



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

TESTOVÁNÍ BEZPEČNOSTI A VÝKONU PROOF-OF-STAKE PROTOKOLŮ POMOCÍ SIMULACE

SECURITY AND PERFORMANCE TESTBED FOR SIMULATION OF PROOF-OF-STAKE PROTOCOLS

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. JURAJ HOLUB

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. IVAN HOMOLIAK, Ph.D.

BRNO 2021

Abstrakt

Do tohoto odstavce bude zapsán výtah (abstrakt) práce v českém (slovenském) jazyce.

Abstract

Do tohoto odstavce bude zapsán výtah (abstrakt) práce v anglickém jazyce.

Klíčové slova

Sem budou zapsána jednotlivá klíčová slova v českém (slovenském) jazyce, oddělená čárkami.

Keywords

Sem budou zapsána jednotlivá klíčová slova v anglickém jazyce, oddělená čárkami.

Citácia

HOLUB, Juraj. *Testování bezpečnosti a výkonu Proof-of-Stake Protokolů pomocí simulace*. Brno, 2021. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Ivan Homoliak, Ph.D.

Testování bezpečnosti a výkonu Proof-of-Stake Protokolů pomocí simulace

Prehlásenie

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana X... Další informace mi poskytli... Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....

Juraj Holub
10. októbra 2021

Podakovanie

V této sekci je možno uvést poděkování vedoucímu práce a těm, kteří poskytli odbornou pomoc (externí zadavatel, konzultant apod.).

Obsah

1	Úvod	3
2	Blockchain	4
2.1	Distribovaná účtovná kniha	4
2.1.1	Vlastnosti blockchainu	4
2.1.2	Aplikačné využitie	5
2.2	Kryptografia v blockchaine	6
2.2.1	Hashovacia funkcia	6
2.2.2	Hash ukazovateľ	6
2.2.3	Digitálny podpis	7
2.2.4	Prahový digitálny podpis	8
2.3	Peer-to-peer sieť	8
2.3.1	Referenčný model	8
2.3.2	Využitie v blockchaine	9
2.4	Datová štruktúra blockchain	10
2.4.1	Blok	10
2.4.2	Transakcia	10
2.4.3	Binárny hashovací strom	11
2.5	Ťažba blokov	11
2.6	Konsenzus	12
2.6.1	Proof-of-Work	13
2.6.2	Proof-of-Stake	13
3	Viacvrstvová abstrakcia	14
4	Útoky na konsenzus	15
4.1	Všeobecné útoky	15
4.1.1	Ovládnutie konsenzu útočníkmi	15
4.1.2	Porušenie synchronného doručovania	15
4.1.3	Útok na časovú synchronizáciu	15
4.1.4	Zdvojnásobenie výdavkov	15
4.1.5	Útok na podskupiny uzlov	16
4.1.6	Útok posledného objaviteľa	16
4.2	Útoky na proof-of-stake	16
4.2.1	Vetvenie bez rizika straty zdrojov	16
4.2.2	Ovplyvnenie volieb	16
4.2.3	Odmietnutie služby lídrovi/výboru	16
4.2.4	Neskoršia korupcia	16

5	Harmony	17
5.1	Konsenzus	17
5.1.1	Protokol	17
5.2	Sharding	18
5.2.1	Rozdelenie hlasovacím podielom	18
5.2.2	Epocha	18
5.2.3	Distribovaná generácia náhodnosti	18
5.2.4	Komunikácie medzi shardami	19
5.3	Model motivácie	19
5.4	Teoretická analýza	19
5.4.1	FBFT konsenzus	19
5.4.2	Sharding	19
5.4.3	Komunikácie medzi shardami	20
6	Solana	21
6.1	Proof-of-history	21
6.2	Konsenzus	21
6.2.1	Voľba proof-of-history generátoru (vodcu)	22
6.2.2	Tower BFT	22
6.2.3	Vetvenie	23
6.3	Model motivácie	23
6.3.1	Vetvenie	23
6.4	Teoretická analýza	23
6.4.1	Proof-of-history	23
7	Prehľad existujúcich simulátorov	24
7.1	SimBlock	24
7.2	Bitcoin Simulator	25
7.3	BlockSim	25
7.4	VIBES	26
7.5	Shadow	26
7.6	FoBSim	27
7.7	Zhodnotenie	27
7.7.1	Porovnanie najvhodnejších nástrojov	27
7.7.2	Výsledná voľba	27
	Literatúra	29

Kapitola 1

Úvod

TODO

Kapitola 2

Blockchain

Táto kapitola vysvetľuje základné koncepty a pojmy spojené z technológiou blockchain, ako aj samotnú dátovú štruktúru blockchain. Sekcia 2.1 vysvetľuje pojmi distribuovaná účtovná kniha a blockchain. Ďalej rozoberá vlastnosti a využitie blockchainu. Sekcia 2.2 vysvetľuje kryptografiu používanú v blockchaine (hashovanie, a asymetrická kryptografia). Sekcia 2.3 popisuje peer-to-peer siete a ich využitie v blockchaine. Nazáver sú v sekcii 2.4 spojené všetky vysvetlené koncepty dokopy a je popísaná samotná dátová štruktúra blockchain.

2.1 Distribuovaná účtovná kniha

Účtovná kniha (anglicky *ledger*) sa v histórii ľudstva dlhodobo používa na záznam rôznych položiek, najčastejšie peňazí a majetku. Príchod digitalizácie a globalizácie presunul tento známy koncept z papierovej podoby do elektronickej. Toto prináša nové výzvy z hľadiska bezpečnosti. *Distribuovaná účtovná kniha* (anglicky *distributed ledger*) je všeobecne technológia, ktorá poskytuje dôveryhodnú a bezpečnú databázu zdieľanú naprieč viacerými inštitúciami, krajinami a to typicky verejne. Najtypickejším odvetvím využitia distribuovanej účtovnej knihy je bankovníctvo. Banka poskytuje centralizovanú autoritu, ktorá zabezpečuje bezpečnú manipuláciu s peniazmi klientov. Tento koncept označujeme ako centralizovaná účtovná kniha. [29]

V roku 2008 bola publikovaná práca [24], ktorá navrhla *decentralizovanú* distribuovanú účtovnú knihu. Práca navrhla koncept elektronickeho platobného systému, ktorého bezpečnosť je založená na kryptografickom dôkaze namiesto dôvere v centralizovanú autoritu. Takáto distribuovaná účtovná kniha sa nazýva **blockchain**.

2.1.1 Vlastnosti blockchainu

Blockchain je dátová štruktúra, ktorá má nasledujúce vlastnosti:

- **Decentralizácia:** Blockchain funguje nad peer-to-peer sieťou, ktorá nepotrebuje centralizovanú dôveryhodnú autoritu.
- **Auditovateľnosť:** Blockchain v sebe nesie celú históriu zmien jeho obsahu a teda každú zmenu stavu dát uložených v blockchaine je možné sledovať.
- **Nemennosť:** Pri správnom použití a dostatočne veľkej sieti nie je možné zmeniť históriu alebo dátový obsah blockchainu.

- **Anonymita:** Užívatelia pracujúci s blockchainom používajú na identifikáciu asymetrickú kryptografiu s digitálnym podpisom. Takýto kryptografický identifikátor neodhaľuje skutočnú identitu užívateľa a pritom umožňuje nepopierateľne určiť vlastníka elektronického zdroja.

Tieto vlastnosti blockchainu sú zabezpečené pomocou peer-to-peer siete na ktorej je blockchain postavený (viď sekcia 2.3) a taktiež pomocou samotnej dátovej štruktúry, ktorá využíva modernú kryptografiu (viď sekcia 2.4). [1]

2.1.2 Aplikačné využitie

Blockchain bol navrhnutý a po prvýkrát implementovaný za účelom poskytnúť elektronickú peňažnú menu nezávislú od centralizovaného bankovníctva. Tento prvý, a najznámejší, blockchain je Bitcoin [24]. Avšak vlastnosti blockchainovej technológie nachádzajú uplatnenie vo veľkom množstve odvetví. Nasledujúci zoznam vymenúva niekoľko aplikácií, ktoré blockchain môže riešiť [18]:

- **Elektronická peňaženka:** Elektronické peňaženky pre obchod s nejakou formou peňazí (typicky v podobe tokenov). Takéto tokeny sú typicky vlastnené pomocou privátneho kľúča, ktorý má uschovaný majiteľ. Majiteľ môže vlastníctvo tokenov presúvať na iné subjekty v danej sieti.
- **Zmenárne:** V dnešnej dobe existuje veľké množstvo elektronických peňažných mien postavených nad blockchainom. Takéto meny všeobecne označujeme ako kryptomeny. Z dôvodu veľkého množstva kryptomien sa prirodzene zvyšuje dopyt po zmenárni medzi jednotlivými kryptomenami. Klasická zmenáreň je riešená tradične centralizovanou autoritou. Avšak blockchain je vhodnou technológiou aj pre decentralizované zmenáreň.
- **Súborové systémy:** V dnešnej dobe už existujú decentralizované súborové systémy založené na peer-to-peer sieťach. Implementácia takéhoto decentralizovaného súborového systému ako blockchain by nám umožnila nepopierateľne a trasovateľne verzovať zmeny v obsahu.
- **Správa identít:** Správa identít je typicky centrálna autorita, ktorá prideluje pre konkrétne entity určité zdroje na ktoré majú právo. Ide o schému podobnú banke. Blockchain by v tomto prípade opäť umožnil náhradu tejto centralizovanej autority za decentralizované siete.
- **Volby:** Elektronické voľby sú ďalším vhodným príkladom, kde sa dá efektívne využiť blockchain. Voliace entity predstavujú decentralizované siete a vlastnosti blockchainu zase poskytujú transparentnosť a verejnú overiteľnosť.
- **Reputačné systémy:** Reputačné systémy slúžia na meranie úrovne dôvery v určité entity. Typickým príkladom je reputácia rôznych predajcov na základe hlasovania zákazníkov. Transparentnosť a nemennosť blockchainovej histórie by znížila možnosť manipulácie s reputáciou v prospech nejakej entity.
- **Aukcie:** Elektronická aukcia je služba veľmi podobná elektronickej peňaženke alebo zmenárni s podobnými bezpečnostnými požiadavkami. Tieto vlastnosti by opäť dokázala pokryť technológia blockchain.

2.2 Kryptografia v blockchaine

Pre pochopenie technológie blockchain je potrebná základná znalosť modernej kryptografie. V tejto sekcii je popísaný kryptografická hashovacia funkcia (pozri 2.2.1) a jej využitie na tvorbu dátových štruktúr zabezpečených proti modifikácii obsahu (viď sekcia 2.2.2). Ďalej je vysvetlený koncept asymetrickej kryptografie a digitálneho podpisu (viď sekcia 2.2.3). Tieto kryptografické primitíva sú základom na ktorom stojí nemennosť, auditovateľnosť a anonymita blockchainu.

