

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

DIPLOMOVÁ PRÁCE

Brno, 2018

Bc. Tomáš Ecler



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

VLIV STOCHASTICKÝCH SELHÁVANÍ LINEK NA PROTOKOL PUSH-SUM

IMPACT OF STOCHASTIC LINK FAILURES ON PUSH-SUM PROTOCOL

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Tomáš Ecler

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Martin Kenyeres

BRNO 2018



Diplomová práce

magisterský navazující studijní obor **Telekomunikační a informační technika**

Ústav telekomunikací

Student: Bc. Tomáš Ecler

ID: 155155

Ročník: 2

Akademický rok: 2017/18

NÁZEV TÉMATU:

Vliv stochastických selhávání linek na protokol push-sum

POKYNY PRO VYPRACOVÁNÍ:

Nastudujte problematiku distribuovaného výpočtu a matematického aparátu sloužícího k modelování distribuovaných systémů. Vyberte si dvě funkcionality zmíněného protokolu a otestujte odolnost protokolu ve zvolených topologiích v programu MATLAB. Otestujte vliv chyby na přesnost a očekávanou deceleraci algoritmu pro různé pravděpodobnosti vzniku chyb alespoň ve 4 topologiích.

DOPORUČENÁ LITERATURA:

[1] KENYERES, Martin, Jozef KENYERES a Vladislav SKORPIL. A MESSAGE FAILURE ANALYSIS OF SYSTEMS EXECUTING AVERAGE CONSENSUS ALGORITHM. European Scientific Journal. 2015, 11(21), 56-62.

[2] KENYERES, Jozef, Martin KENYERES a Markus RUPP. Experimental node failure analysis in WSNs. In: Systems, Signals and Image Processing (IWSSIP), 2011 18th International Conference on. Sarajevo: IEEE, 2011, s. 1-5. ISBN 978-9958-9966-1-0. ISSN 2157-8702.

Termín zadání: 5.2.2018

Termín odevzdání: 21.5.2018

Vedoucí práce: Ing. Martin Kenyeres

Konzultant:

prof. Ing. Jiří Mišurec, CSc.
předseda oborové rady

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Diplomová práce se zabývá problematikou distribuovaného výpočtu a matematického aparátu sloužícího k modelování distribuovaných systémů. Nejprve je popsána problematika distribuovaných algoritmů a vzniklých chyb a následně matematické nástroje pro analýzu distribuovaných algoritmů. V rámci praktické části byl zkoumán vliv stochastických selhávání linek na vybrané parametry protokolu Push-sum a to na odchylku finální hodnoty od skutečné hodnoty průměru, rychlosti konvergence algoritmu, rozdělení konečných stavů a distribuce rychlostí konvergence. Experimenty byly vykonány v simulačním prostředí Matlab na topologii stromu, kruhu, linky, hvězdy a plně konektované mřížky. Byly analyzovány dvě funkcionality protokolu Push-sum, a to odhad průměru a odhad sumy.

KLÍČOVÁ SLOVA

Distribuované algoritmy, Stochastické algoritmy, protokol Push-sum, teorie grafů, selhání linek.

ABSTRACT

This master's thesis deals with the distributed computing and mathematical tools for modelling the distributed systems. Firstly, my attention is focused on a description of the distributed algorithms, characteristic failures for the distributed systems, and mathematical tools for an analysis of the distributed systems. The experimental part is concerned with the impact of stochastic link failures on the chosen parameters of the protocol Push-sum, namely the deviation of the final states from the average value, the convergence rate of the protocol, the distribution of the final states, and the distribution of the convergence rates. My intention is demonstrated using Matlab on a tree, a ring, a line, a star, and a fully-connected mesh topology. Was analyzed two functionalities of the protocol Push-sum, namely an estimation of the average value and an estimation of sum.

KEYWORDS

Distributed algorithms, stochastic algorithms, protocol Push-sum, graph theory, link failures.

ECLER, Tomáš. *Vliv stochastických selhávání linek na protokol push-sum*. Brno, 2017, 79 s. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce: Ing. Martin Kenyeres

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Vliv stochastických selhávání linek na protokol push-sum“ jsem vypracoval(a) samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor(ka) uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil(a) autorská práva třetích osob, zejména jsem nezasáhl(a) nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom(a) následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno

.....

podpis autora(-ky)

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu diplomové práce panu Ing. Martinu Kenyeresovi za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Brno

.....

podpis autora(-ky)



Faculty of Electrical Engineering
and Communication
Brno University of Technology
Purkynova 118, CZ-61200 Brno
Czech Republic
<http://www.six.feec.vutbr.cz>

PODĚKOVÁNÍ

Výzkum popsany v této diplomové práci byl realizován v laboratořích podpořených z projektu SIX; registrační číslo CZ.1.05/2.1.00/03.0072, operační program Výzkum a vývoj pro inovace.

Brno

.....
podpis autora(-ky)



EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI



OBSAH

Úvod	12
1 Techniky odhadu agregáčnících funkcí	13
1.1 Centralizované techniky odhadu	13
1.2 Decentralizované techniky odhadu	13
2 Distribuované systémy	14
2.1 Charakteristika distribuovaných systémů	14
2.1.1 Selhání v distribuovaném systému	15
2.1.2 Modely distribuovaných systémů	16
2.1.3 Porovnání distribuovaného a centralizovaného systému	16
2.2 Distribuované algoritmy	19
2.2.1 Stochastický proces	19
2.2.2 Stochastické algoritmy	19
2.2.3 Drby založené na agregaci algoritmů	20
2.2.4 Push-sum protokol	21
2.2.5 Chyby v distribuovaných sítích	22
2.2.6 Základní prvky distribuovaných algoritmů	23
3 Matematické nástroje pro analýzu distribuovaných algoritmů	24
3.1 Teorie grafů	24
3.1.1 Neorientované grafy	24
3.1.2 Orientované grafy	26
3.1.3 Speciální typy grafů	26
3.2 Lineární algebra	27
3.2.1 Matice výskytu	29
3.2.2 Jednotková matice	30
3.2.3 Stupňovitá matice	30
3.3 Modelování a chybovost linek v distribuovaném systému	31
3.3.1 Bernoulliho rozdělení	31
3.3.2 Vliv selhání linek pomocí Bernoulliho rozdělení	32
3.4 Topologie sítí	32
3.4.1 Linková topologie	33
3.4.2 Kruhová topologie	33
3.4.3 Hvězdicová topologie	33
3.4.4 Stromová topologie	34
3.4.5 Plně konektovaná mřížka	34

4	Vliv stochastického selhání linek pro vybrané topologie - odhad průměru	37
4.1	Vliv chyby na vnitřní stavy uzlů v topologii linky	37
4.2	Odklon finální hodnoty od skutečné hodnoty průměru ve vybraných topologiích	37
4.2.1	Odchylka v topologii linky	39
4.2.2	Odchylka v topologii plně konektovné mřížky	40
4.2.3	Odchylka v kruhové topologii	40
4.2.4	Odchylka v topologii hvězdy	40
4.2.5	Odchylka ve stromové topologii	41
4.2.6	Porovnání dosažených hodnot odchylek ve všech topologiích	41
4.3	Vliv selhání vazeb na rozmanitost konečných výsledků	43
4.4	Vliv selhání linek na počet iterací	43
4.4.1	Topologie linky	43
4.4.2	Topologie plně konektované mřížky	43
4.4.3	Topologie kruhu	51
4.4.4	Topologie hvězdy	51
4.4.5	Topologie stromu	51
4.4.6	Porovnání dosažených hodnot počtu iterací ve zkoumaných topologiích	54
5	Vliv stochastického selhání linek pro vybrané topologie - odhad sumy	56
5.1	Odklon finální hodnoty od skutečné hodnoty ve vybraných topologiích	56
5.1.1	Odchylka v topologii linky	56
5.1.2	Odchylka v topologii plně konektovné mřížky	56
5.1.3	Odchylka v kruhové topologii	56
5.1.4	Odchylka v topologii hvězdy	58
5.1.5	Odchylka ve stromové topologii	58
5.1.6	Porovnání výsledků odklonu zkoumaných topologií	58
5.2	Vliv selhání vazeb na rozmanitost konečných výsledků	60
5.3	Vliv selhání linek na počet iterací	60
5.3.1	Topologie linky	60
5.3.2	Topologie plně konektované mřížky	67
5.3.3	Topologie kruhu	68
5.3.4	Topologie hvězdy	69
5.3.5	Topologie stromu	70
5.3.6	Porovnání dosažených hodnot počtu iterací ve zkoumaných topologiích.	71

5.4	Vzájemné porovnání výsledků pro odhad průměru a sumy.	71
5.4.1	Odklon finální hodnoty od skutečné hodnoty	71
5.4.2	Vliv selhání linek na rychlost konvergence a její distribuci . . .	72
6	Závěr	73
	Literatura	74
	Seznam symbolů, veličin a zkratk	77
	Seznam příloh	78
A	Obsah přiloženého DVD	79

SEZNAM OBRÁZKŮ

2.1	Ukázka distribuovaného systému.	14
2.2	Schéma centralizovaného systému.	17
2.3	Schéma decentralizovaného systému.	18
2.4	Schéma distribuovaného systému.	18
3.1	Ukázka obecného grafu.	25
3.2	Ukázka neorientovaného grafu.	25
3.3	Ukázka orientovaného grafu.	26
3.4	Neorientovaný acyklický graf.	27
3.5	Neorientovaný zakořeněný graf.	27
3.6	Graf pro příklad matice výskytu.	29
3.7	Topologie linky.	33
3.8	Kruhová topologie.	34
3.9	Hvězdicová topologie.	35
3.10	Stromová topologie.	35
3.11	Plně konektovaná mřížka.	36
4.1	Linka $p = 0$	38
4.2	Linka $p = 0,3$	38
4.3	Linka $p = 0,6$	38
4.4	Linka $p = 0,9$	39
4.5	Odklon finální hodnoty v topologii linky.	39
4.6	Odklon finální hodnoty v topologii plně konektované mřížky.	40
4.7	Odklon finální hodnoty v kruhové topologii.	41
4.8	Odklon finální hodnoty v topologii hvězda.	42
4.9	Odklon finální hodnoty ve stromové topologii.	42
4.10	Porovnání vlivu selhání linek v linkové topologii.	44
4.11	Porovnání vlivu selhání linek v kruhové topologii.	45
4.12	Porovnání vlivu selhání linek ve stromové topologii.	46
4.13	Porovnání vlivu selhání linek ve hvězdicové topologii.	47
4.14	Porovnání vlivu selhání linek v plně konektované mřížce.	48
4.15	Vliv selhání linek na počet iterací v topologii linky.	49
4.16	Distribuce počtu iterací v topologii linky.	49
4.17	Vliv selhání linek na počet iterací v topologii plně konektované mřížky.	50
4.18	Distribuce počtu iterací v topologii plně konektované mřížky.	50
4.19	Vliv selhání linek na počet iterací v kruhové topologii.	51
4.20	Zoom na zrychlení algoritmu topologie kruhu.	52
4.21	Distribuce počtu iterací v topologii kruhu.	52
4.22	Vliv selhání linek na počet iterací ve hvězdicové topologii.	53

4.23	Distribuce počtu iterací v topologii hvězdy.	53
4.24	Vliv selhání linek na počet iterací ve stromové topologii.	54
4.25	Distribuce počtu iterací v topologii stromu.	55
5.1	Odklon finální hodnoty v topologii linky.	57
5.2	Odklon finální hodnoty sumy v topologii plně konektované mřížky. . .	57
5.3	Odklon finální hodnoty v kruhové topologii.	58
5.4	Odklon finální hodnoty v topologii hvězda.	59
5.5	Odklon finální hodnoty ve stromové topologii.	59
5.6	Porovnání vlivu selhání linek v linkové topologii.	61
5.7	Porovnání vlivu selhání linek v kruhové topologii.	62
5.8	Porovnání vlivu selhání linek ve stromové topologii.	63
5.9	Porovnání vlivu selhání linek ve hvězdicové topologii.	64
5.10	Porovnání vlivu selhání linek v plně konektované mřížce.	65
5.11	Vliv selhání linek na počet iterací v topologii linky.	66
5.12	Distribuce počtu iterací v topologii linky.	66
5.13	Vliv selhání linek na počet iterací v topologii plně konektované mřížky.	67
5.14	Distribuce počtu iterací v topologii plně konektované mřížky.	67
5.15	Vliv selhání linek na počet iterací v kruhové topologii.	68
5.16	Distribuce počtu iterací v topologii kruhu.	68
5.17	Vliv selhání linek na počet iterací ve hvězdicové topologii.	69
5.18	Distribuce počtu iterací v topologii hvězdy.	69
5.19	Vliv selhání linek na počet iterací ve stromové topologii.	70
5.20	Distribuce počtu iterací v topologii stromu.	70

ÚVOD

Během posledního desetiletí jsme zaznamenali revoluci v souvislosti mezi počítači a výsledným posunem od centralizovaných až po vysoce distribuované systémy. Jedná se o systémy, ve kterých jeden nebo více počítačů či zařízení komunikují s jinými počítači či zařízeními. Distribuovaný systém je z hlediska centralizovaného systému spolehlivější a dostupnější, byť nákladnější a složitější.

S vyšším měřítkem využití roste pravděpodobnost nestability, jako je selhání uzlů a spojení. V mé práci se zaměřím na robustnost protokolu Push-sum, což je multifunkční distribuovaný algoritmus založený na drbech, tedy takzvaných „gossip-based aggregation“ algoritmech, které poskytují výpočetní a komunikační účinnost s vysokou stabilitou při přerušení. V rámci potencionálních chyb v distribuovaných systémech jsem se zaměřil na vliv stochastického selhávání linek na výše zmíněný protokol.

