



Università degli Studi di Roma, Tor Vergata

FACOLTÀ DI SCIENZE MM.FF.NN
Corso di Laurea Magistrale in Informatica

TESI DI LAUREA MAGISTRALE

**Grafi Aleatori Dinamici e Reti P2P Non
Strutturate, Analisi di Due Modelli Ispirati
alla Rete Bitcoin**

Candidato:
Antonio Cruciani
Matricola 0258412

Relatore:
Prof . Francesco Pasquale

Anno Accademico 2018–2019

If you don't believe it or don't get it,
I don't have the time to try to
convince you, sorry.

Satoshi Nakamoto

Contents

1	Introduzione	3
2	Bitcoin	5
2.1	La Rete P2P Bitcoin	6
3	Analisi empiriche sulla rete Bitcoin	12
3.1	Network Analysis Locale	12
3.2	Network Analysis Globale	16
4	Modelli teorici	18
4.1	Edge Dynamic Graph	18
4.2	Vertex Dynamic Graph	19
4.3	Preliminari e notazioni	20
4.4	Analisi Edge Dynamic Graph	22
4.5	Simulazioni per il modello Edge Dynamic	27
4.6	Simulazioni per il modello Vertex Dynamic	34
5	Conclusioni	46
5.1	Sviluppi futuri	47
A	Appendice	48
A.1	Appendice A	48
A.1.1	Catene Di Markov	48
A.1.2	Classificazione degli stati	48
A.1.3	Norme l_1 e l_2	49
A.1.4	ϵ -expander	49
A.2	Appendice B	50
A.2.1	Criterio di convergenza del primo modello	50
A.3	Appendice C	52
A.3.1	Immagini Modello Edge Dynamic	52

1 Introduzione

Bitcoin [1] è una moneta virtuale ideata nel 2008 da uno sviluppatore anonimo conosciuto con lo pseudonimo di Satoshi Nakamoto. Viene definita moneta virtuale in quanto è totalmente dematerializzata e a differenza delle comuni monete a corso legale, non è rilasciata e riconosciuta da alcuna autorità governativa o finanziaria. Con il nome bitcoin si può far riferimento all'intero sistema o all'unità di valuta. In generale si usa la convenzione di riferirsi al sistema con il termine "Bitcoin" e alle quantità monetarie con "bitcoin". In questi ultimi anni si è parlato molto di Bitcoin, infatti in poco tempo si è passati da un sistema utilizzato da pochi utenti per scopi "ludici" o illegali, come il traffico di armi o di droghe nel *Deep Web*, ad un sistema ampiamente utilizzato per effettuare semplici acquisti in negozi online come Etsy, comprare biglietti aerei, prenotare hotel tramite CheapAir e perfino pagare rette universitarie. Infatti il numero di Università che permettono agli studenti di pagare utilizzando Bitcoin è in continuo aumento, alcune tra le tante: ESMT di Berlino, l'Università di Nicosia e il King's College.

La causa di questo cambiamento è dovuta in gran parte all'incremento della capitalizzazione di mercato di bitcoin ottenuta nel 2017, con un aumento di valore di circa 770 miliardi di Dollari, mettendo il sistema al centro dell'attenzione mediatica e finanziaria mondiale, scatenando un boom nell'utilizzo delle criptovalute.

Oltre al suo elevato "valore monetario" l'interesse delle persone verso Bitcoin è dovuto alla sua affidabilità, trasparenza e sicurezza.

Gran parte della fiducia in Bitcoin deriva dal fatto che non richiede alcuna fiducia. Bitcoin è completamente *open source* e decentralizzato. Ciò significa che chiunque ha accesso all'intero codice sorgente in qualsiasi momento. Qualsiasi sviluppatore nel mondo può quindi verificare esattamente come funziona Bitcoin. Tutte le transazioni e i bitcoin emessi possono essere consultati in modo trasparente in tempo reale da chiunque. Tutti i pagamenti possono essere effettuati senza affidamento a terzi e l'intero sistema è protetto da algoritmi crittografici fortemente sottoposti a *peer review* come quelli utilizzati per l'*online banking*. Nessuna organizzazione o individuo può controllare Bitcoin e il sistema rimane sicuro anche se non tutti i suoi utenti possono essere considerati affidabili. La comunicazione avviene tramite una rete *peer-to-peer*, dove non esiste un server centrale, bensì tutti gli utenti della rete contribuiscono equamente al suo corretto funzionamento.

Dato l'enorme successo ricevuto, da diversi anni Bitcoin è arrivato sotto i riflettori della comunità scientifica. Si sta cercando infatti di formalizzare il sistema e di analizzare le sue proprietà al fine di avere una comprensione profonda del sistema e dei suoi limiti. Un fattore che rallenta fortemente i progressi in questa direzione è il forte orientamento alla *security* del sistema, il quale fa sì che molte informazioni riguardanti la rete non possano essere ottenute. Infatti ad esempio è possibile conoscere il numero di utenti attivi nella rete e la nazione da dove si connettono ma non è possibile conoscere le connessioni che intercorrono fra loro. Sarebbe molto interessante poter conoscere la struttura reale della rete P2P Bitcoin per poterne osservare l'evoluzione nel tempo, la struttura, nodi influenti ed altre proprietà interess-

anti che potrebbero essere studiate modellando gli utenti come nodi di un grafo e le connessioni come archi che li connettono.

Conoscere la topologia della rete Bitcoin è un “*hot topic*” che interessa diverse aree di ricerca. Una molto attiva è quella della sicurezza dove sono state proposte diverse tecniche per cercare di scoprire e analizzare la topologia della rete reale, allo scopo di studiarne la robustezza agli attacchi informatici [2],[3], [4]. Infatti la conoscenza della sua struttura potrebbe esporla ad attacchi mirati ad isolare gli agenti o addirittura partizionare la rete. Nell’articolo più recente [4] viene proposta una tecnica di inferenza della rete Bitcoin basata sull’utilizzo di transazioni orfane. I risultati della suddetta tecnica risultano essere molto interessanti. Purtroppo però le analisi svolte da Delgado et al.[4] sono state effettuate su una rete Bitcoin di test e non su quella reale, poiché tale tecnica potrebbe rallentare le transazioni degli utenti e quindi danneggiare la rete Bitcoin stessa.

In un articolo recente [5], viene proposto un modello di grafo aleatorio ispirato dal processo di generazione della rete P2P Bitcoin: un algoritmo distribuito prende in input un grafo Δ -regolare, due interi d , c e costruisce, in $\mathcal{O}(\log n)$ rounds, un grafo più sparso dove ogni nodo ha grado compreso tra d e $c \cdot d$ che con alta probabilità risulta essere un ϵ -expander A.3. Questo risultato è molto interessante, in quanto fornisce delle indicazioni sulla “qualità” della topologia della Rete P2P Bitcoin. Nella sezione “Lavori Futuri” dell’ articolo, gli autori, allo scopo di rendere il modello più realistico possibile, esprimono un forte interesse nello studiare il comportamento di tale algoritmo in uno scenario dinamico, dove i nodi e gli archi entrano o escono nella rete.

In questa tesi affrontiamo lo studio della dinamica della rete P2P da diversi punti di vista.

Nel Capitolo 2 descriveremo brevemente il meccanismo di generazione della rete P2P di Bitcoin

Nel Capitolo 3 verrà illustrata un’analisi dei nodi della Rete Bitcoin effettuata installando un *full node* su una macchina messa a disposizione dall’ Università ed utilizzando i dati forniti da Bitnodes [6], un software che permette di raccogliere “fotografie” della rete ogni cinque minuti.

Nel Capitolo 4 verranno proposte due versioni dinamiche del modello introdotto in [5]. Nella prima, si assumerà di avere archi che entrano ed escono dalla rete e nella seconda, ispirati da [7], lo si assumerà per i nodi della rete. Si dimostrerà che la Catena di Markov associata al primo modello non è reversibile e, mediante i risultati ottenuti dalle simulazioni si potrà osservare come tale processo abbia un *drift* verso un insieme di stati che hanno proprietà di espansione molto vicine a quelle dei grafi random d -regolari. Per il secondo modello verrà proposta un’analisi della misura stazionaria formalizzando il processo stocastico come una Coda Markoviana. Infine, i risultati ottenuti dalle simulazioni mostreranno che i grafi dinamici random, per valori appropriati di d e c , tenderanno ad avere un diametro logaritmico.

2 Bitcoin

In questo capitolo verrà descritto il funzionamento del sistema Bitcoin [1], andando ad enfatizzare gli aspetti legati all'architettura della rete *Peer To Peer* [8].

Il sistema Bitcoin, a differenza dei sistemi bancari tradizionali, è basato su un *trust* decentralizzato. Anziché avere un'unità centrale di fiducia, nel sistema Bitcoin si ottiene come una proprietà derivata dall'interazione dei partecipanti al sistema.

Infatti il sistema finanziario mondiale seppur funzioni discretamente bene, soffre di mancanze dovute al modello basato sulla fiducia. Ovvero fino a quando si utilizza una moneta fisica non si presentano problemi riguardanti le transazioni (scambi di denaro, acquisti) in quanto il passaggio di denaro viene eseguito a "mano". Passando però in uno scenario virtuale, il sistema odierno necessita di unità centrali di fiducia che gestiscano e certifichino i trasferimenti di soldi e gli acquisti da un individuo all'altro. Questa centralizzazione della fiducia rende l'intero sistema fortemente dipendente da queste unità centrali. Un ulteriore problema del sistema odierno, in uno scenario online, è che le transazioni che vengono effettuate non sono irreversibili, il che espone i venditori al rischio di frode.

Il sistema Bitcoin invece permette di accantonare l'idea dell'unità centrale di fiducia e garantisce l'irreversibilità delle transazioni. Questo perché Bitcoin non è altro che un sistema di pagamento elettronico basato sulla sicurezza crittografica anziché sulla fiducia, permettendo a qualsiasi coppia di persone di effettuare transazioni dirette senza il bisogno di un intermediario che "certifichi" la transazione stessa. Inoltre l'intrattabilità computazionale nell'invertire le transazioni effettuate tutela il venditore da possibili frodi.

Nell'articolo originale di Satoshi Nakamoto [1] il sistema viene descritto in nove pagine. Il sistema può essere descritto dalle seguenti proprietà: anonimato degli utenti, transazioni, block-chain, rete P2P, mining e consenso.

Segue una sezione che spiega il funzionamento della rete Bitcoin reale.

Per informazioni circa l'intero sistema Bitcoin si rimanda all'articolo originale [1] e soprattutto al seguente libro [8].

2.1 La Rete P2P Bitcoin

La rete Bitcoin è strutturata come un'architettura *peer-to-peer*. Il termine *peer-to-peer*, o P2P in breve, indica un sistema in cui gli agenti (in questo caso calcolatori) che fanno parte della rete sono pari l'un l'altro, ovvero che sono tutti uguali, senza la presenza di "agenti speciali", e che tutti i nodi della rete condividono l'onere di fornire i servizi della rete. I nodi della rete sono interconnessi fra loro in una rete mesh dove non è presente nessun server, nessun servizio centralizzato e alcun tipo di sistema gerarchico nella rete.

Un generico agente in una rete P2P allo stesso tempo, fornisce e usufruisce di servizi con reciprocità fungendo come incentivo alla partecipazione della rete stessa. Le reti P2P sono intrinsecamente resistenti, completamente decentralizzate e aperte. Oltre a Bitcoin, la più grande applicazione della tecnologia P2P è quella dello scambio di file che ha visto Napster [9] come pioniere di questa idea e BitTorrent [10] come la più recente evoluzione di tale architettura.

Bitcoin è una moneta digitale P2P, dove la sua architettura di rete è sia un riflesso e delle fondamenta per tale caratteristica fondamentale dove la decentralizzazione del controllo gioca un ruolo chiave che può essere ottenuto solo tramite una rete di consenso P2P decentralizzata.

Il termine "Rete Bitcoin" si riferisce all'insieme dei nodi della rete che seguono in protocollo P2P bitcoin. Oltre a tale protocollo, ne esistono altri, ad esempio, come Stratum [11] il quale è usato per effettuare *mining*, gestire nodi di tipo *lightweight* o per gestire i *mobile wallets*. Tali protocolli addizionali sono forniti dai server gateway che accedono alla rete bitcoin utilizzando il protocollo P2P bitcoin estendendo, quindi, tale rete con nodi che eseguono altri protocolli. Ad esempio, i server Stratum connettono i propri nodi che effettuano *mining* alla rete bitcoin principale collegando il protocollo Stratum a quello P2P bitcoin. Useremo il termine "rete bitcoin estesa" per indicare la rete complessiva, che include il protocollo P2P bitcoin, i protocolli di *mining*, il protocollo Stratum, e gli altri protocolli che si occupano di connettere le componenti del sistema bitcoin.

Come precedentemente accennato, la rete bitcoin principale, ovvero, quella che segue il protocollo P2P bitcoin, è composta da nodi che seguono varie versioni del client di riferimento bitcoin detto *Bitcoin Core*. In breve, esso, è un software libero e *open source* che funge da nodo bitcoin (il cui insieme costituisce la rete bitcoin) e fornisce un portafoglio (*wallet*) bitcoin che verifica automaticamente i pagamenti. Essa è considerata l'implementazione di riferimento di bitcoin poiché è quella che è stata proposta da Satoshi Nakamoto [1]. Inoltre la rete è composta da altre centinaia di nodi che eseguono altre implementazioni del protocollo P2P bitcoin, come: *Bitcoin Classic*, *Bitcoin Unlimited*, *BitcoinJ*, *Libbitcoin*, *btcd* e *bcoin*. Una piccola percentuale dei nodi nella rete P2P sono anche *miners*, che competono fra loro nel processo di *mining* validando le transazioni e creando nuovi blocchi da appendere.

La rete bitcoin estesa è quindi composta dalla rete che segue il protocollo P2P bitcoin e dai nodi che seguono diversi protocolli specifici. Connessi alla rete P2P principale ci sono anche *pool servers* e protocolli gateway che servono a connettere alla rete gli altri nodi che eseguono protocolli differenti da quello bitcoin. La maggior

parte di questi nodi sono nodi *miners* e *lightweight wallet client*, i quali non sono in possesso di una copia completa della *blockchain*.

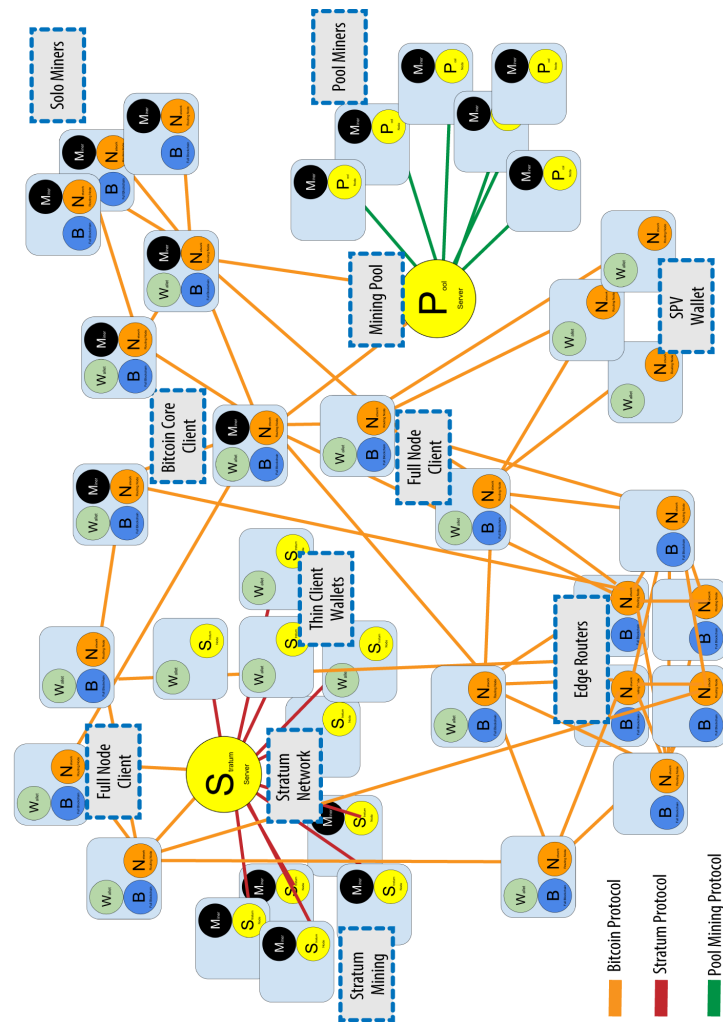


Figure 1: Immagine che rappresenta un esempio della rete bitcoin estesa mostrando diversi tipi di nodi, gateway e protocolli (immagine tratta da [8])

Quando un nuovo nodo accede alla rete P2P bitcoin deve, per poter partecipare alla rete, scoprire e connettersi con altri nodi della rete bitcoin. Più specificamente, un nuovo nodo, affinché possa partecipare alla rete, necessita di almeno un vicino nella rete bitcoin. La locazione geografica degli altri nodi è irrilevante; la topologia della rete bitcoin non è definita geograficamente. Perciò, ogni nodo bitcoin esistente può essere selezionato in modo casuale.