2.2.1 Hashovacia funkcia

Hashovacia funkcia je taká funkcia h , ktorá má ako parameter x reťazec bitov ľubovolnej dĺžky a vracia reťazec y s konštantnou dĺžkou (viď rovnica 2.1). Reťazec y voláme hash. Hashovacia funkcia vracia pre konkrétny vstup vždy rovnaký hash.

$$h(x) = y \quad (2.1)$$

Kryptografická hashovacia funkcia, alebo tiež jednocestná funkcia (anglicky *one way function*), je taká hashovacia funkcia pre ktorú platia nasledujúce tri vlastnosti:

1. Pre daný hash x je výpočetne neozvládnuteľné nájsť správu takú, že $h(x) = y$. Anglicky voláme túto vlastnosť *first preimage resistant*.
2. Pre danú správu je výpočetne neozvládnuteľné nájsť inú správu s rovnakým hashom. Anglicky voláme túto vlastnosť *second preimage resistant*.
3. Pre ľubovoľnú správu je výpočetne neozvládnuteľné nájsť inú správu s rovnakým hashom. Anglicky voláme túto vlastnosť *collision resistant*.

Hashovacie funkcie majú v oblasti počítačovej bezpečnosti dôležité využitie:

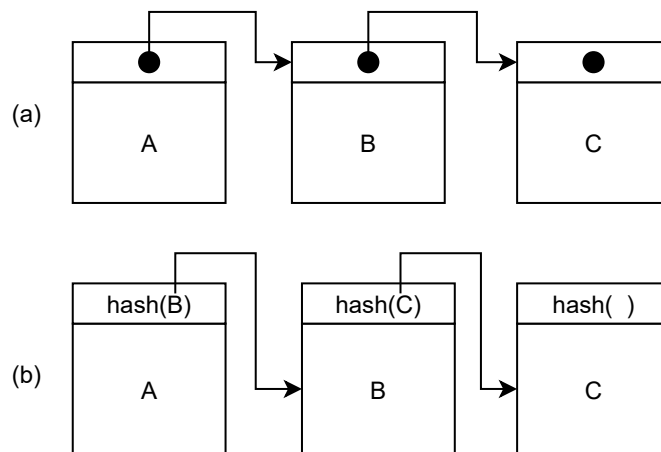
- Bezpečné ukladanie hesiel: Digitálna služba neukladá v databáze heslo, ale len jeho hash. Pri ukradnutí databázy nedochádza k odhaleniu hesiel užívateľov.
- Integrita dát: Hashovacia funkcia môže byť použitá na ochranu integrity ľubovoľných dát. Ak spočítate hash veľkého súboru a bezpečne ho uložíte tak ste schopný detekovať, že niekto tento súbor zmenil.
- Digitálny podpis: Hashovacia funkcia je kryptografické primitívum potrebné pre vytvorenie digitálneho podpisu.

Existuje množstvo hashovacích funkcií. Medzi veľmi známe a používané patrí napríklad MD5 (128 bitový výstup), SHA256 (256 bitový výstup), SHA512 (512 bitový výstup). [22, 31]

2.2.2 Hash ukazovateľ

Hash ukazovateľ (anglicky *hash pointer*) je primitívom pre tvorbu dátových štruktúr s kryptografickým zabezpečením proti manipulácii s obsahom (anglicky *tamper-evident*). Hash ukazovateľ funguje ako klasický ukazovateľ v zozname či strome. Navyše však neumožňuje meniť už pridané prvky. Jediná povolená operácia je prídanie ďalšieho prvku do dátovej štruktúry.

Obrázok 2.1 demonštruje rozdiel medzi zoznamom vytvoreným pomocou klasických ukazovateľov a pomocou hash ukazovateľov. Bežný zoznam umožňuje pozmeniť ľubovoľný už



Obr. 2.1: (a) Zoznam pomocou ukazovateľov (b) Zoznam pomocou hash ukazovateľov

existujúci prvok nezávisle na zvyšku zoznamu. Naopak, hash pointer referencuje pomocou samotného dátového obsahu. Ak by sme zmenili dátový obsah prvku B, tak by sa narušila referencia v predchádzajúcom prvku. [1, 25]

2.2.3 Digitálny podpis

Digitálny podpis (anglicky *digital signature*) je kryptografický koncept používaný na autentifikáciu, autorizáciu a nepopierateľnosť. Digitálny podpis jednoznačne prepojí určitú entitu s informáciou. V technológii blockchain slúži digitálny podpis na určenie vlastníctva zdrojov, ktoré blockchain uchováva. [22, 24]

Moderná kryptografia používa pre zaistenie dôvernosti šifrovanie pomocou tajného kľúča. Pre zašifrovanie a dešifrovanie tajnej správy je potrebná znalosť tajného kľúča. Tento mechanizmus zaistuje dôvernosť avšak nezaistuje nepopierateľnosť pretože obe komunikujúce strany poznajú tajný kľúč a teda nie je možné právne dokázať kto správu napísal. Na zaistenie nepopierateľnosti sa používa asymetrické šifrovanie, ktoré používa dvojicu kľúčov:

- **Privátny kľúč** je tajný a pozná ho len odosielateľ správy. Odosielateľ používa tento kľúč na zašifrovanie správy.
- **Verejný kľúč** je dostupný komukoľvek. Ktokoľvek s týmto kľúčom dokáže dešifrovať správu.

Tieto dva kľúče tvoria dvojicu prepojenú matematickým spôsobom. Zo znalosti verejného kľúča je výpočetne nezvládnuteľné zistiť privátny kľúč. Zašifrovaná správa nie je dôverná pretože ktokoľvek môže použiť verejný kľúč na jej dešifrovanie. Avšak zašifrovaná správa je nepopierateľne napísaná vlastníkom privátneho kľúča.

Tento koncept je základom digitálneho podpisu. Ak chceme nepopierateľne dokázať, že nejaký dátový obsah (napríklad pdf dokument) sme vytvorili mi, tak vypočítame jeho hash (viď sekcia 2.2.1) a zašifrujeme ho naším privátnym kľúčom. Zašifrovaný hash priložíme k dokumentu. Prijemca dokumentu si následne pomocou verejného kľúča dešifruje hash priložený k správe a porovná si ho s tým ktorý vypočítal sám z danej správy. Ak sú hashe rovnaké tak nikto správu nezmenil a dokument je jednoznačne vytvorený vlastníkom tajného kľúča. Najznámejšie algoritmy na digitálny podpis sú RSA, DSA, ECDSA. [22]

2.2.4 Prahový digitálny podpis

Prahový digitálny podpis (anglicky *threshold signatures*) je špeciálna schéma digitálneho podpisu. V tejto schéme je n účastníkov a každý vlastní časť privátneho kľúča. Každý účastník môže použiť svoju časť tajného kľúča na čiastočné podpísanie správy M . Kompletný podpis môže byť zostrojený ak aspoň t účastníkov poskytlo svoju časť podpisu. Potom hovoríme o t -z- n prahovom podpise. Takýto koncept sa dá využiť pri kryptografickom hlasovaní. [7]

2.3 Peer-to-peer sieť

Technológia blockchain je postavená na peer-to-peer sieťach. Peer-to-peer sieť sa podieľa na decentralizovanosti, nemennosti a auditovateľnosti blockchainu.

Peer-to-peer sieť je dynamický súbor nezávislých uzlov (anglicky *peers*), ktoré sú prepojené do grafu. Každý uzol obsahuje zdroje, ktoré zdieľa všetkým ostatným uzlom v sieti. [10, 30] Dôvod existencie peer-to-peer sietí je teda decentralizovaný spôsob zdieľania zdrojov ako sú súbory, fyzické zariadenia, výpočtový výkon alebo aj elektronické finančné zdroje. Dnes existuje množstvo peer-to-peer sietí. Veľmi známe sú napríklad Gnutella, Kazaa alebo BitTorrent. [2]

2.3.1 Referenčný model

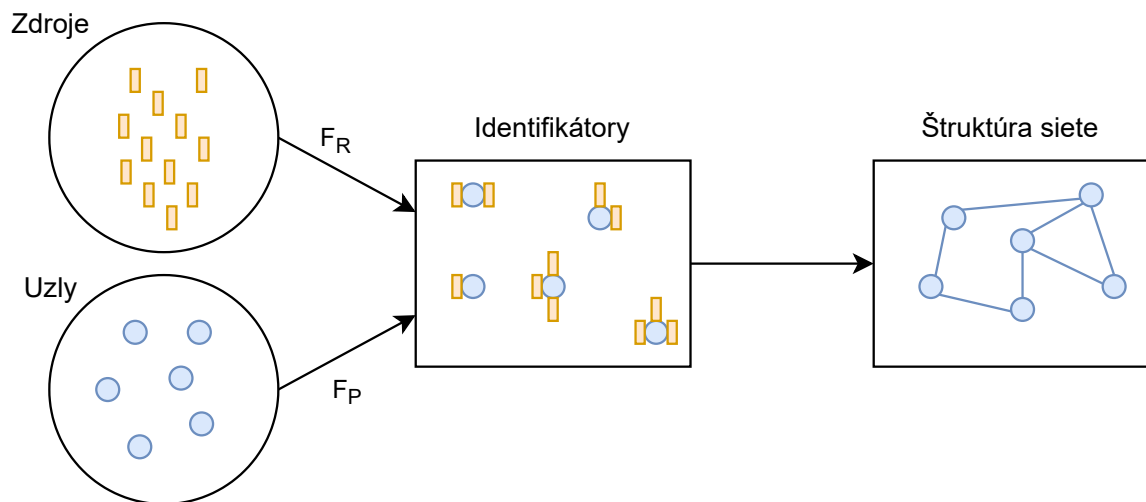
Najbežnejšie technické riešenie peer-to-peer siete je navrstvenie siete (anglicky *overlay network*) na už existujúcu sieť, ktorou je typicky Internet. Takúto sieť potom môžeme definovať ako päťicu (P, R, I, F_P, F_R) , kde:

- P je množina uzlov
- R je množinu zdrojov
- I je priestor identifikátorov
- $F_P : P \rightarrow I$ je funkcia, ktorá mapuje uzoly na identifikátory
- $F_R : R \rightarrow I$ je funkcia, ktorá mapuje zdroje na identifikátory

Obrázok 2.2 ukazuje princíp fungovania takto definovanej siete. Tvorba siete s týmto modelom je potom závislá od šiestich návrhových aspektov:

1. Voľba priestoru identifikátorov.
2. Mapovanie zdrojov a uzlov na identifikátory.
3. Správa priestoru identifikátorov v réžii uzlov siete.
4. Tvorba grafu (štruktúra siete).
5. Stratégia smerovania (anglicky *routing*).
6. Stratégia údržby.

Konkrétne riešenie pre popisovaných šesť aspektov je závislé od požiadaviek na efektivitu, škálovateľnosť, samoorganizovateľnosť, odolnosť voči chybám a kooperáciu. [2]



Obr. 2.2: Referenčný model peer-to-peer siete. [2]

2.3.2 Využitie v blockchaine

Peer-to-peer sieť umožňuje blockchainu uchovávať jeho obsah decentralizovane a pritom bezpečne. Tento koncept si vysvetlíme na prípade blockchainu, ktorý sa využíva ako kryptomena.