Pro určitý popis a analýzu nastalé chyby v distribuované síti, je zapotřebí využít některého z matematických nástrojů a simulačního prostředí. V mém případě lze pro sestavení a výpočet distribuovaného systému využít poznatky z teorie grafů a lineární algebry, díky níž lze grafy reprezentovat pomocí matice sousednosti. V případě simulačního prostředí byl vybrán software Matlab, ve kterém byly realizovány experimenty na vybraných topologiích sítě, jako je topologie linky, kruhu, hvězdy, stromu a plně konektované mřížky.

Tato práce se zabývá vlivem chyby na odchylku finální hodnoty od reálné hodnoty průměru, rychlosti konvergence algoritmu, distribuci finálních stavů a distribuci rychlostí konvergence.

V rámci diplomové práce se zaměřím na zkoumání vlivu selhávání linek na dvě funkcionality protokolu Push-sum, což je odhad průměru a odhad velikosti sumy inicializačních stavů distribuovaným způsobem.

1 TECHNIKY ODHADU AGREGAČNÍCH FUNKCÍ

1.1 Centralizované techniky odhadu

Centralizované techniky odhadu mohou vyžadovat energetickou náročnost komunikace na velké vzdálenosti, z hlediska jednoho centrálního uzlu přes který prochází komunikace, nebo musí obsahovat komplexní multi-hop směrovací protokoly, což by mohlo mít za následek špatnou škálovatelnost. Centralizované techniky jsou také méně robustní a vhodné pro určité výsledky měření, jelikož musí být využito více parametrů pro měření [1].

1.2 Decentralizované techniky odhadu

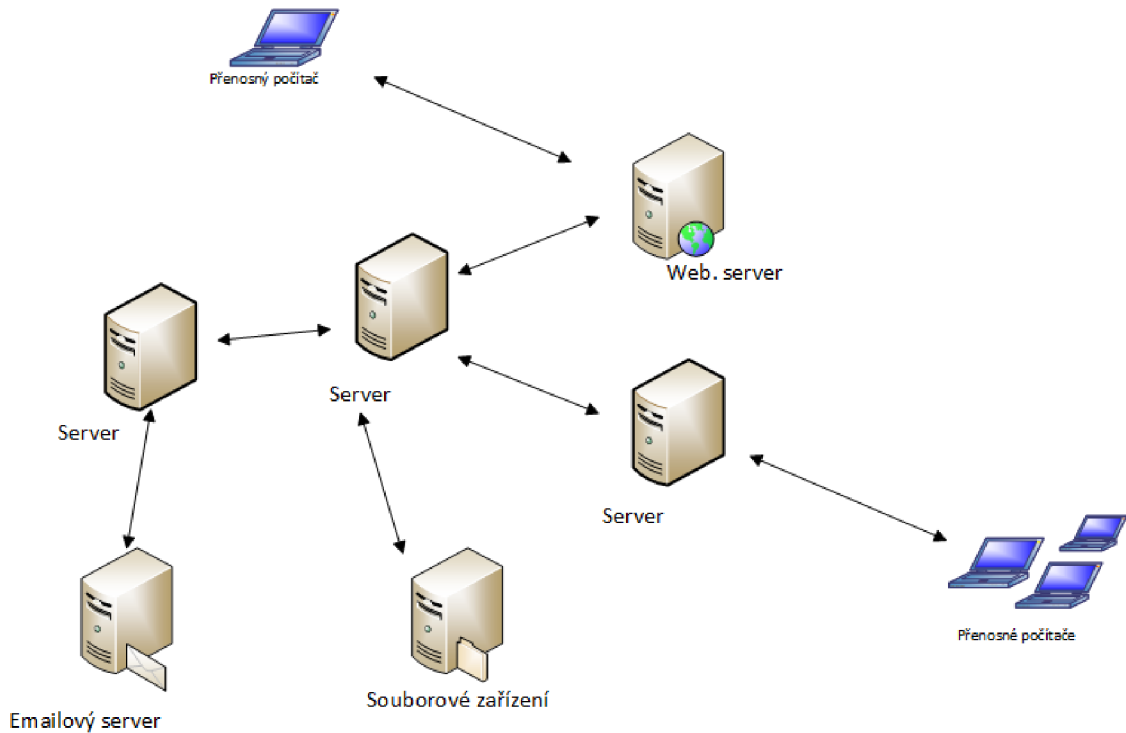
Decentralizované techniky odhadu komunikují způsobem souseď-souseď, což vede k dosažení nízké spotřeby energie a vysoké robustnosti a škálovatelnosti. U této techniky není potřeba znalosti celého systému a topologie sítě a nevyužívají se ani žádné směrovací protokoly.

Existují dvě základní kategorie techniky decentralizovaných odhadů:

1. Informace je přenášena postupně od uzlu, ze kterého byla informace iniciována až k příjemci.
2. Každý uzel rozptyluje své lokální informace iterativně do dalších uzlů, čímž je tato technika robustnější, než předchozí, avšak způsobuje zvýšenou komunikaci. Zahrnuje například metody odhadu součtu, průměru a maxima v síti. Příkladem jsou drby založené na agregaci algoritmů (gossip-based aggregation), součinný algoritmus (consensus algorithm) nebo jejich kombinace [2].

2 DISTRIBUOVANÉ SYSTÉMY

Distribuované systémy jsou specifickým typem paralelních systémů, ve kterých se jednotlivé části procesů vykonávají současně na dvou nebo více prvcích, které spolu komunikují přes síť, zatímco centralizovaný systém spravuje procesy na jediném prvku. Jako reálný příklad použití distribuovaného systému můžeme uvést ilustraci počítačové sítě, jako je na obr. 2.1



Obr. 2.1: Ukázka distribuovaného systému.

2.1 Charakteristika distribuovaných systémů

Návrh a konstrukce distribuovaných systémů je postavena na podstatě sdíleného zdroje. Mohou to být hardwarové komponenty, například tiskárny nebo softwarové komponenty, jako například soubory. Neexistuje globální stav distribuovaného systému, jelikož neexistuje společná paměť, která by dovolila tento stav zachytit. Procesy se řídí vnitřními událostmi a událostmi spojenými s přijetím nebo vysláním zprávy. Distribuovanost je dána způsobem realizace a systémem, na němž je algoritmus realizován [4].

Rozlišujeme:

- Distribuovanost dat

- duplicita dat,
- rozložení dat v jednotlivých místech systému.
- Distribuovanost řízení (neexistuje žádný hlavní proces).

Distribuovaný systém se uživateli jeví jako jeden centralizovaný systém, avšak většinou je rozložen prostřednictvím několika nezávislých systémů, které jsou schopny pracovat spolu, což dělá management a správu mnohem více komplexní v porovnání s centralizovanými systémy. U distribuovaného systému je také obtížnější lokalizovat problémy nebo selhání, jelikož případné chyby mohou být rozloženy v jakékoli části systému, což vyžaduje relativně pokročilou detekci a manipulaci s chybami. Nicméně výhody v mnoha případech vyrovnávají nevýhody.

Výhody a nevýhody distribuovaného systému:

- Výhody:
 - zvýšená dostupnost,
 - vyšší výkonost,
 - rozšiřitelnost.
- Nevýhody:
 - složitost,
 - bezpečnost,
 - vyšší náklady.

2.1.1 Selhání v distribuovaném systému

V distribuovaném systému dochází většinou dochází k částečným poruchám, tedy některé součásti mohou selhat, zatímco ostatní pokračují v práci. Práce s poruchami je tedy obtížnější než u centralizovaného systému. Existují různé techniky pro řešení poruch:

Detekce selhání

Existují způsoby, jak detekovat některé poruchy, zatímco jiné selhání je více nebo méně nemožné zjistit. Jeden ze způsobů, jak zjistit selhání, tedy například poškozená data, je použít kontrolní součet ve zprávách a souborech.

Selhání maskování

Některé zjištěné poruchy mohou být maskované nebo méně závažné. Aby se předešlo chybám, tak se mohou data souborů zapisovat například na dva disky místo

jednoho. Způsob, jakým učinit selhání méně závažným je prostě ztratit poškozenou zprávu, aby příjemce mohl odeslat ztracenou zprávu znovu, pokud odesílatel neobdržel potvrzující zprávu.

Tolerované selhání

Některé chyby mohou být tolerovány, aniž bychom je museli nějakým způsobem maskovat. Například webový prohlížeč, který nemůže získat kontakt s webovým serverem, o tom informuje uživatele, místo aby zkoušel znovu kontaktovat server, zatímco uživatel čeká.

I když existuje mnoho různých typů distribuovaných systémů, tak jsou všechny založeny na základních systémových vzorcích. Nejvíce známé a široce používané distribuované systémy jsou internet, intranet a sítě založené na mobilních zařízeních. Tyto systémy využívají výhod jiných distribuovaných systémů, jako například jmenná služba či distribuovaný souborový systém.

2.1.2 Modely distribuovaných systémů

Procesorové – vytvoří se proces pro zpracování požadavku, umožňuje tak paralelní zpracování více požadavků.

Podle způsobu uchování stavu rozpracovanosti požadavku se dělí na [4]:

1. Stavové – pamatuje si stav rozpracování požadavku.
2. Bezstavové – stav je předán klientu spolu s odpovědí, zpět je přenesen spolu s dalším požadavkem klienta.

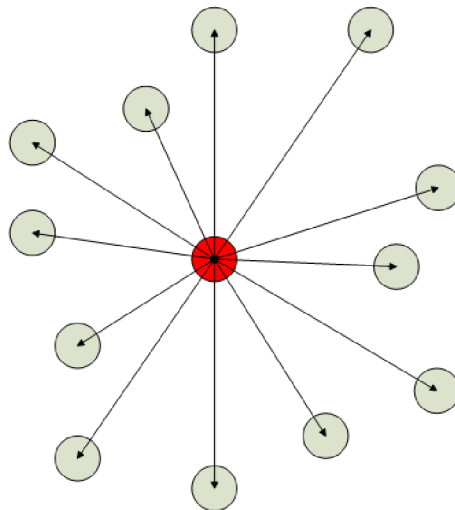
Podle používaných komunikačních služeb se dělí na servery využívající:

1. Spojované služby – virtuální okruhy.
2. Nespojované služby – spolehlivé a nespolehlivé datagramové služby.

2.1.3 Porovnání distribuovaného a centralizovaného systému

Distribuované systémy jsou mnohem flexibilnější než centralizované systémy. Systém lze snadno měnit přidáním nových komponent a aktualizaci lze provést postupně. Vzhledem k tomu, že distribuované systémy jsou založeny na několika počítačích, systémy tak mohou tolerovat selhání na více místech. Pokud dojde k chybě, budou i nadále fungovat, ačkoli zpravidla v omezené míře. Rozdíly v systému centralizovaném, decentralizovaném a distribuovaném jsou patrné z obr. 2.2, obr. 2.3, obr. 2.4. I když se zdá, že systém je spuštěn pouze na jediném procesoru, opak je pravdou, systém může být rozložen pře síť skládající se z několika strojů a procesorů. Počítače

spolupracují jako jeden systém, i když jsou umístěny vzdáleně od sebe. Tímto způsobem jsou zdroje více dostupné a spolehlivé, proto je lze lépe a efektivně využívat. Pokud nastane v jednom počítači chyba a přestane pracovat, ostatní PC pokračují v práci. Uchováním dat na více místech je mnohem spolehlivější. Zdroje mohou být sdíleny tak, aby každý v systému mohl využívat dostupné data kdekoli v celém systému. Distribuovaný systém nepotřebuje funkce centralizovaného systému ve smyslu toho, že pokud například jednomu systému chybí určitá data, může mu je poskytnout jiný systém v síti. Tímto způsobem lze použít poměrně jednoduché počítače. Sdílení hardwarových prostředků, jako jsou tiskárny a pevné disky, výrazně snižují náklady a umožňují sdílení datových zdrojů a dalších softwarových zdrojů, jako jsou například soubory, sdílené databáze či webové stránky. Tyto zdroje jsou všechny implementované na sdílených discích a procesorech [5].



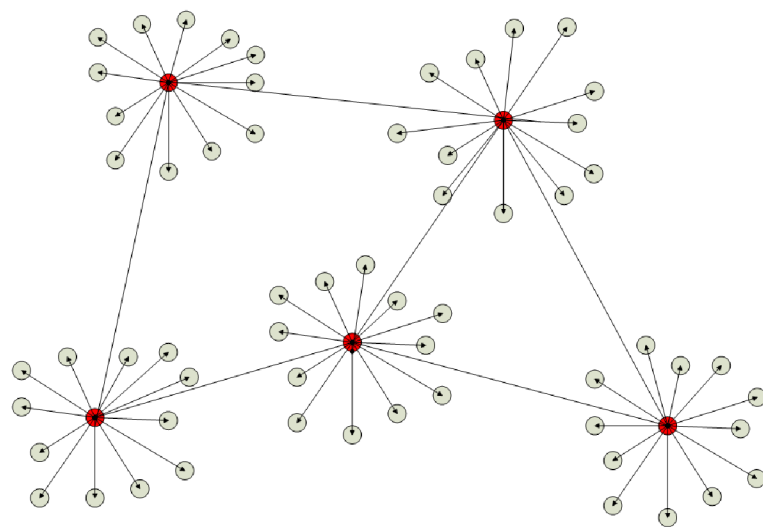
Obr. 2.2: Schéma centralizovaného systému.