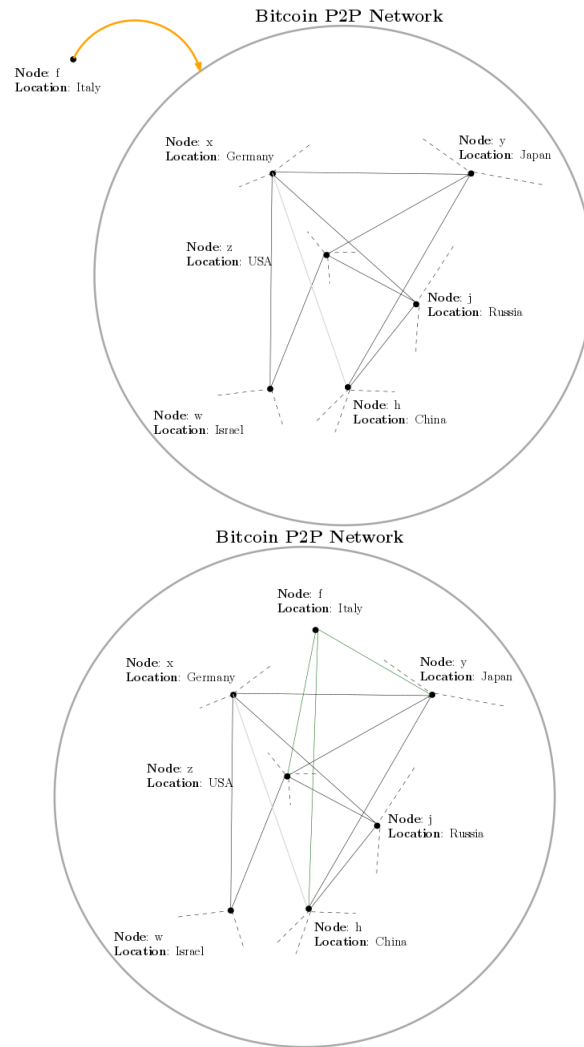


Figure 2: Immagini che rappresentano una porzione della rete P2P Bitcoin e la fase di connessione random di un nuovo nodo.

Per connettersi ad un peer conosciuto, i nodi stabiliscono una connessione TCP, solitamente dalla porta 8333 (la porta generalmente conosciuta come l'unica usata da

bitcoin) o una porta alternativa se previsto. Nel momento in cui una connessione sarà stabilita, il nodo inizierà un “handshake” trasmettendo un *Version Message*, che conterrà le informazioni di identificazione di base:

- Una costante che definisce la versione del protocollo P2P bitcoin che “parla” il client (e.g., 20000)
- Una lista di servizi locali supportati dal nodo
- Il tempo al quale il nodo effettua l’handshake
- L’indirizzo IP del nodo “destinatario ” della richiesta di handshake
- Il proprio indirizzo IP
- Informazioni circa la propria versione del software (eg: Satoshi: 0.9.2.1)
- L’altezza del blocco della *blockchain* posseduta

Il *Version Message* è sempre il primo messaggio inviato da ogni peer ad un altro peer. Un peer che riceve tale messaggio, per prima cosa, esamina la versione del protocollo P2P bitcoin utilizzata dal richiedente connessione e controlla se tale versione è compatibile o meno con se stesso. Se lo è, allora, procede col notificare al nodo richiedente, tramite un messaggio di *acknowledgement* (verack), l’esito e stabilisce la connessione con esso.

Un nodo ha, fondamentalmente, due modi per poter raggiungere gli altri peer nella rete, il primo è quello di interrogare direttamente i così detti “DNS seeds” che forniscono una lista di indirizzi IP di nodi bitcoin. Alcuni di questi DNS rispondono una lista di IP statici di nodi stabili in ascolto della rete bitcoin. Altri, invece, sono delle implementazioni personalizzate del Berkeley Internet Name Daemon (BIND) (citazione?) che ritorna un sottoinsieme casuale di indirizzi da una lista di nodi bitcoin collezionata da un *crawler* oppure da un altro nodo presente nella rete per un lungo periodo di tempo.

Il secondo modo che un nodo, non in possesso di alcuna informazione, per potersi connettere e farsi quindi introdurre alla rete, deve obbligatoriamente connettersi ad almeno un indirizzo IP di un nodo bitcoin conosciuto a priori.

Una volta che il nuovo nodo ha stabilito una o più connessioni nella rete, procede con l’inviare ai suoi vicini un messaggio contenente il proprio indirizzo IP. Questi ultimi, una volta ricevuto tale messaggio, procederanno con il comunicare, a loro volta, ai propri vicini tale indirizzo IP. Così permettendo al nuovo nodo di essere conosciuto nella rete e di stabilire ulteriori connessioni migliorando dunque la qualità della propria connessione Figura 3.

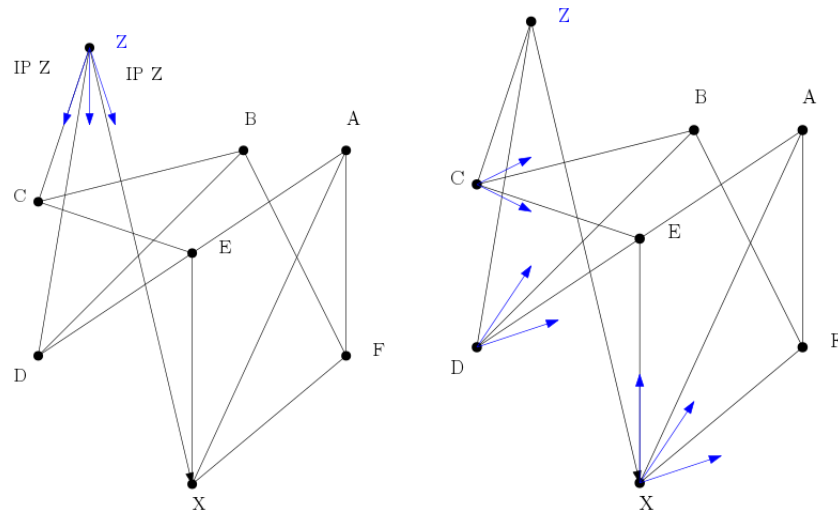


Figure 3: Immagini che rappresentano il processo di diffusione dell'IP che il nodo entrante **Z** innesca nella rete.

Inoltre, il nuovo nodo, può inviare una richiesta, diciamo *getaddr*, ai propri vicini, nella quale richiede una lista di indirizzi IP agli altri *peer*. In questo modo, un nodo può trovare altri agenti nella rete per stabilire una connessione e diffondere più velocemente il proprio IP, facendo sì che la propria “popolarità” aumenti.

Un nodo, per stabilire diversi percorsi all'interno della rete bitcoin, deve connettersi a pochi e a differenti *peer*. I percorsi non sono affidabili, in quanto i nodi nella rete vanno e vengono e ogni nodo quindi deve continuare a cercare nuovi vicini ogni volta che una delle sue connessioni svanisce e ad assistere i nodi entranti nella rete. Notiamo esplicitamente che il numero di connessioni iniziali sufficienti affinché un nuovo nodo entrante si connetta alla rete è pari a uno, in quanto tale primo nodo può occuparsi di introdurre nella rete il nodo entrante mediante i propri vicini. Notiamo inoltre che connettersi ad un grande numero di *peer* non è necessario. Anzi, si può considerare uno “spreco” delle risorse della rete. Ogni nodo attivo nella rete tiene in memoria i *peer* più recenti con i quali ha stabilito una connessione, cosicché nel caso in cui il software del nodo venga riavviato, esso possa connettersi rapidamente alla rete utilizzando i nodi con i quali era connesso prima del riavvio. Nel caso in cui il nodo non riceva alcuna risposta da nessuno dei *peer* salvati in memoria, allora effettua uno dei due procedimenti, illustrati precedentemente, per connettersi con la rete bitcoin.

Abbiamo spesso accennato al fatto che le connessioni possono “cadere”, ma come fa un nodo della rete a capire quando una connessione verso un suo vicino si è interrotta? Semplicemente, sulle connessioni prive di traffico i nodi inviano periodicamente dei messaggi per mantenere le connessioni. Data una connessione, se un *peer* non la utilizza per comunicare per più di 90 minuti, può assumere che essa sia caduta e quindi procedere con la ricerca di un nuovo vicino. In questo modo, la rete si adatta in modo dinamico ai nodi transienti e ai problemi della rete, e può

crescere organicamente e ridurre le proprie dimensioni a seconda delle proprie necessità senza l'ausilio di nessun tipo di controllo centralizzato.

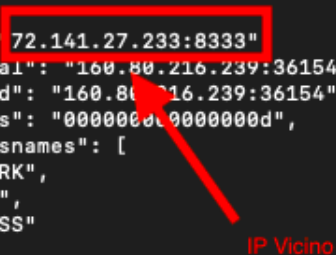
3 Analisi empiriche sulla rete Bitcoin

In questa sezione verranno illustrate le analisi empiriche effettuate sulla rete bitcoin reale. È stato effettuato un task di *Network Analysis* a due “livelli”: locale e globale. Nel primo sono state studiate le proprietà del singolo full node quali, ad esempio: numero di vicini connessi, tempo di persistenza dello stesso vicinato, numero (approssimativo) di nodi client connessi al full node. Per poter effettuare quest’ultima statistica è stato utilizzato *Bitnodes* [6], un software open source dotato di un crawler che permette di ottenere degli snapshot della rete bitcoin, nello specifico, dell’insieme dei full nodes.

3.1 Network Analysis Locale

È stato installato un full node in una macchina fornita dall’Università allo scopo di studiare il comportamento del suo vicinato. Utilizzando un full node con protocollo Bitcoin Core, mediante il comando *getpeerinfo*, è possibile ottenere la lista degli indirizzi IP dei nodi della rete bitcoin connessi ad esso.

```
{
  "id": 7,
  "addr": "72.141.27.233:8333",
  "addrlocal": "160.80.216.239:36154",
  "addrbind": "160.80.216.239:36154",
  "services": "0000000000000000d",
  "servicesnames": [
    "NETWORK",
    "BLOOM",
    "WITNESS"
  ],
  "relaytxes": true,
  "lastsend": 1581464233,
  "lastrecv": 1581464235,
  "bytessent": 4122,
  "bytesrecv": 2261482,
  "conntime": 1581464232,
  "timeoffset": -72,
  "pingtime": 0.163478,
  "minping": 0.163478,
  "version": 70015,
  "subver": "/Satoshi:0.15.1/",
  "inbound": false,
  "addnode": false,
  "startingheight": 616994,
  "banscore": 0,
  "synced_headers": 616938,
  "synced_blocks": 612940,
}
```



IP Vicino

Figure 4: Immagine di un'esecuzione del comando *getpeerinfo*.

Chiaramente, per motivi di sicurezza, non è possibile risalire alla tipologia di un nodo tramite il suo indirizzo ip. A tale scopo è stato sfruttato il software fornito da Bitnodes [6], ovvero sono stati collezionati gli snapshot della rete eseguiti ogni cinque minuti da Dicembre 2019 a metà Gennaio 2020 e grazie a questi dati e alla lista degli IP dei vicini del nostro full node è stato possibile capire quanti dei vicini del nostro nodo fossero full nodes e quanti fossero nodi di tipo diverso.

Dopo aver installato il full node e immediatamente dopo la sincronizzazione della sua blockchain, ad intervalli regolari di 15 minuti, sono stati salvati i seguenti parametri: numero di vicini mediante il comando *getconnectioncount* e una lista di file json contenenti le informazioni riguardanti i nodi connessi al full node tramite il comando *getpeerinfo*. In totale sono stati collezionati 2497 esempi, seguono alcune statistiche preliminari effettuate sul numero di vicini del nostro full node:

Mean	Median	Mode	Std
63.37	57.0	55.0	26.16

Table 1: Tabella delle statistiche riguardanti il numero di vicini del full node

Come si può notare dalla Tabella 3.1 si ha un numero medio di 63 nodi connessi al nostro full node. Eseguendo delle analisi più approfondite si è scoperto che la dimensione del vicinato del nostro nodo cresceva nel tempo. Ovvero, che il vicinato aumentava all'aumentare del tempo.

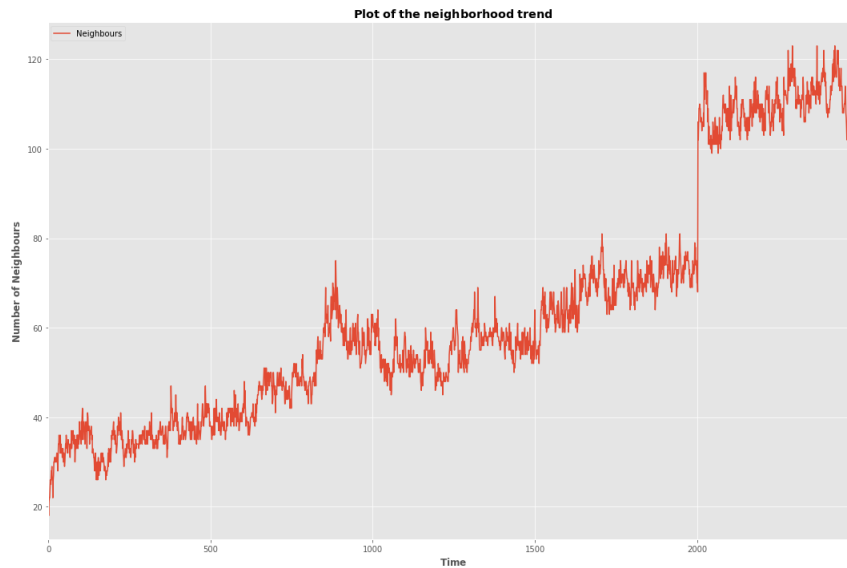


Figure 5: Grafico dell' andamento nel tempo della dimensione del vicinato del full node.

Come si può osservare in Figura 5, il numero di vicini del nodo cresce al crescere del periodo di attività. Questo, intuitivamente, è dovuto al fatto che inizialmente gran parte del vicinato è composto dai full node, che sono stati necessari per la sincronizzazione della blockchain e dai nodi di tipo diverso che si trovavano nella fase di ricerca di un agente in possesso della blockchain. All'aumentare del tempo, però, il nostro nodo viene gradualmente “inglobato” nella vera e propria rete bitcoin avendo un incremento costante di connessioni di nodi che possiamo chiamare “client”. Si può notare che dopo un periodo sufficiente di tempo si ha un netto aumento (da 80 a circa 115) del vicinato e dopo tale picco, il numero di vicini oscilla tra 100 e 120 ma non torna mai ad essere inferiore al minimo di tale intervallo.

È stato inoltre possibile stimare il numero di full node connessi ad ogni *snapshot* del nostro nodo. Nello specifico, sfruttando le informazioni fornite da [6] (lista dei full node nei diversi istanti di tempo) è stato ottenuto il seguente grafico

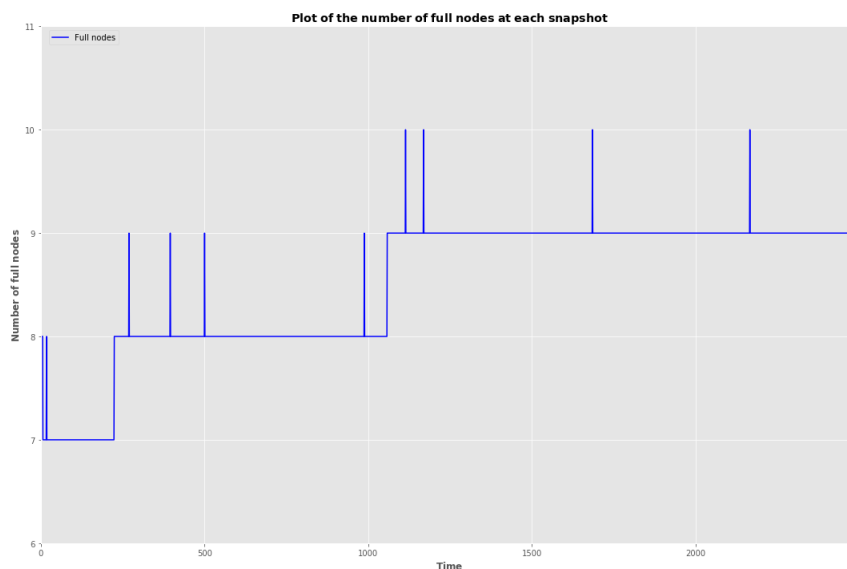


Figure 6: Grafico che mostra il numero di full nodes connessi al nostro nodo bitcoin nel tempo.

Si può osservare come il numero di full node connessi vari all' aumentare del tempo nel range tra 7 e 9. Sono presenti dei picchi di connessioni, questi ultimi potrebbero essere dovuti a dei nuovi nodi entrati nella rete che si connettono per un breve periodo per sincronizzare la propria blockchain.

Dai dati analizzati abbiamo ottenuto che il nostro nodo, su 26 ore di esecuzione, rimane, in media, connesso con gli stessi full nodes per circa 9 ore e con deviazione standard di 11. Per quanto riguarda i nodi client, in media rimangono connessi con il nostro full node per 3 ore con una deviazione standard di 24.

3.2 Network Analysis Globale

In questa sezione analizzeremo il comportamento della rete considerando solo i full nodes, poiché è possibile effettuare uno “snapshot” di tale insieme in un determinato istante di tempo, ma non lo è per quanto riguarda i clients. È stato collezionato un insieme di “snapshot” degli indirizzi IP dei full nodes nella rete utilizzando Bit-nodes [6].