Elektronické financie sú typicky reprezentované pomocou elektronických mincí. Takáto minca je reprezentovaná pomocou nejakej sekvencie bitov. Avšak narozdiel od fyzických mincí, elektronické mince umožňujú jednoduchú falzifikáciu. Útočník skopíruje bitový reťazec danej mince a zaplatí ním viacnásobne rôzne produkty. Tento útok sa volá zdvojnásobenia výdavkov (anglicky *double-spending attack*). Proti tomuto útoku existuje tradičné zabezpečenie pomocou centrálnej autority. Banka je centrálna autorita, ktorá schvaľuje všetky manipulácie s elektronickými mincami a teda neumožní použiť mincu takýmto podvodným spôsobom. Avšak toto riešenie nie je možné použiť v decentralizovanej sieti, kde centrálna autorita neexistuje. V prípade decentralizovanej siete je možné zabrániť tomuto útoku pomocou použitia dátovej štruktúry blockchain. [17]

Kryptomena Bitcoin ako prvá navrhla použitie peer-to-peer siete v spojení s blockchain technológiou pre zabránenie double-spending útoku. V takejto sieti je jediný zdroj na zdieľanie a to je dátová štruktúra blockchain v ktorej sú uložené všetky informácie o elektronických financiách. Zjednodušene môžeme povedať, že majorita uzlov siete zdieľa rovnaký zdroj (rovnakú kópiu blockchainu). Ak chce niektorý uzol vykonať finančnú transakciu tak zašle správu s navrhovanou zmenou blockchainu do siete. Uzly v tejto sieti nie je potrebné identifikovať pretože správy posielané v tejto sieti nie sú smerované na žiadne konkrétne miesto. Keď uzol prijme správu s nejakou modifikáciou tak si overí či ide o validnú požiadavku na finančnú transakciu. Štruktúra blockchainu používa modernú kryptografiu na overenie validnosti transakcie (pozri sekciu 2.2). Blockchain, ktorý vlastní väčšina siete je ten, ktorý sa považuje za pravdu. Útočník by musel teda vlastniť aspoň 51 % uzolov v sieti aby mohol vykonať double-spending útok. Ak je daná sieť dostatočne veľká tak by toho útočník nemal byť schopný dosiahnuť. [24]

2.4 Datová štruktúra blockchain

Blockchain je dátová štruktúra podobná zoznamu (anglicky *linked list*). Blockchain organizuje dáta do podmnožín, ktoré sa volajú bloky. Blok je podobný uzlu v zozname. Každý blok obsahuje referenciu na ďalší blok. Rozdiel medzi zoznamom a blockchainom je v tom, že referencia blockchainu je zabezpečená proti manipulácii (anglicky *tamper-evident*) pomocou modernej kryptografie. Bežný zoznam používa referenciu pomocou ukazovateľov (anglicky *pointers*), ktoré môže ktokoľvek a kedykoľvek pozmeniť bez toho aby pozmenil dátový obsah. Naopak, blockchain vôbec neumožňuje meniť už pridané bloky. Jediná povolená operácia je pridanie ďalšieho bloku na koniec blockchainu. [1]

Každý blok obsahuje dáta, ktoré sú typicky vo forme transakcií. Kryptograficky bezpečný blockchain by mohol fungovať aj tak, že v každom bloku bude uložená práve jedna transakcia. Z dôvodu optimalizácie je ale v jednom bloku uložené množstvo transakcií. Vďaka tejto optimalizácii nemusí celá sieť vytvárať konsenzus po každej transakcii. Samotné transakcie v rámci jedného bloku sú ukladané v ďalšej dátovej štruktúre, ktorá taktiež používa kryptografické hashovanie (viď sekcia 2.4.3). [25]

2.4.1 Blok

Blok sa skladá z hlavičky a tela. Telo bloku obsahuje dáta a hlavička obsahuje metadáta. Dáta v tele bloku sú uložené vo forme transakcií. Transakcie sú popísané v sekcii 2.4.2. Počet transakcií v bloku je typicky obmedzený maximálnou veľkosťou bloku. Hlavička bloku obsahuje metadáta o bloku, kde najdôležitejšie a najbežnejšie sú nasledujúce:

- Hash všetkých transakcií.
- Časové razítko vytvorenia bloku.
- Hash ukazovateľ na predošlý blok v blockchaine.
- Náhodná výzva (anglicky *nonce*), ktorej využitie vysvetľuje sekcia 2.6.1

[36]

2.4.2 Transakcia

Transakcia je elementárna dátová jednotka na ukladanie dáta v blockchaine. Bitcoin, prvý blockchain, použil transakciu na manipuláciu s elektronickými financiami. Takáto transakcia sa skladá z troch častí:

- **Množina vstupov:** Každý vstup má uložený hash predošlej transakcie s ktorej vychádza. Ďalej definuje, ktoré výstupy s predošlej transakcie si nárokuje. Nakoniec obsahuje digitálny podpis, ktorý autorizuje tvorcu transakcie.
- **Množina výstupov:** Každý výstup má hodnotu, ktorá je uchovávaná v blockchaine (typicky minca nejakej kryptomeny). Suma hodnôt všetkých výstupov transakcie musí byť menšia alebo rovná sume všetkých vstupov transakcie. Ak je menšia, tak tento rozdiel je použitý ako odmena pre toho, kto publikoval tento blok blockchainu.
- **Hlavička:** Obsahuje hash transakcie, ktorý je používaný ako unikátny identifikátor pomocou, ktorého sa na transakciu odkazujeme.

2.4.3 Binárny hashovací strom

Binárny hashovací strom alebo tiež Merkle strom (anglicky *Merkle tree*) je datová štruktúra podobná binárnemu stromu, ktorá slúži na efektívne a rýchle vypočítanie hashu veľkého množstva dát. Blockchain používa tento strom na časovo efektívny výpočet hashu všetkých transakcií. Takto vypočítaný hash je uložený v hlavičke bloku.

Merkle strom je vyvážený binarný strom, kde listové uzly obsahujú jednotlivé transakcie uložené v danom bloku blockchainu. Každý nelistový uzol stromu obsahuje hash vypočítaný z jeho potomkov. Koreňový uzol teda obsahuje hash celého stromu a teda aj všetkých transakcií. Pridanie, odobranie, zmena obsahu, alebo zmena poradia transakcií bude teda viesť k zmene koreňového hashu. Konštrukcia stromu, inak povedané výpočet hashu všetkých transakcií, prebieha nasledovne:

1. Všetky transakcie sú uložené do listovej úrovne stromu. Ak je počet transakcií nepárny tak, je posledná vložená dvakrát.
2. Nad každým listovým uzlom je vypočítaný hash.
3. Každý nelistový uzol skonkatenuje hash ľavého a pravého syna, vypočíta nad nimi hash a uloží si ho.

Konštrukcia takéhoto stromu pre n transakcií má časovú zložitosť $O(\log(n))$. Takýto spôsob výpočtu hashu je teda veľmi efektívny pre veľké množstvo transakcií (blok v blockchaine bežne obsahuje stovky transakcií). [9]

Merkle strom umožňuje efektívne šetriť pamäťové nároky blockchainu. Do blockchainu sú neustále pridávané nové bloky, ktoré obsahujú aj rovnaké staré transakcie. Ak už sú transakcie zaznamenané v dostatočne veľkom množstve blokov tak sú z hľadiska bezpečnosti nemenné. V nových blokoch ich už preto nie je potrebné ukladať. Nový blok si preto uloží len hashe starých vetiev stromu, ale ich obsah už nepotrebuje. Takto je zachovaná integrita hashu všetkých transakcií. [24]

2.5 Ťažba blokov

Ťažba (anglicky *minning*) bloku je proces pridania nového bloku na koniec blockchainu. Ťažba bloku zahŕňa validáciu transakcií a blokov. Preto je ťažba kritická pre správne a bezpečné fungovanie blockchainu. Každý uzol siete, ktorý ťaží nové bloky sa nazýva anglickým slovom *miner*. Tieto uzly umožňujú rozširovanie blockchainu. Aby takéto uzly existovali, musia byť motivované. Miner dostane za každý vyťažený blok ako odmenu zdroje uložené v blockchaine.

Ťažba všeobecne pozostáva z nasledujúcich krokov:

1. Miner prijíma požiadavky na transakcie z peer-to-peer siete. Každú transakciu si validuje pomocou kryptografie popísanej v sekcii 2.2.
2. Miner musí taktiež udržiavať aktuálny stav blockchainu. Je potrebné sledovať či nevznikli nové bloky a udržiavať si validný blockchain.
3. Ak miner vlastní validnú a aktuálnu kópiu blockchainu, môže začať vytvárať nový blok. Do nového bloku vloží transakcie, ktoré prijal a boli validné.

4. Novo vytvorený blok je potrebné distribuovať do siete. Ak väčšina siete blok získa a akceptuje, tak bol pridaný do blockchainu. Tento proces zahŕňa problém konsenzu v sieti, ktorý je podrobne vysvetlený v sekcii 2.6.
5. Ak sa podarilo úspešne blok pridať do blockchainu tak miner získava odmenu. Odmena za vyťažený blok je konštantná čiastka zdrojov poskytovaných daným blockchainom. Napríklad Bitcon poskytuje v roku 2021 ako odmenu 6,25 bitcoinov čo približne 300 dolárov ¹. Avšak táto odmena môže byť navýšená o poplatky, ktoré sú v transakciách. Ak teda chcete aby sa vaša transakcia dostala do blockchainu čo najrýchlejšie, poskytnete vyššiu odmenu v podobe poplatku za transakciu. Miner bude potom viac motivovaný pridať práve túto transakciu do bloku.

[25]

2.6 Konsenzus

Konsenzus v blockchaine zabezpečuje, že skupina uzlov (peerov) sa zhodne na rovnakom stave blockchainu. Tradične je konsenzus zabezpečený centrálnou autoritou s ktorou musia byť všetky uzly spojené. Avšak blockchain je decentralizovaný a teda toto riešenie nie je možné. Konsenzus v decentralizovanej sieti blockchainu je zabezpečený pomocou protokolu, ktorý sa snaží nájsť kompromis medzi nasledujúcimi vlastnosťami [16, 35, 21]:

- **Konzistentnosť:** Vždy keď dôjde k potvrdeniu zmeny, celý reťazec sa aktualizuje a všetci čítajú rovnaké hodnoty.
- **Dostupnosť:** Pre každú požiadavku na dáta musí byť poskytnutá odpoveď.
- **Odolnosť voči prerušeniu** (anglicky *partial tolerance*): Sieť funguje aj v prípade, že v nej vznikajú chyby.

Existujú tri najbežnejšie techniky, ktoré sa používajú pre ustavenie konsenzu [18]:

- **Lotéria:** Takéto protokoly náhodne zvolia uzol, ktorý vyprodukuje nový blok. Výhodou tohoto prístupu je jeho jednoduchosť keďže takýto proces nevyžaduje žiadnu interaktivitu. Nevýhodou tohoto prístupu je, že pripúšťa možnosť voľby viacerých uzlov súčasne. V takom prípade sa reťazec rozvetví (anglicky *fork*) a je potrebné určiť ktorá vetva je správna. Typicky sa za správnu vetvu volí tá najdlhšia. Avšak takéto správanie oslabuje konzistentnosť blockchainu. Transakcie v posledných blokoch môžu byť potenciálne zahodené pretože nejde o správnu vetvu. Preto sa za konzistentné transakcie považujú až tie ktoré sú prekryté väčším množstvom nových blokov.
- **Hlasovanie:** Protokoly založené na hlasovaní dosahujú dohodu pomocou hlasovania všetkých zapojených uzlov. Môžeme použiť napríklad protokol Byzantskej chyby (anglicky *Byzant fault tolerance*), ktorý vyžaduje majoritu hlasov k uzavretiu konsenzu (typicky $\frac{2}{3}$). Výhodou je veľmi malá pravdepodobnosť vzniku vetiev reťazca. Na druhej strane, takéto protokoly majú nižšiu priepustnosť, ktorá klesá z narastajúcim počtom uzlov.
- **Kombinovaný prístup:** Tieto protokoly sa snažia kombinovať prístup lotérie a hlasovania s cieľom dosiahnuť výhody oboch prístupov. Napríklad je možné rozdeliť počet uzlov podieľajúcich sa na hlasovaní pomocou lotérie, čím sa zvýši priepustnosť.