Distribuovaný systém je propojená sbírka nezávislých procesů:

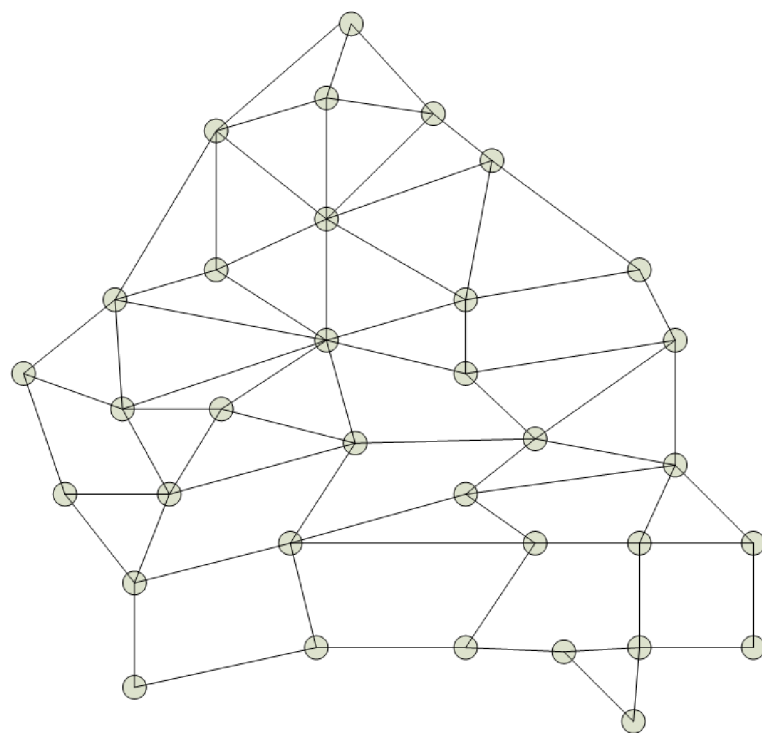
- výměna informací,
- zdroje sdílení,
- zvýšení výkonosti paralelizací,
- zvýšení spolehlivosti replikací.

Centralizovaný systém je založen na jednom centrálním uzlu:

- flexibilita při vyhledávání zdrojů,
- flexibilita při rozšiřování zařízení,
- lepší uživatelské rozhraní,
- snadnější údržba.



Obr. 2.3: Schéma decentralizovaného systému.



Obr. 2.4: Schéma distribuovaného systému.

2.2 Distribuované algoritmy

Distribuované algoritmy jsou využívány ve více prostorovém prostředí, kde jednotlivé procesy komunikují prostřednictvím zpráv. Realizace distribuovaných algoritmů způsobuje z hlediska operačního systému potíže v následujících oblastech:

- při řešení algoritmů vzájemného vyloučení,
- při řešení problémů se zablokováním (uvíznutím) procesů,
- při ukončování procesů,
- při řízení přenosu dat,
- při zajišťování pravidelnosti distribuovaného rozhodování.

Funkce algoritmu:

1. Kořenový proces pošle zprávu všem svým následníkům podle topologie.
2. Po přijetí pošle každý uzel zprávu všem svým následníkům a čeká na odpověď.
3. Po přijetí odpovědi od všech svých následníků, pošle svou odpověď svému předchůdci, je-li proces list stromu, odpovídá okamžitě.
4. Zpracování končí v okamžiku, kdy kořen získá odpověď od všech svých následníků.

Vlastnosti distribuovaných algoritmů:

- stupeň rozdělení,
- odolnost proti poruchám,
- požadavky kladené na síť [6].

2.2.1 Stochastický proces

Stochastický proces je také známý jako náhodný proces. Ve stochastickém nebo náhodném procesu existuje určitá neurčitost, i když je počáteční stav znám, tak existuje několik, často nekonečně mnoho směrů, v nichž se proces může vyvíjet. V mnoha stochastických procesech pohyb do dalšího stavu nebo pozice závisí pouze na aktuálním stavu a je nezávislý na předchozích stavech či hodnotách, které proces uskutečnil. Zatímco v deterministickém procesu, je-li počáteční bod znám, je další krok nebo výsledek předvídatelný [7].

2.2.2 Stochastické algoritmy

Systémy provádějící distribuované algoritmy se skládají z prostorově distribuovaných entit, jejichž cílem je komunikovat s druhými tak, aby plnili specifické funkce jako

celku. Vzájemná komunikace mezi nimi je realizována prostřednictvím předávání zpráv. Rozlišujeme dva typy zpráv [8]:

- Synchronní – systém je tvořen synchronizovanými entitami a jsou vždy připraveny přijímat zprávy od jiných subjektů v systému.
- Asynchronní – systém se může skládat ze zařízení, které nejsou schopny reagovat v reálném čase.

Distribuované systémy mohou být popsány několika specifickými rysy. Mezi nejdůležitější funkce distribuovaného systému můžeme uvést následující:

- entity v systému nesdílejí navzájem hodiny,
- entity jsou jen částečně vědomi dalších prvků v systému,
- entity nemají informace o systému jako celku,
- entity nemusí být ekvivalentní k sobě navzájem,
- entity nejsou informovány o selhání jiné entity v systému.

Entita je prvek, který je schopný měnit stav systému. V mnoha případech především v těch, které zahrnují síťové distribuované systémy. Danými entitami bývají různé druhy zpráv, kterými jednotlivé uzly sítě komunikují a na základě kterých se mění stavy těchto uzlů [9].

2.2.3 Drby založené na agregaci algoritmů

Existuje mnoho kategorií, do kterých lze rozdělit distribuované algoritmy. V této práci se zaměřím na takzvané „gossip-based aggregation“ algoritmy, což jsou v překladu drby založené na agregaci algoritmů.

Gossip algoritmy se objevily jako komunikační a výpočetní modely rozsáhlých síťových systémů. Tyto algoritmy jsou založeny na náhodné komunikaci, která poskytuje pravděpodobnostní záruky na rychlost konvergence a přesnost. Poskytují také robustnost, škálovatelnost, výpočetní a komunikační účinnost a vysokou stabilitu při přerušení.

Tato sada algoritmů je druh komunikace na počítači, inspirovaná šířením informací mezi lidmi přes sociální prostředí. Sadu těchto algoritmů lze charakterizovat třemi hlavními atributy [10]:

1. Rozšířitelnost – růst velikosti sítě výrazně nesníží výkon systému, při provádění algoritmu „gossip-based aggregation“. Každá entita distribuovaného systému odešle pevný počet zpráv bez ohledu na velikost systému, stejně jako to provádí jednoduchý algoritmus pro výměnu informací s malou rychlostí. Aby se mohli naplnit konkrétní funkce, tak konkrétní entity musí být informovány o totožnosti subjektů v sousední oblasti.

2. Adaptabilita – gossip algoritmy umožňují snadno odstranit entity z distribuovaného systému. V některých případech jsou implementovány mechanismy k provedení této akce v algoritmu.
3. Degradace – gossip algoritmy mají určitou prahovou hodnotu f . Pokud není prahová hodnota překročena, tak je zaručena správná funkce algoritmu. Spolehlivost gossip algoritmu je dána pravděpodobností, že nenastane více než f selhání. Tuto pravděpodobnost lze těžko odhadnout, jelikož může být ovlivněna parametry, které se těžko hodnotí. Proto se v praktické implementaci očekává, že tento algoritmus nebude rychle degradován, i za předpokladu překročení prahové hodnoty f .

Z gossip-based aggregation algoritmů jsem se zaměřil na Push-sum protokol.

2.2.4 Push-sum protokol

Push-sum je symetrický protokol pro výpočet hodnot v uzlech sítě a šíření informací v decentralizovaných a asynchronních systémech. Algoritmus je plně decentralizovaný a je vhodný u rozsáhlých sítí. Navrhovaný přístup kombinuje jednoduchost přístupu push a účinnost přístupu pull. Systémy Push-pull nemohou být přímo využité v asynchronních systémech, protože vyžadují synchronní spárované komunikační operace k zajištění jejich přesnosti. Ačkoli Push schémata zaručují přesnost i při asynchronní komunikaci, trpí naopak pomalejší a nestabilní konvergencí. Symetrický Push-sum protokol nevyžaduje synchronní komunikaci a dosahuje rychlosti konvergence podobné jako u režimu Push-pull, při zachování přesnosti stability systému.

V průběhu celého procesu každý uzel uloží dvě hodnoty:

1. Hodnotu vnitřního stavu.
2. Hodnotu váhy.

V případě odhadu průměru je hodnota vnitřního stavu nastavena na výchozí hodnoty uzlu na začátku celého procesu. V případě odhadu sumy jsou vnitřní stavy všech uzlů a počáteční váhy nastaveny na hodnotu nula mimo jednoho uzlu, který je nastaven na hodnotu jedna. [11].

Funkce protokolu Push-sum

Funkce protokolu probíhá následovně, v průběhu každé iterace si každý uzel vybere náhodně jednoho ze svých sousedů. Tomuto sousedu je odeslána dvojice poloviny hodnoty vnitřního stavu a váhy. Tato informace je také uložena ve vnitřní paměti vysílajícího uzlu. Všechny uzly vypočítají odhad průměru výpočtem poměru hodnot parametrů vnitřního stavu a váhy. Tento postup se opakuje, dokud systém nedosáhne

součinnosti, tedy rozdílu mezi maximální a minimální hodnoty v rámci systému, což je menší než 0,00015. Ověření správné funkce Push-sum protokolu lze zkontrolovat globálním součtem všech odhadů v systému, který by měl být konstantní v každé iteraci. Podmínka správné funkce také říká, že součet všech hodnot vnitřního stavu a váhy by měla být stejná během celého procesu [12] [13].

Vliv selhávání linek na „mass conservation theorem“

Nevýhodou algoritmu Push-sum je to, že jeho správnost závisí na takzvané masové ochraně, tedy „mass conservation theorem“. Tento teorém pojednává o tom, že počáteční údaje musí být zachovány během celého výpočtu, či-li jakékoliv selhání systému narušuje „mass conservation theorem“, a proto algoritmus Push-sum nemůže zaručit správné výsledky v případě selhání systému [14].

2.2.5 Chyby v distribuovaných sítích

Mrtvý uzel v síti

Důležitým aspektem v analýze selhání uzlu je topologie sítě. Dopad při selhání uzlu je velmi odlišný, jak u silně propojené sítě, kde má každý uzel velké množství sousedních uzlů, tak u slabě propojené sítě, kde má uzel omezené množství sousedních jednotlivých uzlů.

Za předpokladu, že mrtvý uzel již nevysílá, tak se v síti přizpůsobí nové řešení za předpokladu, že po odstranění mrtvého uzlu síť zůstává připojena, tedy pokud existuje cesta spojující uzly, mezi nimiž byl konkrétní uzel odstraněn. Pokud náhradní cesta mezi uzly neexistuje, dojde k odpojení sítě. Stav odpojení, či nalezení nového řešení záleží na topologii sítě, tedy na podmínce odolnosti 2.1, ve které by měl být určen minimální počet uzlů, jejichž odstranění vede k odpojení. Dále by měla být splněna taková podmínka, kdy při výpadku jednoho uzlu je dočasně nepřístupná jen malá část informací a ostatní informace z jiných uzlů jsou stále k dispozici.

$$|D| < K(N) \wedge n_d \notin C(N); \forall n_d \in D. \quad (2.1)$$

D je počet odebraných uzlů, $K(N)$ je minimální počet uzlů, jejichž odstraněním dojde k odpojení, $C(N)$ počet uzlů sítě N a n_d mrtvý uzel. Dalším důležitým aspektem v případě selhání uzlu je počet sousedů $d(n_i)$ mrtvého uzlu, čím více sousedů, tím větší negativní vliv na výkon sítě.

Aktualizace pevných hodnot

Dalším scénářem při selhání uzlu může být aktualizace pevnou hodnotou. V tomto scénáři má konkrétní uzel N pevné hodnoty ve všech iteracích přenosu. Tato chyba

má devastující účinek v celé síti. Značně se zvýší počet iterací. Pokud je v síti přítomen jeden uzel s pevnou aktualizací hodnot, tak v ostatních přenosech již není dosaženo součinnosti. Již přijímání pevné hodnoty v první iteraci ovlivňuje konečný výsledek, zvyšuje se tak chybovost. Chování sítě je mnohem předvídatelnější než v případě mrtvého uzlu, vliv na výkon sítě závisí na počtu sousedů uzlu s pevnou aktualizací hodnot.

Aktualizace přijatých hodnot

Poslední scénář selhání uzlu opět souvisí s aktualizací hodnot iterace uzlu. V tomto případě poškozený uzel není schopen vypočítat hodnotu, představující stav uzlu po provedení k-té iterace, proto se jeho stav aktualizuje z hodnoty přijaté od svých sousedů předchozí iterace. Uzel sám o sobě není schopen rozpoznat, že neběží správně. Pouze vytvořením kontrolního mechanismu, lze umožnit uzlu předpovědět, jak se mění hodnota uzlu v další iterace. Oproti pevné aktualizaci nemusí tato chyba vést ke katastrofálním výsledkům [15].

2.2.6 Základní prvky distribuovaných algoritmů

V distribuovaném prostředí jsou spolu svázány dva komponenty.

Procesy:

- reagují na určité události.

Komunikační spoje:

- zajišťují informační vazby mezi procesy.

Vlastnosti spojů distribuovaných algoritmů

Statické vlastnosti spojů (topologie):

- úplné polygonální (úplné vzájemné propojení),
- neúplné polygonální (neúplné vzájemné propojení),
- stromové (mezi dvěma uzly existuje právě jedna cesta),
- Hvězdicové (jeden hlavní centrální uzel),
- Kruhové (jednosměrné, obousměrné).

Dynamické vlastnosti spojů (chování při přenosu dat):

- zabezpečení sekvenčního přenosu,
- zabezpečení proti chybám,
- časové parametry při přenosu zpráv,
- zpoždění,
- ochrana proti zahlcení, zablokování [4].

3 MATEMATICKÉ NÁSTROJE PRO ANALÝZU DISTRIBUOVANÝCH ALGORITMŮ

V rámci experimentální části modelujeme funkcionalitu distribuovaného systému použitím matematického aparátu diskutovaného v této kapitole.