Grazie a questi dati è stato possibile studiare il numero medio di nodi che si connettono e si disconnettono dalla rete e anche la permanenza media dei full nodes in essa.

	Mean	Median	Std
In Nodes	79.5	76.0	23.0
Out Nodes	78.3	75.0	24.0

Table 2: Tabella che mostra il numero medio di nodi che entrano ed escono ogni cinque minuti

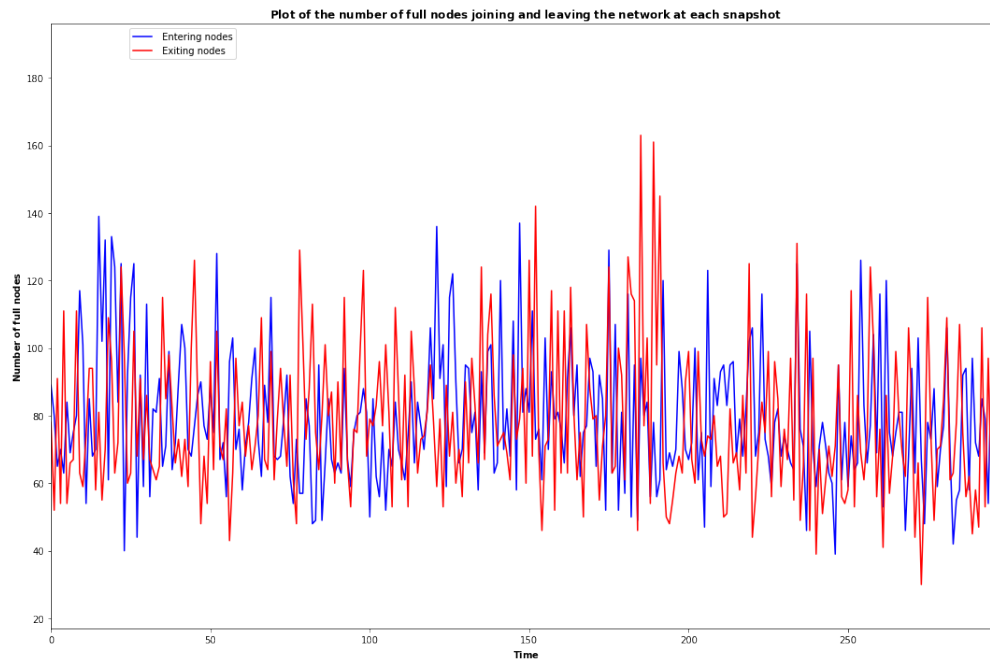


Figure 7: Grafico che mostra il numero di nodi che entrano ed escono ad ogni snapshot.

Nella Tabella 3.2 sono riportate le statistiche base del numero di nodi che entrano ed escono dalla rete. Si può osservare che tali quantità risultano essere “bilanciate” ovvero che, in media, ogni 5 minuti, nella rete, entra ed esce lo stesso numero di agenti. Il grafico illustrato in Figura 7, tratto da un campione dell’insieme dei dati, mostra il numero di agenti entranti ed uscenti ad ogni snapshot.

Successivamente è stato analizzato il tempo di vita dei full nodes nella rete. E dal nostro sample si ottiene che il tempo di vita medio di un full node è di circa 7 ore con un valore mediano di 9, e deviazione standard di 4.3 . Segue il grafico che mostra la distribuzione dei periodi di vita dei full nodes nella rete in termini di numero di osservazioni.

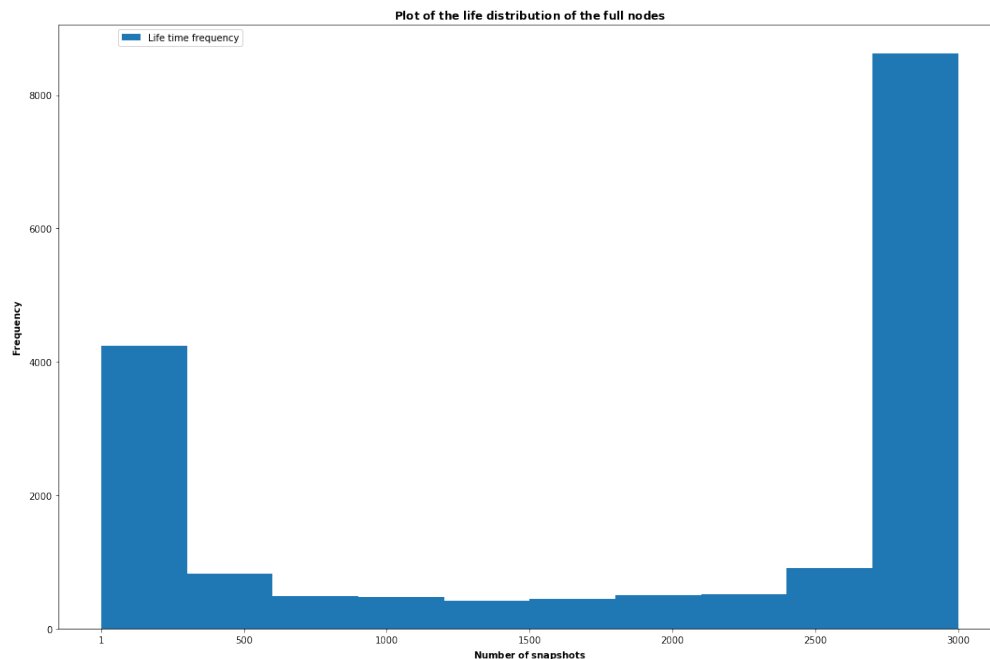


Figure 8: Grafico che mostra la distribuzione dei periodi di vita dei nodi osservati

Dal grafico si evince che i full nodes tendono a rimanere connessi per lunghi periodi di tempo alla rete.

Dall’ analisi effettuata sul nostro fullnode si è ottenuto che in media esso è rimasto connesso con gli stessi full nodes per circa 9 ore, e da quelle effettuate sulla rete globale si è visto che i full nodes tendono a rimanere connessi per periodi superiori alle 10 ore di attività. Questo suggerisce che i full nodes, fra di loro, stabiliscono delle connessioni “fisse”, mentre le connessioni che variano velocemente sono solo quelle dei nodi “client”.

4 Modelli teorici

In questo capitolo verranno illustrati due modelli teorici ispirati alla generazione della rete Bitcoin. Per il modello Edge-Dynamic si assumerà di avere che ad ogni istante di tempo gli archi scompaiano dal grafo con probabilità p . E per il modello Vertex-Dynamic si assumerà di avere nodi che entrano nella rete nella rete seguendo un flusso di Poisson e che si disconnettono con una probabilità q . Nella Sezione 4.4 analizzeremo la reversibilità della Catena di Markov che descrive il modello Edge-Dynamic e nelle Sezioni 4.5 e 4.6 mostreremo rispettivamente i risultati delle simulazioni del modello Edge-Dynamic e Vertex-Dynamic.

4.1 Edge Dynamic Graph

Siano: V l'insieme dei nodi di un grafo, inizialmente definito come $\mathcal{G}(V, \{\emptyset\})$, $d \in \mathbb{N}$, $c > 1$ e $p \in (0, 1)$. Detti, rispettivamente, “target”, “tolleranza”, e “probabilità di cadere di un arco”.

Si assuma che gli istanti di tempo siano discreti $t \in \mathbb{N}$ e senza perdita di generalità che al tempo $t = 0$, $\mathcal{G}(V, \{\emptyset\})$.

Ad ogni istante di tempo t (round), $\forall u \in V$ esegue le seguenti operazioni:

1. Sia $N_u^{inizio} = \{v \in V : u \sim v\}$ all'inizio del round, se $|N_u^{inizio}| < d$, u sceglie un insieme di nodi $X \subseteq V \setminus N_u^{inizio}$ tale che $|X| + |N_u^{inizio}| = d$ con probabilità uniforme e crea gli archi $u \sim v$, $v \in X$.
2. Sia $N_u^{fine} = \{v \in V : u \sim v\}$ alla fine del round, se $|N_u^{fine}| > c \cdot d$ il nodo u sceglie un insieme $Y \subseteq N_u^{fine}$ tale che $|N_u^{fine}| - |Y| = c \cdot d$ con probabilità uniforme ed elimina gli archi $u \sim v$, $v \in Y$.
3. Ogni arco $u \sim v$ nel grafo viene eliminato con probabilità p indipendentemente dagli altri archi in \mathcal{G} .

Osservazione 1. Dato $\mathcal{G}(V, E)$ Δ -Regolare con $\Delta \in \Omega(\log n)$. Se ogni arco $u \sim v$ cade con probabilità $p = \frac{1}{2}$, dato un nodo $u \in V$ esso con alta probabilità non diventerà un nodo isolato.

Proof. Sia $u \in V$ un nodo, allora, per far sì che tale nodo si disconnetta dalla rete devono cadere tutti gli archi a lui incidenti. In altre parole, devono cadere tutti gli archi che lo connettono al suo vicinato $N(u)$, ovvero, devono cadere $|N(u)| \in \Omega(\log n)$ archi.

Allora

$$P(\{\text{Falliscono } |N(u)| \text{ archi}\}) = 2^{-|N(u)|} = 2^{-\Delta} \leq 2^{-c \log n} = n^{-c}$$

■

Si osservi esplicitamente che, sotto questo “setting”, un grafo Δ - regolare con n nodi ha un numero totale di archi pari a $\frac{n\Delta}{2}$. Si può quindi fare la seguente osservazione.

Osservazione 2. La probabilità di un Grafo $\mathcal{G}(V, E)$ Δ -Regolare con $\Delta \in \Omega(\log n)$ di tornare in una configurazione in cui l'insieme degli archi E è vuoto è esponenzialmente bassa.

Infatti si ha che:

$$P(\{\text{Falliscono } \frac{n\Delta}{2} \text{ archi}\}) = 2^{-\frac{n\Delta}{2}} \leq 2^{-\frac{cn \log n}{2}}$$

4.2 Vertex Dynamic Graph

Siano: \mathcal{G} un grafo, inizialmente definito come $\mathcal{G}(\{\emptyset\}, \{\emptyset\})$, $\lambda, d \in \mathbb{N}$, $c > 1$ e $q \in (0, 1)$. Detti, rispettivamente, parametro di intensità del processo di Poisson, “target”, “tolleranza” e probabilità di disconnettersi di un agente.

Si assuma che gli istanti di tempo siano discreti $t \in \mathbb{N}$ e senza perdita di generalità che al tempo $t = 0$, $\mathcal{G}_0(\{\emptyset\}, \{\emptyset\})$.

Ad ogni istante di tempo t (round):

1. Nel grafo accedono $N(t)$ nodi, dove $N(t)$ è un processo di Poisson di parametro λ .
2. Sia $N_u^{\text{inizio}} = \{v \in V : u \sim v\}$ all'inizio del round, se $|N_u^{\text{inizio}}| < d$, u sceglie un insieme di nodi $X \subseteq V \setminus N_u^{\text{inizio}}$ tale che $|X| + |N_u^{\text{inizio}}| = d$ con probabilità uniforme e crea gli archi $u \sim v$, $v \in X$.
3. Sia $N_u^{\text{fine}} = \{v \in V : u \sim v\}$ alla fine del round, se $|N_u^{\text{fine}}| > c \cdot d$ il nodo u sceglie un insieme $Y \subseteq N_u^{\text{fine}}$ tale che $|N_u^{\text{fine}}| - |Y| = c \cdot d$ con probabilità uniforme ed elimina gli archi $u \sim v$, $v \in Y$.
4. Ogni nodo in \mathcal{G} si disconnette dal grafo con probabilità q .

4.3 Preliminari e notazioni

In queste sezioni, ci si riferirà sempre a grafi non orientati, senza self loop e senza archi multipli. Dato un insieme di nodi V di un grafo $\mathcal{G}(V, E)$ con la notazione $u \sim v$ verrà indicato l'arco $\{u, v\} \in E$.

Definizione 4.1. Un grafo $\mathcal{G}(V, E)$ è (d, cd) -regolare se il grado d_u di ogni nodo $u \in V$ è $d_u \in \{d, \dots, c \cdot d\}$.

Si ricordi che un grafo d -regolare con n nodi, si dice (n, d, α) -expander se $|\lambda_2(G)|, |\lambda_n(G)| \leq \alpha d$, dove $d = \lambda_1(G) \geq \lambda_2(G) \geq \dots \geq \lambda_n(G)$ è lo spettro della matrice di adiacenza del grafo \mathcal{G} .

Un vettore $\mathbf{p} \in \mathbb{R}^n$ si dice **vettore di probabilità** se le sue coordinate sono non negative e $\sum_{i=1}^n p_i = 1$. Il vettore di probabilità che corrisponde alla distribuzione uniforme su $\{1, \dots, n\}$ verrà indicato con $\mathbf{u} = \frac{1}{n}(1, \dots, 1)$. In questa sezione verrà mostrato che una Random Walk effettuata sui vertici di un Grafo Expander converge rapidamente alla misura stazionaria. Infatti il principale strumento tecnico utilizzato nelle analisi teoriche e nelle simulazioni che seguiranno sono le Catene di Markov A.1 sfruttate per analizzare i modelli stocastici che verranno proposti e per misurare la qualità dell'espansione durante le simulazioni.

Specificatamente, data una configurazione del modello in un generico istante di tempo $t \in \mathbb{N}$ sono state studiate le proprietà spettrali della matrice della Random Walk definita su Grafo Dinamico al tempo t .

Definizione 4.2. Una Random Walk definita su un grafo finito $\mathcal{G}(V, E)$ è un processo stocastico a tempo discreto (X_0, X_1, \dots) che assume valori in V . Il vertice X_0 viene scelto secondo una distribuzione di probabilità iniziale dall'insieme V , e il generico X_{i+1} viene scelto uniformemente a caso dall'insieme dei vicini di X_i .

Se un grafo \mathcal{G} è d -regolare, allora la matrice di adiacenza A e matrice di adiacenza normalizzata $\bar{A} = \frac{1}{d}A$. Seguono dei commenti intuitivi della random walk definita su \mathcal{G} .

- La random walk su $\mathcal{G}(V, E)$ è una Catena di Markov A.1 con spazio degli stati V e matrice di transizione $P = \bar{A}$
- P è reale, simmetrica e bistocastica
- Sia $1 = \lambda_1 \geq \lambda_2, \dots, \lambda_n \geq -1$ lo spettro di P , allora $\lambda_1 = 1$ e $\max\{|\lambda_2|, |\lambda_n|\} \leq \alpha$
- I corrispondenti autovalori sono anche autovalori di A .
- La misura stazionaria della Random Walk su \mathcal{G} è la distribuzione uniforme, $\mathbf{u}P = P\mathbf{u} = \mathbf{u}$ (Poiché P è simmetrica)

Osserviamo, ora, che se \mathcal{G} è un (n, d, α) -Expander e $\alpha < 1$, allora indipendentemente dalla distribuzione iniziale \mathbf{p} , la Random Walk converge in norma l_1 esponenzialmente veloce alla distribuzione limite (uniforme).

Teorema 4.1. [12] Sia \mathcal{G} un (n, d, α) -Expander con matrice di adiacenza normalizzata $\bar{A} = P$. Allora per ogni vettore di probabilità \mathbf{p} e ogni intero positivo t :

$$\|P^t \mathbf{p} - \mathbf{u}\|_1 \leq \sqrt{n} \cdot \alpha^t$$

Uno strumento più naturale per misurare la distanza tra due distribuzioni di probabilità μ, ν è **distanza in variazione totale** definita come:

$$\|\mu - \nu\|_{TV} = \max_{A \subseteq \Omega} |\mu(A) - \nu(A)| = \frac{1}{2} \|\mu - \nu\|_1$$

In altre parole, se la distanza in l_1 A.1.3 è piccola, allora le due distribuzioni di probabilità μ e ν assegnano quasi la stessa probabilità ad ogni evento nello spazio di probabilità Ω . Lo strumento per misurare la velocità di convergenza di una Random Walk, alla sua misura stazionaria ν è detto **tempo di mixing**:

$$\tau(\epsilon) = \min\{t \mid d(t) \leq \epsilon\}$$

dove;

$$d(t) = \max_i \|P^t(i, \cdot) - \nu\| = \max_i \left[\sum_j |P^t(i, j) - \nu(j)| \right]$$

Utilizzando lo spettro della matrice di transizione della Random Walk P si può trovare un bound al tempo di mixing. Sia $\gamma = \lambda_1 - \lambda_2 = 1 - \lambda_2$ la differenza fra il primo ed il secondo autovalore della matrice di transizione. Esso è detto **Gap Spettrale**. Grazie a tale valore è possibile definire il **tempo di rilassamento** [13] definito come l'inverso del gap spettrale: $t_{rel} = \frac{1}{\gamma}$. Grazie a t_{rel} è possibile fornire un bound al tempo di mixing.

