.47	0.44	19.08
5	60021	49.59	32.95	0.40	17.06
6	69938	50.53	32.17	0.38	16.92
7	73960	48.60	33.68	0.42	17.30
8	74449	49.13	33.88	0.40	16.58
9	74990	49.59	34.13	0.73	15.54
10	83234	50.13	35.28	0.34	14.24

Tabella 8: *Rebalancing Attivo - Modifica 6: statistiche*

A parità di prestazioni sui pagamenti, che rimangano identiche a loro stesse dalla MODIFICA 1 in poi, in questa variante si riscontra un netto miglioramento rispetto all'implementazione precedente nell'esito positivo dei tentativi di rebalancing che passa dal 26.65% al 50.13%, grazie ad una riduzione drastica della percentuale di fallimento per no balance che va da 44.15% a 14.24%. D'altronde la MODIFICA 6 è stata pensata esattamente per far rientrare quest'ultimo valore, poichè tenta di trovare il percorso che collega edge candidato ed edge sbilanciato senza fermarsi al primo stop causato da assenza di balance sufficiente.

La soluzione analizzata tuttavia non risulta particolarmente efficace per via della percentuale di errore per no channel che arriva a superare il 35%.

Nel numero medio di tentativi per pagamento si riscontra un beneficio, dato che ci si ferma a 1.17, con una riduzione del 19.31% rispetto alla precedente implementazione (A.8).

MODIFICA 7 Si combinano i comportamenti messi in piedi nella MODIFICA 6 con quelli relativi alla MODIFICA 2. Cioè la ricerca del percorso per il rebalancing viene condotta partendo non dal primo edge candidato trovato, ma dall'edge candidato con il balance più alto.

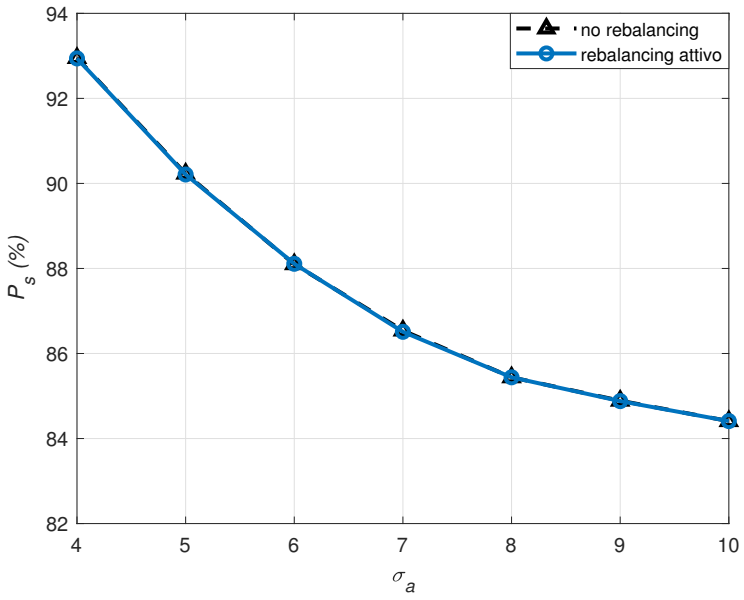

Figura 31: *Rebalancing Attivo - Modifica 7: P_s*

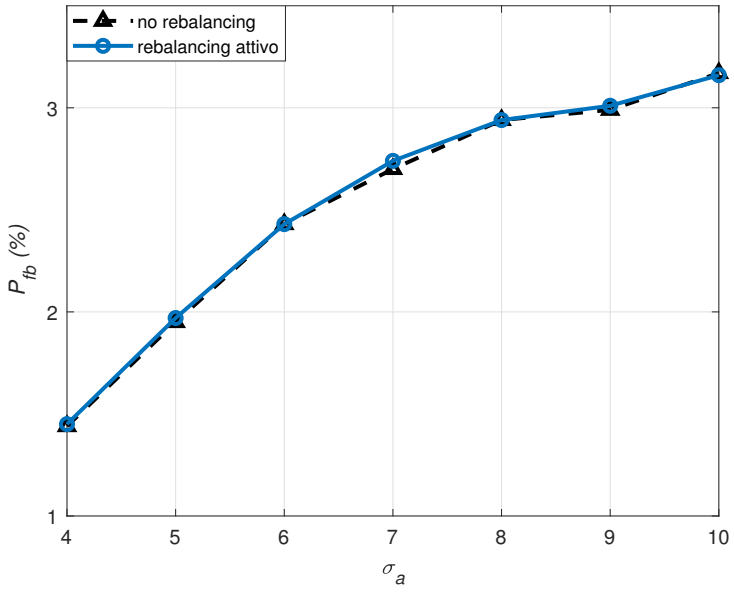

Figura 32: *Rebalancing Attivo - Modifica 7: P_{fb}*

σ_a	N° Tentativi	Successo (%)	No channel (%)	No route (%)	No balance (%)
4	46821	49.86	47.03	0.27	2.84
5	54512	56.91	39.60	0.30	3.18
6	64028	50.91	46.30	0.18	2.61
7	66881	54.22	43.28	0.34	2.16
8	70236	52.29	44.34	0.24	3.13
9	69707	52.70	45.05	0.26	1.98
10	74512	55.47	42.10	0.19	2.24

Tabella 9: *Rebalancing Attivo - Modifica 7: statistiche*

Il contributo da segnalare che viene garantito dalla MODIFICA 7 in merito ai tentativi di rebalancing rispetto alla versione precedente consiste in un abbassamento assoluto di almeno il 10% per quanto riguarda la percentuale dei fallimenti per no balance, con conseguente aumento del 5% della percentuale di successo.

L'attenzione che questa implementazione rivolge nel cercare il percorso migliore a partire dall'edge candidato migliore porta ad ottenere il più importante risultato finora raggiunto in merito al numero di tentativi per pagamento, che si porta a 1.15 con una diminuzione del 31.95% rispetto alla situazione di no rebalancing (A.9).

MODIFICA 8 Nelle versioni passate, l'*amount* trasferito con il pagamento di rebalancing corrisponde alla quantità necessaria affinchè il balance dell'edge sbilanciato raggiunga il 50% della capacità del rispettivo canale. In questa versione del protocollo, partendo dall'implementazione relativa alla MODIFICA 6, si prova a cambiare il valore di questa soglia. Vengono implementate due varianti: la prima con la soglia al 40%, la seconda con la soglia al 60%.

L'obiettivo è controllare se delle modifiche a questa soglia possano tradursi in benefici in termini di performance.

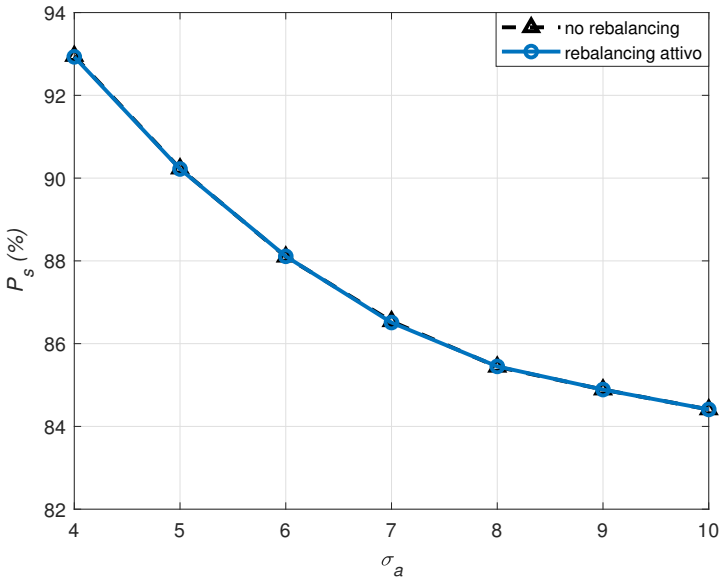

Figura 33: *Rebalancing Attivo - Modifica 8 (soglia 40%): P_s*

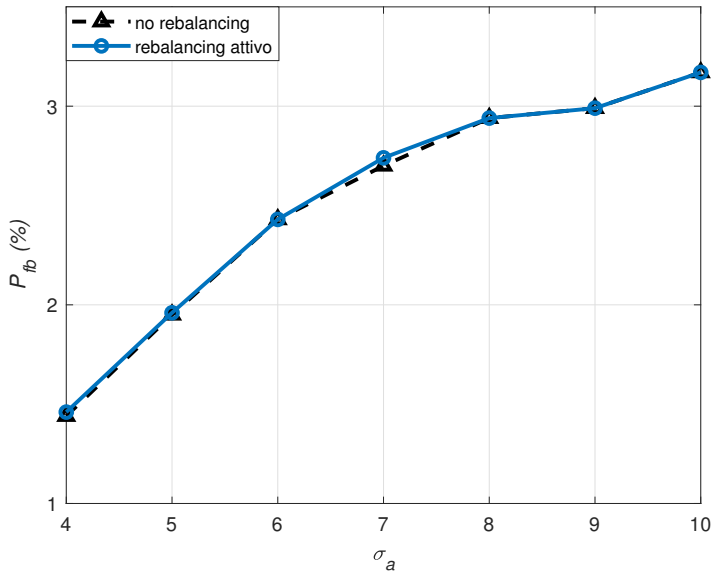

Figura 34: *Rebalancing Attivo - Modifica 8 (soglia 40%): P_{fb}*

σ_a	N° Tentativi	Successo (%)	No channel (%)	No route (%)	No balance (%)
4	53179	50.98	35.07	0.31	13.64
5	65270	49.99	37.42	0.34	12.25
6	78288	49.13	38.42	0.31	12.14
7	77921	48.58	40.30	0.29	10.83
8	86661	47.45	41.27	0.32	10.97
9	83196	48.81	40.21	0.58	10.40
10	89948	48.46	40.68	0.28	10.57

Tabella 10: *Rebalancing Attivo - Modifica 8 (soglia 40%): statistiche*

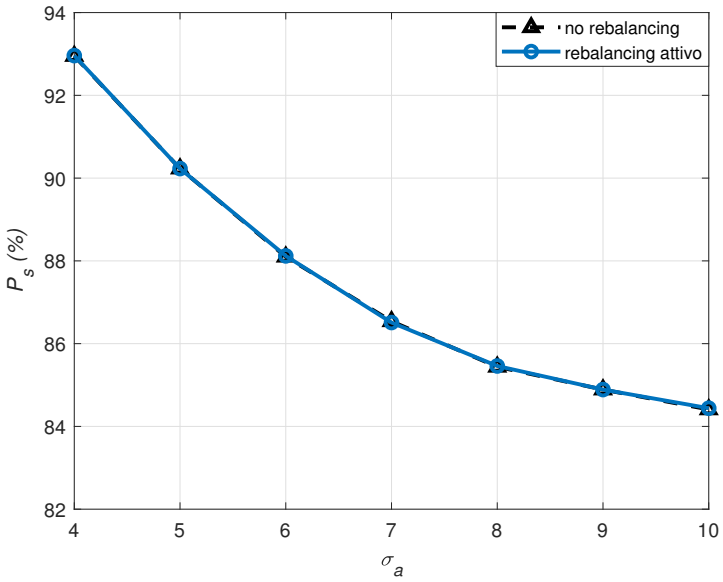

Figura 35: *Rebalancing Attivo - Modifica 8 (soglia 60%): P_s*

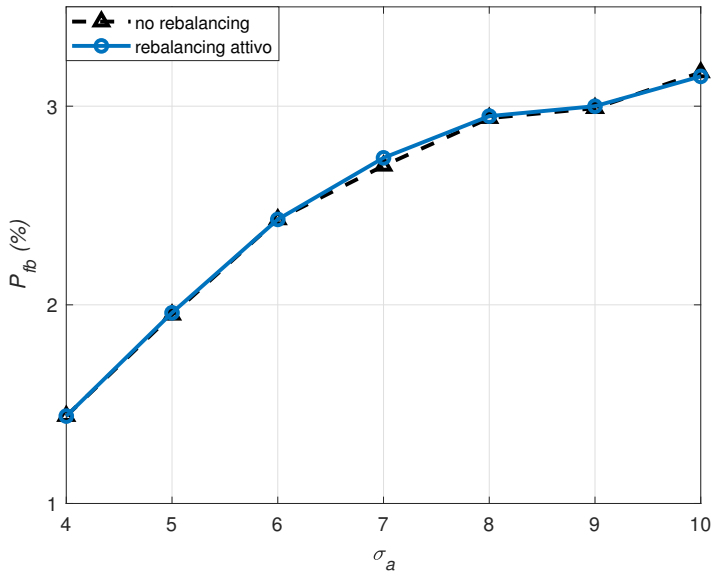

Figura 36: *Rebalancing Attivo - Modifica 8 (soglia 60%): P_{fb}*

σ_a	N° Tentativi	Successo (%)	No channel (%)	No route (%)	No balance (%)
4	42319	45.85	27.81	0.62	25.72
5	52666	45.31	28.72	0.57	25.40
6	64262	46.71	28.32	0.72	24.24
7	64958	48.26	29.11	0.57	22.06
8	67668	48.99	28.58	0.59	21.84
9	69685	48.11	29.67	0.96	21.26
10	72055	48.28	29.76	0.53	21.43

Tabella 11: *Rebalancing Attivo - Modifica 8 (soglia 60%): statistiche*

Entrambe le varianti della MODIFICA 8 non portano a considerevoli miglioramenti, così come per le modifiche precedenti, e in più vedono registrare un calo del 5-10% relativamente alla percentuale di successo dei tentativi di rebalancing rispetto alla modifica precedente. Le due implementazioni portano rispettivamente ad abbassare e aumentare la percentuale di tentativi di rebalancing falliti per no balance, e questa è la chiara conseguenza dell'aver prima ridotto e poi incrementato la quantità di *amount* associato al pagamento di rebalancing.

Anche il numero medio di tentativi per pagamento, attestandosi attorno a 1.17, non concede particolare rilievo alle ultime due implementazioni (A.10 - A.11).

Alla luce di tutto ciò, si evince che la soglia ottimale a cui portare il balance dell'edge sbilanciato sia da mantenere pari al 50% della capacità del rispettivo canale.

MODIFICA 9 Quest'ultima versione di rebalancing attivo è costruita sull'implementazione relativa alla MODIFICA 6. La MODIFICA 9 consente di minimizzare la quota del pagamento di rebalancing quel tanto che basta per consentire al nodo a cui appartiene l'edge sbilanciato di poter effettuare il pagamento. Fino a questo momento la funzione `rebalance()` veniva chiamata nel momento in cui un nodo, *dopo* aver inviato o inoltrato un pagamento, si rende conto di avere un balance al di sotto del 30% rispetto alla capacità totale del canale. In questa versione la `rebalance()` non viene più chiamata in tal modo, bensì *prima* del pagamento qualora ci si accorga che il nodo che deve effettuarlo o inoltrarlo abbia un balance inferiore all'importo del pagamento. Dunque l'*amount* del pagamento di rebalancing in questo modo equivale al necessario che consenta al nodo in questione di effettuare il pagamento.

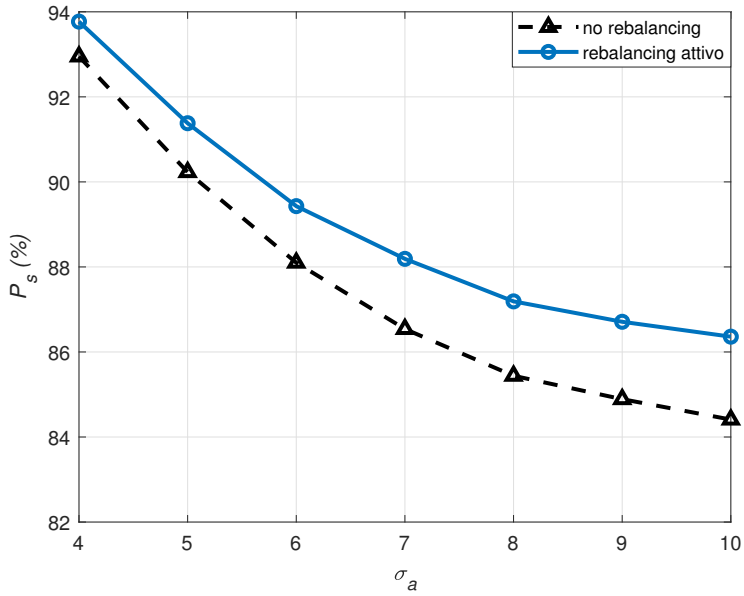

Figura 37: *Rebalancing Attivo - Modifica 9: P_s*

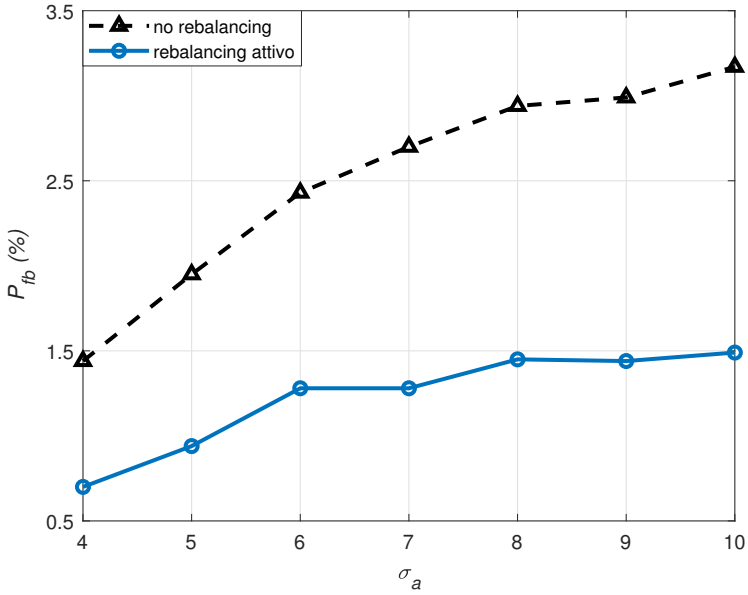

Figura 38: *Rebalancing Attivo - Modifica 9: P_{fb}*

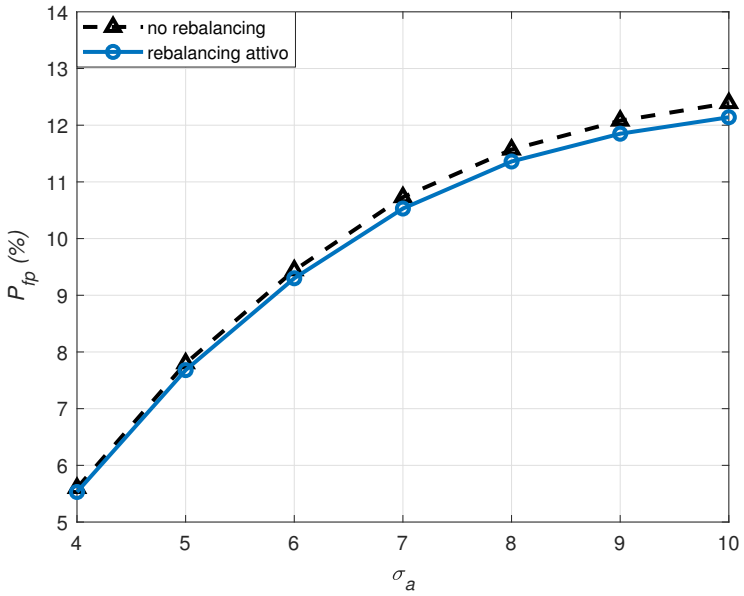

Figura 39: *Rebalancing Attivo - Modifica 9: P_{fp}*

σ_a	N° Tentativi	Successo (%)	No channel (%)	No route (%)	No balance (%)
4	31482	96.39	1.45	0.37	1.80
5	35998	95.01	2.12	0.48	2.38
6	40790	94.31	2.38	0.39	2.91
7	38645	93.38	2.77	0.57	3.27
8	35977	92.85	3.02	0.79	3.34
9	35417	92.80	3.02	0.73	3.45
10	41368	93.18	2.60	0.74	3.48

Tabella 12: *Rebalancing Attivo - Modifica 9: statistiche*

A differenza delle implementazioni precedenti, la MODIFICA 9 assolve a pieno all'obiettivo del rebalancing attivo, con un netto miglioramento delle performance dei pagamenti simulati, i quali vedono un calo relativo del 51.23% per P_{fb} rispetto allo scenario senza rebalancing, che porta ad un aumento dell'1.72% per P_s. Si segnala inoltre una riduzione della percentuale di errore P_{fp} pari al 1.69%.