Distribuovaný systém se sestává z množiny procesů a komunikačních sítí, které mohou komunikovat s ostatními procesy. Každý proces představuje přechodový systém, což znamená, že systém je schopný měnit své (vnitřní) stavy v diskrétních časových okamžicích. Předpokládá se, že procesy jsou navzájem propojené komunikačními linkami, čímž vytváří komunikační síť. Komunikační síť je nejčastěji reprezentovaná jednoduchým neorientovaným grafem $G = (V, E)$, kde V je množina procesů a E je množina komunikačních linek. Ve speciálních případech lze uvažovat i o reprezentaci komunikační sítě orientovaným grafem.

3.1 Teorie grafů

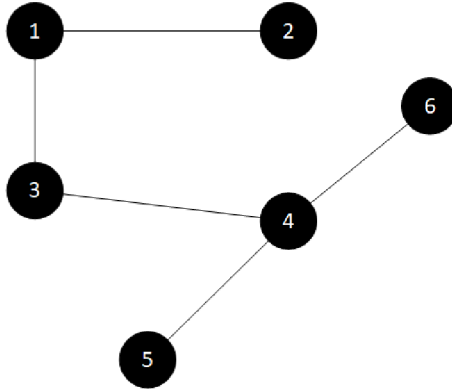
Grafem obvykle znázorníme určité situace, nebo realizaci rozličných problémů, jako například počítačovou síť, postup stavby rodinného domu, sociologické vztahy ve skupině apod.

V základní definici grafu popisujeme hrany mezi dvojicemi vrcholů pomocí dvouprvkových podmnožin těchto vrcholů. Graf je uspořádaná dvojice $G = (V, E)$, kde V je množina vrcholů a E je množina hran. Hranu mezi vrcholy u a v píšeme jako (u, v) . Vrcholy spojené hranou jsou sousední. Grafy si obvykle představujeme v takové analogii, že vrcholy jsou body a hrany jsou čáry mezi nimi, jako na obr. 3.1, kde $V = \{1, 2, 3, 4, 5, 6\}$ a $E = \{(1, 2), (1, 3), (3, 4), (4, 5), (4, 6)\}$. Každý vrchol patří do množiny V a každá hrana patří do množiny E . Některé vrcholy nemusí být spojené žádnou hranou, takové vrcholy nazýváme izolované. Číselné hodnoty specifikují jejich délku. Rozeznáváme grafy orientované a neorientované [16].

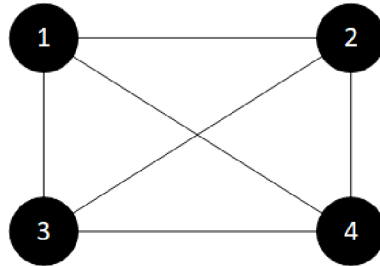
3.1.1 Neorientované grafy

Takzvaný neorientovaný graf, jako na obr. 3.2, je grafem konečným, má tedy konečný počet uzlů a hran. Někdy bývá uvažováno vícero hran mezi dvěma uzly, čímž se jedná o takzvaný multigraf.

Neorientovaný graf nazýváme spojitým, pokud existuje cesta z každého vrcholu do každého jiného vrcholu. Pokud graf neobsahuje smyčky, hrana ve tvaru (u, u) ,



Obr. 3.1: Ukázka obecného grafu.



Obr. 3.2: Ukázka neorientovaného grafu.

nebo více hran mezi stejnými vrcholy, tak se graf nazývá jednoduchý, což znamená, že hrana (u, v) sousedí s vrcholem u a vrcholem v . U takových grafů jdou hrany oběma směry. Například hrana $(1, 3)$ sousedí s vrcholem 3. Každý vrchol má určitý počet stupňů, což je počet hran, které s ním sousedí, tuto skutečnost můžeme vyjádřit vztahem 3.1

$$d(v_i) = |N(v_i)|, \quad (3.1)$$

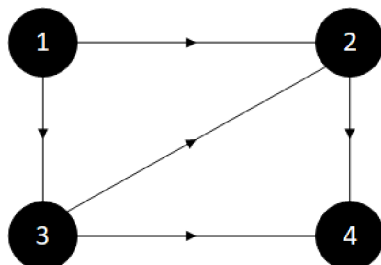
kde písmeno d značí stupeň vrcholu. Například vrchol 3 má stupeň 3, zatímco vrchol 4 má stupeň 2. Vrchol u je soused s vrcholem v , pokud existuje hrana mezi nimi. Například vrchol 1 sousedí s vrcholem 2, 3 a 4.

Pokud si graf vyjádříme pomocí matice sousednosti, která bude popsána níže, můžeme vztah 3.2 zjednodušit vyjádřením d jako

$$d(v_i) = \sum_{j=1}^n [A]_{ij} = \sum_{j=1}^n [A]_{ji}. \quad (3.2)$$

3.1.2 Orientované grafy

Orientovaný graf nazýváme silně souvislým, pokud existuje cesta mezi každými dvěma vrcholy grafu. Někdy je k hranám přiřazen i jejich směr, potom hovoříme o orientovaných grafech. Hrany v orientovaných grafech se znázorňují šipkami, které určují jejich směr, jako na obr. 3.3.



Obr. 3.3: Ukázka orientovaného grafu.

Odcházející stupeň vrcholu, je počet hran, které začínají v daném vrcholu. Přicházející stupeň vrcholu, je naopak počet hran, které končí v daném vrcholu. Například vrchol 4 má přicházející stupeň 2 a odcházející 1. Pokud bychom chtěli určit počet stupňů, nebo-li počet sousedů určitého vrcholu, použijeme vztah 3.3

$$d(v_i) = d_{out}(v_i) + d_{in}(v_i), \quad (3.3)$$

kde d_{out} a d_{in} vypočteme vztahy 3.4 a 3.5

$$d_{out}(v_i) = \sum_{j=1}^n [A]_{ij}, \quad (3.4)$$

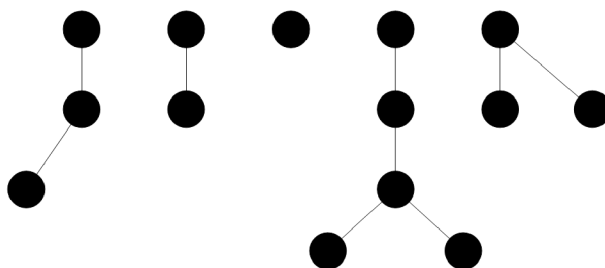
$$d_{in}(v_i) = \sum_{j=1}^n [A]_{ji}. \quad (3.5)$$

3.1.3 Speciální typy grafů

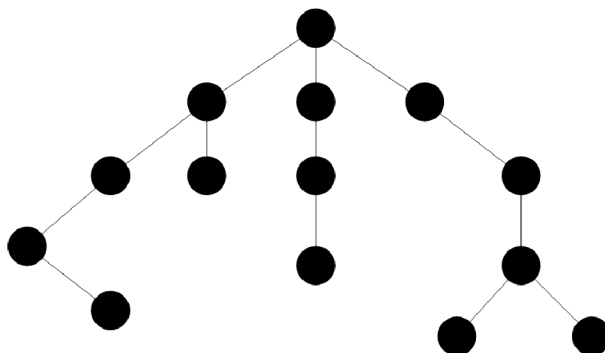
Neorientovaný graf nazýváme stromem, který může být acyklický, jako na obr. 3.4, nebo zakořeněný jako je tomu na obr. 4.12.

Strom je zakořeněný, pokud má jasně vyznačen nejvyšší vrchol \Rightarrow kořen. Každý vrchol má právě jednoho rodiče a může mít libovolný počet dětí.

Úplný graf obsahuje hranu mezi každou dvojicí vrcholů, jako je tomu znázorněno na obr. 4.12.



Obr. 3.4: Neorientovaný acyklický graf.



Obr. 3.5: Neorientovaný zakořeněný graf.

Cesty grafu

Cesta z vrcholu x do vrcholu y je taková postupnost vrcholů (v_0, v_1, \dots, v_k) , že $v_0=x$ a $v_k=y$ a hrany $(v_0, v_1), (v_1, v_2), \dots$ patří do množiny hran E . Celková délka cesty se označuje jako k [17].

Cesta je jednoduchá, pokud neobsahuje více jak dva vrcholy, kromě začátku a konce cesty. Cestu nazýváme cyklem, pokud je to cesta z určitého vrcholu do toho stejného vrcholu.

3.2 Lineární algebra

Lineární algebra je jedna z nejvíce známých matematických disciplín, protože má bohaté teoretické základy a mnoho užitečných aplikací vědy, jako například řešení soustav lineárních rovnic či výpočetní determinant.

Základní pojmy a metody lineární algebry zasahují do téměř všech oblastí moderní matematiky, jako například v technice, informatice, statistice a dalších. Lineární algebra je jazyk, který slouží k vyjádření geometrických ideí, pokud bychom chtěli zapsat graf, tak nejtypičtější zápis grafu je pomocí matice, respektive matice sousednosti. Abychom mohli vyjádřit matici sousednosti, tak se nejprve musí očíslovat všechny vrcholy v grafu. Za rozměr matice se zvolí počet vrcholů, kdybychom měli

například graf se čtyřmi vrcholy, ale matici rozměrů jen 3×3 , nemohli bychom zapsat hrany vedoucí z, nebo do čtvrtého vrcholu. Nechť $G = (V, E)$ je graf s n vrcholy, označené (v_1, \dots, v_n) v libovolném pořadí. Matice sousednosti grafu G je čtvercová matice

$$A_G = (a_{ij})_{ij=1}^n, \quad (3.6)$$

definovaná předpisem:

$$a_{ij} = \begin{cases} 1 & \{v_i, v_j\} \in E \\ 0 & \end{cases} \quad (3.7)$$

Například pro kruhovou síť o čtyřech uzlech by matice vypadala jako na příkladu 3.8

$$A = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix} \quad (3.8)$$

Pro neorientované grafy platí, že jejich matice sousednosti jsou symetrické, tudíž sousední uzly jsou si navzájem rovnocenné, což můžeme vyjádřit matematickým vztahem 3.9

$$N(v_i) = v_j : (v_i, v_j) \in E. \quad (3.9)$$

Pro orientované grafy bude situace zápisu velmi podobná jako u neorientovaných grafů, s tím rozdílem, že si musíme uvědomit, že první číslo matice znamená odkud a druhé kam hrana vede, tato situace je vyjádřena vztahem 3.10 celkového počtu sousedů pro uzly, které jsou v orientovaném grafu nazývány jako rodičovský uzel, což je počáteční bod hrany.

$$\pi(v_i) = v_j : (v_i, v_j) \in E. \quad (3.10)$$

Tento vztah můžeme dále rozšířit o matematický vztah 3.11 celkového počtu sousedů pro takzvaný dětský uzel, což je koncový bod hrany.

$$\lambda(v_i) = v_j : (v_i, v_j) \in E. \quad (3.11)$$

Přestává totiž platit, že hodnota prvku na souřadnicích např. $(1, 2)$ je stejná jako hodnota prvku na pozici $(2, 1)$. Proto matice sousednosti orientovaného grafu nemusí být nutně symetrická.

Další způsob popisu grafu může být za pomoci Laplaceovy matice, která je definována vztahem 3.12

$$L_{ij} = \begin{cases} d(v_i) & i = j \\ -A_{ij} & i \neq j \end{cases} \quad (3.12)$$

Abychom však získali tuto matici, tak musíme nejprve vypočítat rozdíl matic D_G a A_G podle vzorce 3.13

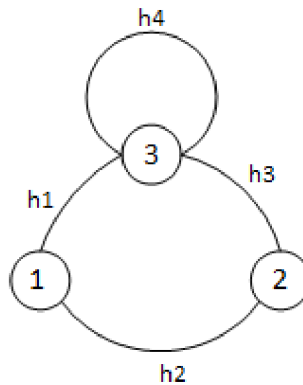
$$L_G = D_G - A_G, \quad (3.13)$$

kde L_G je výsledná Laplaceova matice a D_G je diagonální matice, která znázorňuje, v jaké pozici jsou stupně jednotlivých vrcholů grafu G na diagonále a A_G je matice sousednosti grafu G [18].

Pokud používáme lineární algebru například k pochopení fyzických systémů, často najdeme více struktur v maticích a vektorech. Existuje mnoho aplikací lineární algebry, avšak nejlepší uplatnění pro řešení výše zmíněného problému najdeme ve speciálních typech matic.

3.2.1 Matice výskytu

Matice výskytu bývá také označována jako řídká matice, která představuje speciální třídu matic, jež se skládá převážně z nulových a nenulových prvků. Velikost matice se rovná počtu vrcholů a hran. Mějme obr. 3.6, podle kterého vytvoříme matici výskytu 3.14, kde řádky představují vrcholy a sloupce hrany podle pořadí.



Obr. 3.6: Graf pro příklad matice výskytu.

$$A = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 2 \end{bmatrix} \quad (3.14)$$

Matice se vyplňuje podle toho, kolika hranami je spojen daný vrchol a výsledný počet se umístí jako maticový prvek. V našem případě je vrchol 1 je spojen dvěma hranami h1 a h2, vrchol 2 je spojen hranami h2 a h3 a vrchol 3 je spojen hranami h1, h3 a h4. Za zbývající nevyplněné pozice se dosadí nula. Avšak pokud by byl pro tuto situaci využit větší graf, získali bychom tak větší „řidší“ matici, tedy matice by byla z větší části vyplněna nulami [19].

3.2.2 Jednotková matice

Jednotková matice je čtvercová matice, která má některé jedinečné vlastnosti, například může mít více než jednu totožnou matici, obvykle levou a pravou matici totožnosti, tedy součin jednotkové matice s jinou maticí se rovná stejné matici, což se lze vyjádřit jako vztah 3.15, 3.16

$$I.A = A, \tag{3.15}$$

nebo

$$A.I = A, \tag{3.16}$$

kde I je jednotková matice a A je jiná matice, která se rovná stejné matici A . Jednotková matice obsahuje samé jedničky podél hlavní diagonály, zatímco její ostatní pozice obsahují samé nuly, matice může vypadat jako na příkladu 3.17 [20].