Teorema 4.2. [13] Per ogni matrice di transizione P associata ad una Random Walk, aperiodica, irriducibile, e reversibile si ha che:

$$(t_{rel} - 1) \log\left(\frac{1}{2\epsilon}\right) \leq \tau(\epsilon) \leq t_{rel} \log\left(\frac{1}{2\epsilon \sqrt{\nu_{min}}}\right)$$

Informalmente si può osservare dal teorema che

$$\gamma \text{ piccolo} \iff \text{Tempo di Mixing elevato}$$

Questa osservazione informale fornisce uno strumento per valutare la qualità dell'espansione di un grafo sul quale viene effettuata una Random Walk. Poiché, i grafi Expander sono quelli che hanno il tempo di mixing della Random Walk più veloce di tutti, ovvero logaritmico sull'insieme dei nodi $\tau(G) = \mathcal{O}(\log(|V|))$.

Si osservi che i grafi sui quali verranno effettuate le Random Walk saranno (d, cd) -regolari e quindi la misura stazionaria non sarà quella uniforme ma bensì la seguente:

$$\nu(i) = \frac{|N(i)|}{2|E|}, \quad \forall i \in \Omega$$

4.4 Analisi Edge Dynamic Graph

Studiamo l'evoluzione temporale del grafo in termini di qualità della topologia. Nello specifico analizziamo l'espansione del grafo al variare del tempo, quanto tempo impiega il processo, al variare degli hyperparametri, a raggiungere una situazione di stazionarietà.

Si noti che questo modello stocastico può essere modellato come una Catena di Markov A.1 a tempo discreto e con spazio degli stati finito nel seguente modo: Sia $(X_t)_{t \geq 0}$ la Catena di Markov con spazio degli stati Ω , dove ogni $x \in \Omega$ è una configurazione diversa di \mathcal{G} . Definiamo la dimensione di Ω , sia $n = |\Omega|$, allora il numero di modi in cui si possono connettere n nodi è:

$$\sum_{i=0}^{\binom{n}{2}} \binom{\binom{n}{2}}{i} = 2^{\binom{n}{2}}$$

E' chiaro che $|\Omega| = 2^{\binom{n}{2}}$. Senza perdita di generalità assumiamo che $\{0\} \in \Omega$ sia la configurazione al tempo $t = 0$ ovvero $\mathcal{G}(V, \{\emptyset\})$. Allora, la densità iniziale della catena è Delta di Dirac in $\{0\}$ ($\pi_0(0) = 1$).

Sia $P \in \mathbb{R}^{|\Omega| \times |\Omega|}$, la matrice stocastica rappresentante la Catena di Markov, è chiaro che $P(x, y) > 0$, $x, y \in \Omega$ se e solo se è possibile effettuare una transizione in un passo dallo stato x allo stato y . Ovvero, se è possibile che, seguendo la dinamica, in un istante di tempo, si possa passare da una configurazione x ad una configurazione y , in breve: $\mathcal{G}_x \rightsquigarrow \mathcal{G}_y$.

Possiamo quindi definire la matrice come segue:

$$P(x, y) = \begin{cases} 0 < p_{x,y} \leq 1 & \text{Se } \mathcal{G}_x \rightsquigarrow \mathcal{G}_y. \\ 0 & \text{Altrimenti.} \end{cases}, \quad \forall x, y \in \Omega$$

Si osservi che assumendo di avere la probabilità di fallimento degli archi $p = 0$ si ottiene una catena di Markov con degli stati assorbenti. Osservando il modello, si possono classificare gli stati senza conoscere le probabilità di transizione. E' chiaro che $P(0, 0) = 0$ in quanto, assumendo di non avere $d = 0$, ogni nodo del grafo \mathcal{G}_0 sceglierà uniformemente a caso un insieme di nodi e ci si conatterà. Osserviamo esplicitamente che se $d = 0$ allora $P(0, 0) = 1$ e che quindi la configurazione iniziale è anche configurazione finale del modello. Torniamo al caso in cui $d \neq 0$ ed osserviamo esplicitamente che esiste $\mathcal{C} = C_1 \cup C_2 \cup \dots \cup C_k$ tale che $\forall i \in [k], C_i = \{x_i\} \wedge \mathcal{C} \subseteq \Omega \wedge P(x_i, x_i) = 1$. Ovvero esiste un sottoinsieme di stati della catena che sono stati assorbenti.

Questi sono gli stati rappresentanti tutte le configurazioni \mathcal{G}_i che hanno la seguente proprietà:

$$\forall u \in V, d \leq \text{Deg}(u) \leq c \cdot d$$

Questo modello può essere visto come un'applicazione dell'algoritmo RAES [5] su un grafo completo K_n il quale impiega $\mathcal{O}(\log n)$ round per convergere e con alta

probabilità abbiamo che il sottografo (d, cd) -regolare è un ϵ -expander *w.h.p.*.

Analisi della catena

Si ricordi che se esiste una distribuzione di probabilità π definita su Ω tale che soddisfa le equazioni di bilancio dettagliato

$$\pi(x)P(x, y) = \pi(y)P(y, x), \quad \forall x, y \in \Omega \quad (1)$$

si può dare la seguente definizione

Proposizione 1. *Sia P la matrice di transizione di una Catena di Markov con spazio degli stati Ω . Ogni distribuzione di probabilità π che soddisfa le equazioni di bilancio dettagliato è stazionaria per P*

Proof. Sommando entrambi i lati delle equazioni di bilancio dettagliato su tutti gli $y \in \Omega$ si ottiene:

$$\sum_y \pi(y)P(y, x) = \sum_y \pi(x)P(x, y) = \pi(x)$$

■

Controllare le equazioni di bilancio dettagliato risulta essere spesso conveniente per verificare se una data distribuzione di probabilità π è stazionaria. Questo perché quando vale tale proprietà si può riscrivere

$$\pi(x_0)P(x_0, x_1) \cdots P(x_{n-1}, x_n) = \pi(x_n)P(x_n, x_{n-1}) \cdots P(x_1, x_0)$$

che può essere riscritto come

$$P_\pi(X_0 = x_0 \dots X_n = x_n) = P_\pi(X_0 = x_n, X_1 = x_{n-1} \dots X_n = x_0)$$

In altre parole, se la catena $(X_t)_{t \geq 0}$ soddisfa le equazioni di bilancio dettagliato ed ha una distribuzione iniziale stazionaria, allora la distribuzione su (X_0, X_1, \dots, X_n) è la stessa della distribuzione $(X_n, X_{n-1}, \dots, X_0)$. Per questa ragione la catena viene detta **reversibile**. Esempi di catene reversibili sono l'urna di Ehrenfest [14], la catena di una random walk definita su un grafo e qualsiasi catena di nascita e morte persistente positiva in regime di stazionarietà, ovvero inizializzata con la relativa misura invariante.

Osservazioni sulla distribuzione stazionaria

Assumendo di avere $p \neq 0$ e osservando che la densità iniziale della catena è $\pi(0) = 1$, si ha che, nella catena, esiste un insieme di stati, $\mathcal{D} \subseteq \Omega$ che non possono essere raggiunti. Grazie a questa osservazione si può limitare l'analisi a $\Omega' = \Omega \setminus \mathcal{D}$. Sia \mathcal{P} , la matrice di transizione associata a Ω' , si ha che $\forall x \in \Omega', \mathcal{P}(x, x) > 0$ e $\forall x, y \in \Omega' \exists t > 0 : \mathcal{P}^t(x, y) > 0$. Ossia, \mathcal{P} è aperiodica, irriducibile e persistente positiva. Segue che

1. esiste una distribuzione di probabilità ν su Ω' tale che $\nu = \nu\mathcal{P}$, $\nu(x) > 0 \forall x \in \Omega'$, è unica e inoltre,
2. $\nu(x) = \frac{1}{E_x(\tau_x^+)}$

Dove $\tau_x^+ := \min\{t \geq 1 : X_t = x\}$ quando $X_0 = x$, è il **tempo di primo ritorno** nello stato x . E quindi $E_x(\tau_x^+)$ è il tempo medio di primo ritorno nello stato x .

Analisi della reversibilità

Un primo passo per cercare di calcolare la misura stazionaria ν è quello di verificare se la catena in analisi gode della proprietà di reversibilità. Ricordiamo che una catena è reversibile se soddisfa il Criterio di Kolmogorov.

Teorema 4.3. [15] *Una Catena di Markov, irriducibile, ricorrente positiva e aperiodica con matrice di transizione P è reversibile se e solo se le probabilità di transizione della dinamica soddisfano*

$$p_{i_1, i_2} p_{i_2, i_3} p_{i_3, i_4} \cdots p_{i_{n-1}, i_n} p_{i_n, i_1} = p_{i_1, i_n} p_{i_n, i_{n-1}} \cdots p_{i_4, i_3} p_{i_3, i_2} p_{i_2, i_1}$$

Per ogni sequenza finita di stati $i_1, i_2 \dots i_n \in \Omega$

Tuttavia, si dimostra che la proprietà di reversibilità non vale per la Catena di Markov in analisi, in quanto esiste una sequenza di stati, che inducono un ciclo, per i quali non vale il criterio sopra menzionato.

Teorema 4.4. *La catena $\mathcal{P}4.4$ non è Reversibile.*

Proof. Consideriamo quattro stati della catena descritti dalla seguente immagine:

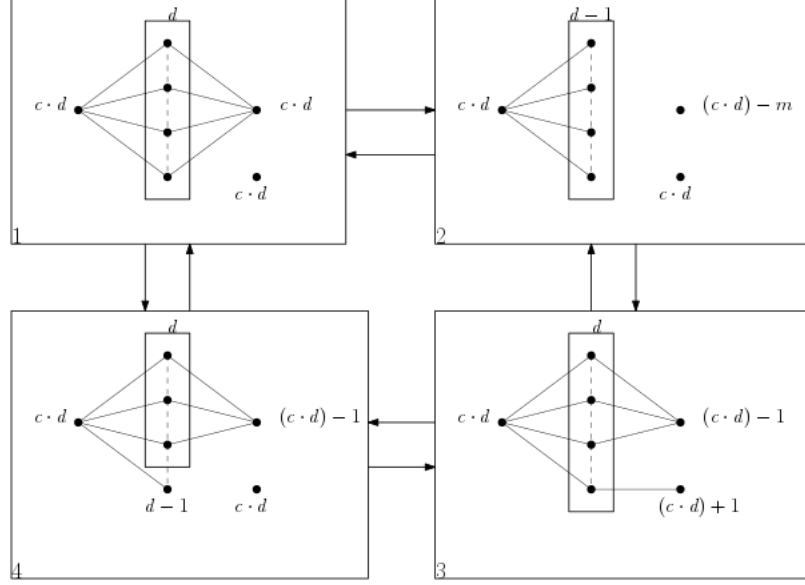


Figure 9: Porzione della catena di Markov che induce un ciclo per il quale non vale il Criterio di Kolmogorov.

Si assuma, senza perdita di generalità, di partire dallo stato indicato nella figura come stato 1, e che ogni nodo del grafo abbia grado compreso tra d e $c \cdot d$ e che p sia la probabilità di un arco di scomparire.

Seguono le descrizioni delle transizioni con le relative probabilità di salto da uno stato all'altro.

Probabilità di transizione dallo stato 1 allo stato 2:

Siano m i nodi centrali della Figura 9, essi hanno esattamente grado d , si assuma inoltre che $(c \cdot d) - m \geq c \cdot d$. Partendo dallo stato 1 l'unico evento che può far transire la catena in un passo allo stato 2 ($P(1,2)$), è il seguente:

“cadono gli m archi che connettono i nodi centrali al nodo di destra e non cade nessun altro arco del grafo”

Definendo come K il numero di archi totale del grafo nella configurazione rappresentata dallo stato 1 si può scrivere $P(1,2) = p^m(1-p)^{K-m}$. Si ha invece che la probabilità di tornare nello stato 1 dallo stato 2 è data dal seguente evento:

“gli m nodi centrali scelgono u.a.r. il nodo di destra e nessun arco del grafo cade”

Sia $\left(\frac{(n-1)-(d-1)}{n-1}\right)^m = a^m$ allora $P(2,1) = a^m(1-p)^K$.

La transizione dallo stato 2 al 3 è data dal seguente evento:

“m-1 nodi si connettono al nodo di destra, un nodo si connette al nodo in basso a destra e nessun arco della rete cade ”

Si può quindi definire $P(2,3) = a^m(1-p)^K$.

Si osservi che nello stato 3 il nodo in basso a destra ha grado $(c \cdot d) + 1$.

Per quanto riguarda $P(3,2)$ l'unico evento che può causare una transizione in un passo dallo stato 3 allo stato 2 è il seguente:

“Il nodo in basso a destra, sceglie u.a.r. dal proprio vicinato il nodo centrale ed elimina l'arco che li connette, inoltre cadono m-1 archi della rete”

Si ha quindi che $P(3,2) = \frac{1}{(c \cdot d)+1} p^{m-1} (1-p)^{K-m-1}$.

Per quanto riguarda $P(3,4)$ il nodo in basso a destra sceglie u.a.r. dal proprio vicinato il nodo centrale cancellando l'arco che li connette e non cade nessun altro arco dalla rete.

Quindi $P(3,4) = \frac{1}{(c \cdot d)+1} (1-p)^{K-1}$ e $P(4,3) = a(1-p)^K$.

Infine $P(4,1) = a(1-p)^K$ e $P(1,4) = p(1-p)^{K-1}$.

Ora a si può verificare se questo ciclo di 4 stati soddisfa il Criterio di Kolmogorov calcolando:

$$P(1,2) \cdot P(2,3) \cdot P(3,4) \cdot P(4,1) = p^m(1-p)^{K-m} a^m(1-p)^K \frac{1}{(c \cdot d)+1} (1-p)^{K-1} a(1-p)^K \\ = p^m(1-p)^{4K-m-1} a^{m+1} \frac{1}{(c \cdot d)+1}$$

$$P(1,4) \cdot P(4,3) \cdot P(3,2) \cdot P(2,1) =$$

$$p(1-p)^{K-1} a(1-p)^K \frac{1}{(c \cdot d)+1} p^{m-1} (1-p)^{K-m-1} a^m(1-p)^K = p^m(1-p)^{4K-m-2} a^{m+1} \frac{1}{(c \cdot d)+1}$$

Abbiamo quindi che

$$P(1,2) \cdot P(2,3) \cdot P(3,4) \cdot P(4,1) \neq P(1,4) \cdot P(4,3) \cdot P(3,2) \cdot P(2,1)$$

Quindi tale dinamica non soddisfa il Criterio di Kolmogorov, e si ha che \mathcal{P} non è reversibile. ■

Il risultato ottenuto, impedisce di utilizzare le equazioni di bilancio dettagliate (Formula 1) per calcolare la misura invariante ν della catena.

4.5 Simulazioni per il modello Edge Dynamic

I modelli analizzati sono stati sviluppati interamente in *Python*, utilizzando uno schema di programmazione ad oggetti. Nello specifico, è stato definito un oggetto *DynamicGraph*, il quale, codifica il processo stocastico che si vuole simulare. Separatamente, sono stati definiti: un oggetto per i protocolli di *Information Spreading* (protocollo di Flooding) e un oggetto per stabilire l'avvenuta convergenza del modello. Per ulteriori informazioni si rimanda all'Appendice A.2 oppure a [16] dove è possibile trovare il codice sorgente sviluppato per le simulazioni.

In queste simulazioni è stato studiato il tempo di convergenza e la qualità dell'espansione del grafo, ovvero si è misurato quanto tempo impiega il modello, al variare dei parametri d, c, p , a raggiungere una configurazione "stabile" e qual è la qualità di espansione di tale configurazione. Sono state poi studiate le proprietà strutturali del grafo, quali : grado e volume, per capire le strutture alle quali converge il modello. Infine sono state analizzate le performance del Protocollo di Flooding per misurare quanto tempo impiega un'informazione a diffondersi in tutto il grafo.

Risultati

In questa sotto sezione verranno presentati i risultati per il modello precedentemente proposto. Esso è stato analizzato al variare di d, c e p . Ogni esperimento è stato ripetuto per $m = 10$ volte. Il grafo dinamico random è stato inizialmente simulato assumendo una probabilità di fallimento degli archi pari a 0 ($p = 0$) e poi aumentando gradualmente tale probabilità. Nelle tabelle che seguiranno saranno riportate le medie dei risultati ottenuti e due ulteriori misurazioni: $\lambda_{gap}^{(max)}, \lambda_{gap}^{(min)}$. Questi ultimi valori rappresentano, rispettivamente, il massimo e il minimo gap spettrale 4.3 della matrice di transizione P_t di una random walk classica definita sulla configurazione \mathcal{G}_t assunta dal grafo nel generico tempo t , al quale il modello converge nelle m simulazioni.

Si ricordi che il gap spettrale della matrice di transizione P di una random walk definita su un grafo è legato al tempo di mixing della catena e di conseguenza alla qualità di espansione del grafo 4.3, [17]. Infatti, più il gap spettrale tende a 1 (e quindi più λ_2 è lontano da 1) e più velocemente la catena converge alla misura stazionaria ν [13]. Per la descrizione del criterio di convergenza adottato si veda l'Appendice A.2.