Questo importante risultato trova risposta nel cambio di paradigma che vede adesso eseguire l'operazione di rebalancing solo quando essa si rende necessaria.

A questi risultati descritti si aggiunge un'impennata nelle percentuali relative ai tentativi di rebalancing, i quali registrano un successo che supera di gran lunga la soglia del 90% per tutti i valori di σ_a.

Un importante risultato è altresì raggiunto dalla media dei tentativi per pagamento che con 1.01 e una riduzione del 40.24% rispetto alla situazione di no rebalancing conosce il suo valore minimo assoluto riscontrato nell'analisi del rebalancing attivo (A.12).

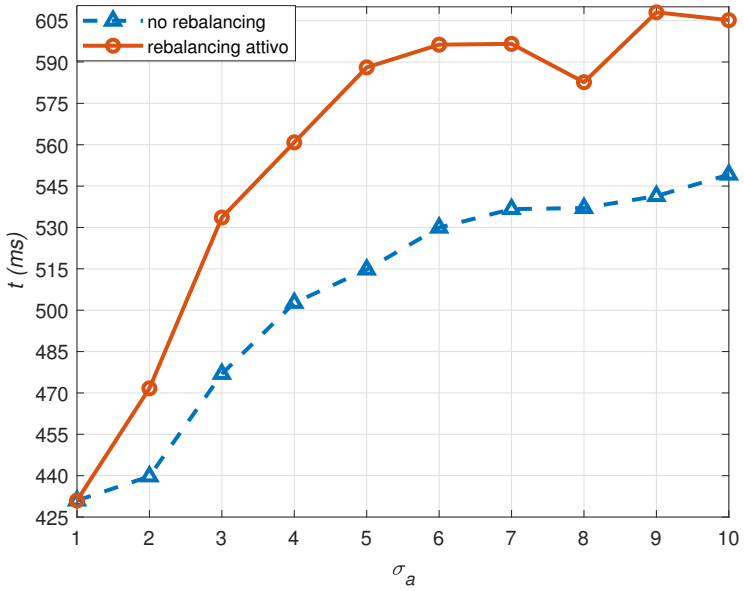

Figura 40: *Tempo medio di esecuzione dei pagamenti: confronto tra no rebalancing e rebalancing attivo*

Questa implementazione dunque inverte l'ordine con cui pagamento e operazione di rebalancing vengono eseguiti. Ciò inserisce un nuovo step tra il momento in cui un pagamento viene schedulato e quando questo viene effettivamente eseguito. Tale variazione, ignorando l'ipotesi iniziale secondo cui i pagamenti di rebalancing vengono eseguiti istantaneamente, rischia in teoria di aumentare il tempo medio impiegato per eseguire i pagamenti.

Nella Figura 40 questa ipotesi viene dimostrata dai dati raccolti che mettono a confronto, al variare di σ_a, il tempo medio di esecuzione dei pagamenti tra l'implementazione di CLoTH senza rebalancing e quella con il rebalancing attivo, nella variante della MODIFICA 9.

Da questi dati si evince che, se per σ_a=1 i tempi si equivalgono, all'aumentare di tale parametro lo scarto del tempo medio di esecuzione dei pagamenti col rebalancing attivo rispetto all'implementazione senza rebalancing si fa sempre più ampio a favore del primo, con l'apice raggiunto in corrispondenza di σ_a=5 dove la differenza è pari a 73.31 ms (A.13).

4.2.3 Riepilogo dei risultati

Nella Tabella 13 vengono riportate in percentuale le variazioni relative di performance registrate da ogni variante del protocollo di re-

balancing attivo rispetto a quelle ottenute nel caso in cui non venga implementata alcuna strategia di rebalancing.

Le statistiche raccolte sono frutto di una media dei risultati ottenuti tra σ_a=4 e σ_a=10, e fanno riferimento alla probabilità di successo dei pagamenti P_s e alla probabilità di fallimento per unbalancing P_{fb}. In ultimo viene riportata la media dei tentativi per ogni pagamento affinchè vada a buon fine per σ_a=10.

	P_s	P_{fb}	Tentativi pagamenti
Impl. iniziale	+0%	+0.11%	-7.10%
MODIFICA 1	+0.02%	-0.1%	-31.36%
MODIFICA 2	+0.01%	+0.11%	-31.36%
MODIFICA 3	-0.01%	+0.23%	-17.16%
MODIFICA 4	+0%	+0.23%	-19.53%
MODIFICA 5	+0%	+0.17%	-14.20%
MODIFICA 6	+0%	+0.19%	-30.77%
MODIFICA 7	-0.01%	+0.51%	-31.95%
MODIFICA 8 (40%)	-0.01%	+0.48%	-30.18%
MODIFICA 8 (60%)	+0.01%	+0.29%	-31.36%
MODIFICA 9	**+1.72%**	**-51.23%**	**-40.24%**

Tabella 13: *Rebalancing attivo: riepilogo dei risultati*

Tutte le modifiche sono state sviluppate in maniera incrementale basandosi ognuna sulle implementazioni precedenti. L'obiettivo è stato quello di variare il comportamento del protocollo per ottenere risultati sempre migliori.

Escludendo la soluzione più performante, quelle che andrebbero segnalate sono la MODIFICA 1 e la MODIFICA 2, le quali introducono un meccanismo che insiste in fase di rebalancing nel cercare l'edge candidato migliore che abbia un percorso garantito verso l'edge sbilanciato. Queste modifiche arrivano a registrare per P_s un miglioramento massimo relativo rispettivamente dello 0.05% e 0.04% rispetto allo scenario senza rebalancing; per P_{fb} le due modifiche riportano un calo massimo relativo dello 0.82% e 0.63%; in entrambi i casi il numero di tentativi per ogni pagamento per andare a buon fine è pari a 1.16. Queste modifiche non si rivelano particolarmente efficaci dato che più di un terzo dei pagamenti di rebalancing fallisce per no channel: cioè il nodo che fa il ribilanciamento non riesce a trovare un edge candidato.

La versione di rebalancing attivo migliore in assoluto è quella relativa alla MODIFICA 9, che arriva a registrare un miglioramento massimo relativo del 2.31% per P_s e un abbassamento massimo relativo per P_{fb} del 53% rispetto allo scenario senza rebalancing. Il numero di

tentativi per ogni pagamento per andare a buon fine si avvicina all'unità con 1.01. La peculiarità della MODIFICA 9 consiste nell'eseguire l'operazione di rebalancing *prima* del pagamento e *solo se è necessaria*.

4.3 REBALANCING PASSIVO

Ogni pagamento attraversa un percorso detto *route* che viene deciso dall'algoritmo di Dijkstra. Esso ha la caratteristica di preferire percorsi formati da edge con commissioni basse.

Il rebalancing passivo ha dunque lo scopo di regolare le policy sulle commissioni imposte dagli edge in modo tale che queste siano inversamente proporzionali ai rispettivi balance.

L'obiettivo è spingere i pagamenti ad attraversare edge con balance alti, limitando quindi il problema dell'unbalancing [10].

4.3.1 Implementazione iniziale

La prima forma di rebalancing passivo che è stata presa in considerazione nei test e su cui sono state effettuate le successive modifiche è quella messa a punto in [10].

Nell'analisi di questa soluzione e delle successive è interessante analizzare non solo P_s e P_{fb}, ma anche l'andamento della media delle tariffe totali associate ad ogni pagamento che ha avuto successo[9], ciascuna frutto della somma delle singole commissioni relative agli edge attraversati dal pagamento considerato.

Questo dato è facilmente correlabile a P_{fp}: può capitare infatti che peggioramenti relativi a P_{fp} possano essere causati da aumenti di commissioni che impediscono ad uno o più pagamenti di seguire un certo percorso.

In questa implementazione l'operazione di rebalancing viene gestita da una funzione apposita chiamata `updateFee()`, la quale viene evocata ogni volta che un nodo invia o riceve un pagamento e che quindi vede modificare il balance di un suo edge. Essa si occupa di aggiornare la commissione associata all'edge stesso in funzione del nuovo valore del rispettivo balance.

9 Per σ_a=10.

Di seguito l'algoritmo:

A. Definizione di K, parametro che determina il valore massimo della nuova commissione;

B. Calcolo del rapporto X tra il balance dell'edge e la capacità del canale corrispondente;

C. Calcolo della nuova commissione Y associata all'edge, dipendente da X, adottando un'apposita funzione inversamente proporzionale che usa K come parametro.

Il valore K scelto è un trade-off tra tutte le commissioni adottate in Lightning Network su cui sono state condotte le simulazioni.

In particolare K è un'astrazione di due misure relative alle commissioni, ossia i valori massimi di *FeeBase* e *FeeProp* (MFB e MFP): FeeBase è la commissione di base che viene applicata da un edge a tutti i pagamenti che l'attraversano; FeeProp è una commissione che dipende dall'importo del pagamento che attraversa l'edge. Il parametro K prevede che MFB=2000 e MFP=2[10].

La funzione inversamente proporzionale passa da due punti: il punto minimo, in corrispondenza del quale X=1 ed entrambe le commissioni sono poste a 0; il punto massimo, per il quale X=0 e le commissioni assumono il rispettivo valore massimo regolato da K.

In questa versione del protocollo la funzione inversamente proporzionale scelta è la *retta* mostrata in Figura 41.

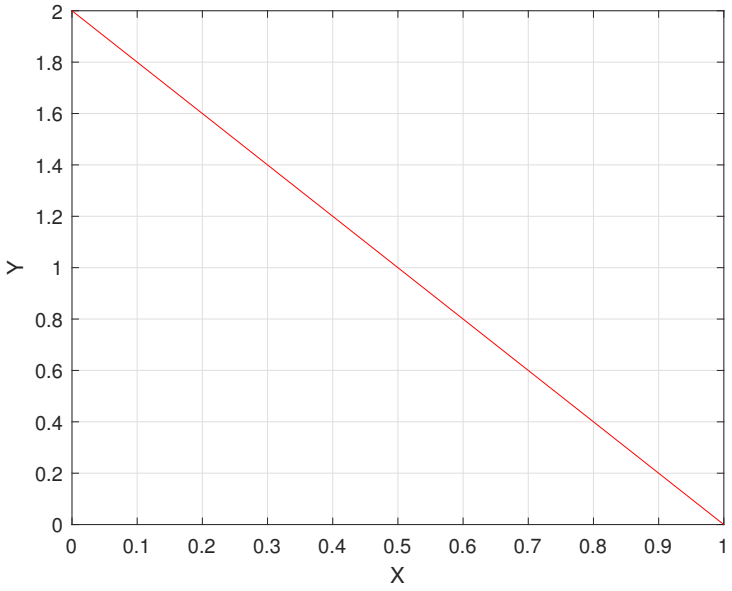

Figura 41: *Retta*

10 Se si considera ad esempio $2K$, allora MFB=4000 e MFP=4.

Di seguito vengono mostrate le prestazioni di questa prima versione di rebalancing passivo.

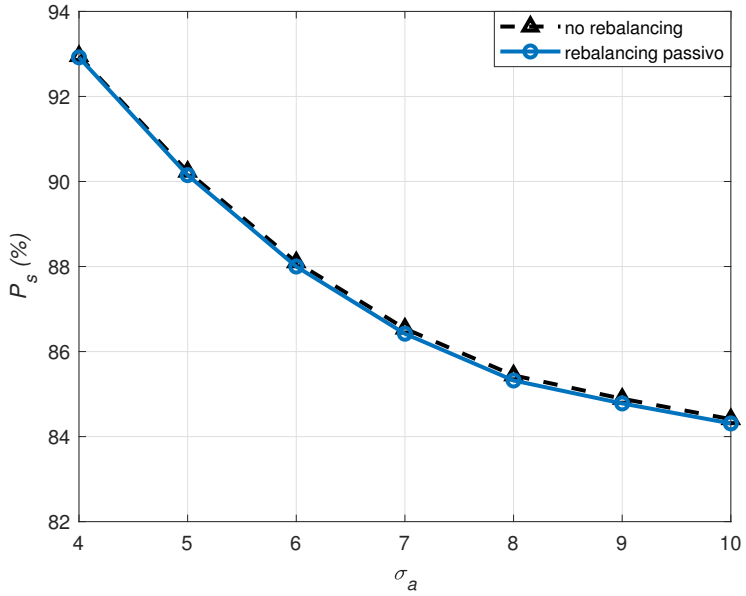

Figura 42: *Rebalancing Passivo - Setup iniziale:* P_s

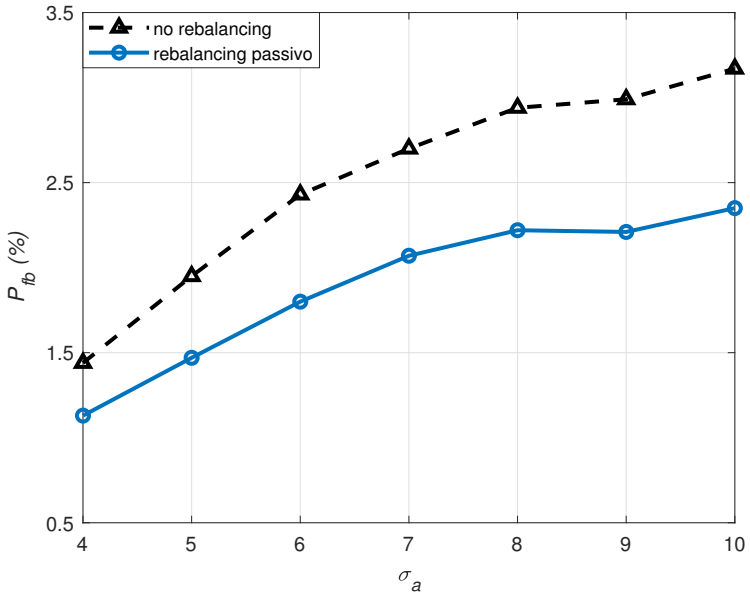

Figura 43: *Rebalancing Passivo - Setup iniziale:* P_{fb}

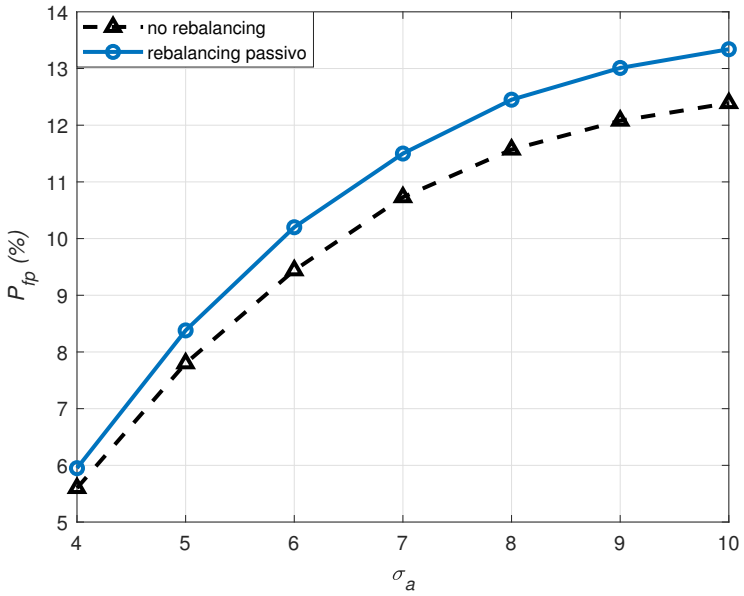

Figura 44: *Rebalancing Passivo - Setup iniziale: P_{fp}*

Questa prima implementazione della strategia di rebalancing passivo fa riscontare rispetto allo scenario senza rebalancing una riduzione relativa della percentuale di errore P_{fb} del 24.55% che, a fronte di un aumento del 7.41% di P_{fp}, determina un lieve decremento per P_s pari allo 0.11%[11] (A.14).

Il peggioramento nei fallimenti per assenza di percorsi è causato dall'aumento delle tariffe di alcuni canali, com'è possibile dedurre da un'analisi dell'output dei pagamenti.

Da notare che il punto di flesso nella Figura 43 relativa a P_{fb} in corrispondenza di σ_a=9, già notato in occasione del rebalancing attivo, qui e nelle prossime modifiche si presenta in forma più marcata, rompendo addirittura la crescita monotonica della percentuale.

Le modifiche a seguire rispettano lo stesso algoritmo dell'implementazione di partenza; esse mirano però ad utilizzare di volta in volta funzioni inversamente proporzionali differenti, mantenendo K costante.

L'obiettivo di questo studio, testando diverse funzioni, è di trovare la forma d'onda che consenta di ottimizzare l'andamento delle commissioni al variare del balance di ciascun edge, in modo tale da limitare il problema dell'unbalancing e migliorare dunque le prestazioni.

11 Tutti i miglioramenti/peggioramenti di performance sono relativi e considerati usando come termine di paragone le prestazioni garantite nello scenario di no rebalancing. Essi inoltre sono calcolati facendo una media al variare di σ_a da 4 a 10.

4.3.2 Simulazioni

Nelle seguenti simulazioni, per ogni modifica, sono state provate varie funzioni che regolano le commissioni in base alla balance. Per ogni modifica sono state avviate delle simulazioni con un rate r_π pari a 100 pagamenti al secondo e con σ_a che varia da 4 a 10.