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.17}$$

3.2.3 Stupňovitá matice

U stupňovité nebo také schodovité matice je vedoucí prvek nenulového řádku první nenulový prvek tohoto řádku. Tato matice musí splňovat následující podmínky:

1. Každý nenulový řádek matice A leží nad každým jejím nulovým řádkem.
2. Vedoucí prvek vyššího řádku leží více vlevo než vedoucí prvek nižšího řádku.
3. Vedoucí prvek každého nenulového řádku je jedna.
4. Ve sloupci, ve kterém se nachází vedoucí prvek určitého řádku, jsou všechny ostatní prvky rovné nula.

Nulový řádek nemá vedoucí prvek. Pokud stupňovitá matice splňuje čtyři výše popsané podmínky jedná se o takzvanou matici redukovaného tvaru, pokud splňuje pouze první dvě, jedná se pouze o stupňovitou matici bez redukovaného tvaru. Stupňovitá matice v redukovaném tvaru může vypadat jako na příkladu 3.18 [20].

$$A = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 2 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (3.18)$$

Výše zmíněné algoritmy mohou najít uplatnění například v bezdrátových senzorových sítích (označené jako WSN), což jsou distribuované systémy sestávající se ze souboru prostorově distribuovaných zařízení, jejichž hlavním cílem je měřit a zpracovávat konkrétní podněty v prostředí. Tato zařízení (lze označit jako uzly) jsou charakterizovány omezenými výpočetními schopnostmi a energií, proto je jejich funkčnost založena na vzájemné spolupráci, respektive komunikaci jednotlivých uzlů [22].

3.3 Modelování a chybovost linek v distribuovaném systému

V této části bude zkoumán vliv vazeb selhání na protokolu push-sum. Selhání přenosu mezi uzly bylo modelováno pomocí Bernoulliho rozdělení.

3.3.1 Bernoulliho rozdělení

Bernoulliho nebo také binomické rozdělení je distribuce pravděpodobnosti, která shrnuje pravděpodobnost, že hodnota bude mít jednu ze dvou nezávislých hodnot, a to „úspěch“ nebo „selhání“. Základní předpoklady binomického rozdělení spočívají v tom, že pro každou studii je pouze jeden výsledek, že každá zkouška má stejnou pravděpodobnost úspěchu, navzájem se vylučuje nebo je navzájem nezávislá.

Binomická distribuce shrnuje počet pokusů nebo pozorování, kdy každá zkouška má stejnou pravděpodobnost dosažení jedné konkrétní hodnoty. Binomické rozdělení určuje pravděpodobnost sledování stanoveného počtu úspěšných výsledků ve stanoveném počtu pokusů. Očekávaná hodnota nebo průměr binomického rozdělení se vypočte vynásobením počtu pokusů pravděpodobností úspěchu. Například, když se vezme očekávaná hodnota počtu stejného otočení mince ve 100 studiích, předpokládaná hodnota je 50, tedy polovina z celkového počtu, jelikož $100 * 0,5$.

3.3.2 Vliv selhání linek pomocí Bernoulliho rozdělení

Mějme určitou pravděpodobnost p , jejíž hodnota určuje pravděpodobnost selhání spojení. Byl zkoumán vliv tohoto selhání silně a slabě propojené topologie. Za účelem zajištění důvěryhodnějších závěrů se každý test skládá ze čtyř sítí topologie hvězdy, stromu, kruhu a plně konektované mřížky. Předpokládejme, že pravděpodobnost selhání může mít deset hodnot: $\{0\%, 10\%, 20\%, 30\%, 40\%, 50\%, 60\%, 70\%, 80\%, 90\%\}$. Nebyla uvažována hodnota rovnající se 100%, jelikož protokol nemůže v takovém scénáři splnit jeho funkčnost. Protokol Push-sum je stochastický protokol, jehož parametry se mohou měnit, když se jeho provádění opakuje, proto každý experiment opakujeme 10 000krát. Konkrétně byl zkoumán vliv dopadu selhání na odchylku finální hodnoty od reálné hodnoty průměru, rychlosti konvergence algoritmu, distribuci finálních stavů a distribuci rychlostí konvergence.

3.4 Topologie sítí

Topologie sítě je uspořádání různých síťových prvků používaných při přenosu dat a vytváření takzvaných uzlů při vzájemném propojení. Právě toto propojení různých prvků je známé jako topologická struktura počítačové sítě. Topologie se dělí na dva typy:

1. Fyzická,
2. logická.

Fyzická topologie je uspořádání různých prvků počítačové sítě, jako jsou například počítače, kabely a další zařízení. Dále pak zobrazuje například rozložení kabeláže sítě, umístění uzlů, propojení kabelů a uzlů a v neposlední řadě udává, jakým způsobem jsou jednotlivá zařízení mezi sebou propojeny. Propojení je realizováno pomocí:

- kroucené dvojlinky,
- koaxiálních kabelů,
- optických kabelů,

nebo bezdrátově pomocí přenosů:

- rádiových,
- mikrovlnných,
- infračervených.

Logická topologie popisuje síťové datové toky z jednoho zařízení na druhý, bez ohledu na tvar uspořádání fyzických komponentů.

V podstatě topologie sítě je uspořádání různých připojených zařízení, kdy toto uspořádání může mít například tvar topologie kruhu, což ale neznamená, že jde o kruhovou topologii, jelikož tvar spojení je založen na tom, jak jsou data přenášena a jak jsou zařízení propojena.

V síti rozdělujeme základní typy topologií, jako jsou linková, kruhová, hvězdicová, stromová a plně konektovaná mřížka. Výše zmíněné topologie pro 16 uzlů využijí k simulaci selhání linek na protokol push-sum.

3.4.1 Linková topologie

Topologie linky jako je na obr. 3.7 má jeden hlavní (sběrniceový) kabel, ke kterému jsou uzly připojeny. Pokud jsou odesílána data mezi uzly, tak ostatní uzly nemohou vysílat. Když je připojeno příliš mnoho uzlů, tak se přenos dat dramaticky zpomalí, jelikož uzly musí déle čekat, než bude sběrnice volná. Signál ze zdroje se pohybuje v obou směrech na všechna zařízení připojená na sběrniceovém kabelu, dokud nenažde určitého příjemce. Pokud se data shodují s adresou počítače, tak jsou data přijata, v opačném případě zařízení tyto data ignoruje.



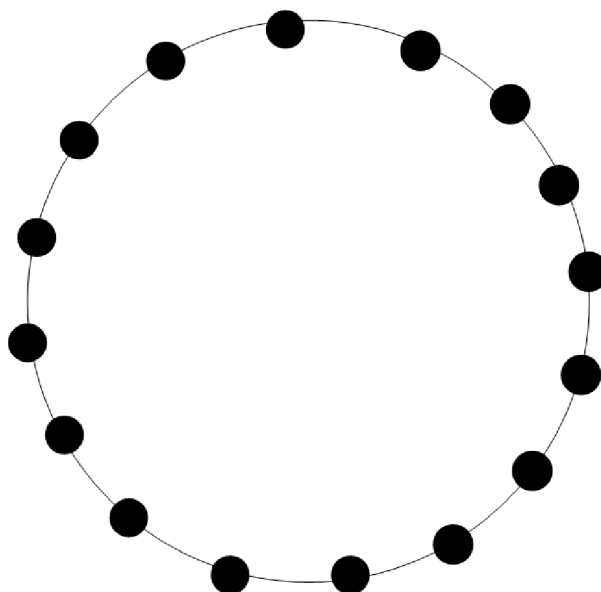
Obr. 3.7: Topologie linky.

3.4.2 Kruhová topologie

Kruhová topologie jako na obr. 3.8 je uspořádána tak, že jednotlivé stanice jsou navzájem propojeny a tvoří uzavřený kruh, ve kterém data procházejí v daném kruhu stanic v jednom směru, kde každé zařízení působí jako opakovač, z důvodů udržení silného signálu při přenosu. Každé zařízení obsahuje přijímač pro příchozí signál a vysílač, který odesílá data na další zařízení v dané cestě. Síť závisí na schopnosti signálu pohybovat se v kruhu, tedy musí procházet každým zařízením v kruhu, dokud nedosáhne svého cíle. Každý uzel je tak kritickým spojením.

3.4.3 Hvězdicová topologie

Topologie je uspořádána tak, aby všechny stanice byly připojeny k hlavnímu centrálnímu rozbočovači. Stanice jsou typicky jako klienti a hlavní centrální rozbočovač je označován jako přepínač. Jedná se o nejjednodušší způsob uspořádání topologie sítě s jedním centrálním hlavním prvkem připojeným k několika dalším funkčním stanicím. Celá síťová komunikace prochází daným centrální ústřednou a celý přenos



Obr. 3.8: Kruhová topologie.

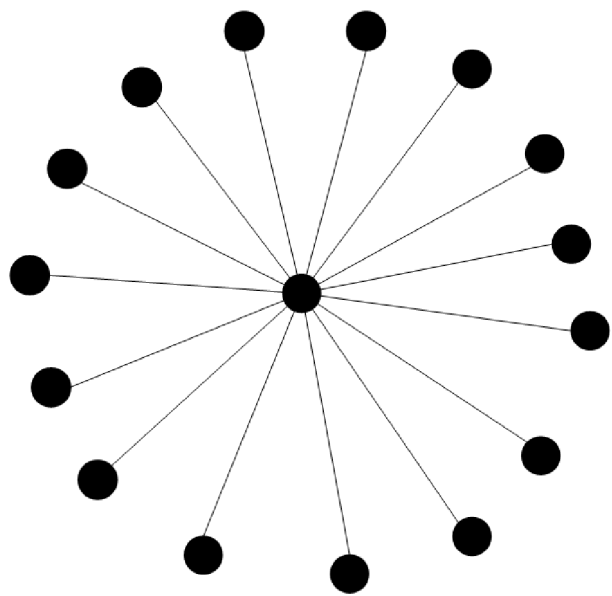
dat probíhá prostřednictvím hlavního přepínače. Hlavní výhodou takového uspořádání je jednoduchost a snadnost, s jakou se mohou přidávat uzly. Nevýhodou však je, pokud dojde k chybě v centrálním uzlu, v tom případě vypne celá síť. Příklad topologie můžeme vidět na obr. 3.9.

3.4.4 Stromová topologie

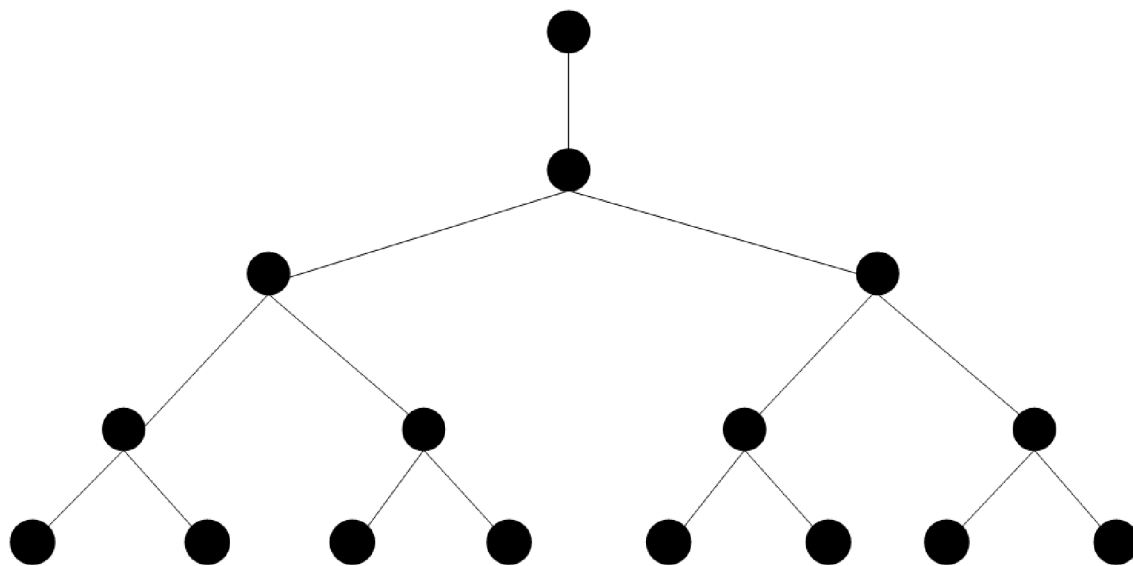
Stromová topologie jako na obr. 3.10 je složena z hlavního funkčního kořenového uzlu, který je pak připojen k dalším prvkům. Jinými slovy je tato topologie spojena jako rozvětvená síť terminálů. Každá úroveň terminálů má určité množství uzlů, které se mohou připojit. Stromové sítě nejsou omezeny počtem úrovní. Topologie je založena na hierarchii uzlů.