Inoltre, per ognuna delle m simulazioni, una volta ottenuta la convergenza dello spectral gap, si è simulato il protocollo di Flooding [18] scegliendo un nodo "initiator" uniformemente a caso dall'insieme dei vertici del Grafo. Si è, infatti, interessati a studiare la qualità del Grafo Dinamico Random in termini di velocità di terminazione del suddetto protocollo, in quanto, il modello proposto, deve sì godere della proprietà di (d, cd) -regolarità, ma deve anche garantire che un messaggio inviato da un agente qualsiasi riesca a "raggiungere" i restanti membri della rete in un numero di round relativamente piccolo.

Si ricordi che:

$$T[\text{Flooding}] = \mathcal{O}(D(\mathcal{G}))$$

Ovvero, il Flooding impiega un tempo minore o uguale al diametro del grafo per informare tutta la rete.

Si è quindi interessati ad ottenere una rete che goda di buone proprietà di espansione e/o abbia diametro logaritmico in $|V|$ per garantire una veloce diffusione dell'informazione come avviene nelle reti reali.

Seguono le tabelle che descrivono le medie dei risultati ottenuti dalle m simulazioni del modello al variare di d, c e p .

Tempo di convergenza ed espansione

Seguono le tabelle riassuntive per le simulazioni effettuate.

Nodes	Range of the Spectral Gap		Convergence Time	
	$\lambda_{gap}^{(max)}$	$\lambda_{gap}^{(min)}$	Mean	Standard Deviation
64	0.045720	0.008424	7	1
256	0.026568	0.017652	9	1
512	0.024401	0.011390	10	1
1024	0.022298	0.013511	12	1

Table 3: Tempi di convergenza e gap spettrali per il modello con: $p = 0, d = 2$ e $c = 1.5$

Nodes	Range of the Spectral Gap		Convergence Time	
	$\lambda_{gap}^{(max)}$	$\lambda_{gap}^{(min)}$	Mean	Standard Deviation
64	0.091676	0.032203	5	0
256	0.054709	0.026314	7	1
512	0.047550	0.033543	8	1
1024	0.044108	0.032252	9	1

Table 4: Tempi di convergenza e gap spettrali per il modello con: $p = 0, d = 2$ e $c = 3$

Nodes	Range of the Spectral Gap		Convergence Time	
	$\lambda_{gap}^{(max)}$	$\lambda_{gap}^{(min)}$	Mean	Standard Deviation
64	0.160664	0.097631	8	1
256	0.116441	0.089971	9	1
512	0.106277	0.097185	10	1
1024	0.102754	0.093414	11	1

Table 5: Tempi di convergenza e gap spettrali per il modello con: $p = 0, d = 3$ e $c = 1.5$

Nodes	Range of the Spectral Gap		Convergence Time	
	$\lambda_{gap}^{(max)}$	$\lambda_{gap}^{(min)}$	Mean	Standard Deviation
64	0.203106	0.118340	6	1
256	0.147694	0.120661	8	1
512	0.138777	0.123922	9	1
1024	0.131522	0.123922	10	1

Table 6: Tempi di convergenza e gap spettrali per il modello con: $p = 0, d = 3$ e $c = 3$

Nodes	Range of the Spectral Gap		Convergence Time	
	$\lambda_{gap}^{(max)}$	$\lambda_{gap}^{(min)}$	Mean	Standard Deviation
64	0.268496	0.198508	8	1
256	0.211848	0.184355	10	1
512	0.205502	0.181030	10	1
1024	0.197613	0.184485	11	1

Table 7: Tempi di convergenza e gap spettrali per il modello con: $p = 0, d = 4$ e $c = 1.5$

Nodes	Range of the Spectral Gap		Convergence Time	
	$\lambda_{gap}^{(max)}$	$\lambda_{gap}^{(min)}$	Mean	Standard Deviation
64	0.272794	0.204814	7	1
256	0.226805	0.197280	9	1
512	0.209376	0.186635	10	1
1024	0.205648	0.188519	11	1

Table 8: Tempi di convergenza e gap spettrali per il modello con: $p = 0, d = 4$ e $c = 3$

Dalle Tabelle 3, 4, 5, 6, 7 e 8 si può osservare che per $d = 2$ il gap spettrale, $1 - \lambda_2$, della matrice di transizione oscilla sempre fra valori molto vicini allo 0, ovvero si ha che $\lambda_2 \simeq 1$. Questo risultato suggerisce che il grafo, ad ogni istante di tempo, potrebbe non essere fortemente connesso oppure potrebbe essere bipartito.

In termini di una random walk classica, tale gap, suggerisce che essa non converge ad una misura stazionaria ν . Si può osservare che all'aumentare di d lo spectral gap si allontana gradualmente dallo zero, indicando, quindi, una migliore connettività del grafo e una migliore topologia in termini di espansione.

Facendo variare la probabilità di fallimento degli archi per valori $0 < p < 1$. In questo "setting" ad ogni istante di tempo, il gap spettrale del grafo è stato misurato due volte: dopo la fase 2) e dopo la fase 3).

Introducendo una probabilità non nulla di fault delle connessioni fra i nodi del grafo ci si aspetta che tali coppie di misurazioni effettuate diferiscano fra loro ad ogni istante di tempo.

Per $p = 0.002$, in media, i risultati ottenuti, in termini di velocità di convergenza del

modello sono molto simili a quelli ottenuti nella precedente simulazione. Si hanno però delle differenze per quanto riguarda la connettività del grafo, infatti, per valori di $d = 2$ il 10% delle volte il grafo risulta essere disconnesso. I gap spettrali, poiché la probabilità di fallimento p è molto bassa, in media, non variano dalla fase 2) alla fase 3) e non si discostano significativamente dai valori delle tabelle precedenti.

In Figura 17 in Appendice A.3, viene mostrato l'andamento, durante le varie simulazioni per $p = 0.002$, dei valori dei gap spettrali, delle convergenze dei modelli, prima della fase 2) e dopo la fase 3) per valori diversi di d e c . Si può osservare che tali gap non variano singnificativamente.

Ulteriori simulazioni sono state effettuate per $p \in [0.1, 0.7]$. Per quanto riguarda i gap spettrali prima della fase 3), per ogni esperimento, si ha che essi non si discostano significativamente da quelli descritti nelle Tabelle 3, 4, 5, 6, 7 e 8. Chiaramente, per le misurazioni effettuate dopo la caduta degli archi il numero di volte in cui il grafo risulta disconnesso (valore dello spectral gap 0) aumenta gradualmente all'aumentare del valore di p .

Confrontando i gap spettrali con quelli di grafi aleatori noti, si ottiene che, a parità della dimensione dell'insieme dei nodi, il modello proposto ha valori molto vicini a quelli dei grafi random Δ -regolari.

Dai risultati ottenuti si evince che il modello, al variare di p , d e c , converge in media in tempo logaritmico ad una classe di configurazioni che hanno le stesse proprietà di espansione. Per $p = 0$ e valori di $d = 2$ si ha che il grafo non converge ad una configurazione con buone proprietà di espansione, mentre per $d = 4$ si ottiene un Grafo Dinamico Random che in media ad ogni round gode delle proprietà di espansione di un grafo 4-regolare. Per questo motivo, nei paragrafi successivi verranno omessi i risultati delle simulazioni per valori di $d = 2, 3$ in quanto non generano un Expander Dinamico.

Proprietà strutturali

In questo paragrafo vengono illustrate le proprietà strutturali del Grafo Dinamico Random per i valori di d e c più significativi in termini di qualità dell'espansione del grafo.

Nodes	Number of (d,cd)-regular nodes		Volume		Degree	
	Mean	Standard Deviation	Mean	Standard Deviation	Mean	Standard Deviation
64	61	3	287	17	6	1
256	243	13	1156	63	6	1
512	485	26	2314	126	6	1
1024	970	53	4622	251	6	1

Table 9: Proprietà strutturali per $p = 0, d = 4$ e $c = 1.5$

Si può osservare dalla Tabella 9 che il migliore Expander Dinamico, al variare del numero di nodi ha circa il 95% dei nodi (d, cd) -regolari.

Vengono di seguito riportate delle ulteriori analisi del modello con $d = 4$ e $c = 1.5$ al variare di p .

Number of nodes				
	64	256	512	1024
p	Average (d,cd)-regular	Average (d,cd)-regular	Average (d,cd)-regular	Average (d,cd)-regular
0	61	243	485	970
0.002	63	254	509	1019
0.1	47	145	460	922
0.2	32	130	263	820
0.3	20	83	166	334
0.4	11	45	90	182
0.5	4	19	40	80
0.6	1	6	13	26
0.7	0	1	3	6

Table 10: Proprietà strutturali al variare di p per $d = 4$ e $c = 1.5$

Nella Tabella 10 vengono riportati i numeri medi di nodi (d, cd) -regolari al variare di n e di p misurati dopo la fase 3). Come si può osservare, all'aumentare della probabilità di fallimento il Grafo Dinamico non riesce a mantenere una struttura (d, cd) -regolare e quindi, di conseguenza, perde le proprietà di espansione ottenute dopo la fase 2). Più precisamente ad ogni round, dopo la fase 2) il modello raggiunge una configurazione (d, cd) -regolare che però, all'aumentare di p , tende a perdere subito dopo la fase 3).

Tempo di Flooding

Per quanto riguarda il protocollo di Flooding, si osserva, che, in media, all'aumentare del valore target d e alla qualità di espansione del grafo si ha una rapida convergenza in termini di velocità di “information spreading”. Si può, infatti, osservare come per $d = 4$ e $c = 1.5$ la velocità media di completamento del protocollo di Flooding sia inferiore a $\log n$.

Seguono i risultati ottenuti per il miglior Expander Dinamico:

p	Number of nodes							
	64		256		512		1024	
	Average Flooding Time	Std Flooding Time	Average Flooding Time	Std Flooding Time	Average Flooding Time	Std Flooding Time	Average Flooding Time	Std Flooding Time
0	4	0	5	1	6	0	7	0
0.002	4	0	5	1	7	2	7	1
0.1	4	1	5	2	7	2	8	1
0.2	5	1	6	1	7	1	9	1
0.3	6	1	7	1	9	1	9	1
0.4	7	1	9	1	10	1	11	1
0.5	8	2	11	1	13	1	14	1
0.6	11	2	16	1	23	3	19	2
0.7	14	3	20	3	24	0	25	2

Table 11: Risultati simulazioni per $d = 4$ e $c = 1.5$

Dalla Tabella 11 si può osservare che, in media, facendo variare la probabilità di fallimento degli archi, il protocollo di Flooding non viene rallentato in modo significativo. Infatti per $p \in [0.002, 0.4]$ esso termina, in media, in circa $\log n$ round. Per $p \in [0.5, 0.7]$ si ha un lieve deterioramento del tempo di completamento del protocollo. Infatti, esso passa da logaritmico a circa \sqrt{n} . Addirittura si ha che negli esperimenti con $n = 1024$ si ottiene, una terminazione dell'algoritmo in meno di \sqrt{n} round. Per gli esperimenti con $d = 4$, $c = 2, 3$ si ottengono dei risultati identici a quelli della Tabella 11. Essi differiscono di al più un round nei casi in cui la probabilità di fallimento degli archi assume valori “estermi”.

Tali risultati confermano che l'informazione diffusa dal nodo “initiator” si propaga più velocemente nella rete in un “setting ” dove le connessioni fra gli agenti non cadono. Ovvero, per quanto riguarda il modello proposto, l'ottimo si ottiene in uno scenario dove, in media, si ha il 99% di nodi (d, cd) -regolari e mano a mano che tale percentuale diminuisce il protocollo di Flooding impiega un numero superiore di round per raggiungere la terminazione.

Conclusioni simulazioni modello Edge Dynamic

In conclusione, dalle analisi si evince che il modello proposto, per valori appropriati di d e c ad ogni istante di tempo $t \geq 0$ prima della fase in cui cadono gli archi, assume una configurazione (d, cd) -regolare che presenta un gap spettrale vicino a quello di un grafo random d -regolare. Questo risultato, suggerisce la presenza di un drift[13] della Catena di Markov, definita sull'insieme delle possibili configurazioni $\{\mathcal{G}_t\}_{t \geq 0}$, in un insieme di stati $E \subset \Omega$ il quale codifica l'insieme delle configurazioni (d, cd) -regolari.

Inoltre, per quanto riguarda i valori dei parametri di d, c studiati il modello, per valori appropriati di $0 \leq p < 0.1$ una volta assunta una configurazione $x \in E$ tende a "mutare" in un'altra configurazione $y \in E$ preservando, quindi, la proprietà di (d, cd) -regolarità.

Per il protocollo di Flooding, eseguito sul modello in continua evoluzione, si ha un tempo di terminazione inferiore a $\log n$ round per le configurazioni che risultano essere nella classe di stati $E \subset \Omega$ e per le simulazioni effettuate con $d = 2$, al variare di p si ha che, in media, l'informazione viene diffusa in tutta la rete in più di $\log n$ round.

Il modello proposto, però, non coglie una proprietà fondamentale di una rete P2P reale. Infatti non modella l'arrivo e l'uscita degli agenti nella rete. A tale scopo segue un ulteriore modello di Grafo Dinamico Aleatorio che presenta due componenti dinamiche: l'insieme dei nodi V e l'insieme degli archi E .

4.6 Simulazioni per il modello Vertex Dynamic

Si osservi che si può analizzare l'evoluzione della componente dinamica riguardante il numero di nodi nel grafo considerando il seguente processo:

$\forall t \geq 0$

1. Nel grafo accedono $N(t)$ nodi, dove $N(t)$ è un processo di Poisson di parametro λ .
2. Ogni nodo in \mathcal{G} si disconnette dal grafo con probabilità q .

Tale processo stocastico può essere modellato come una coda **M/G/∞** [19], come segue: si assuma di analizzare un servizio nel quale i clienti arrivano seguendo un Processo di Poisson di parametro λ e che tale servizio abbia un numero infinito di sportelli dove le persone possono essere servite immediatamente. Si assuma inoltre che i tempi di servizio dei clienti siano definiti da una sequenza di variabili random indipendenti aventi una comune distribuzione di probabilità di media μ . Si osservi che tale coda markoviana non è altro che una formulazione alternativa del processo descritto precedentemente.

Si ha quindi che la distribuzione stazionaria del processo è che vogliamo analizzare è di tipo Poisson:

$$v_i = \frac{(\frac{\lambda}{q})^i}{i!} e^{-\frac{\lambda}{q}}, i \geq 0$$

E che il numero medio di nodi nel grafo è $\frac{\lambda}{q}$.

Assumendo di avere un tempo di permanenza degli agenti della rete di tempo esponenziale di parametro μ si può dimostrare che

Teorema 4.5. [7] Sia $n = \frac{\lambda}{\mu}$ e $\mathcal{G}(V_t, E_t)$ il grafo dinamico al tempo t , allora:

- Per ogni $t = \Omega(n)$, con alta probabilità $|V_t| = \Theta(n)$
- Se $\frac{t}{n} \rightarrow \infty$ allora, con alta probabilità $|V_t| = n + o(n)$

Ulteriori analisi sulla dimensione dell'insieme dei nodi possono essere estese da [7] al modello proposto assumendo di avere un tempo di vita esponenziale dei nodi e per tanto omesse.

Simulazioni

In questa sotto sezione verranno presentati e discussi i risultati ottenuti dalle simulazioni del grafo dinamico random formalizzato in questa sezione. Per diversi valori degli hyper parametri d, c, λ e q sono state effettuati $m = 30$ esperimenti, dove ognuno è descritto dal seguente processo:

Si aspetta che il grafo raggiunga un numero di nodi pari a $\frac{\lambda}{q}$, successivamente si attende che il 90% degli agenti del grafo abbia un numero di vicini compreso fra d e $c \cdot d$. Quando il grafo risulta essere, quindi, (d, cd) -regolare si simula il protocollo di Flooding attendendo la sua terminazione.

Da degli esperimenti preliminari si è osservato che per valori di $q \geq 0.1$ i grafi non riescono ad avere un numero significativo di nodi con grado compreso tra d e $c \cdot d$ (inferiore al 70%) risultando, per la maggior parte degli istanti di tempo, dei grafi disconnessi facendo fallire le simulazioni del protocollo di Flooding. Si è quindi deciso di effettuare test empirici sui modelli con valori di q molto inferiori a 0.1.

Un risultato atteso dagli esperimenti che seguiranno è quello di avere, come per il modello della sezione precedente, una forte relazione tra il diametro del grafo e il tempo di Flooding, in particolare, ci si aspetta che per valori di $d \geq 3$ si avrà l'ottimo e che $d = 2$ indurrà dei Grafi Dinamici che non godranno di una buona connettività e di un piccolo diametro.