MODIFICA 1 La funzione adottata è un'*iperbole equilatera*, mostrata in Figura 45.

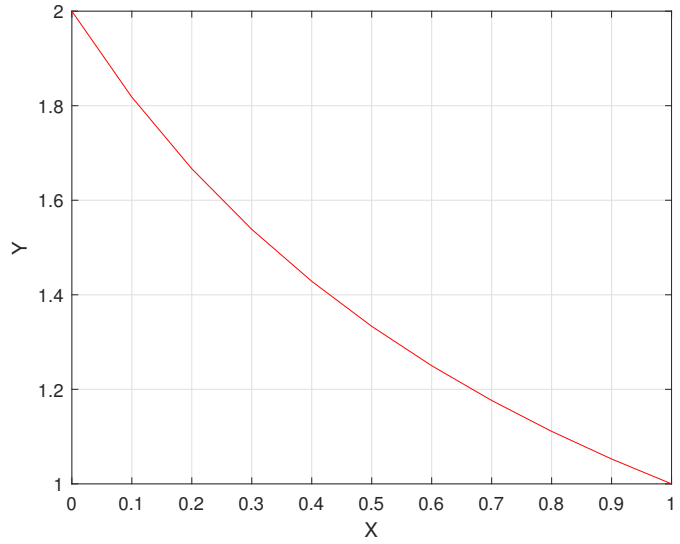

Figura 45: *Iperbole equilatera*

Di seguito le prestazioni:

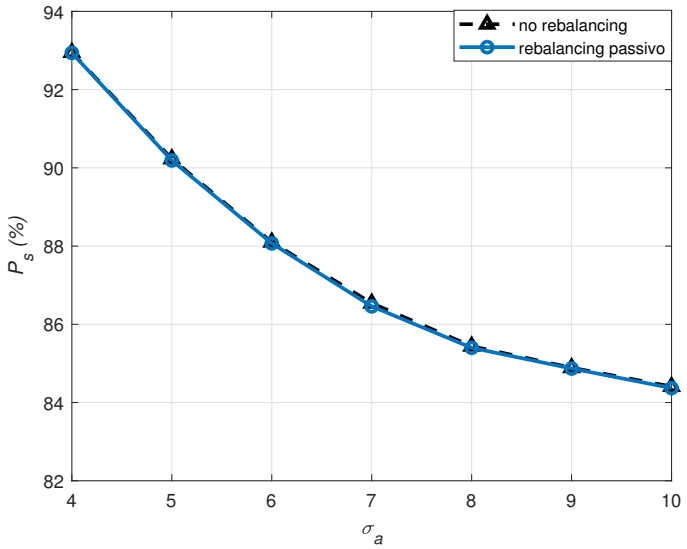

Figura 46: *Rebalancing Passivo - Modifica 1: P_s*

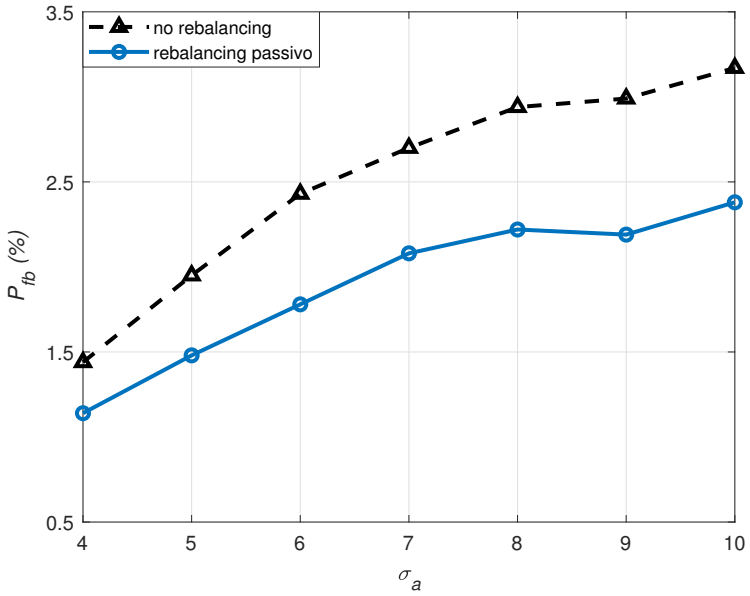

Figura 47: *Rebalancing Passivo - Modifica 1: P_{fb}*

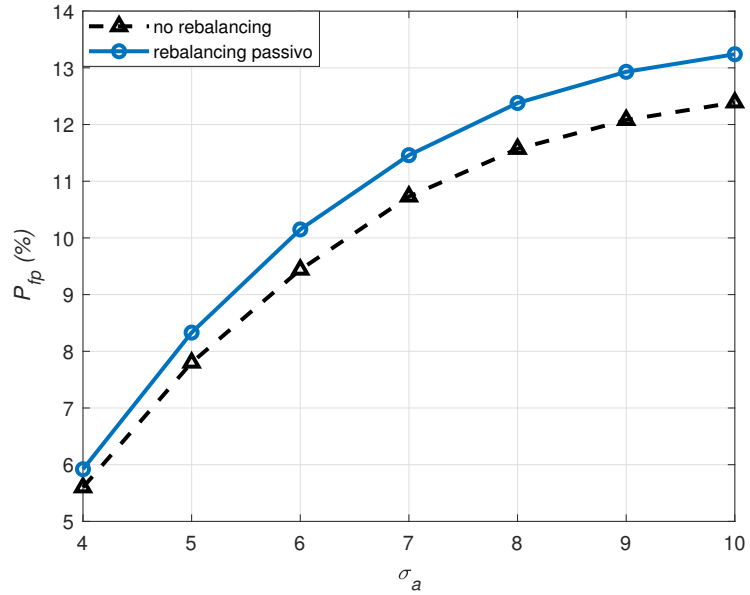

Figura 48: *Rebalancing Passivo - Modifica 1: P_{fp}*

La MODIFICA 1 risulta migliore rispetto all'implementazione iniziale. P_s peggiora dello 0.04% rispetto al caso no rebalancing, mentre P_{fb} ha un calo del 24.40%.

Il valor medio di P_{fp} rientra determinando un aumento del 6.82% rispetto alla situazione senza rebalancing (A.15).

MODIFICA 2 La funzione adottata è una *funzione del quarto ordine*, mostrata in Figura 49.

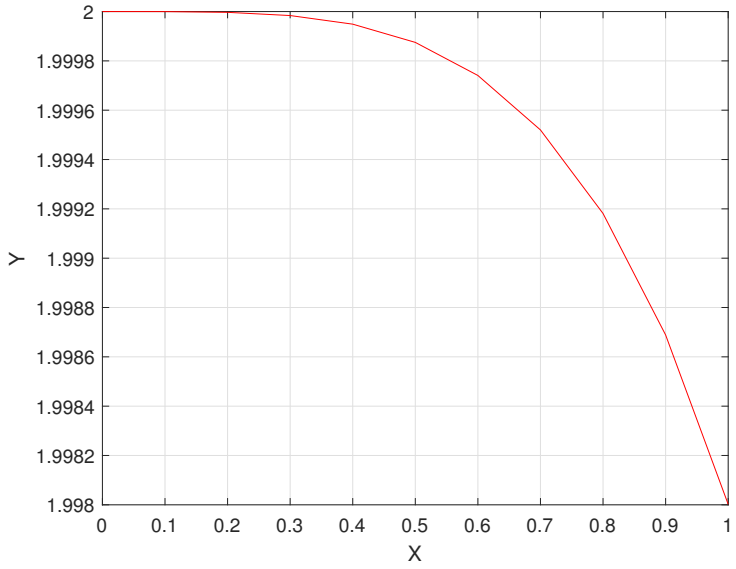

Figura 49: *Funzione del quarto ordine*

Di seguito le prestazioni:

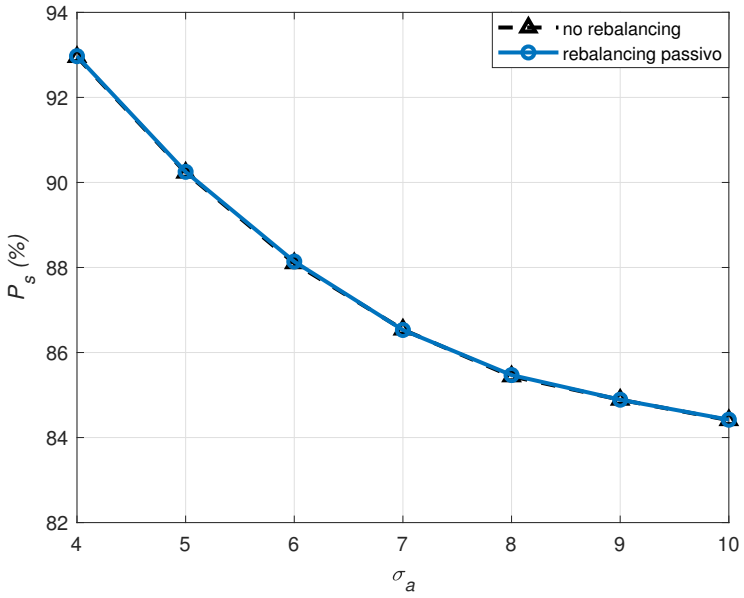

Figura 50: *Rebalancing Passivo - Modifica 2: P_s*

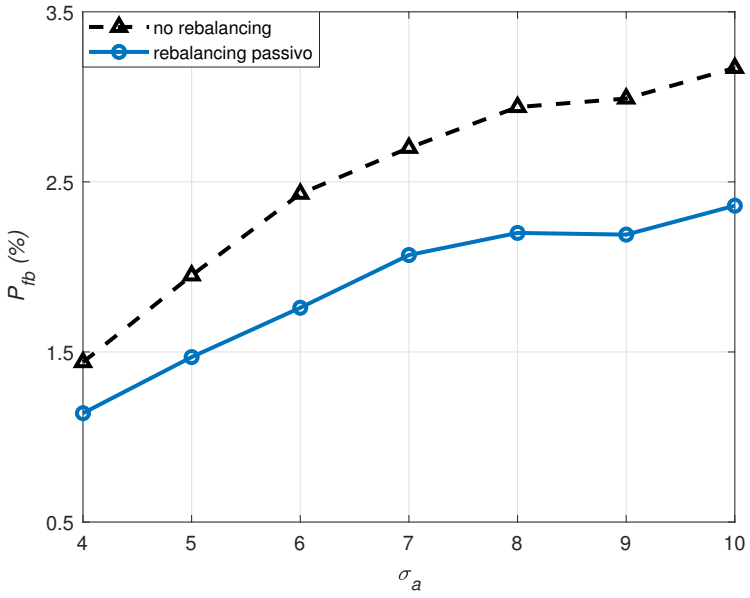

Figura 51: *Rebalancing Passivo - Modifica 2: P_{fb}*

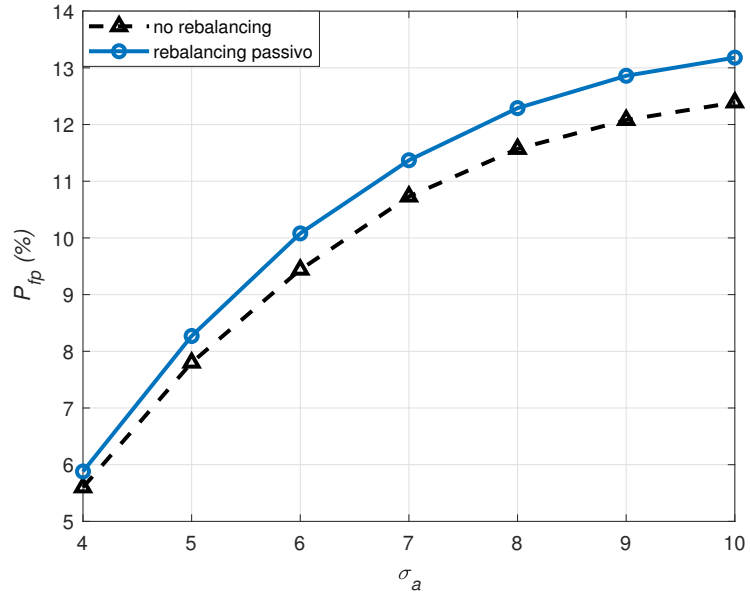

Figura 52: *Rebalancing Passivo - Modifica 2: P_{fp}*

La funzione del quarto ordine determina un'inversione di tendenza, visto che P_s migliora dello 0.02%, mentre P_{fb} ha un decremento del 25.98%. Rispetto alla MODIFICA 1 c'è un miglioramento per P_{fp}, che adesso supera il valore registrato nel caso no rebalancing del 6.12% (A.16).

MODIFICA 3 La funzione adottata è un *esponenziale*, mostrato in Figura 53.

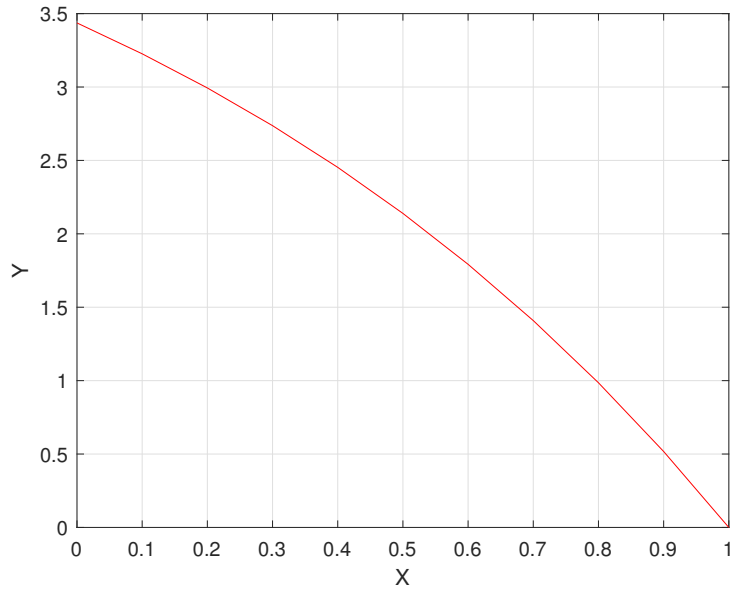

Figura 53: *Funzione esponenziale*

Di seguito le prestazioni:

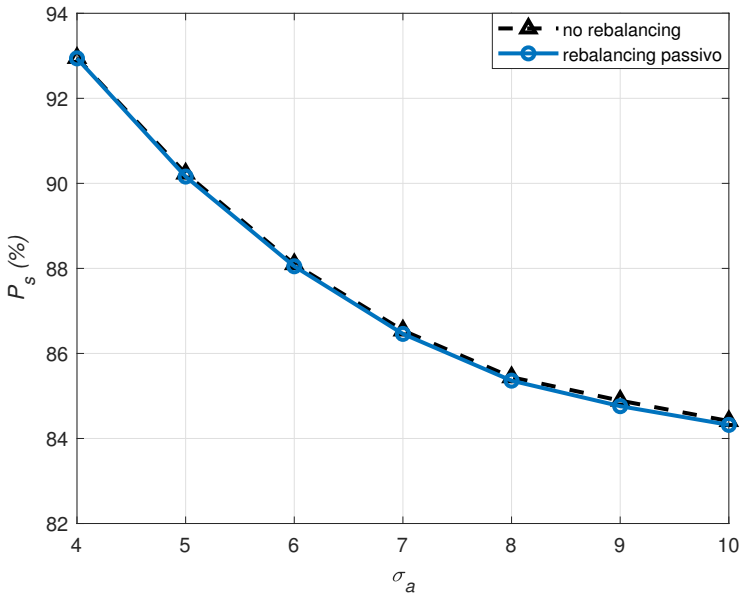

Figura 54: *Rebalancing Passivo - Modifica 3: P_s*

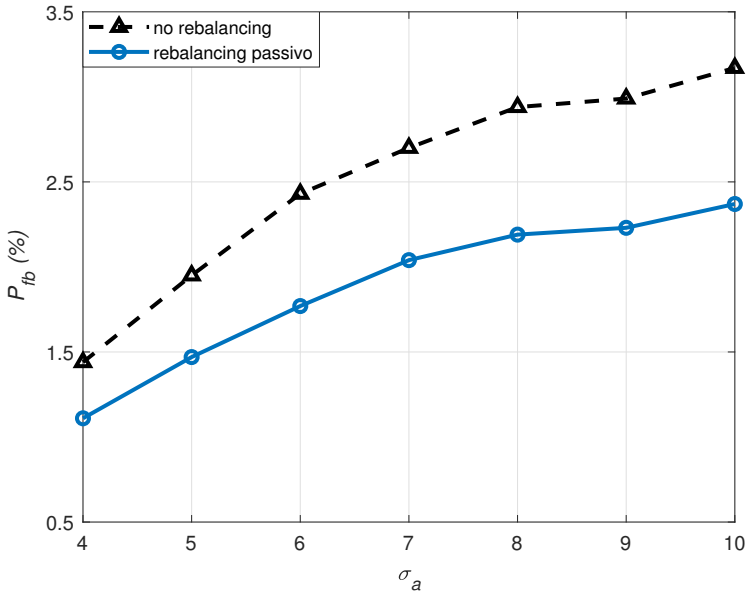

Figura 55: *Rebalancing Passivo - Modifica 3:* P_{fb}

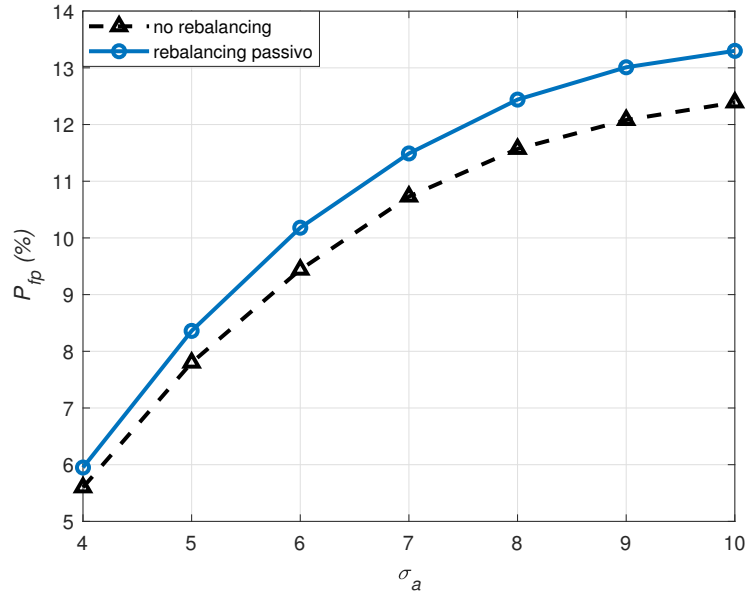

Figura 56: *Rebalancing Passivo - Modifica 3:* P_{fp}

Questa versione di rebalancing passivo riporta le performance dei pagamenti al di sotto della situazione senza rebalancing: P_s peggiora dello 0.08% e P_{fp} del 7.27% (A.17).