¹<https://www.investopedia.com/tech/how-does-bitcoin-mining-work/>

[35, 18]

2.6.1 Proof-of-Work

Dôkaz prácou (anglicky *proof of work*) je najbežnejšia stratégia konsenzus protokolu. Ak chce uzol publikovať nový blok, musí investovať svoj výpočtový výkon do riešenia netriviálneho kryptografického problému. Uzol, ktorý ako prvý vyrieši tento problém má najväčšiu pravdepodobnosť, že bude jeho blok pridaný do reťazca. Samozrejme, je tu možnosť, že problém vyrieši súčasne viacero uzlov. Konečná voľba je teda náhodná. Proof-of-work teda umožňuje, aj keď s oveľa menšou pravdepodobnosťou, že sa reťazec rozvetví. Bezpečnosť takéhoto konsenzu spočíva v tom, že majorita výpočtového výkonu siete (51 %) je vlastnená poctivými uzlami. [21]

Samotný kryptografický problém, ktorý sa rieši spočíva v počítaní hashu (viď 2.2.1) z hlavičky nového bloku (viď 2.4.1). Hlavička obsahuje atribút nonce, ktorý môže miner ľubovoľne nastaviť. Zmenou tohoto atribútu môže miner získať iný hash hlavičky bloku. Konsenzus vyžaduje aby výsledná hash hodnota bola menšia rovná určitej zvolenej hodnote. Miner môže túto podmienku dosiahnuť len tak, že bude inkrementovať hodnotu atribútu nonce až dokedy túto podmienku nesplní. Táto úloha sa teda dá riešiť len pomocou metódy útok hrubou silou (anglicky *brute force*). Miner môže svoju šancu na úspech zvýšiť len tým, že poskytne väčší výpočtový výkon do jej riešenia. Na druhej strane, ostatné uzly môžu overiť, že jeho riešenia je správne veľmi rýchlo a efektívne. [36]

Proof-of-work konsenzus využíva dva typy uzlov. Prvý typ uzla je miner, ktorý vytvára nové bloky tak ako je popísané v sekcii 2.5. Druhý typ uzla je bežný vlastník zdrojov v danom blockchaine, ktorý môže vytvárať transakcie a distribuovať ich do siete. Druhý typ uzla teda nehrá žiadnu rolu v ustanovovaní konsenzu. [21]

Proof-of-work je overený konsenzus protokol, ktorý funguje a používa sa v blockchainoch ako je Bitcoin [24] alebo Ethereum ². Tento protokol má však jeden dlhodobý problém a to je spotreba energie. Uzly ktoré riešia kryptografický problém pre nové bloky spotrebujú veľké množstvo energie čo má nepriaznivý dopad na životné prostredie. Niektoré zdroje ³ napríklad hovoria, že v roku 2021 pokrýva ťažba Bitcoinu 0,5 % celkovej spotreby elektrickej energie na svete. Pre porovnanie, ide o sedemkrát väčšiu spotrebu energie ako má celá spoločnosť Google. [21]

2.6.2 Proof-of-Stake

Dôkaz podielom na vlastníctve (anglicky *proof of stake*) je založený na technike lotérie, kde pravdepodobnosť výhry rastie s množstvom už vlastnených zdrojov. Základnou myšlienkou, je že vlastník veľkého množstva zdrojov v danom blockchaine je veľmi nepravdepodobným útočníkom pretože svoje zdroje nechce ohroziť. Uzly sa teda musia preukázať vlastníctvom zdrojov v danom blockchaine ak chcú publikovať nový blok. Pravdepodobnosť výberu uzlu rastie s množstvom zdrojov, ktoré v sieti vlastní. [18, 26]

Veľkou výhodou proof-of-stake oproti proof-of-work je, že nevyždauje také veľké množstvo energie. Uzly už nemusia súťažiť v riešení výpočtovo náročných úloh. [21]

²<https://ethereum.org/en/whitepaper/>

³<https://www.businessinsider.com/bitcoin-mining-electricity-usage-more-than-google-2021-9>

Kapitola 3

Viacvrstvová abstrakcia

Existuje veľké množstvo blockchain protokolov, ktoré majú rôzne využitie a implementáciu. Avšak všetky tieto implementácie sú založené na spoločnom koncepte distribuovanej účtovnej knihy. Pre ich jednotnú klasifikáciu použijeme nasledujúci model abstrakcie so štyrmi vrstvami: [18]

1. **Sieťová vrstva** predstavuje najnižšiu vrstvu abstrakcie a zaoberá sa peer-to-peer sieťou. Sieť rieši pripájanie nových peerov a komunikácia medzi uzlami v sieti (šírenie transakcií a blokov). Táto vrstva má kritický dosah na výkonnosť blockchainu. Napríklad, verejný blockchain ako je Bitcoin tvorí veľmi rozsiahlu sieť s tisíckami aktívnych uzlov. V takejto sieti sa už vlastnosti ako stratovosť paketov alebo priepustnosť nezanedbateľne prejaví na rýchlosti a stabilite celého blockchainu. [13]
2. **Konsenzus vrstva** definuje protokol pomocou, ktorého sa ustavuje dohoda na stave blockchainu. Táto vrstva kľúčovo ovplyvňuje priepustnosť transakcií (*TPS* - *transaction per second*). Táto metrika je kľúčová napríklad v oblasti kryptomien. Pre porovnanie, centralizovaný platobný systém VISA ¹ má TPS približne 1 500 zatiaľ čo Bitcoin približne 5.
3. **Dátová vrstva** (alebo tiež úložisko) definuje model transakcií (binárny hashovací strom, hashovacie a kryptografické algoritmy).
4. **Aplikačná vrstva** definuje využitie v konkrétnej službe (napríklad kryptomena).

¹<https://towardsdatascience.com/the-blockchain-scalability-problem-the-race-for-visa-like-transaction-speed-5cce48f9d44>

Kapitola 4

Útoky na konsenzus

4.1 Všeobecné útoky

4.1.1 Ovládnutie konsenzu útočníkmi

Tieto útoky narušia decentralizované siete tým, že útočníci dokážu utvoriť konsenzus. V takom prípade sa stáva sieť centralizovaná, kde centrálnou autoritou sú práve útočníci. Príkladom takéhoto útoku pre proof-of-work a proof-of-stake konsenzus je ovládnutie 51 % siete. V prípade protokolov Byzantskej chyby dokáže $\frac{1}{3}$ uzlov spôsobiť, že bude protokol narušený alebo dokonca zastavený. [18]

4.1.2 Porušenie synchronného doručovania

Ak útočník dokáže narušiť synchronne doručovanie správ v protokole, ktorý synchronizáciu predpokladá tak takýto protokol prestane fungovať. Tento útok už nie je možné urobiť na protokole, ktorý umožňuje asynchronnú komunikáciu. Tento útok je možné vykonať napríklad pre protokoly Byzantskej chyby. [18]

4.1.3 Útok na časovú synchronizáciu

Okrem času celého systému majú jednotlivé uzly v sieti vlastný čas. Uzly ho vypočítavajú ako medián všetkých časov získaných od ostatných. Ak je uzol miner tak typicky vloží práve tento čas do hlavičky bloku, ktorý vytvoril. Ostatné uzly v sieti budú pri distribúcii bloku overovať, že čas je dostatočne čerstvý aby bol akceptovaný. Ak útočník disponuje veľkým množstvom uzlov v sieti, tak môže narušiť synchronizáciu času, ktorý bloky získavajú mediánom. Tento útok následne spomaľuje sieť pretože bloky distribuujú bloky s časovou značkou, ktorá už nebude akceptovaná.

4.1.4 Zdvojnásobenie výdavkov

Zdvojnásobenie výdavkov (anglicky *double spending*) je útok, ktorý vzniká vytvorením dvoch alebo viac konfliktných blokov. Tieto bloky vytvárajú tzv. vetvy (anglicky *forks*). S tohoto dôvodu môžu byť niektoré krypto mince dočasne minuté v oboch konfliktných blokoch. Neskôr je síce len jeden z nich validný pretože druhá vetva bude zahodená. Avšak tento útok spomaľuje konzistentnosť blockchainu.

4.1.5 Útok na podskupiny uzlov

Niektoré konsenzus protokoly rozdeľujú celú sieť na podskupiny uzlov (anglicky *shards*). Sharding zvyšuje škálovateľnosť a priepustnosť siete pretože uzly validujú transakcie len v rámci svojej podskupiny. Na druhej strane, tento prístup môže viesť k zníženiu bezpečnosti. Množstvo spolupracujúcich uzlov v jednej takejto podskupine je oveľa menší než v celej sieti. Pre útočníka môže byť preto jednoduchšie ovládnuť takúto podskupinu ako celú sieť.

4.1.6 Útok posledného objaviteľa

last-revealer attack [12]

4.2 Útoky na proof-of-stake

4.2.1 Vetvenie bez rizika straty zdrojov

Generovanie blokov v proof-of-stake nestojí žiadnu energiu v podobe výpočtového výkonu. Uzly preto môžu generovať viacero konfliktných uzlov súčasne a tým zvyšovať pravdepodobnosť, že bude práve ich uzol pridaný do reťazca. Takéto správanie nepredstavuje pre daný uzol žiaden risk v podobe straty zdrojov. Problémom tohoto správania je, že vzniká väčšie množstvo vetiev reťazca. Ako dôsledok sa potom zvyšuje čas do konzistentnosti blockchainu.

4.2.2 Ovplyvnenie volieb

Útočník ovplyvňuje voľby v jeho prospech. Takto zvyšuje pravdepodobnosť, že bude zvolený práve jeho blok.

4.2.3 Odmietnutie služby lídrovi/výboru

Tomuto vobec nerozumiem TODO - najst vhodno literaturu kde sa to vysvetluje

4.2.4 Neskoršia korupcia

Útočník sa snaží získať privátne kľúče uzlov, ktoré mali v minulosti vplyv na reťazec. Útočník ich môže ukradnúť ale taktiež kúpiť. Tieto uzly môžu byť ochotné predáť svoje privátne kľúče pretože tým nič neriskujú. Svoje krypto tokeny môžu kedykoľvek vymeniť za reálne peniaze.

Ak útočník získa kľúče s dostatočným podielom zdrojov, tak môže pozmeniť históriu reťazca.

Kapitola 5

Harmony

Harmony je blockchain používajúci proof-of-stake konsenzus a sharding. Cieľom tohoto protokolu je poskytovať aplikácie, ktoré v minulosti nebolo možné uskutočňovať na technológii blockchain (decentralizované zmenárne, platobný systém rozsahovo porovnateľný s Visa).

Všetky informácie o protokole Harmony popisované v tejto kapitole vychádzajú z oficiálnej dokumentácie publikovanej autormi tohoto protokolu [20]. Sekcia 5.1 popisuje konsenzus protokol. Sekcia 5.2 vysvetľuje sharding používaný v tomto protokole. Na záver je v sekcii 5.4 tento protokol analyzovaný z hľadiska bezpečnosti a výkonnosti.