Seguono, quindi, le tabelle e i grafici degli esperimenti effettuati per $\lambda = 1$ e $q = 0.001$

Proprietà Strutturali e Tempo di Flooding

Nella seguente tabella vengono mostrati i risultati ottenuti dalle simulazioni utilizzando $\lambda = 1$, $q = 0.001$ e quindi con un numero medio di nodi nella rete di $\frac{\lambda}{q} = 1000$. Nello specifico, ogni simulazione del modello al variare di d e c è stata effettuata per 30 volte, la seguente tabella riporta, quindi, le medie dei risultati ottenuti.

d	c	Average (d,cd)-regular	Std (d,cd)-regular	Average V	Std V	Average time disconnected	Std time disconnected	Average Volume	Std Volume	Average Deg	Std Deg	Average Flooding Time	Std Flooding Time
2	1.5	964	33	967	33	0.14	0.01	2369	83	3	0.16	21	2
2	2	964	33	966	33	0.06	0.06	2630	93	4	0.18	15	1.5
2	3	965	33	965	33	0.03	0.03	2836	100	6	0.19	12	1
3	1.5	960	33	965	33	0.12	0.02	3348	115	5	0.23	11	1
3	2	963	33	966	33	0.04	0.04	3892	134	6	0.27	9	0.5
3	3	962	33	965	33	0.00	0.00	4101	143	8	0.29	8	0.5
4	1.5	960	33	965	33	0.10	0.10	4694	163	6	0.32	8	0.8
4	2	961	33	966	33	0.05	0.05	5122	181	8	0.36	7	0.5
4	3	962	33	966	33	0.01	0.01	5302	184	11	0.37	6	0.4

Table 12: Risultati simulazioni per $\frac{\lambda}{q} = 1000$

Una prima osservazione che può essere fatta, da questo primo esperimento, è che in media, il tempo di completamento del protocollo di Flooding diminuisce al crescere del grado medio dei nodi del grafo e soprattutto che in media si hanno i risultati migliori quando il grafo risulta essere connesso per la maggior parte dei round delle simulazioni. Dagli m test è emerso, inoltre, che per $d = 4$ e $c \in \{1.5, 2, 3\}$, in media, il grafo ha un diametro logaritmico. Oltre agli score medi delle simulazioni, sono stati analizzati quelli che possono essere considerati come i casi peggiori ottenuti. I quali sono quelli in cui il Flooding fallisce, ovvero quando una porzione degli agenti della rete non viene mai informata, o quando il protocollo termina correttamente impiegando, però, un tempo superiore rispetto a tutte le altre simulazioni.

Una prima ipotesi per giustificare le basse prestazioni del Flooding è quella in cui il grafo risulta essere disconnesso per la maggior parte del tempo in cui il protocollo è in azione. È interessante anche studiare le cause che garantiscono elevate prestazioni in termini temporali del protocollo di information spreading. Come per il caso peggiore, un'ipotesi è che il comportamento del protocollo dipenda fortemente dal fatto di essere eseguito su una topologia dinamica che non presenti componenti disconnesse per la maggior parte degli istanti di tempo. E un'altra ipotesi è che tali performance aumentino quando il grafo è connesso ma è anche (d, cd) -regolare.

Seguono i grafici che illustrano le analisi effettuate sui casi peggiori e i casi migliori delle simulazioni per il modello con i parametri precedentemente definiti.

Plot for $d:2$ $c:1.5$

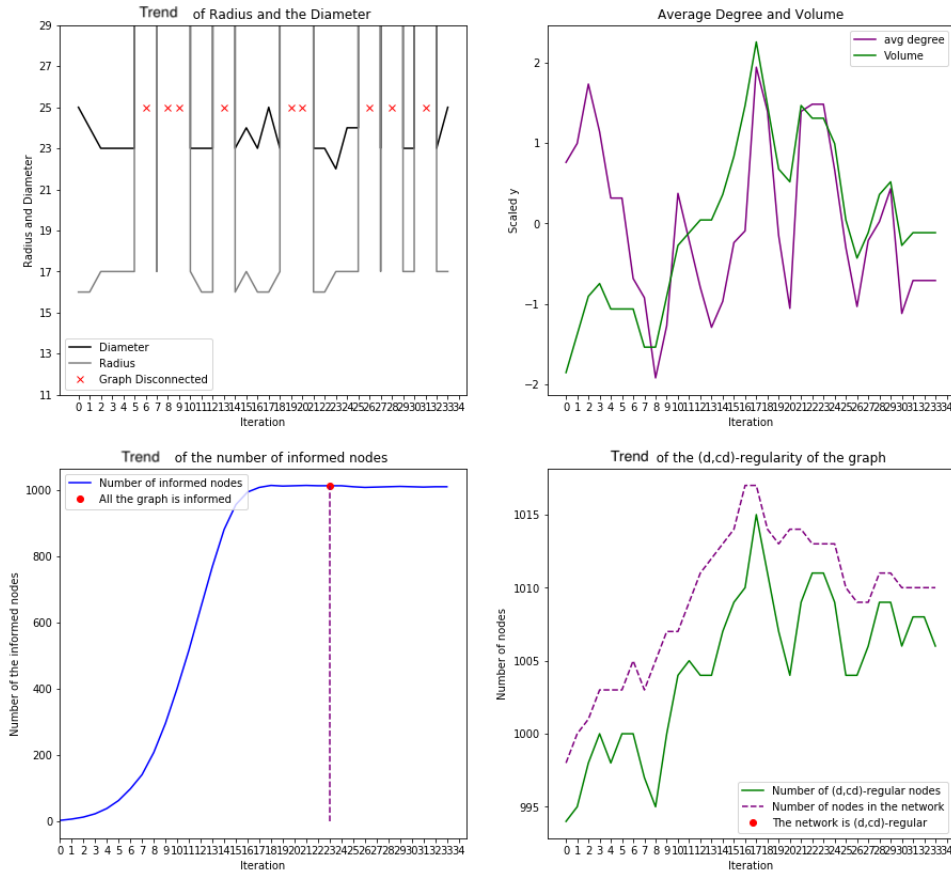


Figure 10: **In alto a sinistra:** grafico che mostra l'andamento del diametro e del raggio del grafo nel tempo. **In alto a destra:** grafico che mostra la relazione tra grado medio e volume del grafo nel tempo. **In basso a sinistra:** grafico che mostra il numero di nodi informati dal protocollo di flooding ad ogni istante di tempo. **In basso a destra:** grafico che mostra il numero di nodi (d, cd) -regolari e il numero totale di nodi nella rete per ogni istante di tempo

In Figura 10 sono mostrate le misurazioni effettuate sulla simulazione peggiore in termini di tempo di esecuzione del protocollo di flooding. Si osservi che il caso peggiore si ha per una simulazione del modello con $d = 2$ e $c = 1.5$, poiché si ottiene un grafo dinamico che per la maggior parte del tempo di esecuzione del protocollo di flooding risulta o essere disconnesso o avere un diametro molto elevato compreso tra 23 e 25. Dalla figura si può osservare che il protocollo di information spreading

informa tutta la rete in 23 istanti di tempo. Si osservi, inoltre, che circa 90% dei della rete è (d, cd) -regolare ma che non esiste un istante di tempo in cui lo è tutta la rete.

Segue il grafico che illustra la simulazione migliore ottenuta in termini di tempo richiesto dal flooding per informare tutta la rete.

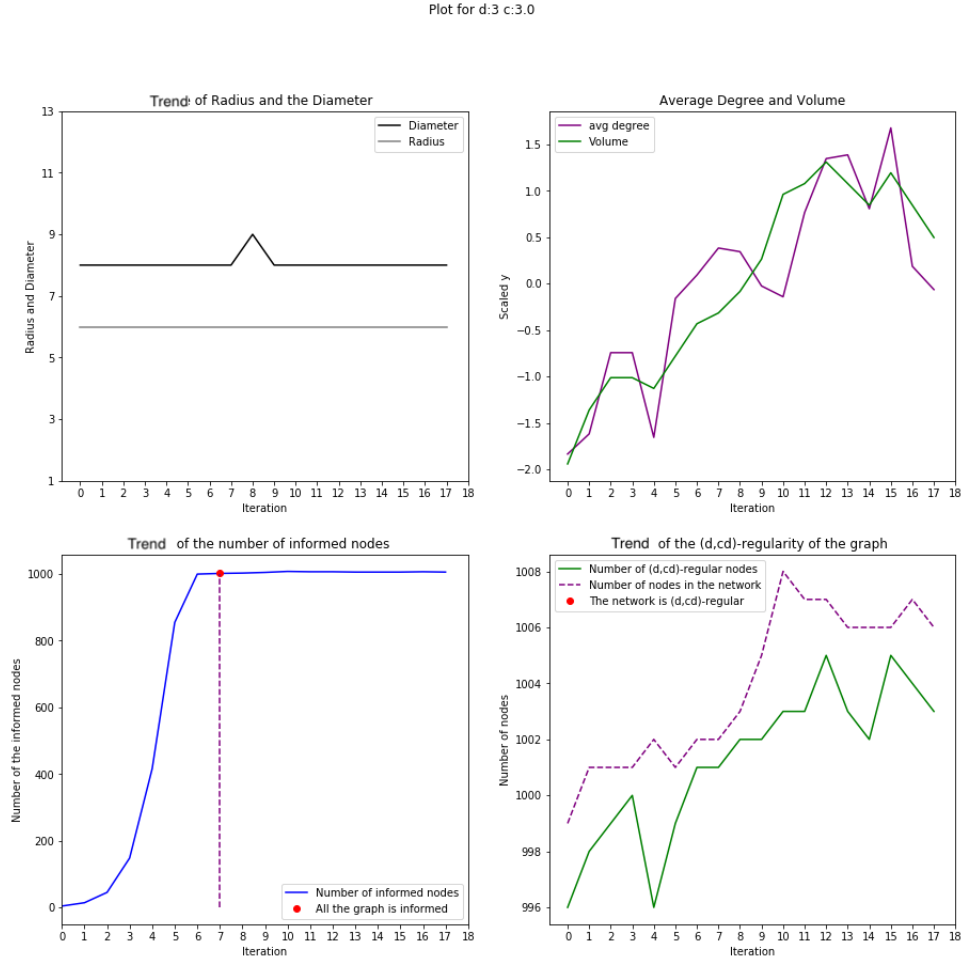


Figure 11: Figura relativa alla simulazione migliore ottenuta

In Figura 11 si può osservare che il protocollo di Flooding, eseguito su un grafo dinamico con $d = c = 3$, impiega solo 7 istanti di tempo per informare tutta la rete, che essa risulta essere sempre connessa con un diametro di 8, logaritmico in $|V|$, e che per tutti gli istanti di tempo dell'esecuzione del protocollo il 99% dell'insieme dei nodi è (d, cd) -regolare. Nell'immagine successiva, sono riportate le misurazioni effettuate sulla simulazione che ha generato un grafo dinamico (d, cd) -regolare per

la maggior parte degli istanti di tempo.

Plot for d:2 c:3.0

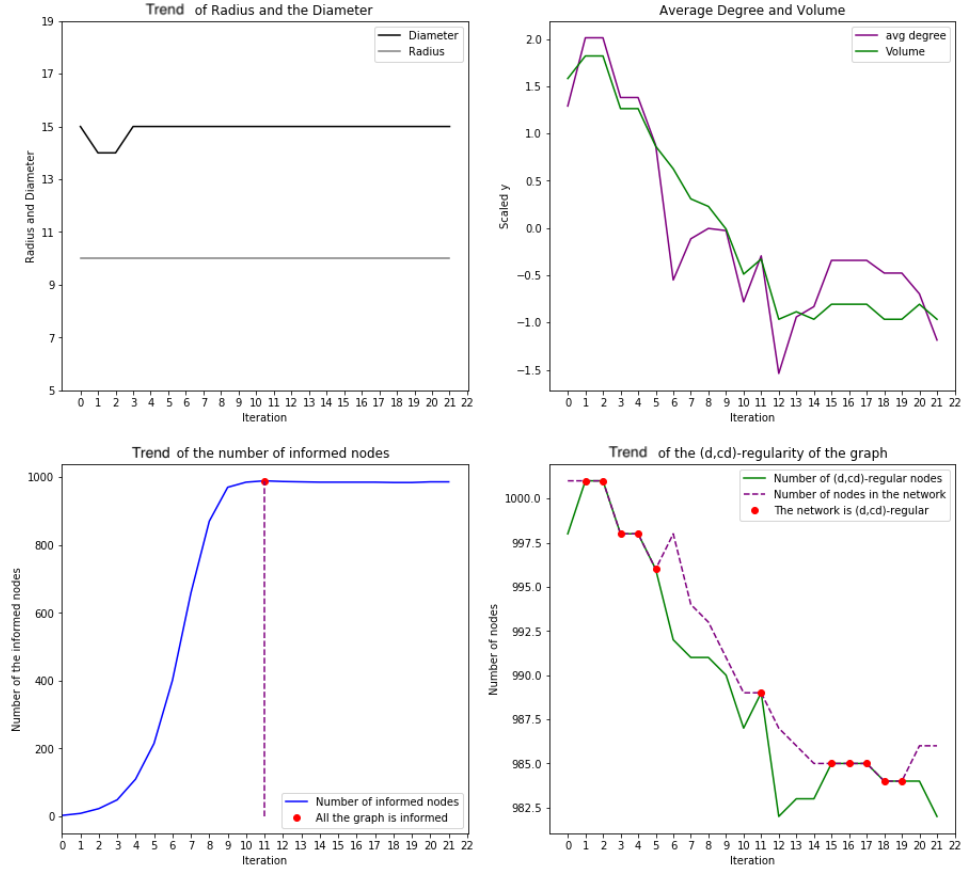


Figure 12: Figura relativa alla simulazione che ha generato il miglior grafo (d,cd) -regolare

In Figura 12 si può osservare che nonostante il grafo risulti spesso essere (d,cd) -regolare si ha un diametro superiore a quello di Figura 11 e un tempo di Flooding di 11 istanti tempo, peggiore quindi del precedente esempio mostrato.

Dalle simulazioni effettuate, si evince che il protocollo di Flooding termina più velocemente in topologie dinamiche con $d > 2$, nello specifico l'ottimo si ha per $d = 4$ e $c = 3$.

Sono stati effettuati ulteriori esperimenti fissando $\lambda = 1$ e aumentando gradualmente q . Ottenendo quindi un numero medio $\frac{\lambda}{q}$ di nodi nel grafo inferiore rispetto

a quello degli esperimenti precedentemente descritti.

Seguono i risultati ottenuti per le simulazioni dove si sono scelti $\lambda = 1$ e $q = 0.01$

d	c	Average (d,cd)-regular	Std (d,cd)-regular	Average [V]	Std [V]	Average time disconnected	Std time disconnected	Average Volume	Std Volume	Average Deg	Std Deg	Average Flooding Time	Std Flooding Time
2	1.5	92	4	95	5	0.2	0.0	228	12	3	0.2	14	2
2	2	93	5	95	5	0.1	0.1	257	14	4	0.2	10	1
2	3	94	5	95	5	0.1	0.1	279	15	6	0.2	8	1
3	1.5	91	5	96	5	0.1	0.1	326	18	5	0.2	8	2
3	2	93	4	96	4	0.1	0.1	383	21	6	0.3	7	1
3	3	92	5	96	5	0.1	0.1	401	25	8	0.3	6	1
4	1.5	91	5	96	5	0.1	0.0	457	24	6	0.4	6	1
4	2	91	4	96	4	0.1	0.1	500	24	8	0.4	6	1
4	3	92	4	97	4	0.1	0.1	524	27	11	0.4	5	1

Table 13: Risultati simulazioni per $\frac{\lambda}{q} = 100$

Come si può vedere dalla Tabella 13, in media il miglior tempo di completamento del protocollo di Flooding si ha per $d = 4$ e $c = 3$. Questo è dovuto al fatto che in questi m esperimenti per i valori di $d = 4$ e $c = 3$ si ha che il grafo risulta quasi sempre connesso e soprattutto che, mediamente, il suo diametro è logaritmico.

Anche per queste simulazioni sono stati analizzati i casi peggiori e i casi migliori in termini di tempo di completamento del protocollo di information spreading e (d, cd) -regolarità del grafo.

Plot for $d:2$ $c:1.5$

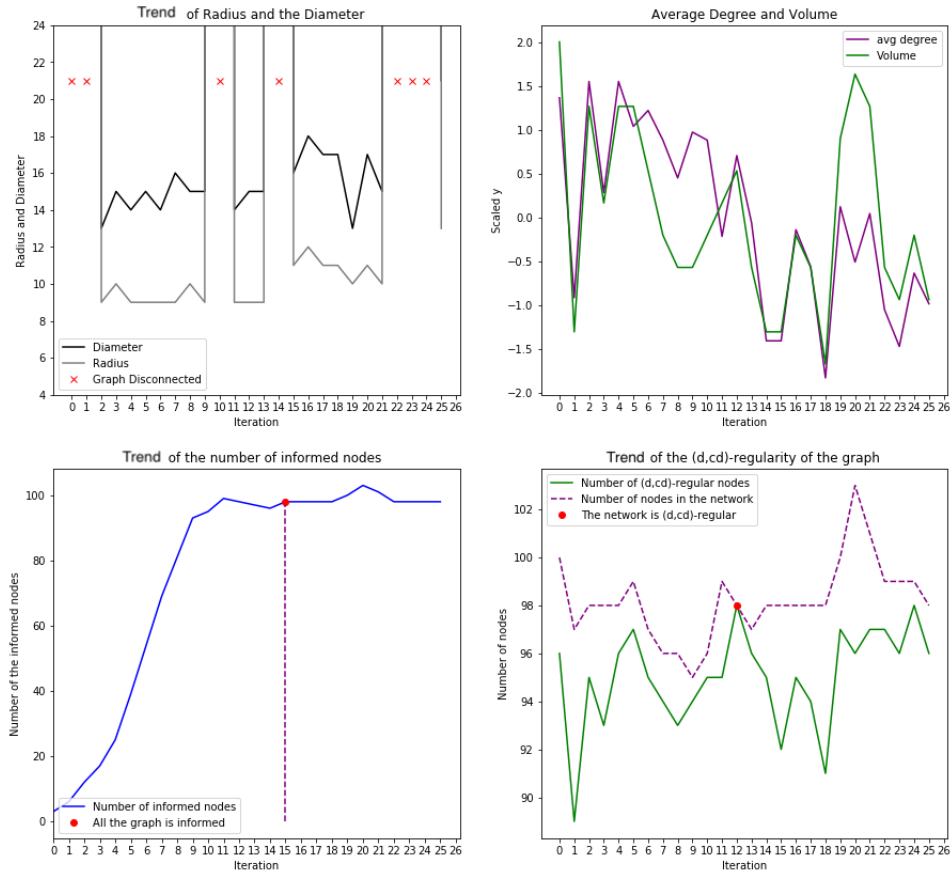


Figure 13: Figura relativa alla simulazione peggiore in termini di tempo di completamento del protocollo di Flooding

Dalla Figura 13 si evince chiaramente che il Flooding è molto lento per via del fatto che il Grafo Dinamico Random è spesso disconnesso e quando invece risulta essere connesso ha un diametro molto grande. Dalla figura in basso a sinistra si può osservare come il numero di nodi informati ad ogni round aumenti molto lentamente. Ancora una volta, la peggiore simulazione è per valori di $d = 2$ e $c = 1.5$.