3.4.5 Plně konektovaná mřížka

Plně konektovaná mřížka nebo také úplná topologie na obr. 3.11. Jedná se o topologii, ve které je každé zařízení propojené s každým „úplná“ nebo může být použita alternativa, kdy se některé spoje vynechají „částečná“. Data přenášena z jednoho zařízení jsou odesílána a přijímána všemi ostatními. Tímto způsobem jsou data přenášena v cestě, která není jednosměrná, ale je řešena ve více přístupových cestách, kdy všechny pracovní stanice fungují při přijímání a odesílání dat najednou. Nevýhodou je zbytečná kapacita přenosu dat, stejně jako složitost a vysoké náklady při sestavování. Výhodou této topologie je velká spolehlivost. Když některý spoj

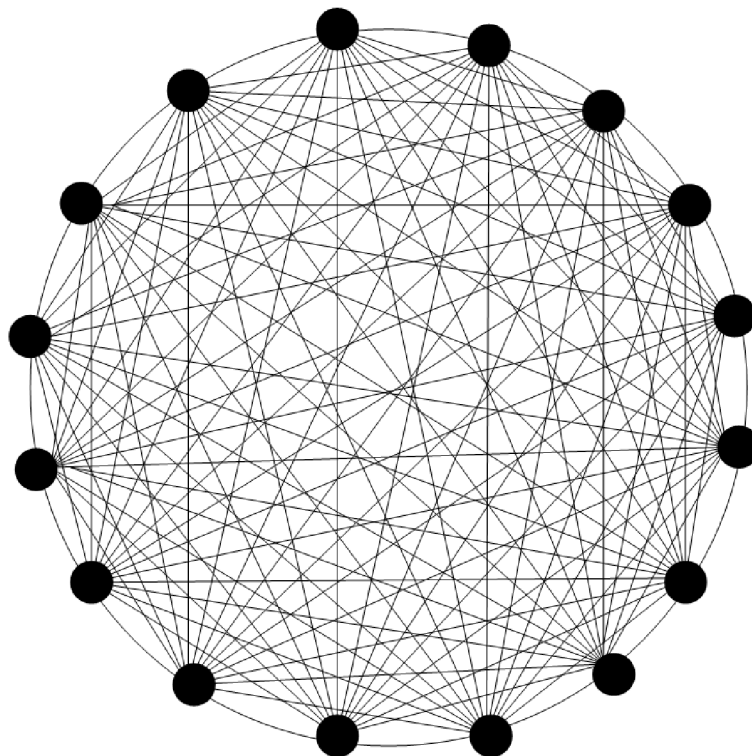


Obr. 3.9: Hvězdicová topologie.



Obr. 3.10: Stromová topologie.

vypadne data si k cíli najdou jinou cestu. Tato topologie se využívá především v bezdrátových sítích [23].



Obr. 3.11: Plně konektovaná mřížka.

4 VLIV STOCHASTICKÉHO SELHÁNÍ LINEK PRO VYBRANÉ TOPOLOGIE - ODHAD PRŮMĚRU

Cílem této části byl praktický experiment zaměřený na dopad selhání spojení v distribuovaném systému pro výše vybrané topologie, a to linkovou, kruhovou, hvězdicovou, stromovou a plně konektovanou mřížku. V případě odhadu průměru byl zkoumán vliv chyby na odchylku finální hodnoty od reálné hodnoty průměru, rychlosti konvergence algoritmu, distribuce finálních stavů a distribuce rychlostí konvergence. Pokud došlo k selhání propojení mezi dvěma danými uzly, tak to ještě nemusí znamenat, že určitá zpráva nebude doručena, i za podmínky, že byla zpráva ztracena v důsledku poruchy spojení.

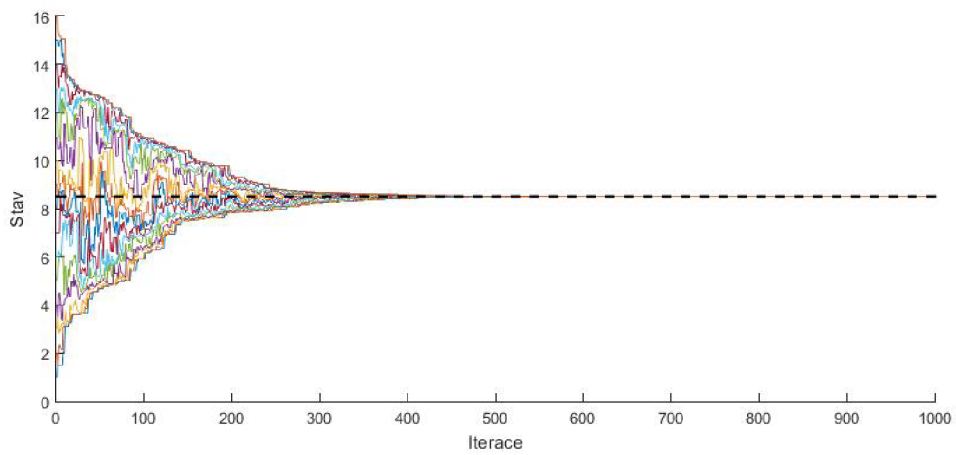
4.1 Vliv chyby na vnitřní stavy uzlů v topologii linky

V následující části byl zkoumán vliv chyby na vnitřní stavy uzlů a hodnot, ke které konvergují. Jako objekt zájmu byla vybrána topologie linky, ve které byl zvolen bezchybový scénář, tedy s výskytem 0 % chyby, a dále pak scénáře s pravděpodobností výskytu chyby 30 %, 60 %, 90 %, kde pro každou pravděpodobnost byl experiment zopakován 3 x, avšak kvůli limitovanému rozsahu práce byl zobrazen jen jeden graf pro každou pravděpodobnost chyby.

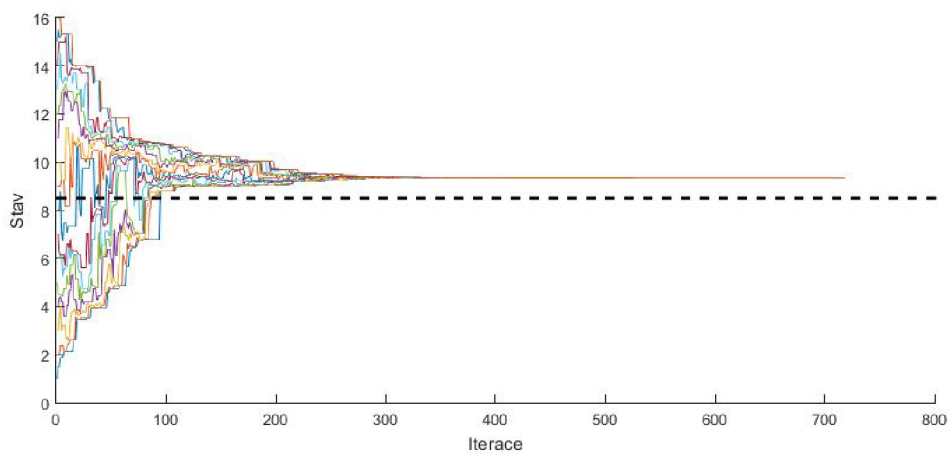
Na obr. 4.1 lze pozorovat bezchybový stav linek, které konvergují k aritmetickému průměru inicializačních hodnot uzlů, což je 8,5. V dalších experimentech s pravděpodobností výskytu chyby 30 % (obr. 4.2) a 60 % (obr. 4.3) si nelze nepovšimnout snižujícího se nárůstu počtu iterací a také jiné hodnotě, ke které síť konverguje, což je způsobeno stochastickou náhodnou chybou linky na trase. Při 90 % (obr. 4.4) pravděpodobnostním výskytu chyby pozorujeme výrazné zpomalení algoritmu a nejvyšší chybu způsobenou selháváním linek.

4.2 Odklon finální hodnoty od skutečné hodnoty průměru ve vybraných topologiích

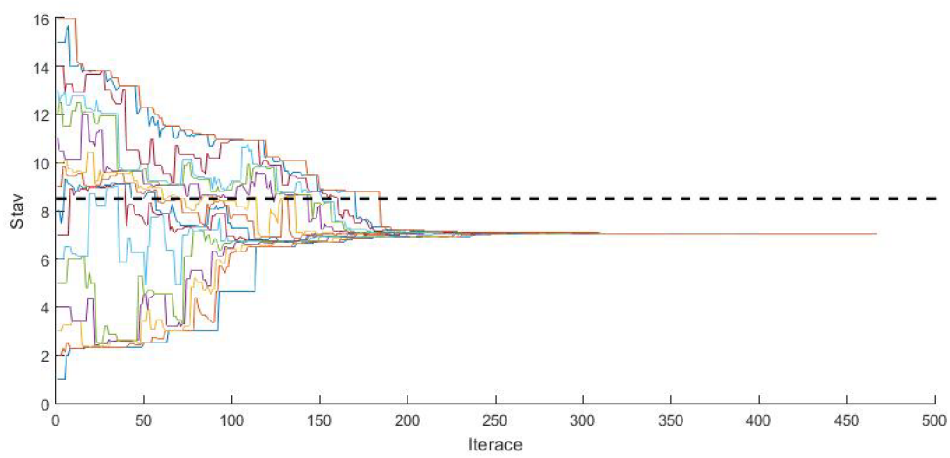
V této části byl zkoumán odklon finální hodnoty od skutečného aritmetického průměru všech hodnot v důsledku selháváním linek.



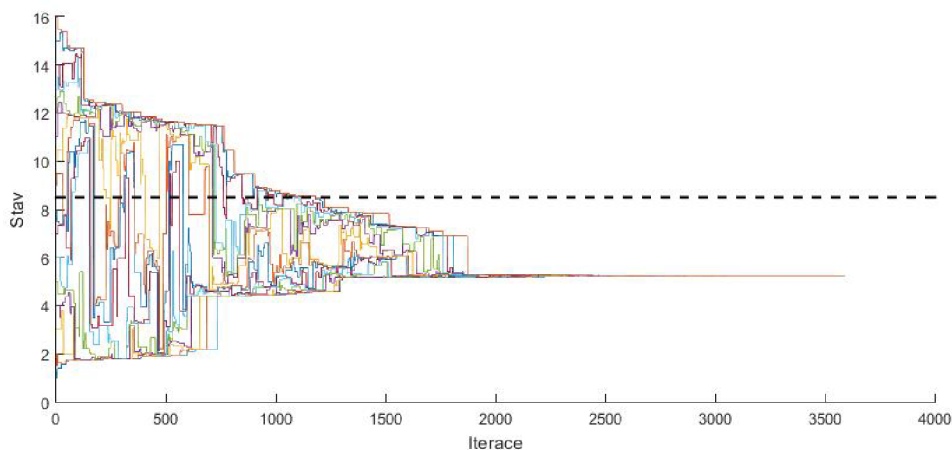
Obr. 4.1: Linka $p = 0$.



Obr. 4.2: Linka $p = 0,3$.



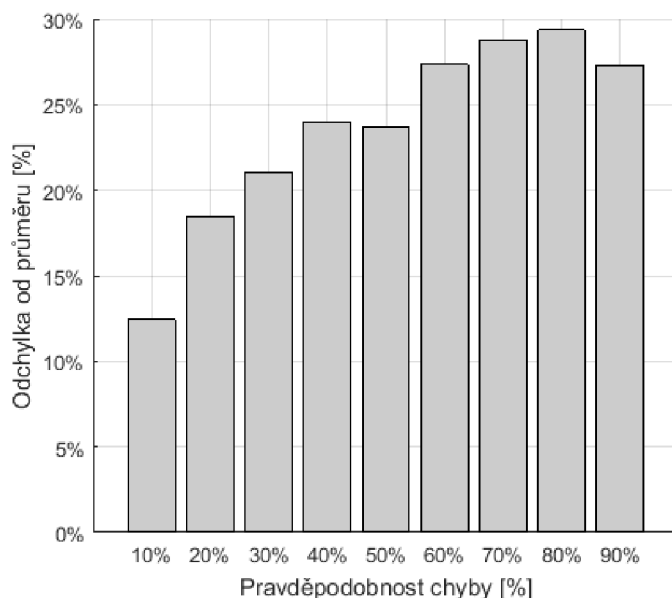
Obr. 4.3: Linka $p = 0,6$.



Obr. 4.4: Linka $p = 0,9$.

4.2.1 Odchylka v topologii linky

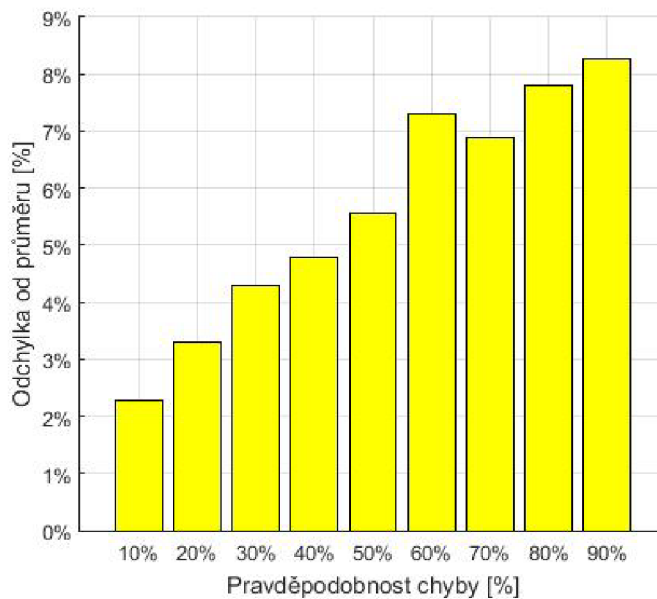
Na obr. 4.5 je znázorněna odchylka konvergence (odklonu) od aritmetického průměru, ve které lze vidět, že se vzrůstající pravděpodobností výskytu chyby (0 % až 90 %) vzrůstá i odchylka od průměru. Avšak můžeme vidět dvě výjimky a to při pravděpodobnosti 50 % a 90 %, kdy algoritmus dosáhl menší odchylky, než v případě nižší pravděpodobnosti nastalé chyby. Nejvyšší odchylky se dosáhlo při pravděpodobnosti 80 %, kdy odchylka činila téměř 30 % od průměru.



Obr. 4.5: Odklon finální hodnoty v topologii linky.

4.2.2 Odchylka v topologii plně konektované mřížky

Další experiment byl zaměřen na plně konektovanou mřížku, kde charakter odchylky průměru (obr. 4.6) je podobný jako v předchozím případě u topologie linky. Při výskytu 70 % chyby vznikla menší odchylka, než v případě nižší pravděpodobnosti nastalé chyby. Při porovnání odchylky u topologie linky a plně konektované mřížky bylo zjištěno, že u topologie plně konektované mřížky je algoritmus výrazně přesnější, než u topologie linky. Maximální hodnota odchylky činila něco málo přes 8 %.