5.1 Konsenzus

Konsenzus v Harmony protokole je založený na algoritme PBFT (*Practical Byzantine Fault Tolerance*) [11]. Pri PBFT voľbách je jeden uzol vodca a ostatné validátory. PBFT voľby prebiehajú v dvoch fázach: príprava a potvrdenie. Vo fáze prípravy rozošle vodca návrh všetkým validátorom pomocou broadcastu. Následne každý validátor pošle broadcastom svoj hlas. Takto môže každý uzol v sieti spočítať počet hlasov. Fáza prípravy končí keď uzol videl viac ako $\frac{2}{3}$ hlasov. Následuje fáza potvrdenia, ktorá zahŕňa podobný proces hlasovania pomocou broadcastu. Celková časová zložitosť PBFT je $O(n^2)$ pre n uzlov.

5.1.1 Protokol

Harmony používa úpravu PBFT s lineárnou škálovateľnosťou, ktorú nazýva FBFT (*Fast Byzantine Fault Tolerance*). FBFT namiesto zasielania hlasov pomocou broadcastu používa prahový digitálny podpis (pozri sekciu 2.2.4). FBFT konsenzus prebieha nasledovne:

1. Vodca vytvorí blok a rozošle jeho hlavičku validátorom pomocou broadcastu. Súčasne rozošle aj dátový obsah bloku.
2. Validátori overia hlavičku bloku, podpíšu ju svojim digitálnym podpisom a pošlú späť vodcovi. Obsah bloku je zatiaľ ignorovaný.
3. Keď vodca prijme aspoň $\frac{2}{3}$ podpisov tak ich agreguje do jediného prahového digitálneho podpisu. Tento podpis rozošle pomocou broadcastu spolu s bitmapou indikujúcou validátorov ktorý podpísali.
4. Každý validátor overí, že prahový podpis obsahuje požadované $\frac{2}{3}$ hlasov. Až v tejto chvíli validátor overí transakcie v dátovom obsahu bloku, ktorý bol zasielaný už v kroku 1. Ak všetko súhlasí, tak podpíše správu s kroku 3 a pošle ju späť vodcovi.

5. Vodca čaká na $\frac{2}{3}$ podpisov validátorov s predošlého kroku (môžu sa líšiť od podpisov z kroku 3). Opäť ich agreguje do prahového podpisu a spolu s bitmapou účastníkov rozošle pomocou broadcastu nový blok na potvrdenie všetkým validátorom.

Je dôležité podotknúť, že Harmony je proof-of-stake konsenzus. Preto v celej tejto schéme neplatí, že jeden validátor sa rovná jednému hlasu. Jeden účastník môže mať viac hlasovacích lístkov (viď sekcia 5.2.1). Vodca v skutočnosti nečaká na $\frac{2}{3}$ podpisov, ale len na toľko aby ich vlastníci spoločne vlastnili $\frac{2}{3}$ celkového podielu hlasovacích lístkov.

5.2 Sharding

Harmony protokol používa *sharding* aby dosiahol lepšiu škálovateľnosť a priepustnosť transakcií. Niektoré zdroje ¹ deklarujú, že Harmony má priepustnosť priemerne 2000 TPS. Aktuálne používa 4 shardy a autori tvrdia, že pridaním každej ďalšej sa zvýši priepustnosť o 500 TPS. Táto sekcia vysvetľuje celý koncept shardingu v protokole Harmony.

5.2.1 Rozdelenie hlasovacím podielom

Ak sa chce uzol stať účastníkom Harmony blockchainu, musí vložiť určité množstvo tokenov (koncept proof-of-stake). Tento podiel určuje koľko bude mať hlasovacích lístkov pri validovaní. Tieto hlasovacie lístky sú následne náhodne rozdelené medzi všetky shardy (viď sekcia 5.2.3) po dobu jednej epochy (viď sekcia 5.2.2).

5.2.2 Epocha

Epocha je časový interval počas, ktorého je štruktúra každého shardu nemenná. Epocha odpovedá času potrebnému na vygenerovanie 32 768 blokov v beacon sharde, čo je približne 18,2 hodiny ² (koncept beacon shard bude vysvetlený v sekcii 5.2.4). Na začiatku epochy je vygenerované náhodné číslo (pozri sekcii 5.2.3) na základe ktorého sa vytvorí rozdelenie do shardov. Účastníci, ktorí chcú validovať v nasledujúcej epoche musia v aktuálnej vložiť svoj podiel tokenov.

5.2.3 Distribuovaná generácia náhodnosti

Harmony rozdeľuje hlasovacie lístky do shardov pomocou náhodnej voľby (anglicky *randomness-based sharding*). Na vygenerovanie náhodného čísla sa používa algoritmus VRF (*Verifiable Random Function* [15]) a VDF (*Verifiable Delay Function* [6]). Tento výpočet vykonáva beacon shard (koncept beacon shard bude vysvetlený v sekcii 5.2.4):

1. Vodca pošle hash posledného bloku všetkým validátorom.
2. Každý validátor vypočíta s prijatého hashu pomocou VRF náhodné číslo, ktoré pošle späť vodcovi.
3. Keď vodca prijme $\frac{1}{3}$ náhodných čísel, tak nad nimi urobí XOR. Výslednú hodnotu *pRand* vloží do nového bloku pomocou FBFT konsenzu popísaného v sekcii 5.1.1.

¹<https://www.investing.com/news/cryptocurrency-news/harmony-one-slow-and-steady-wins-the-race-2609585>

²<https://docs.harmony.one/home/network/validators/definitions/epoch-transition>

4. Keď je $pRand$ potvrdená v bloku, vypočíta vodca z tejto hodnoty finálne náhodné číslo rnd pomocou VDF.
5. VDF garantuje, že výpočet rnd zaberie toľko času aby sa stihlo vyprodukovať špecifické množstvo nových blokov. Keď vodca vypočíta rnd tak ho pomocou FBFT vloží do nového bloku.

5.2.4 Komunikácie medzi shardami

Každý shard spravuje vlastný reťazec (anglicky *shard chain*), ktorý spracováva vlastné transakcie. Jeden shard má špeciálnu úlohu a nazývame ho *beacon shard*. Tento shard generuje náhodné číslo a rozdeľuje na základe neho hlasovacie lístky do shardov (viď 5.2.3). Beacon shard taktiež prijíma tokeny na hlasovanie v ďalšej epoche (viď 5.2.2).

Harmony umožňuje komunikáciu medzi shardami. Každá taká komunikácia predstavuje broadcast na úrovni celej siete. Pomocou tejto komunikácie si môžu užívatelia presúvať svoje zdroje medzi shardami.

Vždy keď niektorý shard potvrdí nový blok, zašle jeho hlavičku na beacon chain. Ten ju validuje a uloží do svojho nového bloku. Nový blok v beacon chaine je následne pomocou broadcastu zaslaný všetkým ostatným shardom, ktoré si ho uložia. Vďaka tomu vedia jednotlivé shardy validovať transakcie z iných shardov.

5.3 Model motivácie

Za každý nový blok v reťazci sú validátori odmenení v podobe protokolom definovaného počtu tokenov. Každý validátor dostane podiel z celkovej odmeny priamo úmerný množstvu jeho hlasovacích lístkov.

Na druhej strane, každé zlomyseľné chovanie je potrestané odobraním časti vlastných tokenov. Napríklad, ak bude dokázané, že validátor podpísal neplatný blok, budú mu odobrané všetky zdroje v danom sharde.

5.4 Teoretická analýza

V tejto sekcii teoreticky zanalyzujeme protokol Harmony z hľadiska bezpečnosti a výkonnosti.

5.4.1 FBFT konsenzus

FBFT konsenzus je obdobou PBFT a teda platí, že aby sieť fungovala, musí v nej byť menej ako $\frac{1}{3}$ škodlivých uzlov. Namiesto hlasovania broadcastom sa používa prahový digitálny podpis čo je kryptograficky bezpečný spôsob hlasovania. Tento konsenzus protokol teda neznižuje bezpečnosť a zároveň zvyšuje výkonnosť keďže znižuje časovú zložitosť z $O(n^2)$ na $O(n)$.

5.4.2 Sharding

Na rozdelenie do shardov sa používa lotéria, ktorá je postavená na kryptograficky bezpečnom algoritme VRF. Náhodné rozdelenie do shardov je všeobecne najbezpečnejší spôsob, ktorý efektívne bráni útoku na podskupinu uzlov (viď sekcia 4.1.5). Navyše sa do shardov



Obr. 5.1: Pravdepodobnosť ovládnutia jedného shardu útočníkom.

nerozdeľujú uzly ale samotné hlasovacie lístky. Vďaka zaručenej náhodnosti rozdelenia hlasovacích lístkov do shardov je možné určiť pravdepodobnosť ovládnutia shardu pomocou distribučnej funkcie kumulatívneho hypergeometrického rozdelenia $H(N, K, n, k)$ [28, 20], kde:

- N je celkové množstvo hlasovacích lístkov,
- $K = \frac{N}{3}$ je maximálne množstvo škodlivých hlasovacích lístkov
- $n = \frac{N}{n_{shard}}$ je množstvo lístkov v každom sharde, kde $n_{shard} = 4$ je počet shardov,
- $k = \frac{K}{n_{shard}}$ je množstvo škodlivých lístkov v jednom sharde.

Obrázok 5.1 ukazuje pravdepodobnosť ovládnutia jedného shardu v závislosti na celkovom podiele hlasovacích lístkov vlastnených útočníkom. Tento výpočet je založený na už spomínanom hypergeometrickom rozdelení a parametroch blockchainu Harmony [20]. Môžeme vidieť, že útočník má zanedbateľnú pravdepodobnosť ovládnuť shard pokiaľ nevlastní približne $\frac{1}{3}$ z celkového množstva hlasovacích lístkov. S toho plynie, že sharding v tomto protokole neumožňuje útočníkovi využiť útok na podskupinu uzlov (viď sekcia 4.1.5).

Harmony sharding používa algoritmus VDF na oneskorenie odhalenia náhodného čísla čím zabraňuje útoku na posledného objaviteľa (viď sekcia 4.1.6).

5.4.3 Komunikácie medzi shardami

Autori deklarujú vysokú škálovateľnosť pomocou shardingu. Avšak samotné shardy sú všetky úzko závislé na beacon sharde z dôvodu medzishardovej komunikácie. Každý nový blok v ľubovoľnom sharde musí byť zaslaný do beacon shardu. Ten musí jeho hlavičku pomocou konsenzu uložiť do svojho nového bloku a tento blok broadcastom poskytnúť všetkým ostatným shardom. Táto závislosť na beacon sharde môže potenciálne predstavovať výkonnostné úzke hrdlo. Aktuálne Harmony používa len 4 shardy a teda nie je overené ani jasné nakoľko bude tento protokol efektívny ak by pracoval s desiatkami až stovkami shardov.

Kapitola 6

Solana

6.1 Proof-of-history

Proof-of-history je kryptografický algoritmus, ktorý usporiada dátové záznamy v čase tak ako vznikajú. Takýto algoritmus pomáha jednoznačne a nepopierateľne usporiadať transakcie a konsenzus hlasovania v čase pre decentralizovanú sieť ako je blockchain. Dôkaz o usporiadaní je založený na kryptografickej hash funkcii (pozri sekciu 2.2.1), ktorá spĺňa kritérium *collision resistant*.