MODIFICA 4 La funzione adottata è una *funzione logaritmica*, mostrata in Figura 57.

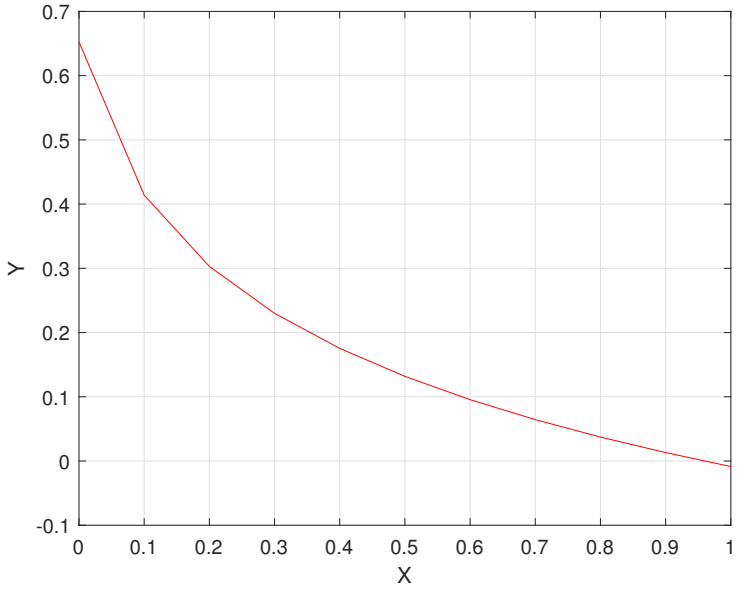

Figura 57: *Funzione logaritmica*

Di seguito le prestazioni:

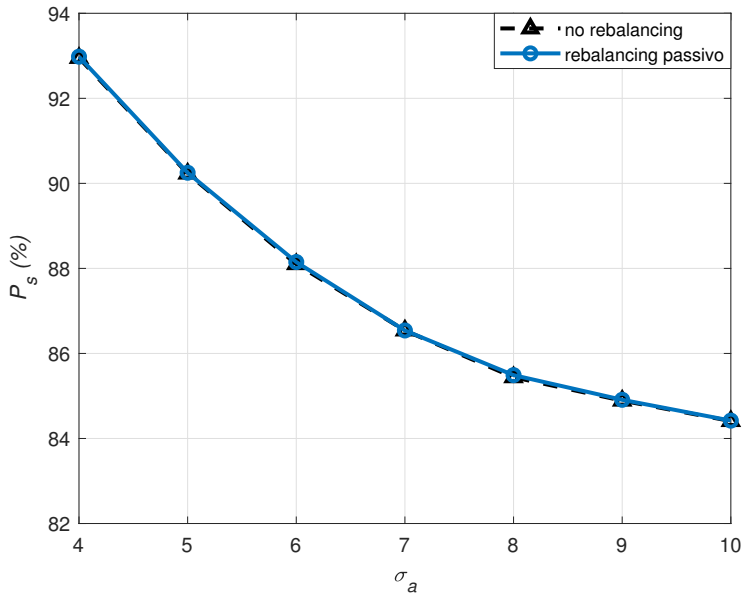

Figura 58: *Rebalancing Passivo - Modifica 4: P_s*

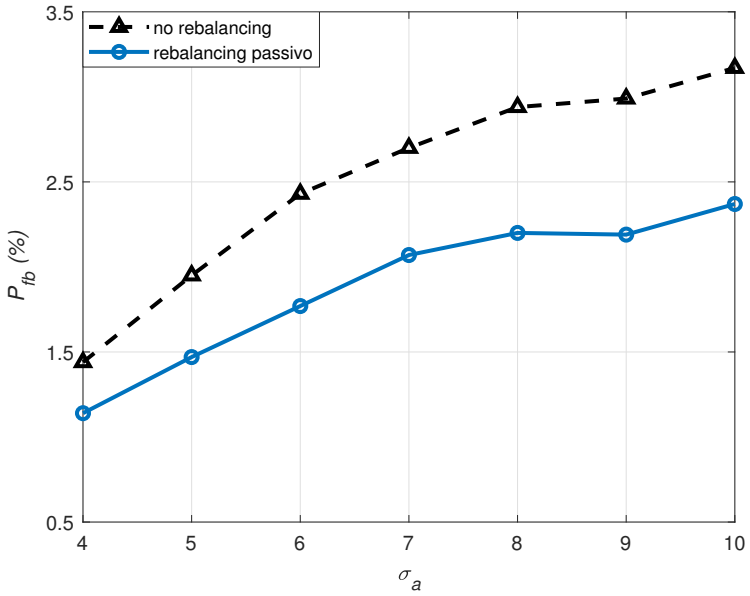

Figura 59: *Rebalancing Passivo - Modifica 4: P_{fb}*

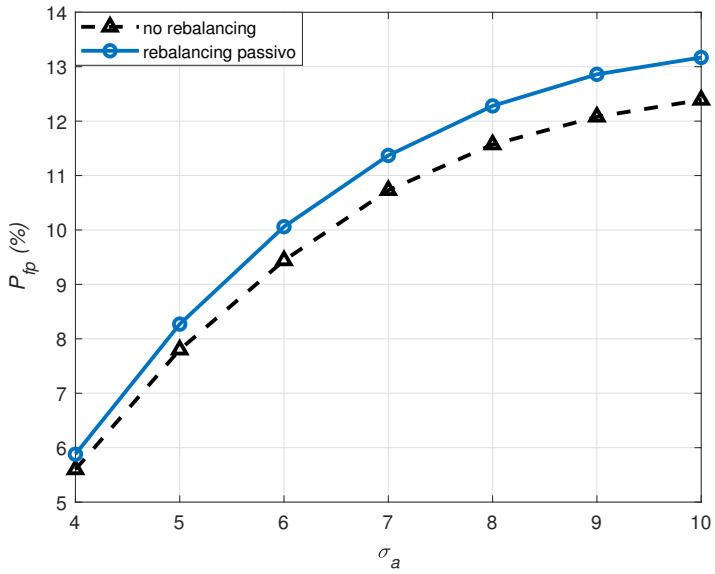

Figura 60: *Rebalancing Passivo - Modifica 4: P_{fp}*

La funzione logaritmica garantisce il risultato migliore finora registrato relativamente al rebalancing passivo, seppur solo leggermente superiore a quello ottenuto nella MODIFICA 2 con la funzione del quarto ordine: P_s migliora dello 0.03% rispetto alla situazione di no rebalancing, mentre P_{fb} del 24.73%. P_{fp} riporta il deficit più piccolo finora registrato, peggiorando solo del 6.07% (A.18).

MODIFICA 5 La funzione adottata è una *parabola*, mostrata in Figura 61.

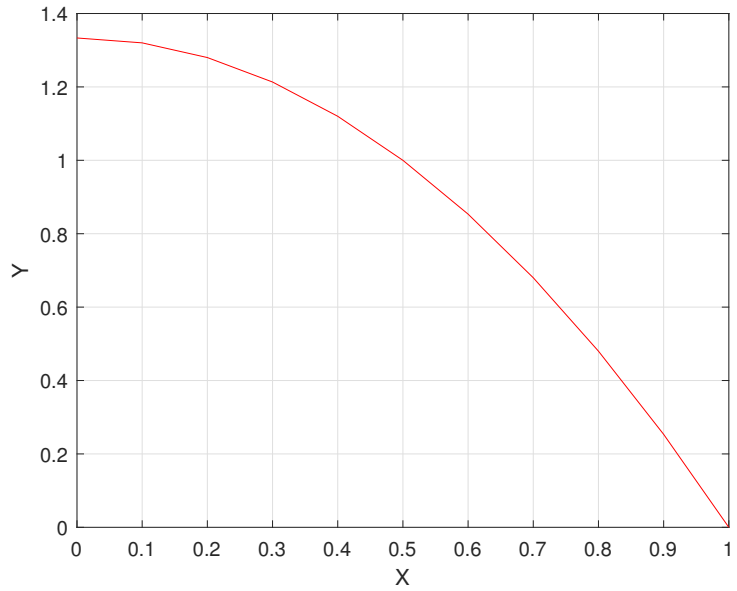

Figura 61: *Parabola*

Di seguito le prestazioni:

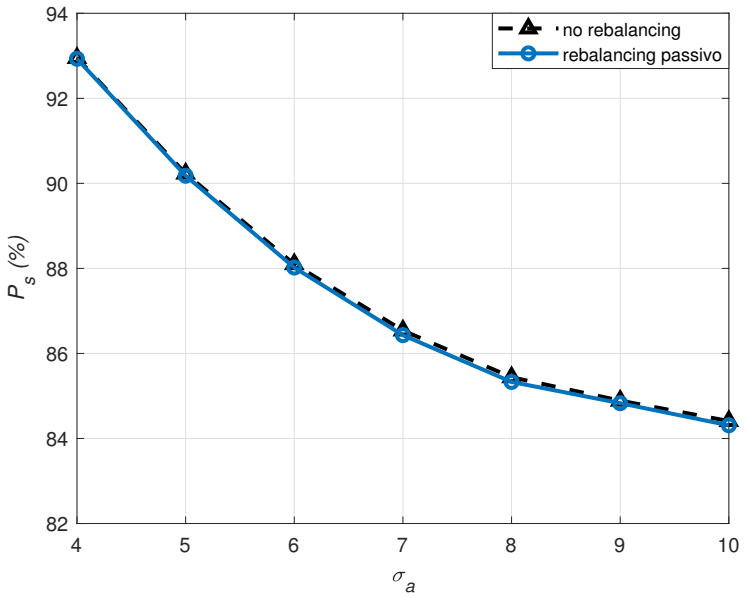

Figura 62: *Rebalancing Passivo - Modifica 5: P_s*

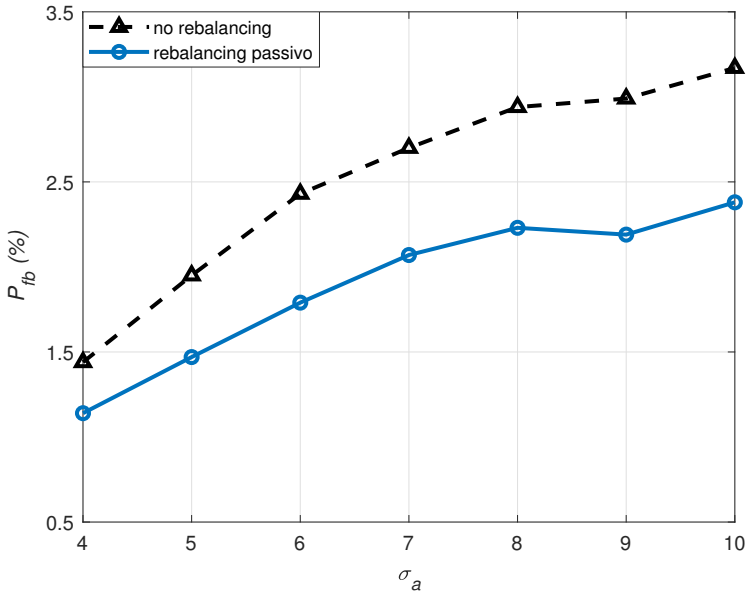

Figura 63: *Rebalancing Passivo - Modifica 5:* P_{fb}

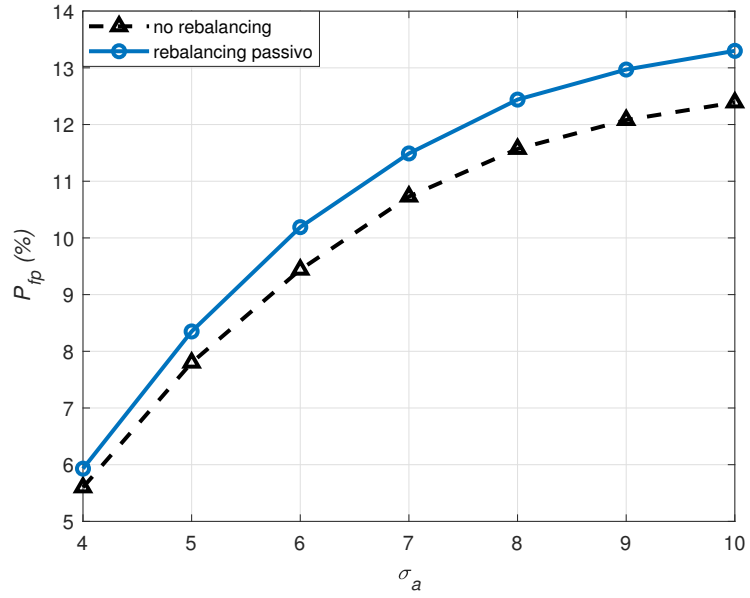

Figura 64: *Rebalancing Passivo - Modifica 5:* P_{fp}

Questa implementazione fa peggio di tutte le precedenti, dato che P_s si abbassa dello 0.09%. A ciò si accompagna il miglioramento di P_{fb} del 24.42% e l'involuzione del 7.17% di P_{fp} (A.19).

MODIFICA 6 La funzione adottata è una *funzione sigmoidea*, mostrata in Figura 65.

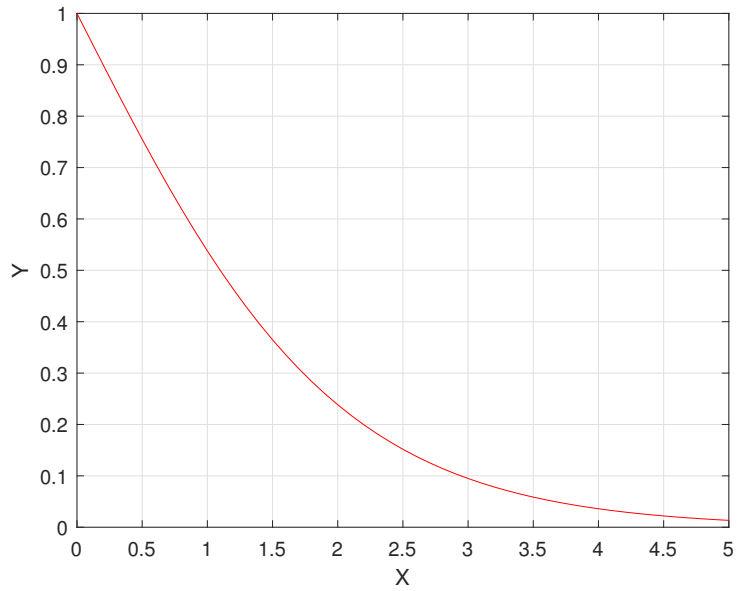

Figura 65: *Sigmoide*

Di seguito le prestazioni:

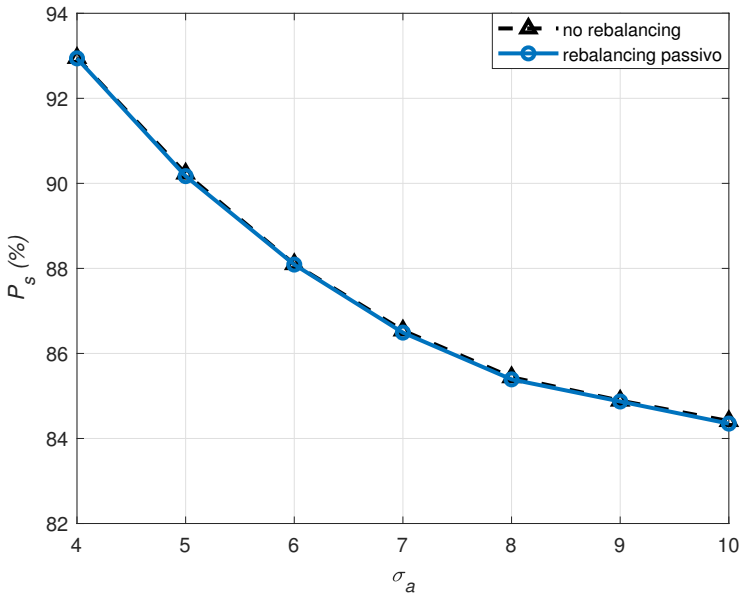

Figura 66: *Rebalancing Passivo - Modifica 6: P_s*

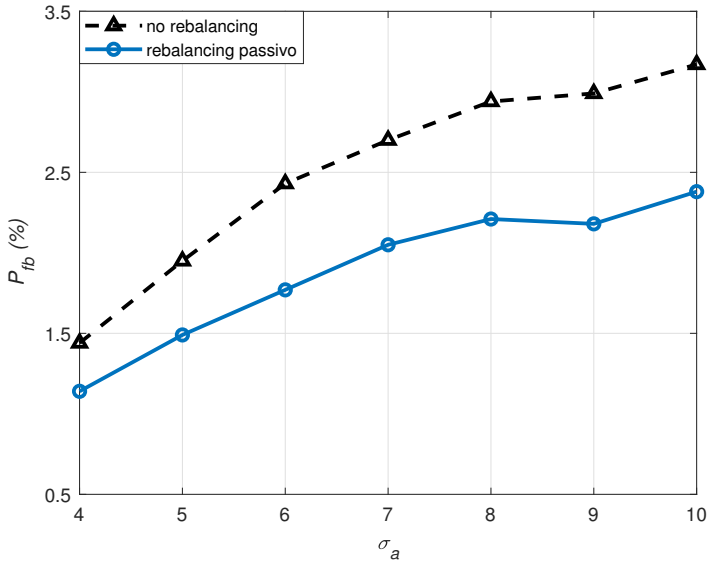

Figura 67: *Rebalancing Passivo - Modifica 6:* P_{fb}

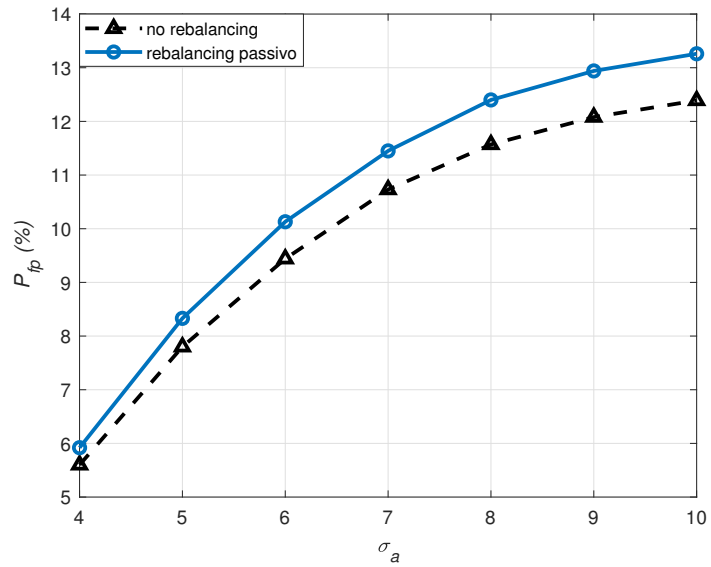

Figura 68: *Rebalancing Passivo - Modifica 6:* P_{fp}

Il sigmoide migliora le performance registrate nella MODIFICA 5, ma mantiene P_s al di sotto dello 0.04% rispetto allo scenario senza rebalancing. Un lieve miglioramento si riscontra anche per P_{fb} che cala del 24.64%, e per P_{fp} che fa rientrare il proprio incremento al 6.83% (A.20).