Plot for d:4 c:3.0

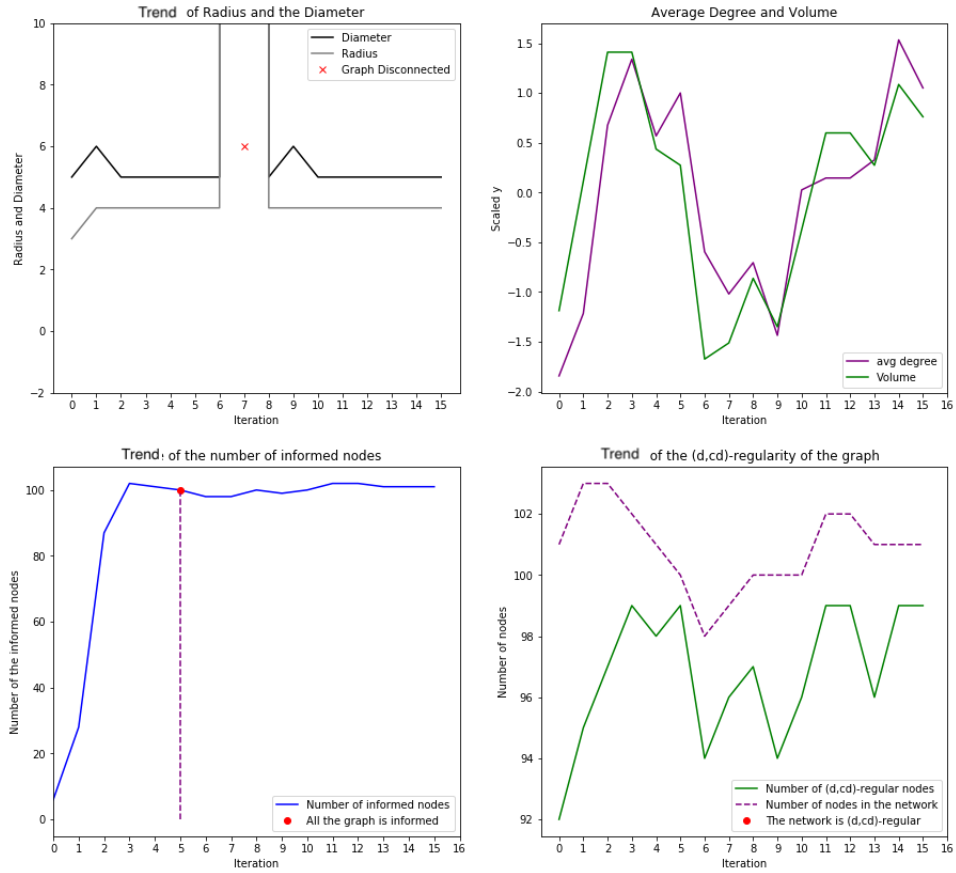


Figure 14: Figura relativa alla simulazione migliore in termini di tempo di completamento del protocollo di Flooding

In Figura 14 si osserva come la dimensione del diametro sia molto importante per ottenere un' esecuzione del protocollo di Flooding ottimale. Infatti, si può notare la relazione tra il diametro e il tempo di completamento. Entrambi sono logaritmici in $|V|$. I grafici che seguiranno mostreranno come l'avere un numero di nodi (d, cd) -regolari molto elevato e un diametro superiore a $\log n$ non migliori

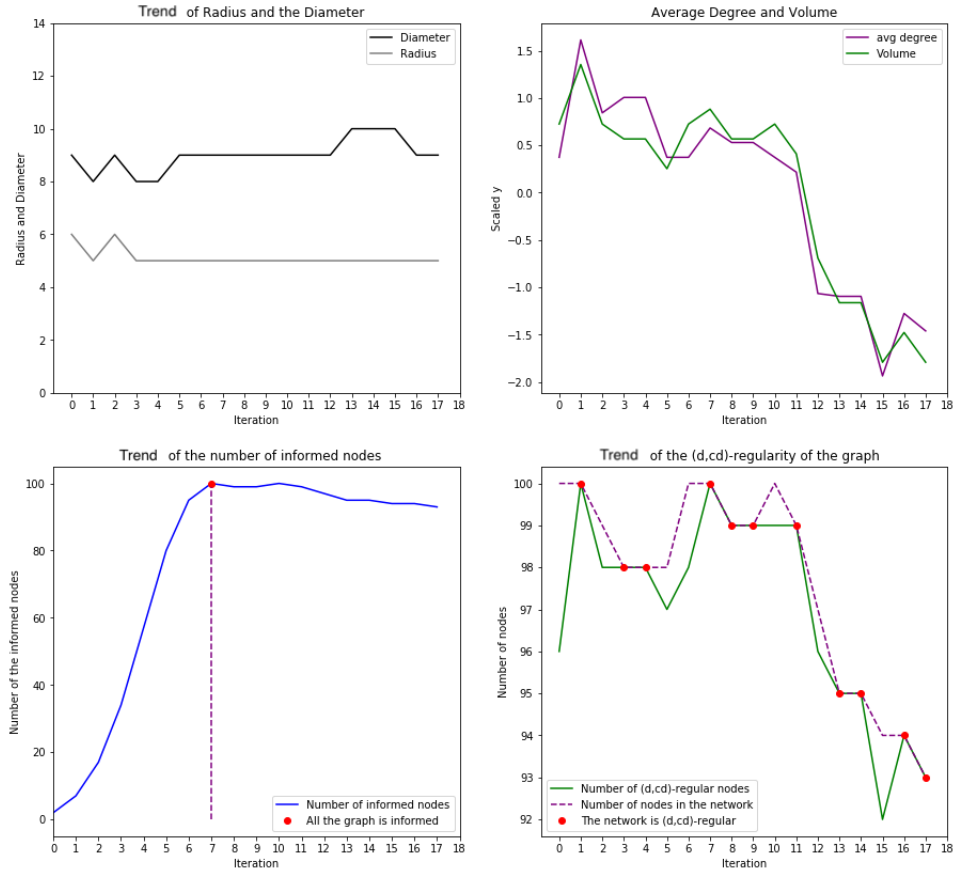


Figure 15: Figura relativa alla simulazione che ha generato il miglior grafo (d, cd) -regolare

Da queste due serie di esperimenti si può osservare come anche per questo modello, con $\lambda = 1$ e $q = 0.01$, il miglior valore per d risulti essere 4 poiché permette di avere un Grafo Dinamico Random con un diametro logaritmico, quasi sempre connesso e con un numero di nodi (d, cd) -regolari che in media non scende mai sotto il 90% .

Successivamente, sono stati fatti variare i valori del parametro di intensità del Processo di Poisson ottenendo dei risultati interessanti per quanto riguarda la velocità di convergenza del modello. Si è potuto osservare, infatti, che un parametro di intensità elevato, a parità dei valori di q , implica un tempo più lungo per raggiungere il 90% di (d, cd) -regolarità. Infatti per esperimenti effettuati con $\lambda = 10$, $q = 0.01$ si ha che il modello, in media, per i diversi valori di d e c , impiega circa 200 istanti di tempo in più rispetto al quello con $\lambda = 1$ ed il medesimo q .

Per quanto riguarda il Flooding time, all'aumentare del parametro di intensità, λ , del processo di Poisson si è ottenuto un deterioramento significativo per le simulazioni dei Grafi con $d = 2$ ed uno molto più lieve per i restanti valori target.

Nelle seguenti tabelle vengono presentati i risultati che mostrano il Flooding time medio e il numero medio di nodi informati ad ogni round.

d	c	$\lambda = 10, q = 0.02$			$\lambda = 10, q = 0.01$			$\lambda = 20, q = 0.02$		
		Average Flooding Time	Std Flooding Time	Average Informed Nodes at Each Time Step	Average Flooding Time	Std Flooding Time	Average Informed Nodes at Each Time Step	Average Flooding Time	Std Flooding Time	Average Informed Nodes at Each Time Step
2	1.5	56	16	7	46	10	17	237	73	4
2	2	25	5	14	23	5	30	50	13	16
2	3	17	3	18	15	3	39	25	6	27
3	1.5	37	13	10	33	8	22	143	51	6
3	2	16	3	19	13	2	44	29	7	24
3	3	12	3	22	11	2	49	18	5	24
4	1.5	18	5	16	15	3	41	41	11	18
4	2	14	4	21	11	2	49	21	5	31
4	3	12	3	22	10	1	51	17	4	35

Table 14: Tabella che riassume i risultati in termini di tempo medio di Flooding e numero medio di nodi informati ad ogni round per gli esperimenti con $\lambda \in \{10, 20\}$.

Dalla Tabella 14 si può vedere che, in media, i modelli con i parametri di $d = 3$, $c \in \{2, 3\}$ e $d = 4$ permettono di ottenere un Flooding time appena superiore a $\log n$. Infatti, in tali simulazioni, per gli istanti di tempo nei quali le configurazioni risultavano connesse, si avevano Grafi con un diametro molto vicino al logaritmico.

Proprietà di Espansione e Protocollo di Flooding

Per quanto riguarda la qualità dell'espansione delle configurazioni, ovvero la misura dei gap spettrali calcolati sulla matrice di transizione della Random Walk definita su una configurazione del modello, sono stati misurati nel seguente modo: una volta che nel grafo sono presenti $\frac{\lambda}{q}$ nodi e circa un 90% di essi è (d, cd) -regolare viene calcolato lo spectral gap.

Dalle misurazioni effettuate si è ottenuto che il massimo gap spettrale che il modello riesce ad assumere per valori di $d = 2, 3$ è molto vicino allo 0 e per i restanti valori di d si ha che assume un valore massimo di 0.3. Si ha quindi che anche in questo caso le configurazioni migliori hanno uno spectral gap molto vicino a quello di un grafo random 4-regolare. Infatti in tali configurazioni si ha anche l'ottimo in termini di tempo di completamento del protocollo di Flooding, di diametro e di numero medio di nodi informato ad ogni round.

In quest'ultima tabella viene mostrato l'andamento, su tutte le m simulazioni, del numero medio di nodi informati ad ogni istante di tempo dal protocollo di Flooding. Si può osservare che per valori di $d = 4$ si ha il maggior numero medio di nodi informati. Infatti dalle altre misurazioni effettuate si ha che per tale valore di d , in media, si hanno le migliori configurazioni in termini di spectral gap e di diametro del grafo.

Average Informed Nodes at Each Time Step						
		$\lambda = 1$		$\lambda = 10$		$\lambda = 20$
d	c	$q = 0.001$	$q = 0.01$	$q = 0.01$	$q = 0.02$	$q = 0.02$
2	1.5	32	4	17	7	4
2	2	40	5	30	14	16
2	3	45	5	39	18	27
3	1.5	47	5	23	10	7
3	2	54	6	44	19	25
3	3	56	6	49	22	35
4	1.5	56	6	41	16	19
4	2	60	6	49	21	31
4	3	60	7	51	22	35

Table 15: Numero medio di nodi informato dal Flooding ad ogni istante di tempo

In conclusione, dalle analisi effettuate si evince che il modello proposto, per $d = 4$ e $c \in \{1.5, 2, 3\}$ tende ad assumere delle configurazioni con un gap spettrale molto vicino a quello dei grafi d -regolari mantenendo un diametro logaritmico, o quasi, e quindi garantendo ad un generico processo di information spreading quale il Flooding di terminare con successo in $\log n$ rounds.

5 Conclusioni

In questo lavoro di tesi sono stati proposti due modelli di Grafi Dinamici Random ispirati al processo di generazione della rete P2P bitcoin. Nello specifico, si è cercato di ampliare il lavoro svolto da Becchetti et al [5] indagando e proponendo due versioni dinamiche del modello da loro proposto.

Per quanto concerne l'analisi effettuata sul primo Grafo Dinamico Random si sono ottenuti dei risultati interessanti per quanto riguarda l'esistenza di una sotto dinamica della Catena di Markov, definita sullo spazio, finito, degli stati, Ω , che rappresenta tutte le possibili configurazioni del modello. Infatti, si è mostrato che tale sotto dinamica \mathcal{P} è irriducibile, aperiodica e persistente positiva suggerendo, quindi, l'esistenza di un'unica misura invariante ν definita come:

$$\nu(x) = \frac{1}{E_x(\tau_x^+)}, \forall x \in \Omega'$$

Dove $\Omega' \subseteq \Omega$ è il sottoinsieme di stati raggiungibili dalla dinamica con densità iniziale $\pi_0(0) = 1$.

Successivamente, in ordine di analizzare tale misura stazionaria, è stato mostrato che la sotto dinamica in analisi \mathcal{P} non gode della proprietà di reversibilità, impedendo quindi di utilizzare l'equazione di bilanciamento dettagliato per calcolarne il valore e quindi dare un bound sul mixing time della sotto catena in analisi.

Questo risultato negativo suggerisce una notevole difficoltà di analisi teorica del mixing time o perlomeno che richiede un approccio elaborato per fornire tale bound.

È stato mostrato inoltre che una volta che il modello raggiunge una configurazione Δ -regolare, con $\Delta \in \Omega(\log n)$, assumendo di avere una probabilità di fallimento degli archi $p = \frac{1}{2}$, con alta probabilità non diventa un grafo disconnesso.

Dalle sperimentazioni effettuate si evince che, per valori di $d = 4$ e $c \in \{2, 3\}$ si ottiene un buon grafo dinamico (d, cd) -regolare. Nello specifico, si è notato che per i suddetti valori di d e c lo spectral gap del grafo è molto vicino a quello di un grafo random d -regolare ottenendo, ad ogni istante di tempo, delle configurazioni di diametro logaritmico e resistenti ai "faults" delle connessioni, in termini di tempo di esecuzione del protocollo di information spreading. Si noti come questa proprietà sia fondamentale nella rete P2P bitcoin.

Inoltre, dagli esperimenti, si è osservato che la dinamica, presenta un drift in una classe di stati che codificano delle configurazioni (d, cd) -regolari, tale proprietà è quella che garantisce una (d, cd) -regolarità, del modello ad ogni istante di tempo.

Successivamente, si è fatto notare come tale Grafo Dinamico Random non simuli perfettamente una rete P2P, in quanto si ha che la componente dinamica è definita sugli archi ma non sull'insieme dei nodi. È stato quindi pensato un secondo modello che presentasse tale proprietà assumendo che gli agenti arrivino seguendo un flusso di Poisson di parametro λ e che escano dalla rete con probabilità q . Anche in questo modello, la scelta del vicinato viene effettuata selezionando d vicini con probabilità uniforme. Per quanto riguarda l'analisi teorica, si è mostrato come i nodi seguano un

processo stocastico $M/G/\infty$, ovvero una coda Markoviana con un flusso di entrata di tipo Poisson, un tempo di servizio definito da una distribuzione di probabilità arbitraria e un numero di sportelli infinito.

Dalle analisi empiriche si è potuto osservare, al variare di λ, q , come la maggior parte dei nodi della rete debba rimanere connessa per un periodo non troppo breve. Questo per poter garantire una buona connettività del grafo e una buona, in termini di numero di round, esecuzione del protocollo di Flooding. Infatti si è notato che per valori di $q \geq 0.1$ le configurazioni del Grafo Dinamico Random, per la maggior parte degli istanti di tempo, risultano essere disconnesse.

Anche per quanto riguarda questo modello si è osservato che i valori ottimi, tra quelli analizzati, di d e c sono, rispettivamente, 4 e $\{2, 3\}$, poiché garantiscono, in media, uno spectral gap delle configurazioni vicino allo 0.3 (vicino quindi a quello di un grafo random 4-regolare) e un tempo di esecuzione logaritmico del protocollo di Flooding.

Il modello, per i suddetti valori di d e c permette all'informazione, che parte da un generico agente, di diffondersi e raggiungere velocemente tutti i nodi della rete.

Questo Grafo Dinamico Random studiato è una rappresentazione più fedele della rete P2P bitcoin rispetto al primo proposto, in quanto presenta entrambe le componenti dinamiche della rete reale ovvero: le connessioni e gli agenti.