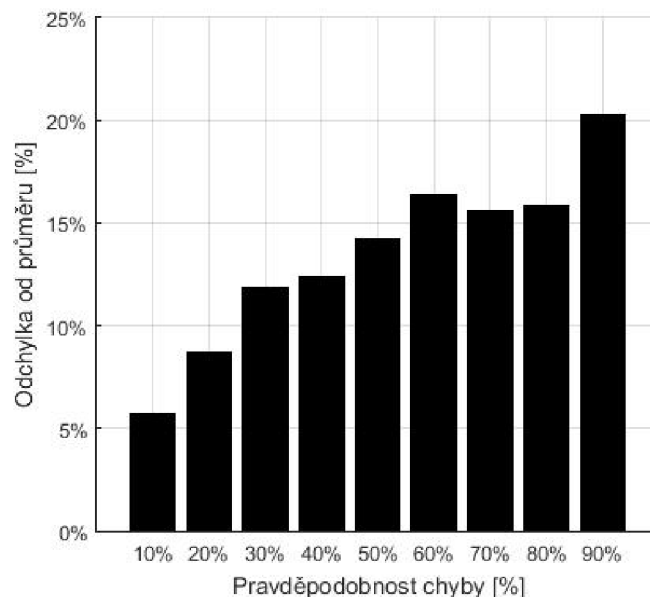
Obr. 4.6: Odklon finální hodnoty v topologii plně konektované mřížky.

4.2.3 Odchylka v kruhové topologii

Další zkoumaná topologie byla tvaru kruh, kdy charakter průběhu (obr. 4.7) je podobný, jako u předešlé topologie plně konektované mřížky, tedy odchylka není rostoucí. Odchylka od průměru pro pravděpodobnost nastalé chyby rovné 60 % byla vyšší než pro 70 % a 80 %. Maximální odchylka při pravděpodobnostní chybě 90 % činila přes 20 %.

4.2.4 Odchylka v topologii hvězdy

U topologie hvězdy je odchylka od průměru všech inicializačních hodnot rostoucí, tedy čím vyšší pravděpodobnost výskytu chyby, tím vyšší odchylka od průměru. Výsledek experimentu lze pozorovat na obr. 4.8. Maximální odchylka nedosáhla ani 8 %.



Obr. 4.7: Odklon finální hodnoty v kruhové topologii.

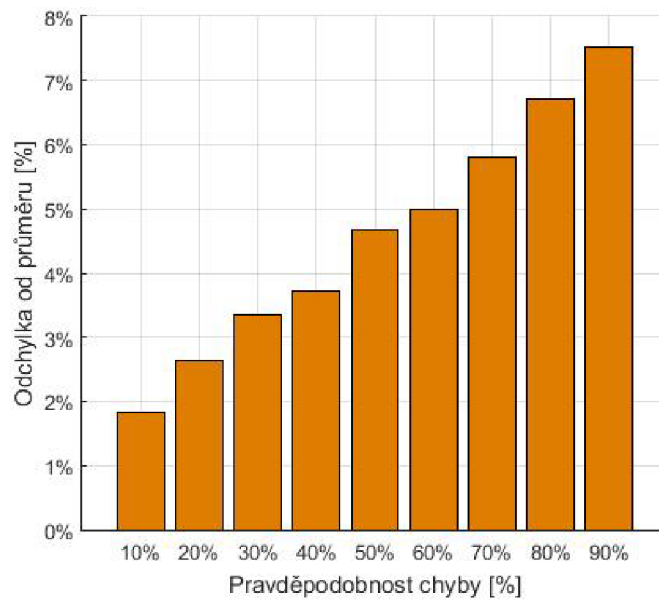
Algoritmus byl v této topologii nejpřesnější v porovnání s ostatními zkoumanými topologiemi.

4.2.5 Odchylka ve stromové topologii

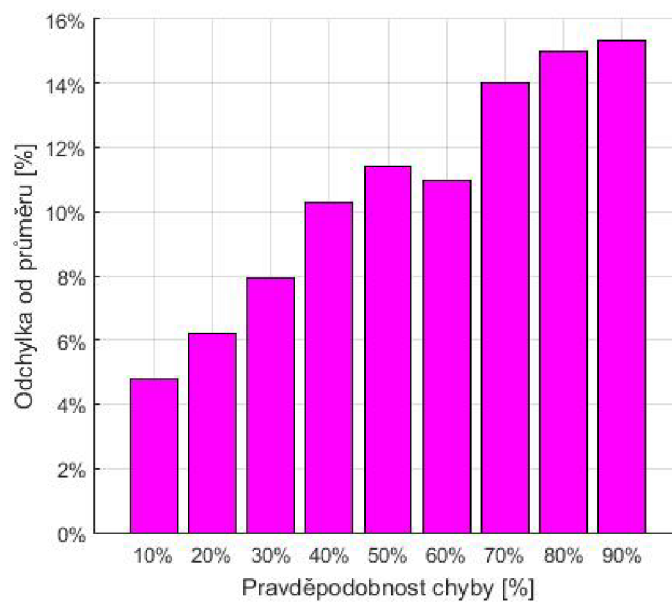
Na obr. 4.9 odklonu finální hodnoty konvergence ve stromové topologii lze pozorovat, že průběh se vzrůstající pravděpodobností výskytu chyby je rostoucí, kromě narušení charakteru vizualizace při 60 % pravděpodobnosti, kde hodnota odklonu činila o přibližně 0,5 % méně, než u předcházející pravděpodobnosti chyby. Maximální odklon finální hodnoty při 90 % pravděpodobnosti chyby činil necelých 16 %.

4.2.6 Porovnání dosažených hodnot odchylek ve všech topologiích

Z experimentů bylo zjištěno, že průměrná míra konvergence, je menší pro malé hodnoty pravděpodobnosti. Dále bylo zjištěno, že nejpřesnější byl algoritmus u struktury hvězdy, kde odchylka činila maximálně 7,5 % a plně konektované mřížky s odchylkou maximálně 8,3 %. Nižší hodnoty konvergence jsou způsobeny lepším propojením dané struktury. Nejvyšší hodnotu odklonu měla topologie linky s necelými 30 % a pak topologie kruhu s 20 %.



Obr. 4.8: Odklon finální hodnoty v topologii hvězda.



Obr. 4.9: Odklon finální hodnoty ve stromové topologii.

4.3 Vliv selhání vazeb na rozmanitost konečných výsledků

Následující experiment je zaměřen na zkoumání vlivu selhání spojení na rozmanitost konečných výsledků, ke kterým uzly konvergují. Experiment byl prováděn pro hodnoty pravděpodobnosti 10 %, 50 % a 90 % pro výše zmíněné topologie, tedy topologie linky (obr. 4.10), kruhu (obr. 4.11), stromu (obr. 4.12), hvězdy (obr. 4.13) a plně konektované mřížky (obr. 4.14).

Z výsledků lze zaznamenat, že charakter pozorovaného jevu je pro všech pět typů zkoumaných sítí stejný. Vyšší hodnota pravděpodobnosti způsobí, že konečné výsledky budou rozmanitější, tedy širší. Ve slabě propojených strukturách je rozptyl významnější.

4.4 Vliv selhání linek na počet iterací

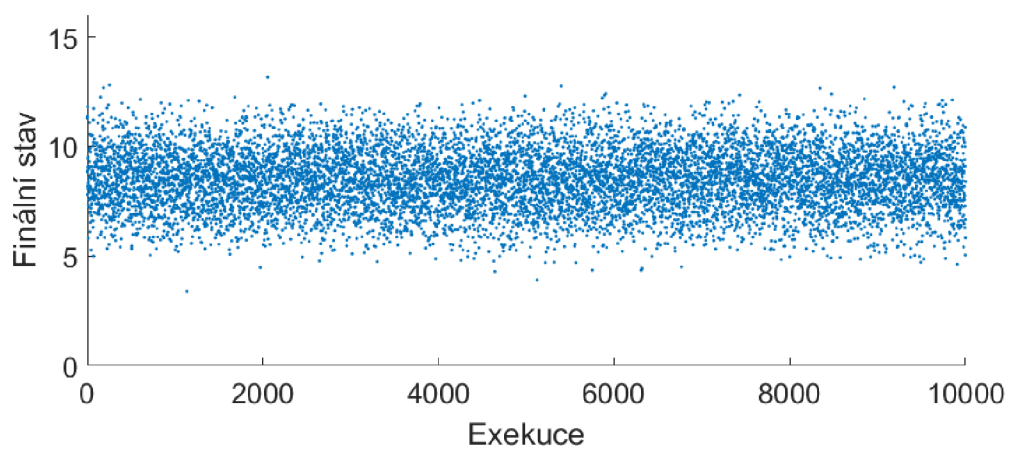
V následující části byl zkoumán vliv selhání linek na výše uvedených topologiích pro pravděpodobnosti 10 %, 20 %, 30 %, 40 %, 50 %, 60 %, 70 %, 80 % a 90 % na počet provedených iterací, tedy zdali dojde ke zpomalení algoritmu, nebo naopak k jeho zrychlení. Dále pak byla posuzována distribuce rychlostí konvergence u 10 000 pokusech opakování pro pravděpodobnosti výskytu chyby 0 %, 20 %, 60 %, 80 %.

4.4.1 Topologie linky

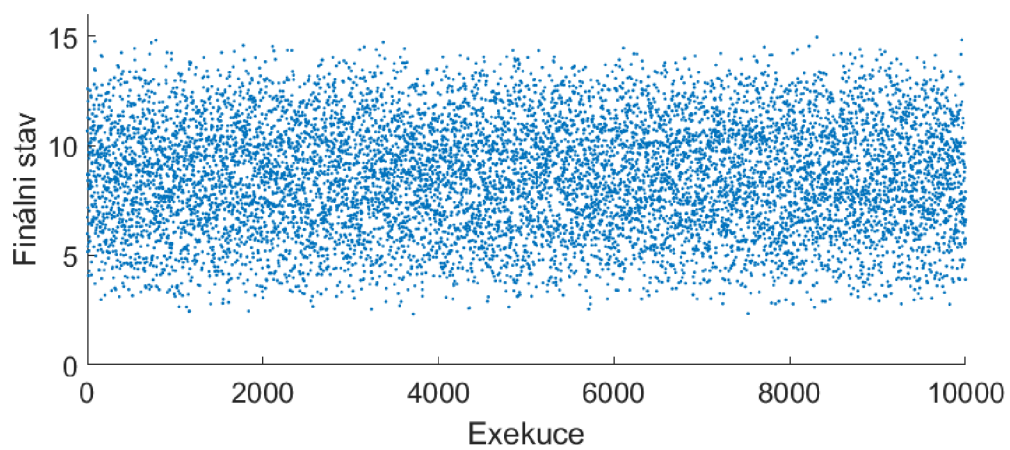
Na obr. 4.15 lze pozorovat, že při pravděpodobnosti výskytu chyby 10 % až 70 % dojde ke zrychlení algoritmu v průměru okolo 15 %, což je způsobeno slabě propojenou strukturou. Velký nárůst počtu iterací je až při 80 % a 90 %, kdy dojde téměř až ke 130 % zpomalení algoritmu. Na obr. 4.16 lze vidět četnost korespondujících rychlostí pro 10 000 opakování. Nejvyšší rozmanitost výsledků a také nejvyšší nárůst iterací lze zaznamenat při nejvyšší pravděpodobnostní chybě. Nejvíce stabilní rychlost konvergence je u bezchybného stavu linky.

4.4.2 Topologie plně konektované mřížky

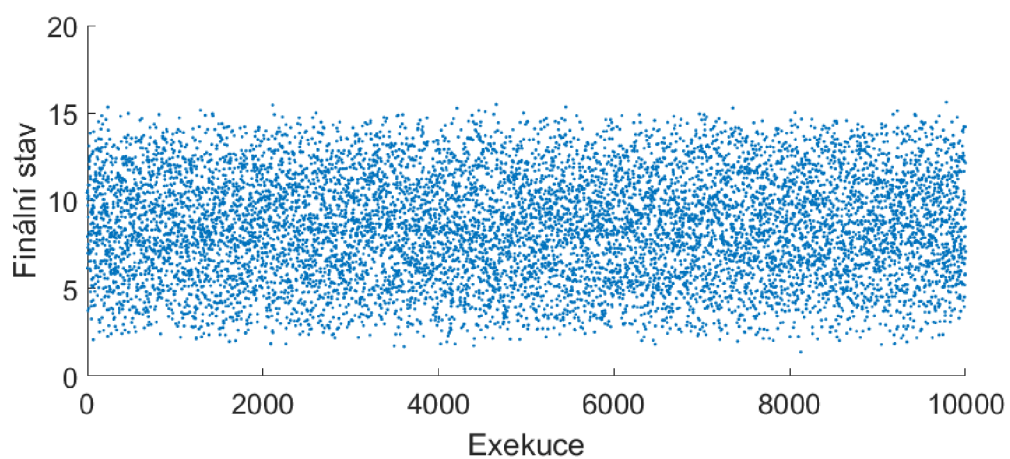
V topologii plně konektované mřížky má zpomalení algoritmu rostoucí charakter, tedy od nejnižší pravděpodobnosti výskytu chyby až po nejvyšší, což můžeme vidět na obr. 4.17, kdy v plně propojené síti dojde k velkému nárůstu iterací a tedy zpomalení algoritmu téměř až o 600 %. Pokud porovnáme četnost výsledků na počet iterací (obr. 4.18), tak zjistíme, že se vzrůstající pravděpodobností výskytu chyby se zvyšuje jak počet iterací, tak i rozmanitost výsledků.