MODIFICA 7 Considerando che la funzione inversamente proporzionale che ha garantito risultati migliori è la *funzione logaritmica* adottata nella MODIFICA 4, nelle Tabelle A.21 - A.22 - A.23 - A.24 vengono mostrate le prestazioni della rete adottando questa funzione al

variare di K.

Dai dati raccolti si evince che le performance migliori si ottengono quando il parametro utilizzato è *16K*: anche se P_s mantiene il proprio incremento stabile a 0.03%, la media delle commissioni relative ai pagamenti andati a buon fine risulta essere la più bassa.

Tali prestazioni rimangono tuttavia abbondantemente inferiori rispetto a quelle registrate in corrispondenza della miglior soluzione di rebalancing attivo (MODIFICA 9).

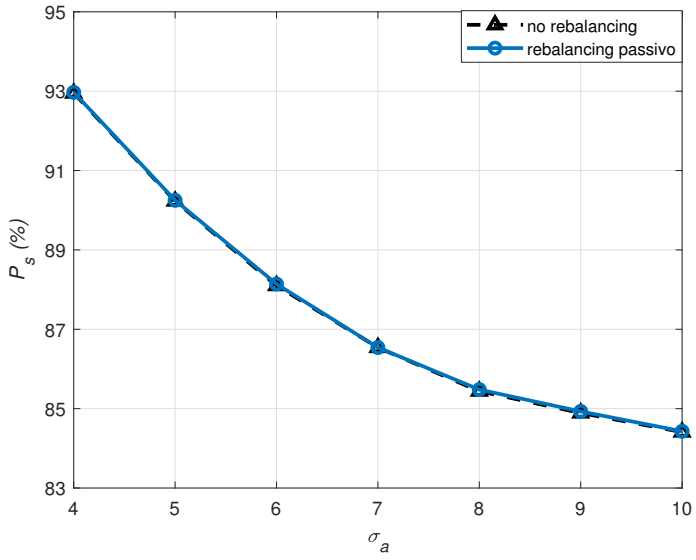

Figura 69: *Rebalancing Passivo - Modifica 7 con K/2: P_s*

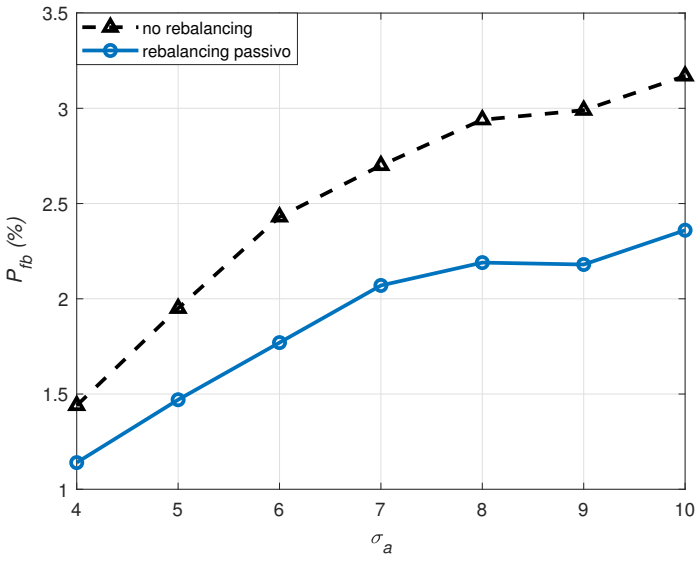

Figura 70: *Rebalancing Passivo - Modifica 7 con K/2: P_{fb}*

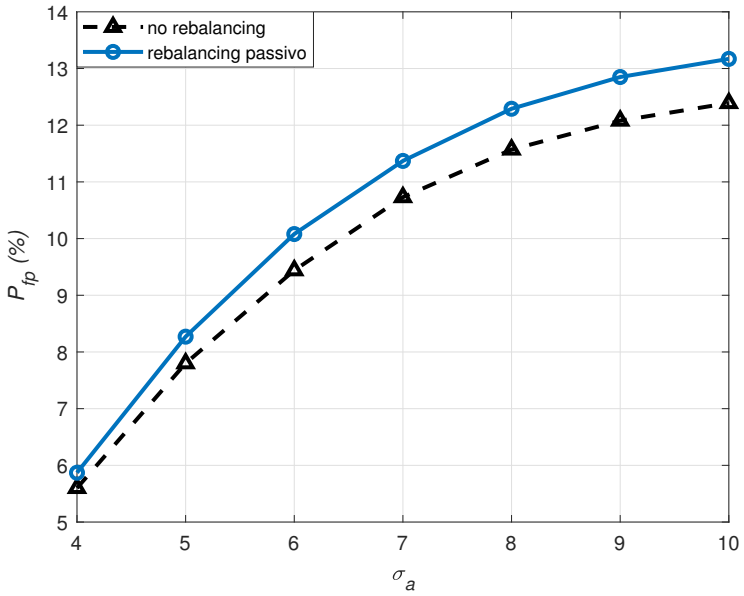

Figura 71: *Rebalancing Passivo - Modifica 7 con K/2:* P_{fp}

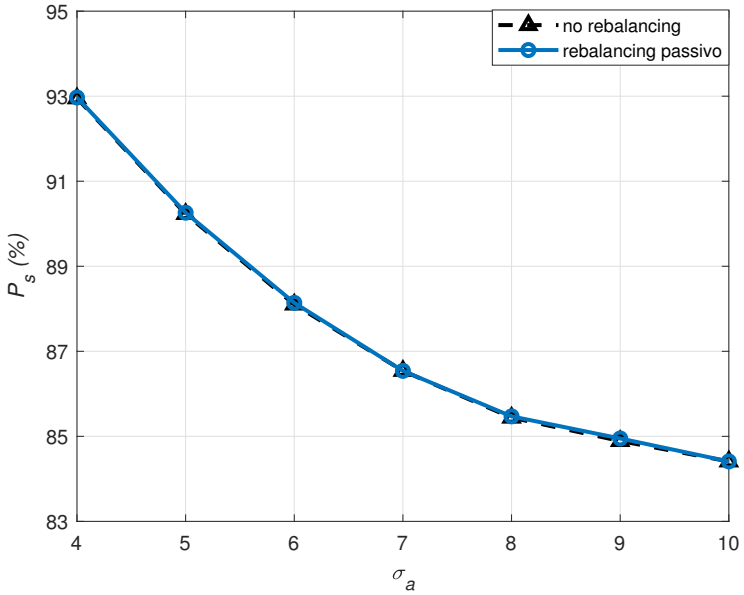

Figura 72: *Rebalancing Passivo - Modifica 7 con 2K:* P_s

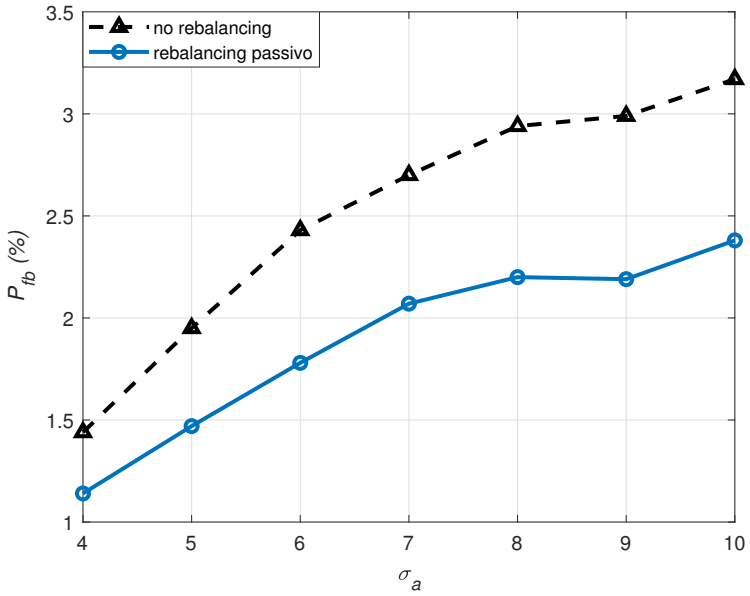

Figura 73: *Rebalancing Passivo - Modifica 7 con 2K: P_{fb}*

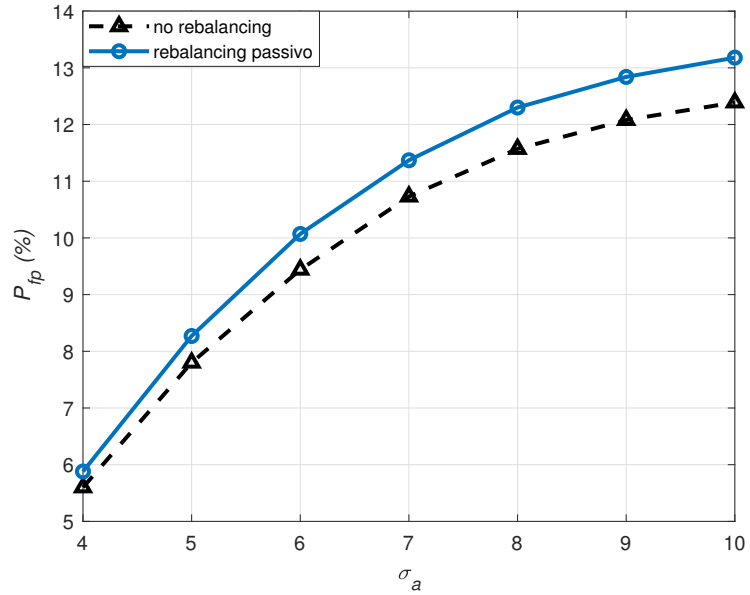

Figura 74: *Rebalancing Passivo - Modifica 7 con 2K: P_{fp}*

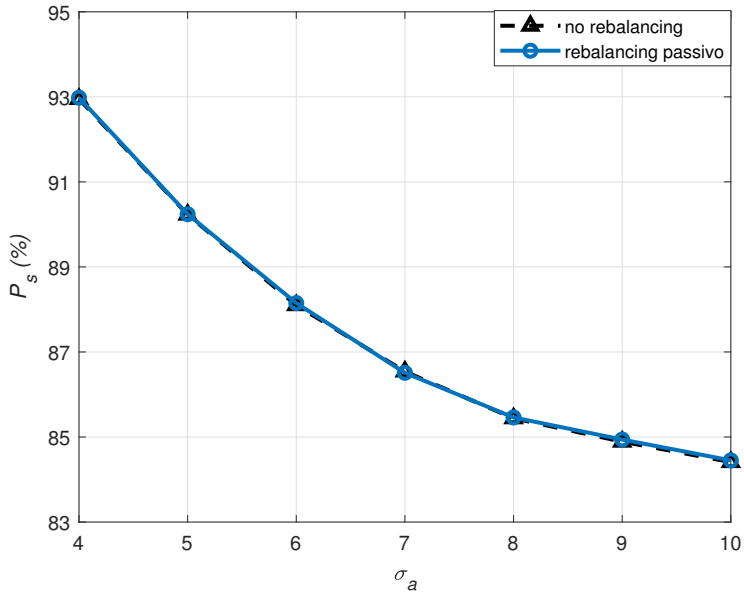

Figura 75: *Rebalancing Passivo - Modifica 7 con 16K: P_s*

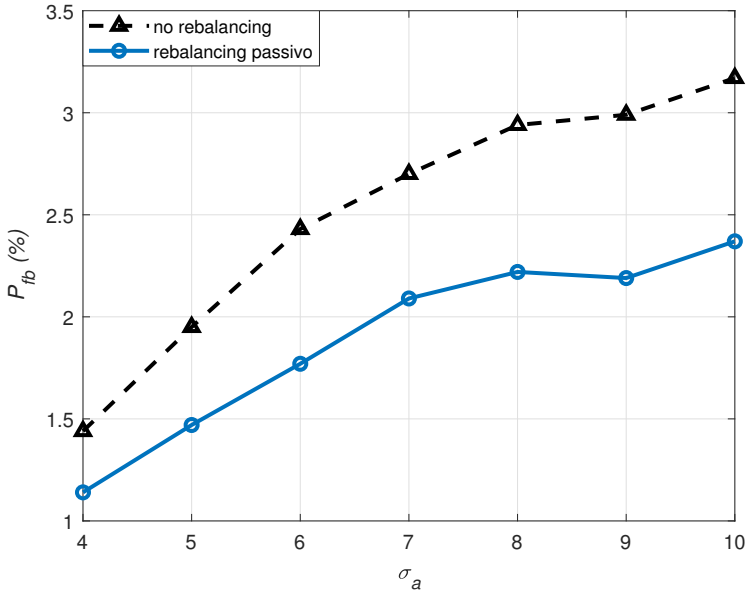

Figura 76: *Rebalancing Passivo - Modifica 7 con 16K: P_{fb}*

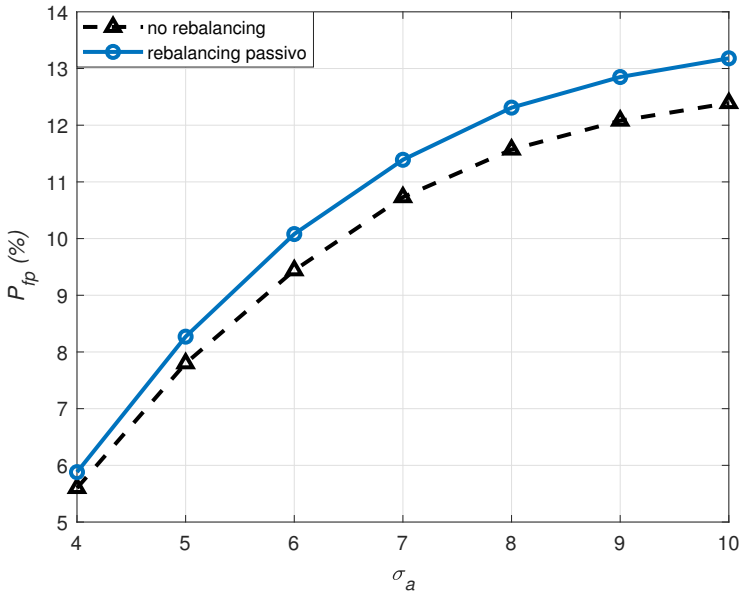

Figura 77: *Rebalancing Passivo - Modifica 7 con 16K: P_{fp}*

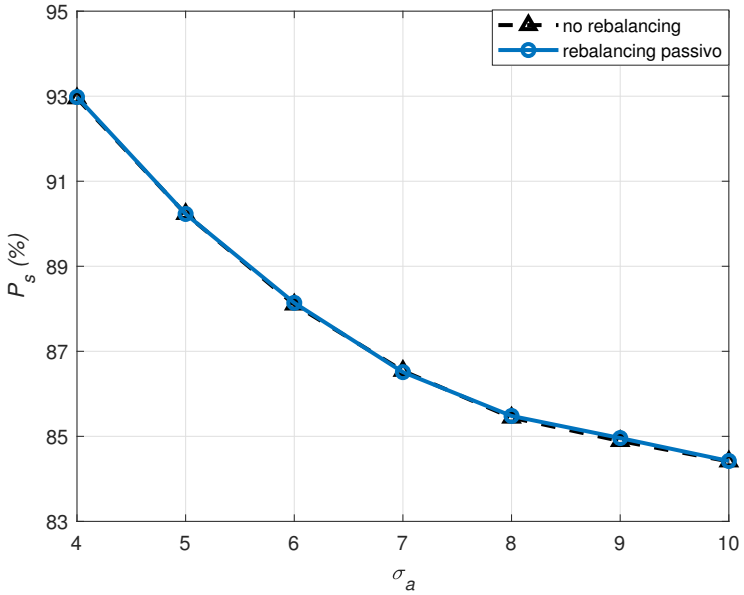

Figura 78: *Rebalancing Passivo - Modifica 7 con 32K: P_s*

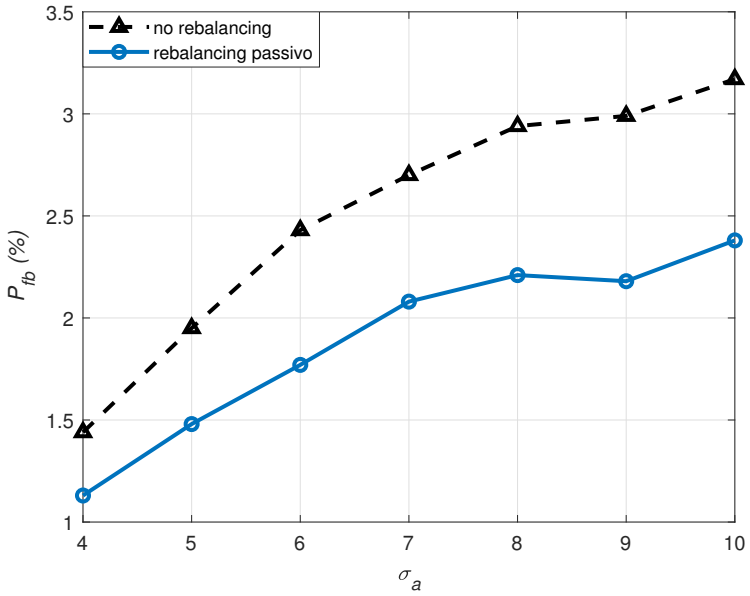

Figura 79: *Rebalancing Passivo - Modifica 7 con 32K:* P_{fb}

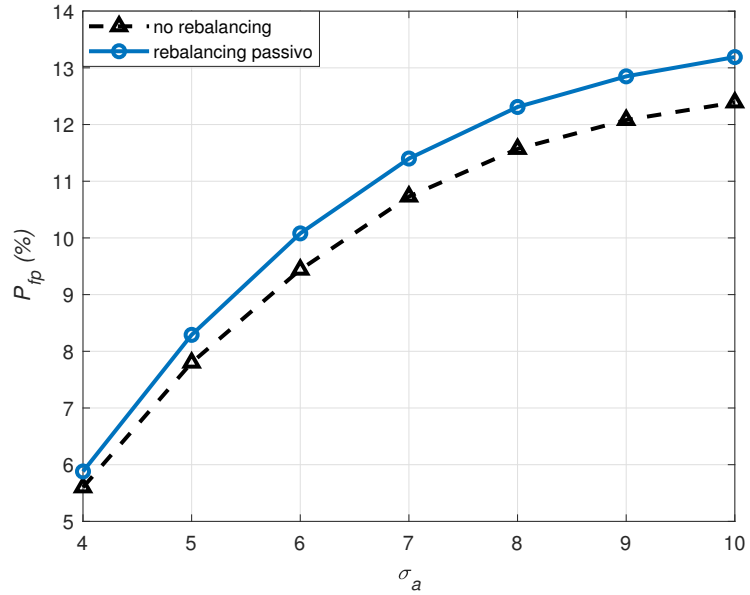

Figura 80: *Rebalancing Passivo - Modifica 7 con 32K:* P_{fp}

4.3.3 Riepilogo dei risultati

Nella Tabella 14 vengono riportate in percentuale le variazioni relative di performance registrate da ogni variante del protocollo di rebalancing passivo rispetto a quelle ottenute nel caso in cui non venga implementata alcuna strategia di rebalancing.