Obrázok 6.1 demonštruje princíp fungovania proof-of-history. V sieti vždy existuje jeden uzol, ktorý nazývame *proof-of-history generátor*. Ten neustále (najrýchlejšie ako dokáže) pridáva nové záznamy do hash pointer zoznamu (pozri sekciu 2.2.2). Pridávanie nových hashov do zoznamu je sekvenčné a podmienené existenciou predchádzajúceho hashu. Tento princíp sa nazýva *Verifiable Delay Function* [6]. Každý hash teda predstavuje časový bod a vzdialenosť dvoch hashov predstavuje časový interval.

Generátor je aktuálny vodca v konsenzus protokole. Všetky transakcie a hlasy od rôznych klientov sú zasielané na generátor. Ten ich serializuje do proof-of-history reťazca. Vždy keď generátor prijme dáta od klienta (transakciu alebo hlas) pridá ich do zoznamu. Ak práve nemá dáta tak generuje prázdny záznam. Na obrázku môžeme vidieť v ktorých hashoch, alebo tiež časových momentoch, bola transakcia pridaná do reťazca.

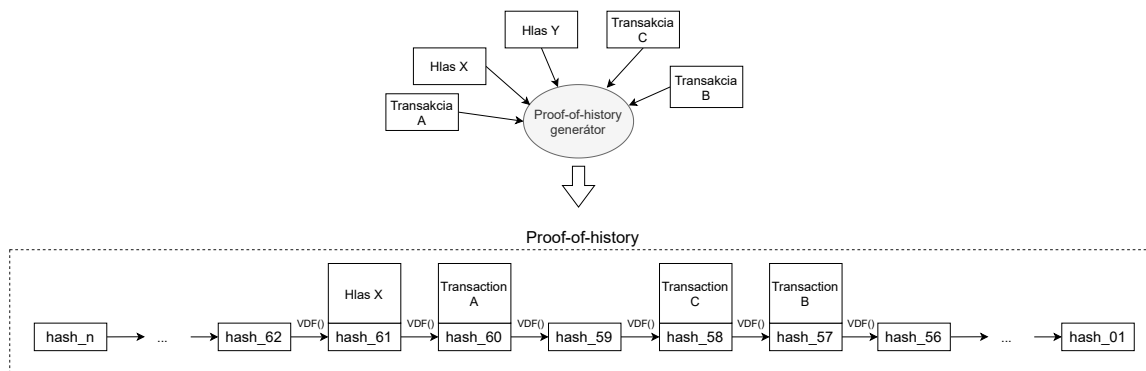
Generátor rozosiela proof-of-history sekvenciu do celej siete. Každý uzol potom môže jednoznačne určiť v akom čase boli vygenerované transakcie a hlasy konsenzu. Synchronizácia v blockchaine Solana je založená na tomto koncepte.

Existuje teoretický útok pri ktorom útočník zmení poradie v reťazci. Predpokladajme, že útočník pozná všetky transakcie a hlasy a má väčší výpočtový výkon než aktuálny generátor. Potom by dokázal vypočítať reťazec s alternatívnym usporiadaním a publikovať ho skôr. Práve kvôli tomuto generátor reťazec vždy podpíše a až potom distribuuje (pozri sekciu 2.2.3).

[34]

6.2 Konsenzus

Solana používa proof-of-stake konsenzus založený na PBFT [11]. Konsenzus sa používa na zvolenie proof-of-history generátoru (sekcia 6.2.1), hlasovanie o validnosti proof-of-history sekvencie (sekcia 6.2.2) a je na ňom založený aj model motivácie (odmeňovanie a tresty).



Obr. 6.1: Reprezentácia času v decentralizovanej sieti pomocou proof-of-history.

6.2.1 Voľba proof-of-history generátoru (vodcu)

Solana sieť sa rozdeľuje do tzv. zhlukov. Zhluk je množina validátorov, ktoré spolupracujú na konsenze. Zhluky môžu mať prieniky. Každý zhluk má vodcu, ktorý generuje záznamy v proof-of-history. Vodca sa mení po fixnom počte vygenerovaných záznamov. Tento interval voláme *slot*. Nový vodca sa určí na základe už dopredu stanoveného rozvrhu. Rozvrh je stanovený na fixný počet slotov, ktorý voláme *epocha*. Rozvrh na aktuálnu epochu sa vypočíta už v predchádzajúcej. Samotný výpočet rozvrhu prebieha tak, že sa zoberú všetci validátori, ktorý hlasovali v posledných n záznamoch, kde n je hodnota stanovená zhlukom. Títo validátori dostanú v rozvrhu pridelené vodcovské sloty na základe váhy ich podielu. Inak povedané, validátor s najväčším podielom bude v nasledujúcej epoche najčastejšie vodcom (princíp proof-of-stake).

6.2.2 Tower BFT

Solana používa konsenzus protokol *tower BFT*, ktorý je založený na $\frac{2}{3}$ zhode hlasov validátorov (BFT):

1. Klienti siete zasielajú transakcie na aktuálneho vodcu (proof-of-history generátor).
2. Vodca pridáva transakcie do proof-of-history sekvencie a túto dátovú štruktúru distribuuje validátorom.
3. Validátori overia transakcie v proof-of-history sekvencii a vykonajú ich v stanovenom poradí. Takto získajú nový stav.
4. V špecifikovanom čase, po uplynutí určitého počtu proof-of-history záznamov, pošlú validátori svoje hlasy vodcovi. Hlas je hash stavu validátora po vykonaní transakcií. Tento hash je navyše podpísaný privátnym kľúčom validátora.
5. Vodca pridáva prijaté hlasy do proof-of-history sekvencie.
6. Validátori sledujú hlasy, ktoré sa následne vyskytujú v proof-of-history sekvencii. Každý validátor počíta hlasy a ak dosahujú $\frac{2}{3}$ celkového počtu validátorov, tak bol dosiahnutý konsenzus.

Solana používa proof-of-stake a teda váha hlasu validátora je priamo úmerná jeho podielu. Pre konsenzus nie je nutné dosiahnuť $\frac{2}{3}$ celkového počtu hlasov validátorov, ale len toľko hlasov aby ich vlastníci vlastnili $\frac{2}{3}$ celkového podielu.

[33]

6.2.3 Vetvenie

Uzly v Solana sieti tolerujú dočasnú stratu spojenia s aktuálnym vodcom (poh generátorom). Pri strate spojenia predpokladajú len prázdne záznamy, ktoré neobsahujú žiadne dáta. Takýto poh reťazec si dokážu dopočítat aj sami bez komunikácie zo sieťou z posledného hashu minulého slotu.

Vetvenie v Solana konsenze vzniká pri zmene vodcu. Nový vodca nemusel zachytiť hlasy uložené v poslednom slotu. Tento slot teda nahradí prázdny záznamami. Vetvenie slotu je teda binárne (vetva s dátami a prázdna vetva).

Každý validátor musí sledovať celý binárny strom možných vetiev a uchovávať si ich stav. Voľbu vetvy urobí validátor tak, že vodcovi v danej vetve pošle svoj hlas čím vetvu potvrdí. Validátor potom nemôže hlasovať v žiadnej inej vetve po dobu (pevný počet slotov) ktorá sa nazýva *lockout*. Lockout perióda sa navyše zdvojnásobuje každým ďalším pridaným hlasom v danej vetve až do maximálnej hodnoty (aktuálne je maximum 32 hlasov). Ak chce validátor hlasovať, za inú vetvu, musí počkať do konca periódy lockout. Následovne vykoná návrat späť (anglicky *rollback*) do posledného spoločného stavu medzi starou a novou vetvou.

6.3 Model motivácie

6.3.1 Vetvenie

Hlasy v jednej vetve sa ukladajú do pomyselného zásobníku s obmedzenou kapacitou na 32 položiek. Každý nový pridaný hlas do zásobníka zdvojnásobí lockout všetkých predošlých hlasov v zásobníku. Keď zásobník dosiahne maximálnu kapacitu tak je najstarší hlas odstránený (FIFO) a majiteľ hlasu je odmenený.

Validátor je potrestaný za súbežné hlasovanie v rôznych vetvách. Ak hlasoval v jednej vetve, tak je zaviazaný na hlasovanie len v tejto vetve po dobu lockoutu. Ak validátor poruší lockout a bude mu to preukázané, budú mu odobrané jeho zdroje.

Validátori sú odmeňovaní za výber vetvy, ktorá je najčastejšia a najpoužívannejšia. Validátor je odmenený za každý hlas, ktorý dosiahne v danej vetve maximálny lockout.

6.4 Teoretická analýza

6.4.1 Proof-of-history

Why Proof-of-History is not (yet) the consensus of the future? ¹

¹<https://anycoindirect.eu/en/blog/what-is-proof-of-history>

Kapitola 7

Prehľad existujúcich simulátorov

Pre simuláciu zvolených proof-of-stake protokolov nebude vytvorený úplne nový simulátor. Simulácia celej technológie blockchainu s dostatočne presným model je rozsiahla úloha nad rámec tejto práce. Navyše už boli vytvorené simulačné nástroje, ktoré simulujú rôzne blockchainové protokoly. K týmto simulátorom existujú práce ktoré experimentálne vyhodnotili ich presnosť (viď[27, 13]). Preto bude použitý jeden s týchto nástrojov. Zvolený nástroj bude následne upravený a rozšírený tak aby bolo možné simulovať zvolené proof-of-stake protokoly.

Táto kapitola analyzuje niekoľko aktuálne dostupných simulátorov blockchainu. V závere kapitoly je vykonané porovnanie najvhodnejších nástrojov a je zvolený jeden pre ďalšiu prácu.

7.1 SimBlock

SimBlock je simulátor založený na diskretnej simulácii. Je implementovaný v programovacom jazyku Java a jeho zdrojový kód je voľne dostupný (anglicky *open source*)¹. Simulátor podporuje proof-of-work protokoly Bitcoin, Litecoin, a Dogecoin. Avšak umožňuje aj modulárne zmeniť konsenzus protokol. Posledná stabilná verzia pridala jednoduchú implementáciu proof-of-stake konsenzu².

Autori tohoto projektu v rámci vyhodnocovania presnosti simulátora vykonali experiment ktorý porovnal ich prácu s podobným už existujúcim simulátorom. Obe simulácie spustili s rovnakými parametrami a to pre protokoly Bitcoin, Litecoin, a Dogecoin. Výsledky oboch simulátorov boli veľmi podobné. Na základe tohoto experimentu autori zhodnotili, že ich simulátor napodobuje reálne blockchainové systémy s dobrou presnosť.

Autori ďalej navrhli úpravu algoritmu na voľbu susedných uzlov (anglicky *neighbor node selection algorithm*) v protokole Bitcoin. Navrhnutú úpravu odsimulovali a vyhodnotili, že ich vylepšenie algoritmu zvyšuje priepustnosť transakcií. Touto simuláciou bol demonštrovaný význam tohoto simulačného nástroja a to je zlepšovanie blockchainových protokolov z hľadiska výkonnosti a bezpečnosti pomocou simulácie. [4]

Výhodou tohoto simulačného nástroj je schopnosť simulovať aj rozsiahle siete s viac ako 10 000 uzlami. Ďalšou výhodou je, že simulácia ustanovuje medzi uzlami priame spojenie (anglicky *point-to-point*) uvažujúce geografickú lokalitu jednotlivých uzlov. Simulácia

¹Apache License, Version 2.0

²<https://github.com/dsg-titech/simblock/releases/tag/v0.8.0>

teda umožňuje zohľadňovať sieťové oneskorenie (anglicky *latency*) a šírku pásma (anglicky *bandwidth*).