Si può quindi concludere che il secondo modello sia quello designato per effettuare ulteriori analisi a livello teorico e sperimentale.

5.1 Sviluppi futuri

Seppur concluso che il primo modello non sia una rappresentazione fedele dello scenario reale, è interessante continuare l'analisi teorica cercando di dare un bound al mixing time, calcolando il drift che fa tendere la catena in una classe di stati $E \subset \Omega' (d, cd)$ -regolari.

Per quanto concerne il secondo modello, sarebbe interessante approfondire a livello teorico l'evoluzione di tale dinamica, che purtroppo non è stata analizzata in questo lavoro di tesi. Un'idea per continuare questo lavoro è quella di modificare il secondo Grafo Dinamico Random sulla scelta del vicinato. Definendo un processo nel quale gli agenti non scelgono a caso un numero fissato di vicini, ma scelgono in modo casuale un numero di vicini secondo una qualche distribuzione di probabilità, e.g.: $\text{Poisson}(\lambda)$, con un parametro di intensità appropriato.

A Appendice

In questa sezione verranno illustrati gli strumenti matematici utilizzati per analizzare i modelli proposti (Appendice A) e le tecniche definite per stabilire la convergenza dei grafi dinamici random (Appendice B).

A.1 Appendice A

A.1.1 Catene Di Markov

Definizione A.1. (Catena Di Markov): Una catena di Markov a stati finiti è un processo stocastico che si muove attraverso gli elementi di un insieme finito Ω seguendo la seguente dinamica: quando ci si trova in un $x \in \Omega$, la prossima posizione viene scelta seguendo una distribuzione di probabilità $\mathcal{P}(x, \cdot)$ fissata. Ovvero, sia $(X_i)_{i \geq 1}$ una sequenza di variabili aleatorie, tale sequenza si dice processo stocastico Markoviano (o catena di Markov) con lo spazio degli stati definito da Ω e matrice di transizione $\mathcal{P} \in \mathbb{R}^{|\Omega| \times |\Omega|}$ se $\forall x, y \in \Omega, t \geq 1, [Z_{t-1} = \bigcap_{i=0}^{t-1} \{X_i = x_i\} : P(Z_{t-1} \cap \{X_t = x\}) > 0]$ vale la seguente proprietà:

$$P(X_{t+1} = y \mid Z_{t-1}, \{X_t = x\}) = P(X_{t+1} = y \mid X_t = x) = (\mathcal{P})_{x,y} = p_{x,y}$$

Tale proprietà è spesso chiamata “perdita di memoria” della catena, o più formalmente: proprietà di Markov. Essa ci dice che muoversi nello stato y condizionata al fatto di trovarsi nello stato x non dipende da dove ci trovavamo prima di essere in tale stato. La sola matrice \mathcal{P} detta matrice di transizione è sufficiente per descrivere la catena. Essa deve essere matrice stocastica, ovvero deve valere:

$$\sum_{i \in \Omega} p_{i,j} = 1 \quad \forall j \in \Omega$$

A.1.2 Classificazione degli stati

Diciamo che uno stato j è raggiungibile da i ($i \rightarrow j$) se $\exists h > 0 : (\mathcal{P}^h)_{i,j} > 0$, ovvero: $i \rightarrow j$ se è possibile muoversi partendo da i e arrivando a j in un numero finito di passi. Chiaramente, se $i \rightarrow k \wedge k \rightarrow j \Rightarrow i \rightarrow j$. Uno stato $i \in \Omega$ si dice **persistente** se $\forall j \in \Omega : i \rightarrow j$ si ha anche che $j \rightarrow i$ e uno stato, invece, si dice **ricorrente** se $\exists j \in \Omega : i \rightarrow j$ ma $j \nrightarrow i$. Inoltre uno stato i si dice **assorbente** se $p_{i,i} = 1$. Seguono due lemmi utili alla classificazione degli stati della catena.

Lemma A.1. Se i è uno stato persistente e $i \rightarrow j$ allora anche j è uno stato persistente.

Lemma A.2. Ogni catena di Markov con spazio degli stati finito ha almeno uno stato persistente.

Dalla classificazione dei singoli stati di una catena di Markov si può procedere con il classificare le classi di stati, ovvero: un sottoinsieme \mathcal{C} di Ω si dice classe chiusa di stati se qualsiasi sia $i \in \mathcal{C}$ si ha che $\sum_{j \in \mathcal{C}} p_{i,j} = 1$. E' chiaro che data una classe chiusa \mathcal{C} , $\forall i \in \mathcal{C}, p_{i,j} = 0, j \in \mathcal{C}^c$. Una classe chiusa si dice **irriducibile** se tutti gli stati al suo interno sono comunicanti. E' chiaro che lo spazio degli stati Ω di una catena di Markov può essere decomposto come $\Omega = \mathcal{T} \cup \mathcal{C}_1 \cup \mathcal{C}_2 \cup \dots$ dove $\mathcal{C}_i, i = 1, 2, \dots$ sono classi disgiunte irriducibili persistenti e \mathcal{T} è l'insieme degli stati transienti.

A.1.3 Norme l_1 e l_2

Sia $\mathbf{x} \in \mathbb{R}^n$ un vettore, allora le norme l_1 ed l_2 sono definite come segue:

- $\|\mathbf{x}\|_1 = \sum_{i=1}^n |x_i|$
- $\|\mathbf{x}\|_2 = (\mathbf{x}^T \mathbf{x})^{\frac{1}{2}} = \sum_{i=1}^n x_i^2$

A.1.4 ϵ -expander

Definizione A.2. Sia $\mathcal{G}(V, E)$ un grafo non orientato e sia $U \subseteq V$ un sottoinsieme di nodi del grafo. Il volume di tale sottoinsieme è definito come $\text{Vol} = \sum_{v \in U} d_v$

Definizione A.3. Un grafo $\mathcal{G}(V, E)$ è un ϵ -expander se, per ogni sottoinsieme $U \subset V$ con $|U| \leq \frac{n}{2}$, la grandezza del taglio (in termini di numero di archi nel taglio) $|C(U, V \setminus U)|$ è almeno $\epsilon \cdot \text{Vol}(U)$.

A.2 Appendice B

A.2.1 Criterio di convergenza del primo modello

Studiando l'andamento del modello si può osservare come lo spectral gap, dopo un numero finito di passi, si stabilizzi, per un lungo periodo, in un determinato range di valori che differiscono fra di loro di al più ϵ .

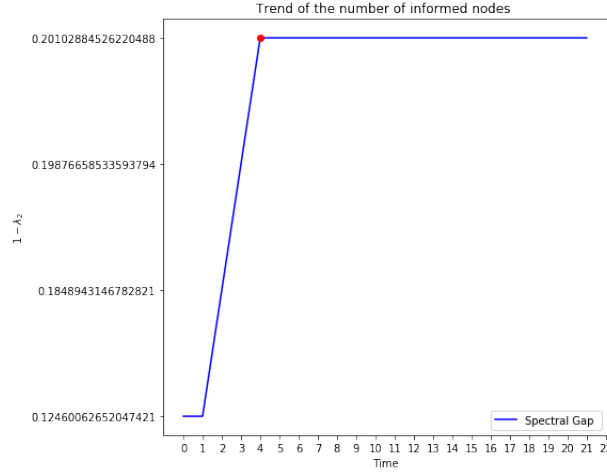


Figure 16: Figura che mostra l'andamento dello spectral gap per una simulazione del modello definito per $|V| = 1024$

Dalla Figura 16 si può notare che $1 - \lambda_2$ dopo quattro istanti di tempo si stabilizza su 0.2. Da tale osservazione segue la definizione di convergenza del modello. Lo scopo è quello di stabilire se un dato modello dopo un determinato numero di istanti di tempo evolve in una specifica topologia e soprattutto se questa configurazione risulta essere stabile negli istanti successivi. Formalmente siamo interessati a verificare se il processo stocastico definito su Ω transisce in una classe di stati $E \subset \Omega$ e per un lungo periodo di tempo rimane in essa. Data tale classe di stati, rappresentante un insieme di configurazioni del grafo dinamico, si vuole, inoltre, studiare la “qualità”, in termini di espansione, della topologia. A tale scopo, ad ogni istante di tempo t , data la matrice di adiacenza del grafo, $A_t = A(\mathcal{G}_t)$ abbiamo misurato il gap spettrale della matrice di transizione della random walk definita su tale configurazione come $P = D_t^{-1} A_t$, dove D_t^{-1} è l'inversa della matrice delle somme delle righe della matrice di adiacenza al tempo t (ovvero l'inversa della matrice diagonale dei gradi del grafo). Si osservi esplicitamente che nel caso in cui la generica configurazione \mathcal{G}_j sia un grafo con un nodo disconnesso, si ottiene una matrice dei gradi D_j non invertibile. Quest'ultima osservazione non crea nessun problema, in quanto un grafo disconnesso non è un expander, quindi, per ogni configurazione disconnessa, senza perdita di generalità, il gap spettrale è stato definito come segue: $1 - \lambda_2 = 0$. In questo setting dinamico, tale gap spettrale oltre a fornire una “descrizione” della qualità dell'espansione delle configurazioni del grafo, permette di ca-

pire di quanto è cambiata la struttura dal generico tempo j al tempo $j + 1$. Segue il concetto di distanza tra due generiche configurazioni del processo in termini di differenza in modulo dei gap spettrali associati: $|(1 - \lambda_2^{(t)}) - (1 - \lambda_2^{(t+1)})|$. Si osservi che tale quantità è direttamente proporzionale alla differenza in termini di qualità dell'espansione tra \mathcal{G}_t e \mathcal{G}_{t+1} . Siano quindi $\sigma_i, \sigma_{i+1}, \dots, \sigma_{i+k}$ i gap spettrali delle configurazioni al tempo $i, i + 1, \dots, i + k$, si può utilizzare il concetto di distanza tra configurazioni in termini di gap spettrali per capire se una sequenza di stati rappresenta delle configurazioni ϵ -distanti tra di loro.

Definizione A.4. *Una sequenza di configurazioni $\mathcal{G}_t, \mathcal{G}_{t+1}, \dots, \mathcal{G}_{t+k}$ con relativi gap spettrali $\sigma_t, \sigma_{t+1}, \dots, \sigma_{t+k}$ delle matrici di transizione $P_i = D_i^{-1} A_i$, $i = t, \dots, t + k$ è ϵ -distante se $|\sigma_i - \sigma_{i+1}| \leq \epsilon$, $i = t, \dots, t + k - 1$*

Definiamo, quindi, una struttura dati di tipo coda Q , ovvero una struttura dati di tipo FIFO (First In First Out), assumendo, inoltre, che essa possa contenere $\log n$ elementi. Possiamo utilizzare Q per capire quando fermare le simulazioni dei processi stocastici nel seguente modo:

Ad ogni istante di tempo t ,

- Se Q non contiene $\log n$ elementi, calcola il gap spettrale del processo al tempo t ed aggiungilo alla coda.
- Se Q contiene $\log n$ elementi e gli elementi di Q non definiscono una sequenza di stati ϵ -Distante, calcola il gap spettrale del processo al tempo t , rimuovi il primo elemento della coda, appendi il gap spettrale calcolato a Q e verifica se la nuova sequenza è ϵ -Distante.
- Se Q contiene $\log n$ elementi ed essi sono ϵ -distanti, allora il processo è andato a convergenza.

A.3 Appendice C

A.3.1 Immagini Modello Edge Dynamic

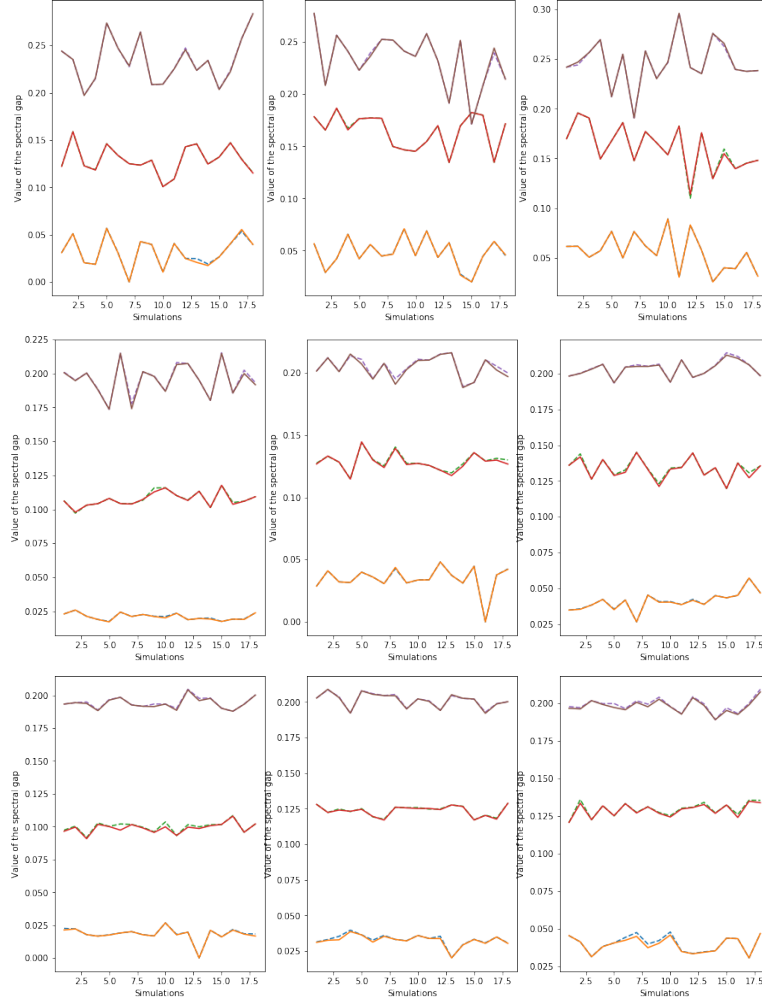


Figure 17: Figura che illustra l'andamento dello spectral gap nelle diverse simulazioni. **Dall'alto verso il basso:** Grafici degli esperimenti per $|V| = 64, 256, 512$. **Da sinistra verso destra:** plotting per i diversi valori di $d = 2, 3, 4$. **Curva blu tratteggiata:** valori del gap spettrale durante la fase 2) del processo. **Curve gialle, rosse e marroni:** rappresentano, rispettivamente, l'andamento dello spectral gap dopo la fase 3) per valori di $c = 1.5, 2, 3$.

References

- [1] S. Nakamoto, “Bitcoin: A peer-to-peer electronic cash system - white paper,” 2008.
- [2] A. Miller, J. Litton, A. Pachulski, N. Gupta, D. Levin, N. Spring, and B. Bhattacharjee, “Discovering bitcoin’s public topology and influential nodes,” *et al*, 2015.
- [3] T. Neudecker, P. Andelfinger, and H. Hartenstein, “Timing analysis for inferring the topology of the bitcoin peer-to-peer network,” in *2016 Intl IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDDCom/IoP/SmartWorld)*, pp. 358–367, IEEE, 2016.
- [4] S. Delgado-Segura, S. Bakshi, C. Pérez-Solà, J. Litton, A. Pachulski, A. Miller, and B. Bhattacharjee, “Txprobe: Discovering bitcoin’s network topology using orphan transactions,” in *International Conference on Financial Cryptography and Data Security*, pp. 550–566, Springer, 2019.
- [5] L. Becchetti, A. Clementi, E. Natale, F. Pasquale, and L. Trevisan, “Finding a bounded-degree expander inside a dense one,” in *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 1320–1336, SIAM, 2020.
- [6] “bitnodes.earn.com.”
- [7] G. Pandurangan, P. Raghavan, and E. Upfal, “Building low-diameter peer-to-peer networks,” *Selected Areas in Communications, IEEE Journal on*, vol. 21, pp. 995 – 1002, 09 2003.
- [8] A. M. Antonopoulos, *Mastering Bitcoin: Unlocking Digital Crypto-Currencies*. O’Reilly Media, Inc., 1st ed., 2014.
- [9] “www.napster.com.”
- [10] “www.bittorrent.com.”
- [11] “www.github.com/aeternity/protocol/blob/master/stratum.md.”
- [12] S. Hoory, N. Linial, and A. Wigderson, “Expander graphs and their applications,” *Bulletin of the American Mathematical Society*, vol. 43, no. 4, pp. 439–561, 2006.
- [13] D. A. Levin and Y. Peres, *Markov chains and mixing times*, vol. 107. American Mathematical Soc., 2017.
- [14] A. Calzolari, “Modelli stocastici a valori discreti,” 2018.

- [15] W. J. Stewart, *Probability, Markov chains, queues, and simulation: the mathematical basis of performance modeling*. Princeton university press, 2009.
- [16] A. Cruciani, “[www.github.com/antonio-cruciani/dynamic-random-graph-generator](https://github.com/antonio-cruciani/dynamic-random-graph-generator),” 2020.
- [17] L. Trevisan., “Lecture notes on expansion, sparsest cut, and spectral graph theory,” 2003.
- [18] N. Santoro, *Design and Analysis of Distributed Algorithms (Wiley Series on Parallel and Distributed Computing)*. USA: Wiley-Interscience, 2006.
- [19] H. C. Tijms, *A first course in stochastic models*. John Wiley and sons, 2003.