Le statistiche raccolte sono frutto di una media dei risultati ottenuti tra σ_a=4 e σ_a=10, e fanno riferimento alla probabilità di successo

dei pagamenti P_s, alla probabilità di fallimento per unbalancing P_{fb} e assenza di percorsi P_{fp}, e alla media delle commissioni legate ai pagamenti andati a buon fine.

	$\mathbf{P_s}$	$\mathbf{P_{fb}}$	$\mathbf{P_{fp}}$	**Commissioni**
Impl. iniziale	-0.11%	-24.55%	+7.41%	-68.24%
MODIFICA 1	-0.04%	-24.40%	+6.82%	-61.95%
MODIFICA 2	+0.02%	-25.98%	+6.12%	-82.68%
MODIFICA 3	-0.08%	-25.04%	+7.27%	-50.32%
MODIFICA 4	+0.03%	-24.73%	+6.07%	-80.92%
MODIFICA 5	-0.09%	-24.42%	+7.17%	-71.42%
MODIFICA 6	-0.04%	-24.64%	+6.83%	-73.70%
MODIFICA 7 (K/2)	+0.03%	-24.87%	+6.07%	-81.57%
MODIFICA 7 (2K)	+0.03%	-24.62%	+6.09%	-76.87%
MODIFICA 7 (16K)	**+0.03%**	**-24.53%**	**+6.16%**	**-84.37%**
MODIFICA 7 (32K)	+0.03%	-24.66%	+6.22%	-77.47%

Tabella 14: *Rebalancing passivo: riepilogo dei risultati*

In ogni implementazione, testando diverse funzioni, l'obiettivo principale è stato quello di trovare la forma d'onda che consentisse di ottimizzare l'andamento delle commissioni al variare del balance di ciascun edge, in modo tale da limitare il problema dell'unbalancing e migliorare dunque le prestazioni.

Tra la MODIFICA 1 e la MODIFICA 6 sono state provate diverse funzioni, e tra tutte quella che ha portato alla più alta percentuale di P_s è stata la funzione logaritmica della MODIFICA 4 con +0.03% rispetto allo scenario senza rebalancing.

Nella MODIFICA 7 dunque è stata mantenuta la funzione logaritmica andando a variare il valore del parametro K.

Per quanto piccole siano le variazioni di prestazioni riportate, tra le alternative proposte quella che garantisce i risultati migliori è la variante con *16K*; essa infatti riporta la riduzione maggiore, pari all'84.37%, nelle commissioni dei pagamenti andati a buon fine. P_s arriva a registrare in questa variante un miglioramento massimo relativo dello 0.06% rispetto alla situazione senza rebalancing, P_{fb} riporta un decremento massimo relativo del 27.16%, mentre P_{fp} cresce al massimo del 6.78% rispetto allo scenario iniziale.

4.4 REBALANCING ATTIVO E PASSIVO

Per chiudere il cerchio, è stato condotto un ultimo ciclo di simulazioni con CLoTH adottando la miglior soluzione trovata del rebalancing attivo (MODIFICA 9) insieme all'alternativa più performante del rebalancing passivo (MODIFICA 7 con parametro *16K*).

A seguire vengono illustrati i risultati.

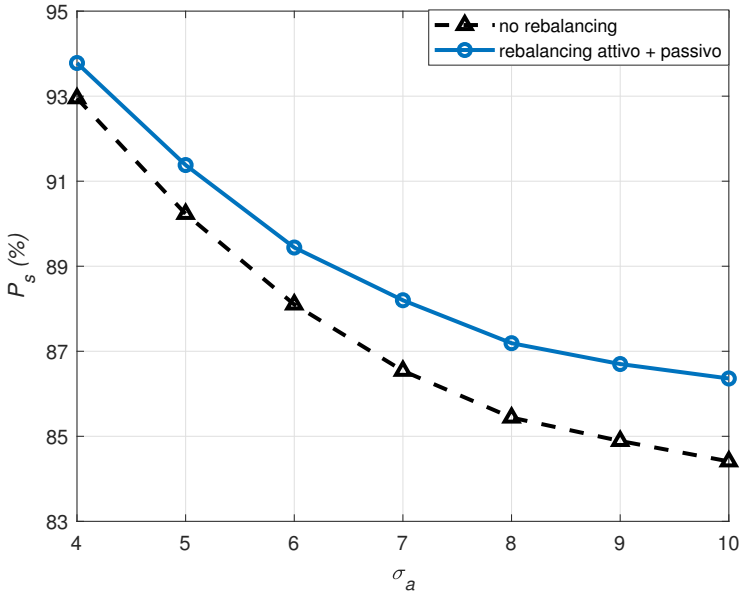

Figura 81: *Rebalancing Attivo + Passivo: P_s*

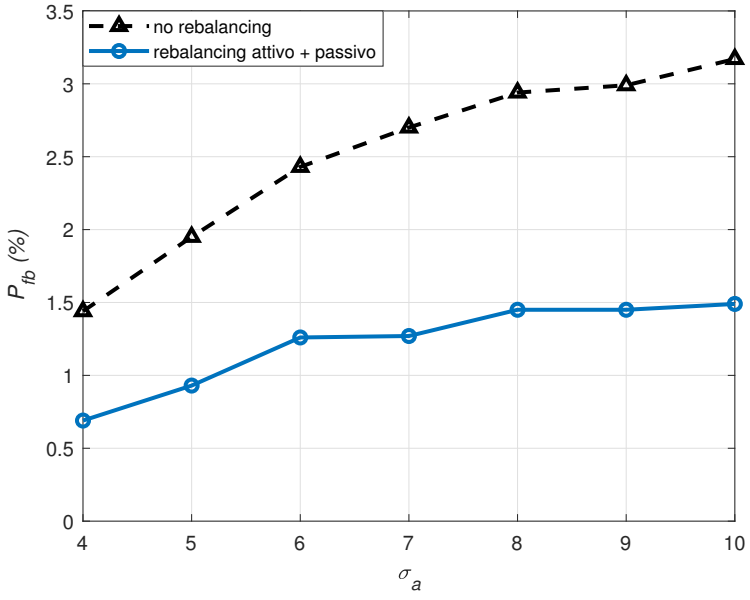

Figura 82: *Rebalancing Attivo + Passivo: P_{fb}*

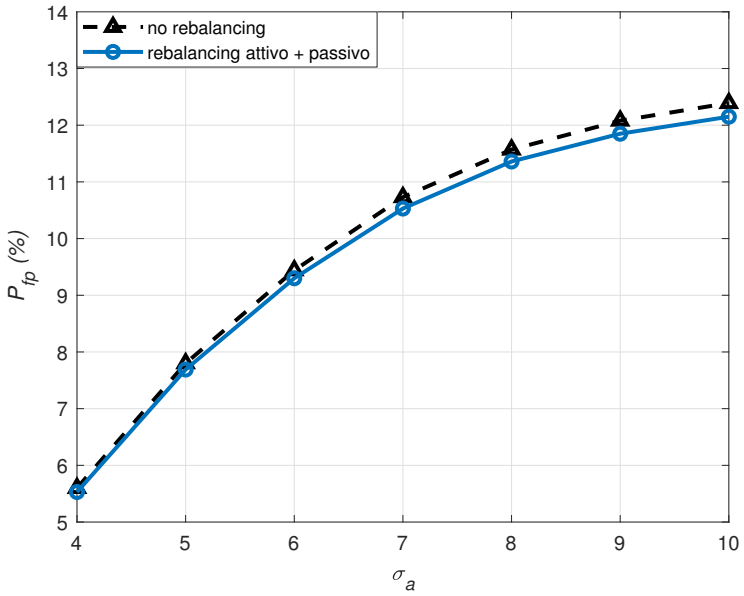

Figura 83: *Rebalancing Attivo + Passivo: P_{fp}*

σ_a	N° Tentativi	Successo (%)	No channel (%)	No route (%)	No balance (%)
4	30610	96.08	1.49	0.34	2.09
5	35942	95.15	2.13	0.40	2.32
6	34760	93.64	2.72	0.47	3.16
7	38267	94.19	2.81	0.48	2.51
8	38252	93.22	2.88	0.72	3.19
9	37738	93.45	2.81	0.54	3.19
10	41125	92.74	2.67	0.62	3.98

Tabella 15: *Rebalancing Attivo + Passivo: statistiche*

I risultati ottenuti si avvicinano, senza superarli, a quelli ricavati in corrispondenza della miglior soluzione di rebalancing attivo (A.25).

Questa prova suggerisce che la MODIFICA 9 relativa alla strategia di rebalancing attivo rappresenta l'implementazione che garantisce le più alte prestazioni nei pagamenti.

4.5 SCENARIO SERVICE-PROVIDER

Uno dei contesti in cui Lightning Network viene più largamente utilizzata è quello service-provider, in cui i pagamenti sono indirizzati dalla maggior parte dei nodi della rete sempre verso gli stessi pochi nodi della rete (*i service providers*) i quali vengono pagati come erogatori di qualche servizio.

Studiare l'efficacia delle strategie di rebalancing nel contesto service-provider è molto interessante, essendo questo scenario particolarmente soggetto al problema dello sbilanciamento dei canali. Esso è dovuto al fatto che i pagamenti nel contesto service-provider vengono indirizzati sempre verso gli stessi nodi, e in questo modo i canali tendono a sbilanciarsi più facilmente.

L'obiettivo del prossimo set di simulazioni è valutare le performance dei pagamenti nella rete in un contesto di questo tipo implementando la miglior soluzione di rebalancing trovata, ossia la MODIFICA 9 del rebalancing attivo; la stessa analisi viene condotta anche con la scelta migliore relativa al rebalancing passivo, cioè la MODIFICA 7 con parametro *16K*.

Per poter configurare al meglio lo scenario di riferimento, in CLoTH vengono definite tre classi di nodi:

- Payees: i service-providers, ossia i nodi che ricevono i pagamenti. Questi sono nodi direttamente collegati ad uno degli hub della rete, condizione favorevole affinchè i pagamenti possano giungere con più facilità. Inoltre, ciascun payee viene scelto in modo tale che, nel canale con l'hub, il balance dell'hub sia più elevato del balance del payee. Questo serve ad evitare che i pagamenti diretti ai payees e che attraversano l'hub possano fallire per insufficienza di balance. I payees sono in totale 162;
- Payers: i nodi che inviano i pagamenti. Anche questi nodi sono direttamente collegati ad un hub, in modo da velocizzare la loro comunicazione con i nodi payees. Inoltre, ciascun payer viene scelto in modo tale che, nel canale con l'hub, il balance del payer sia più elevato del balance dell'hub. Questo serve ad evitare che i pagamenti che partono dai payers e che attraversano l'hub possano fallire per insufficienza di balance. I payers sono in totale 1547;
- Nodi ibridi: i nodi che sia ricevono sia inviano pagamenti, e che quindi non appartengono nè alla categoria dei payees nè a quella dei payers. Essi sono in totale 2738.

Nelle Figure 84 - 85 - 86 si può apprezzare come nello scenario service-provider le prestazioni migliorino di molto nel momento in cui viene adottata la strategia di rebalancing attivo rispetto alla situazione di no rebalancing (A.26).

Adottando la miglior strategia di rebalancing attivo (MODIFICA 9) si riscontra un miglioramento delle performance progressivo passando tra un valore di σ_a e il successivo. Nella situazione più critica, con σ_a=10 e dunque un valore medio degli importi dei pagamenti più alto, P_s registra un incremento del 32% rispetto a quanto garantito nello scenario senza rebalancing (A.27).

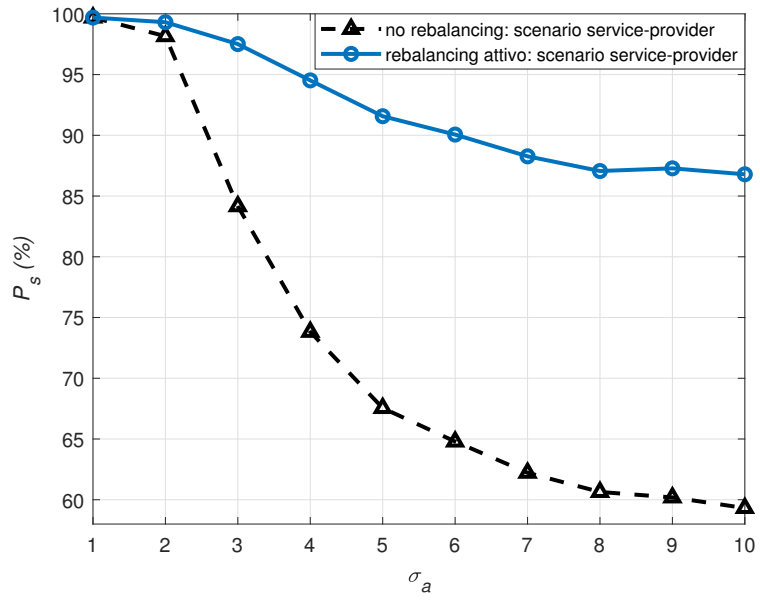

Figura 84: *Confronto tra no rebalancing e rebalancing attivo in scenario service-provider: P_s*

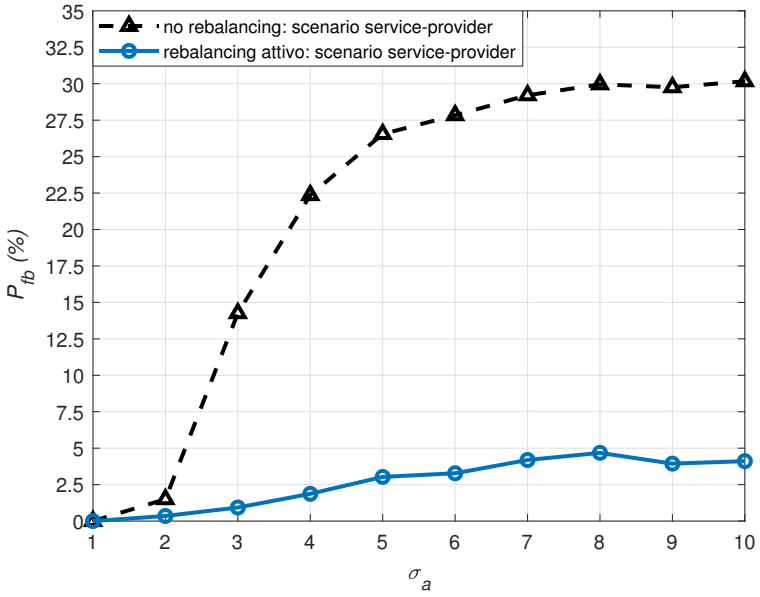

Figura 85: *Confronto tra no rebalancing e rebalancing attivo in scenario service-provider: P_{fb}*

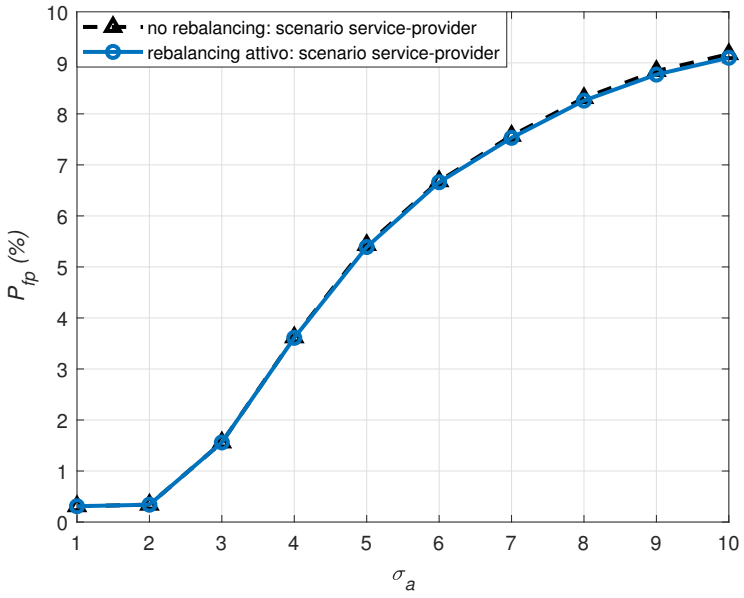

Figura 86: *Confronto tra no rebalancing e rebalancing attivo in scenario service-provider: P_{fp}*

Non adottando il rebalancing attivo, la causa maggiore di fallimento per i pagamenti è proprio il channel unbalancing; P_{fb} infatti per il valore più alto di σ_a raggiunge la soglia del 30%. Col rebalancing attivo invece P_{fb} si attesta su un valore di poco superiore al 4%.

Questi numeri conferiscono da soli una grande importanza allo studio condotto: la MODIFICA 9 del rebalancing attivo assicura una più che concreta protezione contro il problema dello sbilanciamento, e rappresenta il contributo più tangibile di questo lavoro.

La validità di questa strategia di rebalancing nel contesto service-provider si evince anche dal numero medio di tentativi per pagamento, che per σ_a=10 passa da 4.07 nello scenario di no rebalancing a 1.09 qualora si adotti il rebalancing attivo.

Le Figure 87 - 88 - 89 confermano infine quanto visto in precedenza riguardo la tecnica del rebalancing passivo, che se implementata in uno scenario service-provider assicura un miglioramento medio della probabilità di successo P_s di appena 1.32% e della probabilità di fallimento per unbalancing P_{fb} del 6.35%, contrastata da P_{fp} che per σ_a=10 arriva registrare un peggioramento del 22% (A.28).