Na druhú stranu, simulátor predpokladá, že všetky uzly sú poctivé. V aktuálnej implementácii teda neumožňuje experimentovanie so zlomyseľným správaním niektorých uzlov. Pre analýzu útokov ako je double-spending alebo selfish-mining by bolo potrebné tento simulačný nástroj rozšíriť. Ďalšou možnou nevýhodou je, že simulácia prebieha na úrovni blokov a propagácia transakcií zatiaľ nie je uvažovaná. [13]

7.2 Bitcoin Simulator

Bitcoin Simulator je open source ³ simulačný nástroj implementovaný v programovacom jazyku C++. Simulátor je postavený nad platformou NS-3 ⁴, ktorá slúži na diskretnú simuláciu Internetových systémov. Na tejto platforme je teda postavená simulácia peer-to-peer siete blockchainu. Tento simulátor je určený a bol vyvinutý na experimentovanie s proof-of-work protokolmi. Aktuálne podporuje protokoly Bitcoin, Litecoin, a Dogecoin. Simulátor teda nepodporuje proof-of-stake konsenzus. Avšak v minulosti už bol tento nástroj použitý tretou stranou, ktorá rozšírila implementáciu o proof-of-stake protokoly Algorand, Casper FFG a Gasper (viď [8]).

Autori simulátoru vykonali experiment na vyhodnotenie presnosti ich nástroja. Tri vyššie spomenuté protokoly boli simulované na rozsiahlej sieti a boli namerané mediány propagačného času bloku. Mediány získané zo simulácie boli relatívne podobné mediánom skutočných sietí postavených na týchto troch protokoloch. S tohoto hľadiska bol simulátor vyhodnotený ako pomerne presný. [14]

Medzi výhody tohoto simulačného nástroja patrí jeho rozsiahlosť, ktorá poskytuje veľkú škálu vstupných parametrov a taktiež veľké množstvo nameraných výstupných metrík. Z hľadiska simulácie proof-of-work protokolov umožňuje simulátor rozlišovať rôzne typy uzlov (bežný uzol a miner uzol). Rozlišovanie rôznych uzlov umožňuje tomuto nástroju simulovať aj zlomyseľné chovanie niektorých uzlov. Tento simulátor teda umožňuje analyzovať aj bezpečnosť proof-of-work konsenzu z hľadiska ťažby blokov (útoky selfish-mining a double-spending). [13]

7.3 BlockSim

BlockSim je open source ⁵ simulátor vytvorený v programovacom jazyku Python3 určený na diskretnú simuláciu blockchainu. Autori definujú tri základné ciele tohoto simulátora: všeobecnosť, rozširovateľnosť a jednoduchosť. Všeobecnosť architektúry simulátora umožňuje simuláciu veľkého množstva blockchain systémov. Simulátor by malo byť možné jednoducho rozšíriť o ďalšie protokoly. Oba predošlé ciele majú viesť k tomu aby bol tento nástroj jednoduchý na použitie.

Simulátor poskytuje všeobecnú abstrakciu blockchainu, ktorú autori nazvali *base model*. Base model je zdieľaná vrstva pre všetky konkrétne protokoly pretože pokrýva všetky základné prvky blockchainu (uzly, transakcie, bloky, reťazec blokov, fork reťazca). Nad touto vrstvou je potom možné vytvoriť implementáciu konkrétnych protokolov. Autori demonštrovali túto flexibilnú architektúru tak, že vytvorili nad base modelom simuláciu dvoch

³<https://arthurgervais.github.io/Bitcoin-Simulator/index.html>

⁴<https://www.nsnam.org/>

⁵<https://github.com/maher243/BlockSim>

proof-of-work protokolov (Bitcoin a Ethereum). Simulátor by teda mal byť priamočiaro rozširiteľný aj o simuláciu proof-of-stake protokolov. Avšak takáto simulácia by neumožnila analýzu špecifickej sekvencie správ proof-of-stake konsenzu. [3]

Najväčšou výhodou tohoto simulátoru je jeho jednoduchá architektúra, ktorá umožňuje rozširiteľnosť o ďalšie blockchainové protokoly. Simulátor pokrýva sieťovú, dátovú a konsenzus vrstvu. Z hľadiska sieťovej vrstvy je možné simulovať oneskorenie a šírku pásma pomocou geografickej distribúcie uzlov siete. Ďalšou výhodou je pomerne veľké množstvo konfigurovateľných vstupných parametrov simulácie.

Naopak nevýhodou simulátoru je, že nedokáže efektívne simulovať rozsiahle siete. Tento simulátor sice podporuje simulácie transakcií avšak tento proces je limitovaný, keďže implementácia neobsahuje žiaden model účtov (napríklad UTXO ⁶ pre Bitcoin). [13]

7.4 VIBES

VIBES je open source ⁷ nástroj pre simuláciu blockchain technológie. Simulátor implementuje výhradne proof-of-work konsenzus. Podľa autora je možná implementáciu rozšíriť o proof-of-stake konsenzus. Tento simulátor je zameraný na všeobecnejšiu simuláciu blockchainu v peer-to-peer sieti a neobsahuje implementáciu žiadneho konkrétneho protokolu. [32]

Výhodou tohoto nástroja je možnosť simulácie škodlivých uzlov. Simulátor je teda možné použiť na bezpečnostnú analýzu konsenzus protokolov. Ďalej, je podporovaná simulácia na úrovni dát (transakcie). Avšak rovnako ako pri nástroji BlockSim (viď sekcia 7.3), transakcie neobsahujú žiaden model účtov.

Nevýhodou tohoto nástroja je, že používa konštantné oneskorenie (anglicky *delay*) propagácie blokov a transakcií v sieti. Dôsledkom je nepresnosť simulácie z hľadiska výkonnostnej analýzy blockchainu. Autor tvrdí, že simulátor umožňuje simuláciu rozsiahlych sietí s viac ako 10 000 uzlami. Avšak v skutočnosti je takáto simulácia pomerne náročná pretože tento nástroj používa centrálny koordinátor, ktorý tvorí úzke hrdlo (anglicky *bottleneck*) simulácie. [13]

7.5 Shadow

Shadow ⁸ je paralelný diskretný simulátor, ktorý umožňuje beh reálnych aplikácií ako doplnkov. Tento nástroj teda simuluje vrstvu sieťovej komunikácie a na samotných uzloch siete spúšťa reálnu aplikáciu. Samotnú aplikáciu je pritom nutné modifikovať len minimálne. Takýmto spôsobom je pomocou tohoto nástroja simulovaná napríklad sieť Tor. [19]

Nad týmto simulátorom bol vytvorený doplnok, ktorý umožňuje priamo spustiť implementáciu referenčného klienta Bitcoinu. Vďaka rôznym optimalizáciám umožňuje tento doplnok spustiť tisíce takýchto Bitcoin uzlov na jedinom stroji.

Nespornou výhodou tohoto nástroja je presná simulácia, keďže implementácia nevytvára žiadnu abstrakciu Bitcoinu, ale spúšťa jeho natívnu implementáciu. Tento nástroj je vhodný na štúdium jemných rozdielov softvéru distribuovaného systému.

Nevýhodou je nízka flexibilita tohoto nástroja. Spomínaný doplnok je použiteľný jedine pre simuláciu protokolu Bitcoin. Na simuláciu ľubovoľného iného protokolu by bolo potrebné vytvoriť nový doplnok. [23]

⁶<https://river.com/learn/terms/u/unspent-transaction-output-utxo/>

⁷<https://github.com/i13-msrg/vibes>

⁸<https://shadow.github.io/>

7.6 FoBSim

FoBSim je open source ⁹ simulátor, ktorého model pokrýva dve oblasti:

- Fog Computing: Je fyzický lebo virtuálny zdroj vložený medzi koncového užívateľa a tradičné dátové centrum (anglicky *cloud*). Toto rozšírenie cloudu má zvýšiť efektivitu a bezpečnosť. Táto služba využíva tradične centrálnu autoritu.
- Blockchain: Technológia blockchain integrovaná s fog computing by mala viesť k decentralizácii tejto služby.

Rozoberať prečo je snaha integrovať tieto dve služby do jednej nie je predmetom tejto práce. Avšak tento simulačný nástroj je vyvíjaný práve za účelom experimentovania v tejto oblasti. Pre našu prácu je zaujímavá len časť, ktorá simuluje blockchain.

Simulátor poskytuje model blockchainu, ktorý obsahuje nasledujúcu funkcionality: sieťová vrstva, konsenzus algoritmy, odmeňovanie užívateľov za vytváranie nových blokov (anglicky *incentive mechanism*), paralelná ťažba blokov, distribúcia správ v sieti. [5]

Autori deklarujú, že simulátor implementuje tri konsenzus protokoly: proof-of-work, proof-of-stake a proof-of-authority. Avšak nikde nedefinujú o aké konkrétne protokoly ide. Presnosť simulácie je nejasná keďže ide o nový projekt a neexistuje zatiaľ žiadna práca, ktorá by sa zaoberala jeho vyhodnocovaním. Samotná implementácia predstavuje asi 3 000 riadkov kódu v pythone ¹⁰ čo je dramaticky menej voči iným nástrojom.

7.7 Zhodnotenie

Táto sekcia zhodnocuje simulačné nástroje popísané v rámci tejto kapitoly. Z týchto nástrojov sú porovnané vlastnosti troch najvhodnejších. Na základe týchto vlastností je na záver zvolený simulátor, ktorý bude v rámci tejto práce rozšírený o simuláciu požadovaných proof-of-stake protokolov.

7.7.1 Porovnanie najvhodnejších nástrojov

Pre potreby tejto práce definujeme niekoľko požiadaviek na simulačný nástroj ktorý použijeme. V prvom rade požadujeme dostupnosť zdrojového kódu s licenciou open source. Ďalej požadujeme aby simulátor vykazoval relatívne dostatočnú presnosť v porovnaní s reálnou blockchainovou sieťou. Dostatočnou presnosťou z hľadiska našej práce sa myslí, že simulátor vierohodne napodobuje sieťovú a konsenzuálnu vrstvu. Zaoberáme sa analýzou konsenzus protokolov a teda dátová vrstva už nie je pre simuláciu kľúčová.

Tieto požiadavky spĺňajú v najväčšej miere tri simulačné nástroje: SimBlock, Bitcoin Simulator a BlockSim. Tabuľka 7.1 zhrňuje vlastnosti týchto troch simulátorov na sieťovej, konsenzus a dátovej vrstve. Na základe tohoto zhodnotenia môžeme povedať, že tieto tri nástroje poskytujú podobne rozsiahle modely.