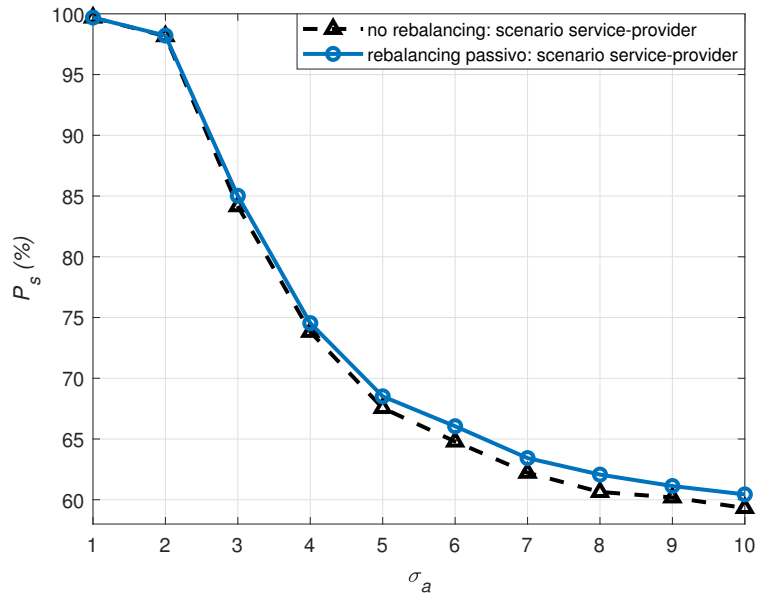

Figura 87: *Confronto tra no rebalancing e rebalancing passivo in scenario service-provider: P_s*

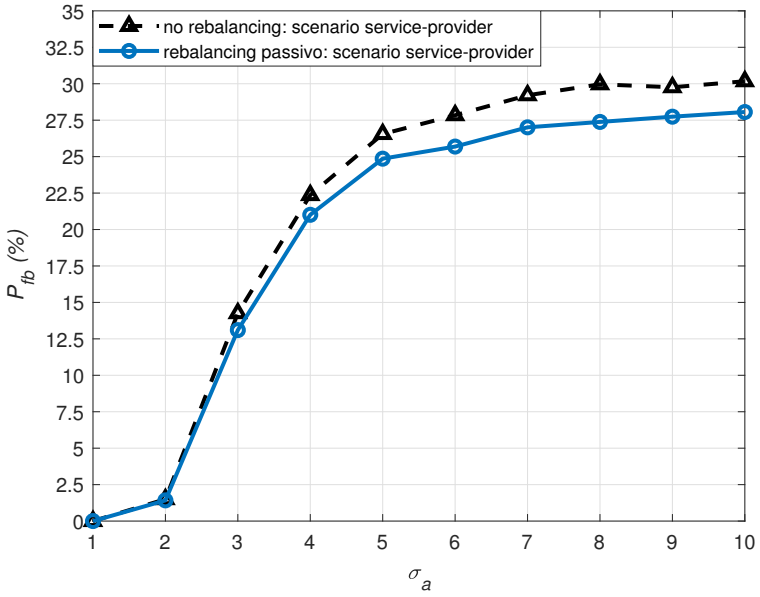

Figura 88: *Confronto tra no rebalancing e rebalancing passivo in scenario service-provider: P_{fb}*

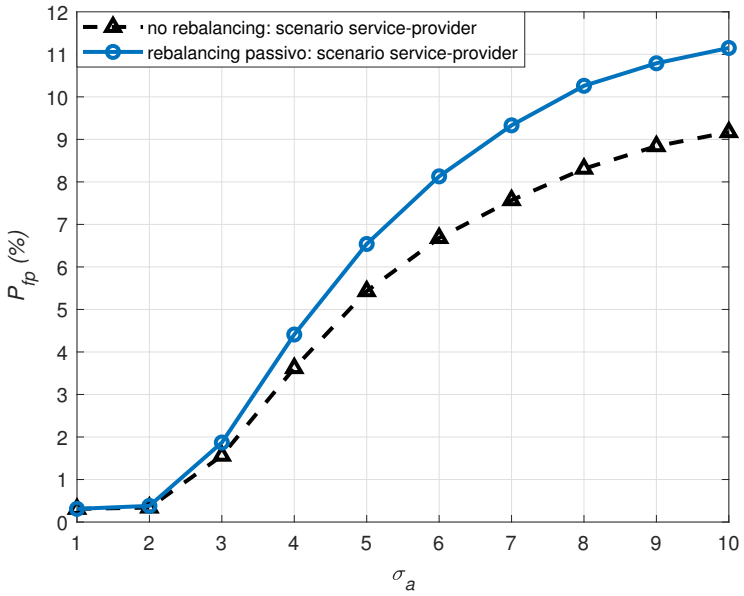

Figura 89: *Confronto tra no rebalancing e rebalancing passivo in scenario service-provider:* P_{fp}

Si può conludere che, anche in uno scenario service-provider, la scelta vincente per contrastare il problema dello sbilanciamento dei canali è rappresentata dal rebalancing attivo.

5 CONCLUSIONI E SVILUPPI FUTURI

Il contesto da cui nasce questo lavoro è quello di esplorare delle soluzioni che possano limitare il problema della scalabilità della blockchain. In tal senso le payment channel network rappresentano l'espediente più approfondito: esse consentono infatti di mettere in piedi reti di pagamento dove è possibile eseguire pagamenti off-blockchain. Lightning Network è il protocollo che corrisponde alla declinazione più nota e approfondita delle payment channel network; esso usa i contratti HTLC per eseguire pagamenti senza il coinvolgimento della blockchain. HTLC fa inoltre da deterrente per i disonesti dato che eventuali tentativi da parte di un nodo di ingannare la controparte vengono puniti con la requisizione di tutti i bitcoin in suo possesso.

Lightning Network si trova tuttavia nei suoi primi stadi di sviluppo e pertanto dimostra alcune criticità, come ad esempio lo sbilanciamento dei canali che causa il fallimento di alcuni pagamenti. Il rebalancing rappresenta uno dei metodi che possono essere adottati per contrastare questo problema.

L'obiettivo dello studio è stato dunque simulare diverse strategie di rebalancing utilizzando CLoTH, un simulatore in grado di riprodurre l'esecuzione di pagamenti in una rete di pagamento HTLC e di restituire misure di performance come la probabilità di successo dei pagamenti o il tempo medio impiegato per portarli a compimento.

Le due serie principali di simulazioni che sono state condotte afferiscono rispettivamente a differenti metodi di rebalancing attivo e passivo. Il rebalancing attivo si basa sull'eseguire un pagamento con lo scopo di ribilanciare i canali, mentre il rebalancing passivo consiste nel regolare la policy sulle commissioni da corrispondere ad un nodo nel momento in cui esso faccia da intermediario in un pagamento.

Dai risultati ottenuti emerge che la migliore implementazione trovata adotta il rebalancing attivo ed è in grado di ridurre di oltre la metà la probabilità di fallimento dei pagamenti per unbalancing.

L'eccezionalità di tali risultati, che rappresentano il più che tangibile contributo offerto da questo lavoro, trovano più voce se analizzati in un contesto service-provider, dove la probabilità di successo dei pagamenti registra un miglioramento addirittura del 32%.

Questa tesi rappresenta il naturale proseguimento del lavoro avviato in [10], ma ancora molte esplorazioni potrebbero essere condotte utilizzando il simulatore CLoTH. Qui lo studio si è concentrato ad analizzare modifiche al protocollo che implementassero strategie di

rebalancing: in futuro potranno essere studiate altre tipologie di modifiche al protocollo. Ad esempio si potrebbe seguire un approccio che divida i grandi pagamenti in pagamenti più piccoli, in modo da ridurre i fallimenti dovuti ai limiti di capacità dei canali. Inoltre sarebbe interessante simulare scenari di attacco a Lightning Network e provare a scoprire tecniche per garantire una valida difesa.

Infine lo stesso CLoTH si presta ad essere migliorato, ad esempio introducendo il concetto di blockchain e quindi simulando le interazioni tra quest'ultima e la rete di pagamento; inoltre sarebbe utile rendere il simulatore completamente multi-thread, al fine di distribuire il carico computazionale su più macchine.

Questi perfezionamenti migliorerebbero le performance di CLoTH consentendo di condurre nuove tipologie di simulazioni, tutte quante orientate a garantire un'evoluzione alle payment channel network e quindi un futuro sostenibile al sistema Bitcoin.

A RISULTATI COMPLETI DELLE SIMULAZIONI

A.1 NO REBALANCING

σ_a	P_s(%)	P_{fb}(%)	P_{fp}(%)	P_{ft}(%)	T (ms)	N_{att}	R (hops)
1	98.97	0.00	1.03	0.00	430.98	1.00	4.01
2	98.92	0.03	1.05	0.00	439.74	1.08	3.93
3	96.78	0.56	2.66	0.00	476.89	1.30	3.86
4	92.95	1.44	5.60	0.01	502.65	1.42	3.83
5	90.23	1.95	7.80	0.02	514.73	1.51	3.80
6	88.10	2.43	9.44	0.03	529.93	1.58	3.79
7	86.54	2.70	10.73	0.03	536.66	1.63	3.77
8	85.44	2.94	11.57	0.04	537.03	1.66	3.77
9	84.89	2.99	12.08	0.03	541.39	1.68	3.78
10	84.41	3.17	12.39	0.04	549.14	1.69	3.77

Tabella A.1: *No Rebalancing: risultati completi*

A.2 REBALANCING ATTIVO

σ_a	P_s(%)	P_{fb}(%)	P_{fp}(%)	P_{ft}(%)	T (ms)	N_{att}	R (hops)
1	98.97	0.00	1.03	0.00	430.98	1.00	4.01
2	98.92	0.03	1.05	0.00	438.26	1.04	3.97
3	96.77	0.56	2.66	0.00	458.37	1.13	3.98
4	92.95	1.44	5.60	0.01	477.00	1.24	3.92
5	90.21	1.97	7.80	0.01	494.36	1.32	3.95
6	88.11	2.44	9.43	0.02	547.73	1.51	3.99
7	86.54	2.70	10.73	0.03	505.91	1.39	3.87
8	85.47	2.91	11.59	0.03	504.49	1.38	3.90
9	84.87	3.01	12.09	0.02	559.94	1.58	3.93
10	84.40	3.17	12.39	0.04	554.97	1.57	3.94

Tabella A.2: *Rebalancing attivo: risultati completi setup iniziale*

σ_a	P_s(%)	P_{fb}(%)	P_{fp}(%)	P_{ft}(%)	T (ms)	N_{att}	R (hops)
1	98.97	0.00	1.03	0.00	430.98	1.00	4.01
2	98.92	0.03	1.05	0.00	437.92	1.00	4.08
3	96.78	0.56	2.66	0.00	444.34	1.02	4.09
4	92.95	1.44	5.60	0.01	450.82	1.05	4.08
5	90.25	1.95	7.80	0.01	458.12	1.08	4.07
6	88.14	2.41	9.43	0.02	468.01	1.12	4.07
7	86.53	2.73	10.73	0.02	469.88	1.13	4.07
8	85.47	2.93	11.57	0.03	472.27	1.14	4.06
9	84.90	2.99	12.09	0.02	473.09	1.15	4.06
10	84.44	3.15	12.38	0.03	476.51	1.16	4.06

Tabella A.3: *Rebalancing attivo: risultati completi modifica 1*

σ_a	P_s(%)	P_{fb}(%)	P_{fp}(%)	P_{ft}(%)	T (ms)	N_{att}	R (hops)
1	98.97	0.00	1.03	0.00	430.97	1.00	4.01
2	98.92	0.03	1.05	0.00	437.94	1.00	4.08
3	96.78	0.56	2.66	0.00	445.02	1.02	4.10
4	92.94	1.45	5.60	0.02	459.79	1.08	4.09
5	90.24	1.95	7.80	0.01	461.81	1.09	4.08
6	88.12	2.42	9.44	0.03	484.22	1.18	4.07
7	86.53	2.73	10.73	0.02	483.60	1.18	4.06
8	85.47	2.93	11.57	0.03	500.07	1.25	4.05
9	84.87	3.00	12.09	0.03	492.80	1.23	4.04
10	84.43	3.15	12.39	0.03	477.01	1.16	4.07

Tabella A.4: *Rebalancing attivo: risultati completi modifica 2*

σ_a	P_s(%)	P_{fb}(%)	P_{fp}(%)	P_{ft}(%)	T (ms)	N_{att}	R (hops)
1	98.97	0.00	1.03	0.00	430.97	1.00	4.01
2	98.92	0.03	1.05	0.00	443.23	1.03	4.06
3	96.77	0.56	2.66	0.00	454.80	1.13	3.93
4	92.93	1.45	5.60	0.00	485.27	1.26	3.92
5	90.23	1.96	7.80	0.01	488.85	1.30	3.88
6	88.09	2.44	9.45	0.03	502.16	1.36	3.88
7	86.53	2.70	10.74	0.03	496.28	1.33	3.87
8	85.46	2.93	11.58	0.03	509.90	1.38	3.92
9	84.87	3.01	12.10	0.03	502.17	1.38	3.85
10	84.41	3.16	12.39	0.04	512.21	1.40	3.86

Tabella A.5: *Rebalancing attivo: risultati completi modifica 3*

σ_a	P_s(%)	P_{fb}(%)	P_{fp}(%)	P_{ft}(%)	T (ms)	N_{att}	R (hops)
1	98.97	0.00	1.03	0.00	430.97	1.00	4.01
2	98.92	0.03	1.05	0.00	437.99	1.00	4.08
3	96.77	0.56	2.66	0.01	453.21	1.12	3.96
4	92.94	1.45	5.60	0.01	470.46	1.16	4.02
5	90.22	1.97	7.80	0.01	485.53	1.28	3.89
6	88.08	2.45	9.44	0.03	492.04	1.27	3.95
7	86.53	2.71	10.74	0.03	499.99	1.32	3.92
8	85.45	2.93	11.58	0.03	521.56	1.38	4.00
9	84.90	2.98	12.09	0.03	499.63	1.31	3.98
10	84.41	3.15	12.40	0.04	509.06	1.36	3.93

Tabella A.6: *Rebalancing attivo: risultati completi modifica 4*

σ_a	P_s(%)	P_{fb}(%)	P_{fp}(%)	P_{ft}(%)	T (ms)	N_{att}	R (hops)
1	98.97	0.00	1.03	0.00	431.08	1.00	4.01
2	98.92	0.03	1.05	0.00	437.95	1.00	4.08
3	96.77	0.57	2.66	0.00	452.09	1.09	4.01
4	92.95	1.44	5.60	0.01	473.54	1.23	3.91
5	90.21	1.97	7.80	0.02	484.32	1.30	3.90
6	88.09	2.45	9.44	0.02	500.95	1.36	3.91
7	86.55	2.70	10.73	0.03	499.26	1.36	3.87
8	85.46	2.93	11.58	0.03	501.42	1.37	3.88
9	84.89	2.99	12.10	0.02	516.44	1.39	4.00
10	84.41	3.16	12.39	0.03	532.20	1.45	3.96

Tabella A.7: *Rebalancing attivo: risultati completi modifica 5*

σ_a	P_s(%)	P_{fb}(%)	P_{fp}(%)	P_{ft}(%)	T (ms)	N_{att}	R (hops)
1	98.97	0.00	1.03	0.00	431.01	1.00	4.01
2	98.92	0.03	1.05	0.00	437.95	1.00	4.08
3	96.78	0.56	2.66	0.00	444.44	1.02	4.09
4	92.95	1.43	5.60	0.01	450.84	1.05	4.08
5	90.23	1.96	7.80	0.02	460.91	1.10	4.07
6	88.11	2.44	9.43	0.02	472.53	1.14	4.08
7	86.53	2.73	10.72	0.02	474.63	1.15	4.07
8	85.46	2.93	11.57	0.02	473.48	1.15	4.07
9	84.89	3.00	12.09	0.02	477.37	1.17	4.06
10	84.41	3.17	12.39	0.03	479.69	1.17	4.07

Tabella A.8: *Rebalancing attivo: risultati completi modifica 6*

σ_a	P_s(%)	P_{fb}(%)	P_{fp}(%)	P_{ft}(%)	T (ms)	N_{att}	R (hops)
1	98.97	0.00	1.03	0.00	430.97	1.00	4.01
2	98.92	0.03	1.05	0.00	437.92	1.00	4.08
3	96.78	0.56	2.66	0.00	446.61	1.03	4.10
4	92.94	1.45	5.60	0.01	458.24	1.07	4.09
5	90.21	1.97	7.81	0.01	459.23	1.08	4.08
6	88.11	2.43	9.44	0.03	472.90	1.13	4.08
7	86.51	2.74	10.73	0.02	472.06	1.13	4.08
8	85.44	2.94	11.59	0.03	471.42	1.14	4.07
9	84.88	3.01	12.09	0.02	474.73	1.15	4.06
10	84.41	3.16	12.40	0.03	475.77	1.15	4.07

Tabella A.9: *Rebalancing attivo: risultati completi modifica 7*

σ_a	P_s(%)	P_{fb}(%)	P_{fp}(%)	P_{ft}(%)	T (ms)	N_{att}	R (hops)
1	98.97	0.00	1.03	0.00	430.98	1.00	4.01
2	98.92	0.03	1.05	0.00	437.93	1.00	4.08
3	96.78	0.56	2.66	0.00	444.87	1.02	4.09
4	92.93	1.46	5.60	0.01	452.52	1.06	4.08
5	90.22	1.96	7.80	0.01	461.28	1.10	4.07
6	88.11	2.43	9.44	0.02	472.73	1.14	4.07
7	86.51	2.74	10.73	0.02	473.15	1.15	4.06
8	85.45	2.94	11.58	0.03	476.69	1.17	4.06
9	84.89	2.99	12.09	0.02	478.76	1.18	4.05
10	84.41	3.17	12.40	0.03	480.87	1.18	4.06

Tabella A.10: *Rebalancing attivo: risultati completi modifica 8 (soglia 40%)*

σ_a	P_s(%)	P_{fb}(%)	P_{fp}(%)	P_{ft}(%)	T (ms)	N_{att}	R (hops)
1	98.97	0.00	1.03	0.00	431.00	1.00	4.01
2	98.92	0.03	1.05	0.00	441.06	1.01	4.09
3	96.78	0.56	2.66	0.00	444.06	1.02	4.09
4	92.96	1.44	5.59	0.01	450.16	1.05	4.08
5	90.23	1.96	7.79	0.00	459.68	1.09	4.08
6	88.12	2.43	9.42	0.03	469.25	1.12	4.08
7	86.51	2.74	10.73	0.02	469.88	1.13	4.07
8	85.46	2.95	11.58	0.02	472.57	1.14	4.07
9	84.89	3.00	12.08	0.03	474.89	1.16	4.06
10	84.44	3.15	12.38	0.03	477.13	1.16	4.07

Tabella A.11: *Rebalancing attivo: risultati completi modifica 8 (soglia 60%)*

σ_a	P_s(%)	P_{fb}(%)	P_{fp}(%)	P_{ft}(%)	T (ms)	N_{att}	R (hops)
1	98.97	0.00	1.03	0.00	430.97	1.00	4.01
2	98.94	0.01	1.05	0.00	437.72	1.00	4.08
3	97.13	0.24	2.64	0.00	439.60	1.00	4.09
4	93.77	0.70	5.53	0.00	439.19	1.00	4.08
5	91.38	0.94	7.68	0.00	439.71	1.01	4.08
6	89.43	1.28	9.30	0.00	440.36	1.01	4.08
7	88.19	1.28	10.53	0.00	440.54	1.01	4.07
8	87.19	1.45	11.36	0.00	439.47	1.01	4.07
9	86.71	1.44	11.85	0.00	439.23	1.01	4.06
10	86.36	1.49	12.14	0.00	440.02	1.01	4.06