7.7.2 Výsledná voľba

Predošlá klasifikácia určila simulátory, ktoré majú podporu požadovaných vlastností pre potreby tejto práce. Avšak táto klasifikácia bola navysokej úrovni a zjednodušovala kom-

⁹GNU General Public License v3.0

¹⁰<https://github.com/sed-szeged/FobSim>

	Popis vlastnosti	Protokol		
		SimBlock	Bitcoin Simulator	BlockSim
Sieťová vrstva	Velkosť bloku	1	1	1
	Velkosť transakcie	0	0	1
	Počet uzlov	1	1	1
	Geografická distribúcia uzlov	1	1	1
	Bandwith	1	1	1
	Latency	1	1	1
Konsenzus vrstva	Proof-of-Work	1	1	1
	Proof-of-Stake	0	0	1
	Množstvo vygenerovaných blokov	1	1	1
Dátavá vrstva	Generovanie transakcií	0	0	1
	Vzťah transakcie k uzlu	0	0	0

Tabuľka 7.1: Podporované vlastnosti zvolených simulátorov na sieťovej, konsenzus a dátovej vrstve.

plexnosť jednotlivých vrstiev. Napríklad sieťová vrstva nástroja SimBlock je značne komplexnejšia než simulácie sieťovej vrstvy nástroja BlockSim. Tabuľka 7.2 zhrnuje a porovnáva vlastnosti týchto troch simulátorov z hľadiska náročnosti rozširiteľnosti pre potreby tejto práce. Ako môžeme vidieť, nástroj BlockSim má problém simulovať rozsiahle siete čo je veľkým nedostatkom. Bitcoin Simulator je zase projekt už 5 rokov nevyvíjaný. V tejto práci preto bude použitý nástroj SimBlock, ktorý bude rozšírený o požadovanú funkcionálnosť.

	SimBlock	Bitcoin Simulator	BlockSim
Velkosť simulovaných sietí	rozsiahle	rozsiahle	malé
Programovací jazyk	Java	C++	Python
Dostupnosť	open source	open source	open source
Vytvorenie projektu	Jún 2019	Apríl 2016	Apríl 2019
Posledná zmena v repozitári	Február 2021	Október 2016	Máj 2021
Podpora Proof-of-Stake	áno (s ukážkou)	nie	áno (bez ukážky)
Simulácia útočiacich uzlov	nie	áno	nie je jasné
Podporované protokoly	Bitcoin	Bitcoin	Bitcoin
	Litecoin	Litecoin	Ethereum
	Dogecoin	Dogecoin	

Tabuľka 7.2: Porovnanie vlastností troch najrozsiahlejších simulátorov.

Literatúra

- [1] *Horizen Academy - Blockchain as a data structure* [<https://academy.horizen.io/technology/expert/blockchain-as-a-data-structure/>]. Accessed: 2021-06-03.
- [2] ABERER, K., ALIMA, L., GHODSI, A., GIRDZIJAUSKAS, S., HARIDI, S. et al. The Essence of P2P: A Reference Architecture for Overlay Networks. In: . Január 2005, s. 11– 20. DOI: 10.1109/P2P.2005.38. ISBN 0-7695-2376-5.
- [3] ALHARBY, M. a MOORSEL, A. van. BlockSim: An Extensible Simulation Tool for Blockchain Systems. *Frontiers in Blockchain*. 2020, zv. 3, s. 28. DOI: 10.3389/fbloc.2020.00028. ISSN 2624-7852. Dostupné z: <https://www.frontiersin.org/article/10.3389/fbloc.2020.00028>.
- [4] AOKI, Y., OTSUKI, K., KANEKO, T., BANNO, R. a SHUDO, K. SimBlock: A Blockchain Network Simulator. In: *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. 2019, s. 325–329. DOI: 10.1109/INFOCOMW.2019.8845253.
- [5] BANIATA, H. a KERTÉSZ, A. FoBSim: An extensible open-source simulation tool for integrated Fog-Blockchain systems. In: . Október 2020. DOI: 10.36227/techrxiv.13148390.v1.
- [6] BONEH, D., BONNEAU, J., BÜNZ, B. a FISCH, B. Verifiable delay functions. In: Springer. *Annual international cryptology conference*. 2018, s. 757–788.
- [7] BONEH, D., LYNN, B. a SHACHAM, H. Short Signatures from the Weil Pairing. In: *Proceedings of the 7th International Conference on the Theory and Application of Cryptology and Information Security: Advances in Cryptology*. Berlin, Heidelberg: Springer-Verlag, 2001, s. 514–532. ASIACRYPT '01. ISBN 3540429875.
- [8] BORČÍK, F. *Testování bezpečnosti a výkonu Proof-of-Stake Protokolů pomocí simulace*. Brno, CZ, 2021. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Dostupné z: <https://www.fit.vut.cz/study/thesis/22846/>.
- [9] BOSAMIA, M. a PATEL, D. Current Trends and Future Implementation Possibilities of the Merkel Tree. *INTERNATIONAL JOURNAL OF COMPUTER SCIENCES AND ENGINEERING*. August 2018, zv. 6, s. 294–301. DOI: 10.26438/ijcse/v6i8.294301.
- [10] BUFORD, J., YU, H. a LUA, E. P2P Networking and Applications. *P2P Networking and Applications*. Január 2009. DOI: 10.1016/B978-0-12-374214-8.X0001-3.

- [11] CASTRO, M. a LISKOV, B. Practical Byzantine Fault Tolerance. In: *Proceedings of the Third Symposium on Operating Systems Design and Implementation*. USA: USENIX Association, 1999, s. 173–186. OSDI '99. ISBN 1880446391.
- [12] DWORZANSKI, P. A note on committee random number generation, commit-reveal, and last-revealer attacks. In: Dostupné z: http://paul.oemm.org/commit_reveal_subcommittees.pdf.
- [13] FAN, C., GHAEMI, S., KHAZAEI, H. a MUSILEK, P. Performance Evaluation of Blockchain Systems: A Systematic Survey. *IEEE Access*. 2020, zv. 8, s. 126927–126950. DOI: 10.1109/ACCESS.2020.3006078.
- [14] GERVAIS, A., KARAME, G. O., WUST, K., GLYKANTZIS, V., RITZDORF, H. et al. On the Security and Performance of Proof of Work Blockchains. In: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. New York, NY, USA: Association for Computing Machinery, 2016, s. 3–16. CCS '16. DOI: 10.1145/2976749.2978341. ISBN 9781450341394. Dostupné z: <https://doi.org/10.1145/2976749.2978341>.
- [15] GILAD, Y., HEMO, R., MICALI, S., VLACHOS, G. a ZELDOVICH, N. Algorand: Scaling Byzantine Agreements for Cryptocurrencies. In: New York, NY, USA: Association for Computing Machinery, 2017, s. 51–68. SOSP '17. DOI: 10.1145/3132747.3132757. ISBN 9781450350853. Dostupné z: <https://doi.org/10.1145/3132747.3132757>.
- [16] GILBERT, S. a LYNCH, N. Brewer's Conjecture and the Feasibility of Consistent, Available, Partition-Tolerant Web Services. *SIGACT News*. New York, NY, USA: Association for Computing Machinery. jún 2002, zv. 33, č. 2, s. 51–59. DOI: 10.1145/564585.564601. ISSN 0163-5700. Dostupné z: <https://doi.org/10.1145/564585.564601>.
- [17] HOEPMAN, J.-H. Distributed Double Spending Prevention. Marec 2008.
- [18] HOMOLIAK, I., VENUGOPALAN, S., HUM, Q., REIJSBERGEN, D., SCHUMI, R. et al. The Security Reference Architecture for Blockchains: Towards a Standardized Model for Studying Vulnerabilities, Threats, and Defenses. Október 2019.
- [19] JANSEN, R. a HOPPER, N. Shadow: Running Tor in a Box for Accurate and Efficient Experimentation. In: *Proceedings of the 19th Symposium on Network and Distributed System Security (NDSS)*. Internet Society, February 2012.
- [20] LAN, R. et al. Harmony: Technical Whitepaper. In: Dostupné z: <https://harmony.one/whitepaper.pdf>.
- [21] LEPORE, C., CERIA, M., VISCONTI, A., RAO, U. P., SHAH, K. A. et al. A Survey on Blockchain Consensus with a Performance Comparison of PoW, PoS and Pure PoS. *Mathematics*. 2020, zv. 8, č. 10. DOI: 10.3390/math8101782. ISSN 2227-7390. Dostupné z: <https://www.mdpi.com/2227-7390/8/10/1782>.
- [22] MENEZES, A. J. *Handbook of Applied Cryptography*. Taylor & Francis Inc, 1996. ISBN 0849385237ID.

- [23] MILLER, A. a JANSEN, R. Shadow-Bitcoin: Scalable Simulation via Direct Execution of Multi-Threaded Applications. In: *Proceedings of the 8th USENIX Conference on Cyber Security Experimentation and Test*. USA: USENIX Association, 2015, s. 7. CSET'15.
- [24] NAKAMOTO, S. Bitcoin: A Peer-to-Peer Electronic Cash System. *Cryptography Mailing list* at <https://metzdowd.com>. Marec 2009.
- [25] NARAYANAN, A., BONNEAU, J., FELTEN, E., MILLER, A. a GOLDFEDER, S. *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction*. Princeton University Press, 2016. ISBN 9780691171692.
- [26] NGUYEN, C., DINH THAI, H., NGUYEN, D., NIYATO, D., NGUYEN, H. et al. Proof-of-Stake Consensus Mechanisms for Future Blockchain Networks: Fundamentals, Applications and Opportunities. *IEEE Access*. Jún 2019, PP, s. 1–1. DOI: 10.1109/ACCESS.2019.2925010.
- [27] PAULAVICIUS, R., GRIGAITIS, S. a FILATOVAS, E. A Systematic Review and Empirical Analysis of Blockchain Simulators. *IEEE Access*. 2021, zv. 9, s. 38010–38028.
- [28] RICE, J. *Mathematical Statistics and Data Analysis*. Cengage Learning, 2006. Advanced series. ISBN 9780534399429. Dostupné z: <https://books.google.sk/books?id=EKA-yeX2GVgC>.
- [29] ROBLEH ALI, R. B. et al. *Distributed Ledger Technology: beyond block chain*. London, 1 Victoria Street, 2016.
- [30] SCHOLLMEIER, R. A Definition of Peer-to-Peer Networking for the Classification of Peer-to-Peer Architectures and Applications. In: . September 2001, s. 101 – 102. DOI: 10.1109/P2P.2001.990434. ISBN 0-7695-1503-7.
- [31] SMART, N. *Cryptography: An Introduction*. McGraw-Hill, 2003.
- [32] STOYKOV, L. VIBES: fast blockchain simulations for large-scale peer-to-peer networks. In: . Technische Universität München, Marec 2018.
- [33] YAKOVENKO, A. Solana: Documentation. In: . San Francisco, California: Solana Labs. Dostupné z: <https://docs.solana.com/>.
- [34] YAKOVENKO, A. Solana: A new architecture for a high performance blockchain v0.8.13. In: . 2017. Dostupné z: <https://solana.com/solana-whitepaper.pdf>.
- [35] ZHANG, S. a LEE, J.-H. Analysis of the main consensus protocols of blockchain. *ICT Express*. 2020, zv. 6, č. 2, s. 93–97. DOI: <https://doi.org/10.1016/j.ict.2019.08.001>. ISSN 2405-9595. Dostupné z: <https://www.sciencedirect.com/science/article/pii/S240595951930164X>.
- [36] ZHENG, Z., XIE, S., DAI, H.-N., CHEN, X. a WANG, H. An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends. In: . Jún 2017. DOI: 10.1109/BigDataCongress.2017.85.