Tabella A.12: *Rebalancing attivo: risultati completi modifica 9*

σ_a	no rebalancing (ms)	rebalancing attivo (ms)
1	430.98	430.98
2	439.74	471.66
3	476.89	533.60
4	502.65	560.85
5	514.73	588.04
6	529.93	596.27
7	536.66	596.57
8	537.03	582.67
9	541.39	608.00
10	549.14	605.16

Tabella A.13: *Tempo medio di esecuzione dei pagamenti: confronto tra no rebalancing e rebalancing attivo*

A.3 REBALANCING PASSIVO

σ_a	P_s(%)	P_{fb}(%)	P_{fp}(%)	P_{ft}(%)	T (ms)	N_{att}	R (hops)
1	98.97	0.00	1.03	0.00	338.94	1.00	3.09
2	98.91	0.03	1.06	0.00	340.52	1.00	3.10
3	96.74	0.46	2.80	0.00	345.05	1.00	3.14
4	92.92	1.13	5.95	0.00	350.53	1.01	3.18
5	90.15	1.47	8.38	0.00	353.43	1.02	3.20
6	88.00	1.80	10.20	0.00	356.77	1.03	3.21
7	86.42	2.07	11.50	0.00	358.02	1.03	3.22
8	85.32	2.22	12.45	0.00	359.66	1.03	3.23
9	84.78	2.21	13.01	0.00	359.67	1.03	3.22
10	84.31	2.35	13.34	0.00	361.31	1.03	3.23

Tabella A.14: *Rebalancing passivo: risultati completi setup iniziale*

σ_a	P_s(%)	P_{fb}(%)	P_{fp}(%)	P_{ft}(%)	T (ms)	N_{att}	R (hops)
1	98.97	0.00	1.03	0.00	334.43	1.00	3.05
2	98.92	0.03	1.06	0.00	334.99	1.00	3.05
3	96.74	0.45	2.80	0.00	334.85	1.01	3.05
4	92.94	1.14	5.92	0.00	336.49	1.01	3.05
5	90.19	1.48	8.33	0.00	337.89	1.02	3.05
6	88.07	1.78	10.15	0.00	340.49	1.03	3.05
7	86.46	2.08	11.46	0.00	340.87	1.03	3.05
8	85.40	2.22	12.38	0.00	340.93	1.03	3.05
9	84.87	2.19	12.93	0.00	340.92	1.03	3.05
10	84.37	2.38	13.24	0.00	342.55	1.04	3.05

Tabella A.15: *Rebalancing passivo: risultati completi modifica 1*

σ_a	P_s(%)	P_{fb}(%)	P_{fp}(%)	P_{ft}(%)	T (ms)	N_{att}	R (hops)
1	98.97	0.00	1.03	0.00	337.48	1.00	3.07
2	98.92	0.03	1.05	0.00	339.49	1.00	3.09
3	96.77	0.45	2.78	0.00	355.60	1.06	3.17
4	92.97	1.14	5.88	0.01	384.18	1.18	3.24
5	90.25	1.47	8.27	0.01	397.00	1.24	3.29
6	88.14	1.76	10.08	0.02	414.38	1.31	3.31
7	86.53	2.07	11.37	0.03	426.69	1.39	3.33
8	85.47	2.20	12.29	0.04	432.70	1.43	3.34
9	84.89	2.19	12.86	0.07	437.06	1.45	3.34
10	84.42	2.36	13.18	0.04	445.50	1.49	3.25

Tabella A.16: *Rebalancing passivo: risultati completi modifica 2*

σ_a	P_s(%)	P_{fb}(%)	P_{fp}(%)	P_{ft}(%)	T (ms)	N_{att}	R (hops)
1	98.97	0.00	1.03	0.00	339.81	1.00	3.09
2	98.91	0.03	1.06	0.00	340.39	1.00	3.09
3	96.75	0.44	2.81	0.00	342.87	1.01	3.12
4	92.94	1.11	5.95	0.00	347.91	1.01	3.15
5	90.16	1.47	8.36	0.00	351.60	1.02	3.17
6	88.05	1.77	10.18	0.00	353.89	1.03	3.18
7	86.46	2.04	11.49	0.00	356.22	1.03	3.19
8	85.36	2.19	12.44	0.00	355.95	1.03	3.19
9	84.76	2.23	13.01	0.00	357.70	1.03	3.19
10	84.32	2.37	13.30	0.00	358.42	1.04	3.20

Tabella A.17: *Rebalancing passivo: risultati completi modifica 3*

σ_a	P_s(%)	P_{fb}(%)	P_{fp}(%)	P_{ft}(%)	T (ms)	N_{att}	R (hops)
1	98.97	0.00	1.03	0.00	335.97	1.00	3.06
2	98.92	0.03	1.05	0.00	337.86	1.00	3.07
3	96.77	0.45	2.78	0.00	350.45	1.05	3.13
4	92.98	1.14	5.88	0.00	372.49	1.15	3.19
5	90.25	1.47	8.27	0.01	384.92	1.21	3.22
6	88.15	1.77	10.06	0.02	398.18	1.25	3.25
7	86.54	2.07	11.37	0.03	413.21	1.35	3.26
8	85.49	2.20	12.28	0.03	414.93	1.36	3.27
9	84.91	2.19	12.86	0.04	427.24	1.44	3.28
10	84.42	2.37	13.17	0.04	422.32	1.39	3.28

Tabella A.18: *Rebalancing passivo: risultati completi modifica 4*

σ_a	P_s(%)	P_{fb}(%)	P_{fp}(%)	P_{ft}(%)	T (ms)	N_{att}	R (hops)
1	98.97	0.00	1.03	0.00	336.73	1.00	3.06
2	98.92	0.03	1.06	0.00	337.46	1.00	3.07
3	96.75	0.45	2.81	0.00	340.20	1.01	3.10
4	92.93	1.14	5.93	0.00	344.29	1.01	3.12
5	90.18	1.47	8.35	0.00	347.73	1.02	3.14
6	88.02	1.79	10.19	0.00	350.47	1.03	3.15
7	86.43	2.07	11.49	0.00	351.68	1.03	3.16
8	85.33	2.23	12.44	0.00	352.80	1.03	3.16
9	84.83	2.19	12.97	0.00	353.79	1.03	3.16
10	84.31	2.38	13.30	0.00	354.71	1.03	3.17

Tabella A.19: *Rebalancing passivo: risultati completi modifica 5*

σ_a	P_s(%)	P_{fb}(%)	P_{fp}(%)	P_{ft}(%)	T (ms)	N_{att}	R (hops)
1	98.97	0.00	1.03	0.00	334.90	1.00	3.05
2	98.92	0.03	1.05	0.00	335.23	1.00	3.05
3	96.75	0.44	2.80	0.00	335.24	1.01	3.05
4	92.94	1.14	5.92	0.00	337.48	1.02	3.05
5	90.17	1.49	8.33	0.00	339.71	1.02	3.06
6	88.09	1.77	10.13	0.00	340.91	1.03	3.06
7	86.49	2.05	11.45	0.00	341.93	1.03	3.06
8	85.39	2.21	12.40	0.00	343.03	1.03	3.06
9	84.87	2.18	12.94	0.00	343.49	1.04	3.06
10	84.35	2.38	13.26	0.00	343.63	1.04	3.06

Tabella A.20: *Rebalancing passivo: risultati completi modifica 6*

σ_a	P_s(%)	P_{fb}(%)	P_{fp}(%)	P_{ft}(%)	T (ms)	N_{att}	R (hops)
1	98.97	0.00	1.03	0.00	337.48	1.00	3.07
2	98.92	0.03	1.05	0.00	339.51	1.00	3.09
3	96.77	0.44	2.78	0.00	355.94	1.06	3.17
4	92.97	1.14	5.87	0.01	384.56	1.19	3.24
5	90.25	1.47	8.27	0.01	398.48	1.25	3.29
6	88.14	1.77	10.08	0.02	414.49	1.32	3.31
7	86.54	2.07	11.37	0.02	430.56	1.42	3.33
8	85.48	2.19	12.29	0.04	434.96	1.44	3.34
9	84.93	2.18	12.85	0.05	438.66	1.48	3.34
10	84.43	2.36	13.17	0.04	439.17	1.46	3.35

Tabella A.21: *Rebalancing passivo: risultati completi modifica 7 con K/2*

σ_a	P_s(%)	P_{fb}(%)	P_{fp}(%)	P_{ft}(%)	T (ms)	N_{att}	R (hops)
1	98.97	0.00	1.03	0.00	336.04	1.00	3.06
2	98.92	0.03	1.05	0.00	336.28	1.00	3.06
3	96.76	0.45	2.79	0.00	347.32	1.04	3.11
4	92.97	1.14	5.88	0.01	367.14	1.13	3.16
5	90.26	1.47	8.27	0.00	376.17	1.17	3.19
6	88.14	1.78	10.07	0.01	386.75	1.22	3.21
7	86.54	2.07	11.37	0.02	399.99	1.29	3.22
8	85.47	2.20	12.30	0.03	403.30	1.30	3.23
9	84.95	2.19	12.84	0.02	406.06	1.33	3.23
10	84.41	2.38	13.18	0.02	410.81	1.35	3.24

Tabella A.22: *Rebalancing passivo: risultati completi modifica 7 con 2K*

σ_a	P_s(%)	P_{fb}(%)	P_{fp}(%)	P_{ft}(%)	T (ms)	N_{att}	R (hops)
1	98.97	0.00	1.03	0.00	336.07	1.00	3.06
2	98.92	0.03	1.05	0.00	335.32	1.00	3.06
3	96.76	0.44	2.79	0.00	339.74	1.01	3.07
4	92.98	1.14	5.88	0.00	347.78	1.05	3.09
5	90.24	1.47	8.27	0.00	353.01	1.07	3.11
6	88.15	1.77	10.08	0.01	360.02	1.10	3.12
7	86.51	2.09	11.39	0.00	365.53	1.13	3.12
8	85.46	2.22	12.31	0.01	367.75	1.14	3.13
9	84.94	2.19	12.85	0.01	369.49	1.16	3.13
10	84.45	2.37	13.18	0.01	371.42	1.16	3.13

Tabella A.23: *Rebalancing passivo: risultati completi modifica 7 con 16K*

σ_a	P_s(%)	P_{fb}(%)	P_{fp}(%)	P_{ft}(%)	T (ms)	N_{att}	R (hops)
1	98.97	0.00	1.03	0.00	335.83	1.00	3.05
2	98.92	0.03	1.05	0.00	335.39	1.00	3.05
3	96.76	0.44	2.79	0.00	339.22	1.01	3.07
4	92.98	1.13	5.88	0.00	345.17	1.04	3.08
5	90.23	1.48	8.29	0.00	349.02	1.06	3.09
6	88.14	1.77	10.08	0.00	354.64	1.08	3.10
7	86.51	2.08	11.40	0.00	357.28	1.09	3.10
8	85.48	2.21	12.31	0.00	360.17	1.11	3.11
9	84.96	2.18	12.85	0.00	361.22	1.12	3.11
10	84.42	2.38	13.19	0.00	363.60	1.12	3.11

Tabella A.24: *Rebalancing passivo: risultati completi modifica 7 con 32K*

A.4 REBALANCING ATTIVO E PASSIVO

σ_a	P_s(%)	P_{fb}(%)	P_{fp}(%)	P_{ft}(%)	T (ms)	N_{att}	R (hops)
1	98.97	0.00	1.03	0.00	430.94	1.00	4.01
2	98.94	0.01	1.05	0.00	437.69	1.00	4.08
3	97.13	0.23	2.64	0.00	439.54	1.01	4.08
4	93.78	0.69	5.53	0.00	439.06	1.01	4.07
5	91.38	0.93	7.69	0.00	438.95	1.01	4.07
6	89.44	1.26	9.30	0.00	439.25	1.01	4.07
7	88.20	1.27	10.53	0.00	438.67	1.01	4.06
8	87.19	1.45	11.36	0.00	438.53	1.01	4.06
9	86.70	1.45	11.85	0.00	437.97	1.01	4.05
10	86.36	1.49	12.15	0.00	439.05	1.02	4.06

Tabella A.25: *Rebalancing attivo + passivo: risultati completi*

A.5 SCENARIO SERVICE-PROVIDER

σ_a	P_s(%)	P_{fb}(%)	P_{fp}(%)	P_{ft}(%)	T (ms)	N_{att}	R (hops)
1	99.69	0.00	0.31	0.00	349.36	1.00	3.29
2	98.16	1.50	0.34	0.00	367.82	1.10	3.26
3	84.16	14.26	1.56	0.02	487.21	1.60	3.38
4	73.82	22.36	3.62	0.20	676.95	2.41	3.46
5	67.55	26.55	5.43	0.47	798.48	2.95	3.46
6	64.79	27.84	6.68	0.69	888.63	3.32	3.47
7	62.22	29.21	7.57	1.00	963.42	3.66	3.49
8	60.64	29.96	8.31	1.09	963.11	3.63	3.48
9	60.19	29.76	8.84	1.21	1046.15	4.03	3.49
10	59.32	30.17	9.17	1.34	1055.49	4.07	3.48

Tabella A.26: *Scenario service-provider con no rebalancing: risultati completi*

σ_a	P_s(%)	P_{fb}(%)	P_{fp}(%)	P_{ft}(%)	T (ms)	N_{att}	R (hops)
1	99.69	0.00	0.31	0.00	349.36	1.00	3.29
2	99.31	0.35	0.34	0.00	351.72	1.00	3.32
3	97.51	0.93	1.56	0.00	353.16	1.01	3.32
4	94.52	1.87	3.61	0.00	359.74	1.04	3.34
5	91.57	3.03	5.39	0.00	363.66	1.07	3.35
6	90.06	3.28	6.66	0.00	362.89	1.08	3.34
7	88.27	4.19	7.53	0.00	365.81	1.08	3.35
8	87.06	4.68	8.26	0.00	364.43	1.09	3.36
9	87.28	3.94	8.77	0.00	380.36	1.14	3.38
10	86.79	4.11	9.10	0.00	364.13	1.09	3.35

Tabella A.27: *Scenario service-provider con rebalancing attivo: risultati completi*

σ_a	P_s(%)	P_{fb}(%)	P_{fp}(%)	P_{ft}(%)	T (ms)	N_{att}	R (hops)
1	99.69	0.00	0.31	0.00	286.94	1.00	2.64
2	98.21	1.42	0.38	0.00	288.82	1.02	2.64
3	85.02	13.10	1.87	0.00	336.62	1.23	2.70
4	74.54	21.01	4.41	0.03	389.48	1.50	2.75
5	68.51	24.86	6.54	0.09	441.55	1.74	2.79
6	66.05	25.69	8.13	0.13	472.89	1.88	2.81
7	63.43	27.01	9.33	0.23	503.13	2.03	2.82
8	62.07	27.38	10.26	0.30	528.26	2.16	2.82
9	61.13	27.74	10.79	0.34	521.14	2.12	2.83
10	60.44	28.06	11.15	0.35	539.35	2.20	2.85

Tabella A.28: *Scenario service-provider con rebalancing passivo: risultati completi*

BIBLIOGRAFIA

[1] Satoshi Nakamoto. *Bitcoin: A peer-to-peer electronic cash system*. 2008.

[2] Yonatan Sompolinsky e Aviv Zohar. *Accelerating bitcoin's transaction processing.Fast Money Grows on Trees, Not Chains*. 2013.

[3] Joseph Poon e Thaddeus Dryja. *The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments*. 2016.

[4] Christian Decker e Roger Wattenhofer. «A fast and scalable payment network with bitcoin duplex micropayment channels». In: *Symposium on Self-Stabilizing Systems. Springer* (2015), pp. 3–18.

[5] Andrew Miller et al. «Sprites: Payment Channels that Go Faster than Lightning». In: *CoRR abs/1702.05812* (2017).

[6] *Raiden Network*. (visitato il 31/07/2018). URL: https://raiden.network/.

[7] Christian Decker Conrad Burchert e Roger Wattenhofer. «Scalable funding of Bitcoin micropayment channel networks». In: *Royal Society open science 5.8* (2018), p. 180089.

[8] Rami Khalil e Arthur Gervais, cur. *Revive: Rebalancing off-blockchain payment networks*. Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, pp. 439-453. ACM. 2017.

[9] Pavel Prihodko et al. *Flare: An Approach to Routing in Lightning Network*. 2016.

[10] Marco Conoscenti. *Capabilities and Limitations of Payment Channel Networks for Blockchain Scalability*. Tesi di Dottorato. Politecnico di Torino, 2019.

[11] Joseph Poon e Thaddeus Dryja. *The Bitcoin Lightning Network: ScalableOff-Chain Instant Payments*. 2016.

[12] *Lightning Network Search and Analysis Engine*. (visitato il 30/11/2019). URL: https://1ml.com/.

[13] Andreas M. Antonopoulos. *Mastering Bitcoin: programming the open blockchain*. 2ª ed. O'Reilly Media, Inc., 2017.

[14] Andreas M. Antonopoulos. *Mastering Bitcoin: programming the open blockchain. Traduzione Italiana*. Independently published, 2019.

[15] David Chaum. «Blind signatures for untraceable payments». In: *Advances in cryptology. Springer* (1983), pp. 199–203.

[16] *Maximum transaction rate*. (visitato il 31/07/2018). URL: https://en.bitcoin.it/wiki/Maximum_transaction_rate.

[17] Manny Trillo. *Stress Test Prepares VisaNet for the Most Wonderful Time of the Year*. (visitato il 25/04/2019). URL: https://www.visa.com/blogarchives/us/2013/10/10/stress-test-prepares-visanet-for-the-most-wonderful-time-ofthe-year/index.html.

[18] Kyle Croman et al. «On scaling decentralized blockchains». In: *International Conference on Financial Cryptography and Data Security. Springer* (2016), pp. 106–125.

[19] Arthur Gervais et al. «On the security and performance of proof of work blockchains». In: *IProceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. ACM* (2016), pp. 3–16.

[20] Ittay Eyal et al. «Bitcoin-ng: A scalable blockchain protocol». In: *13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16)* (2016), pp. 45–59.

[21] Yossi Gilad et al. «Algorand: Scaling byzantine agreements for cryptocurrencies». In: *Proceedings of the 26th Symposium on Operating Systems Principles. ACM.* (2017), pp. 51–68.

[22] Aggelos Kiayias et al. «Ouroboros: A provably secure proof-of-stake blockchain protocol». In: *Annual International Cryptology Conference. Springer.* (2017), pp. 357–388.