(a) pravděpodobnost 10 %

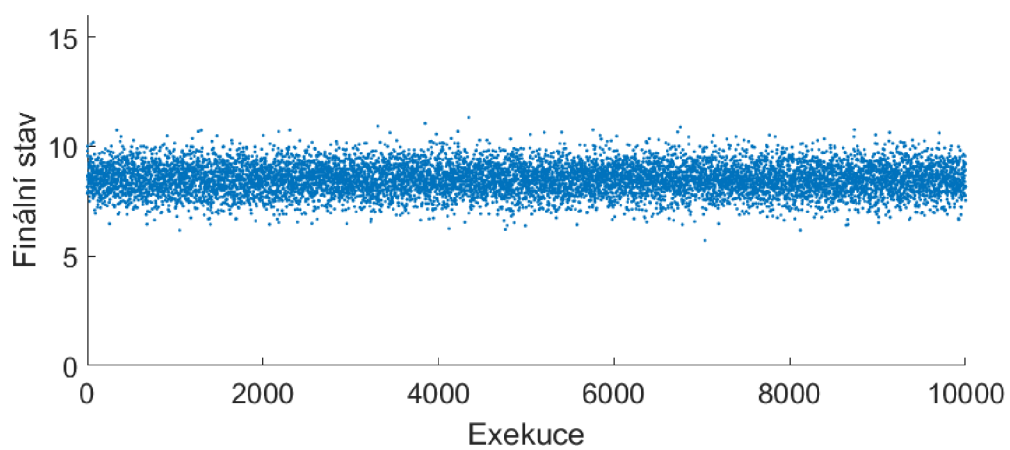


(b) pravděpodobnost 50 %

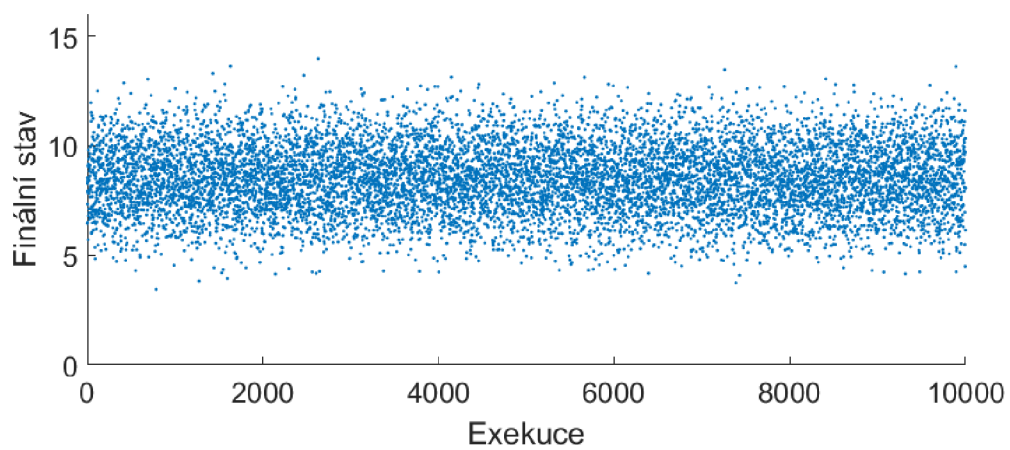


(c) pravděpodobnost 90 %

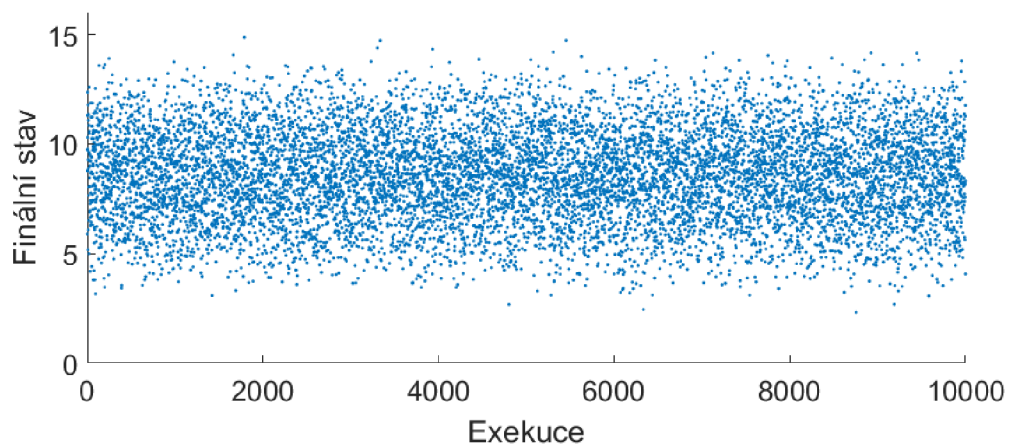
Obr. 4.10: Porovnání vlivu selhání linek v linkové topologii.



(a) pravděpodobnost 10 %

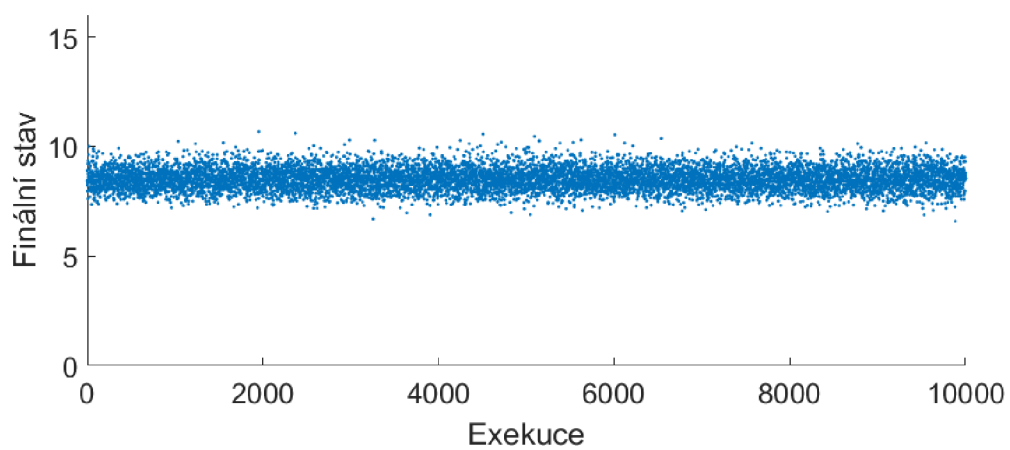


(b) pravděpodobnost 50 %

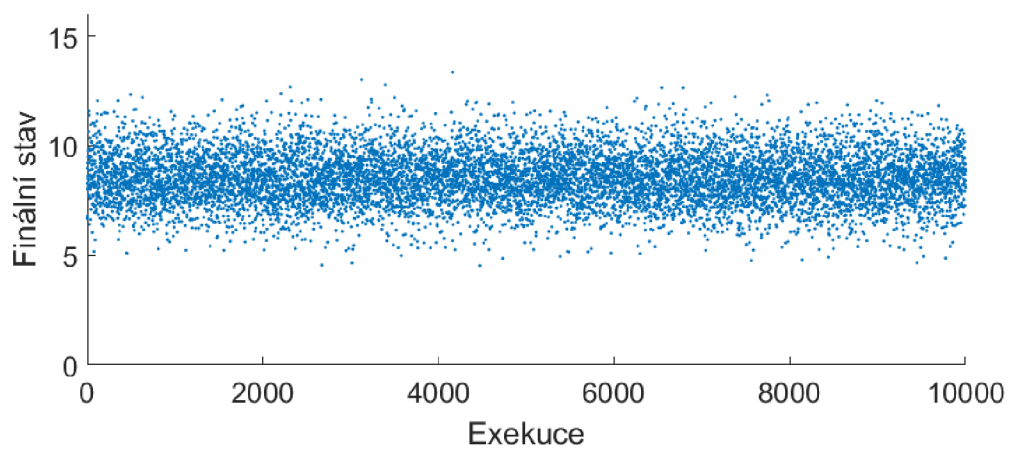


(c) pravděpodobnost 90 %

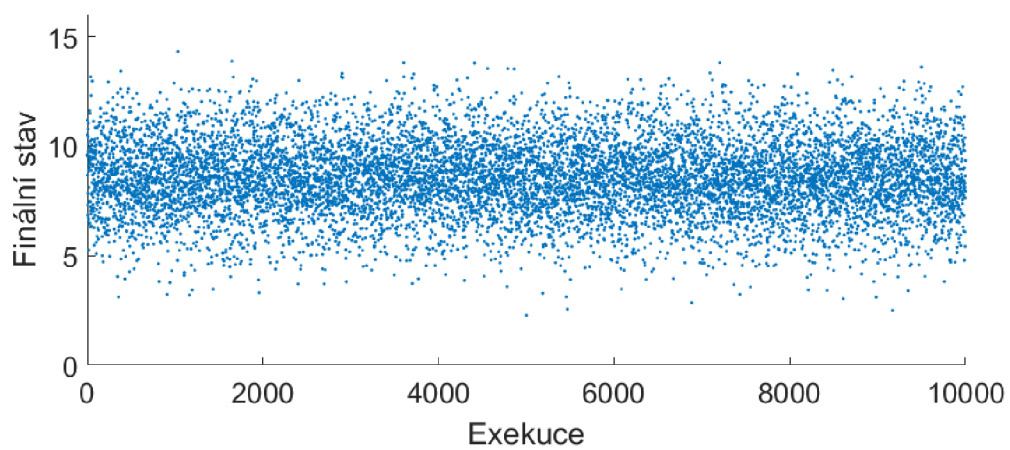
Obr. 4.11: Porovnání vlivu selhání linek v kruhové topologii.



(a) pravděpodobnost 10 %

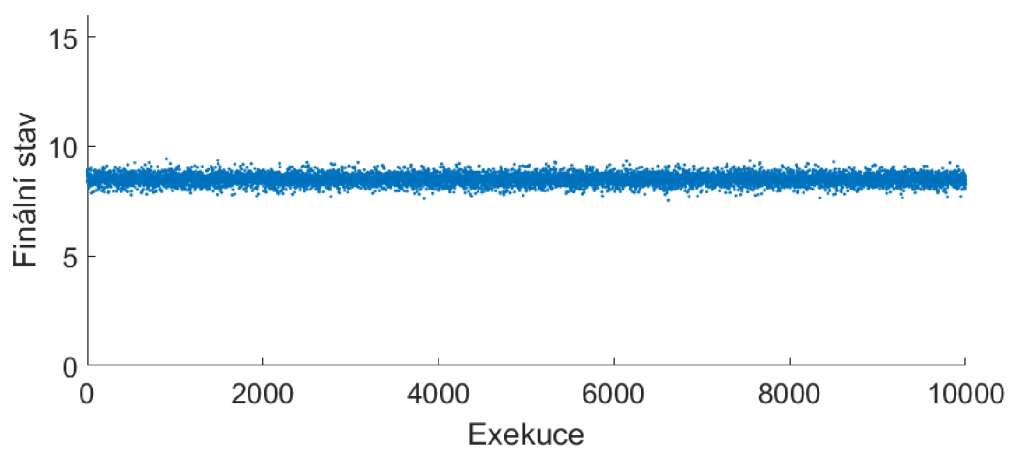


(b) pravděpodobnost 50 %

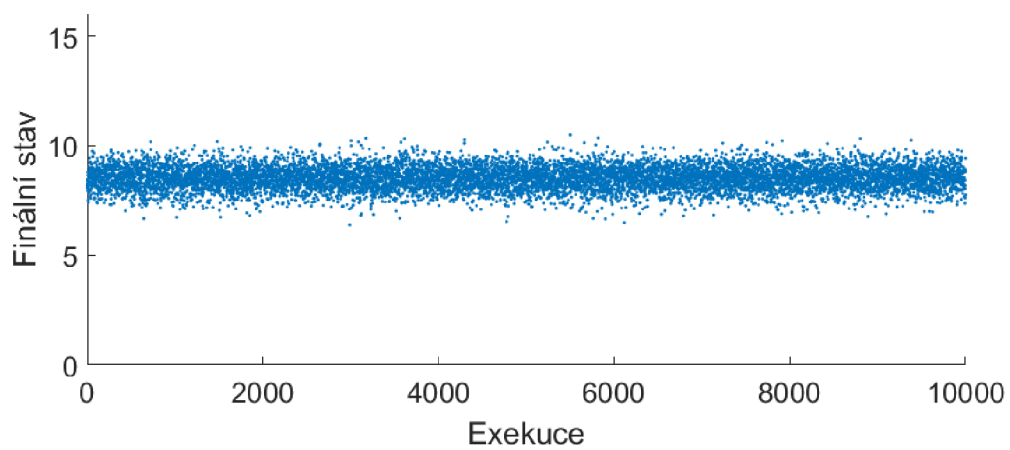


(c) pravděpodobnost 90 %

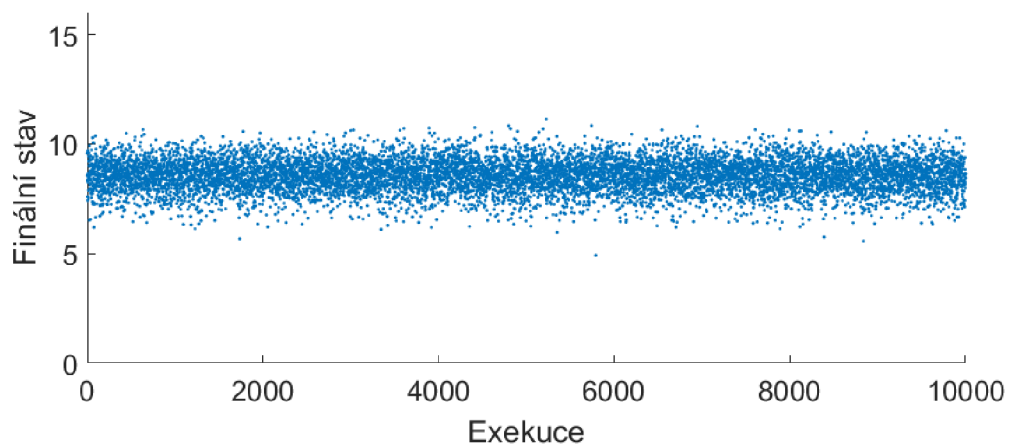
Obr. 4.12: Porovnání vlivu selhání linek ve stromové topologii.



(a) pravděpodobnost 10 %

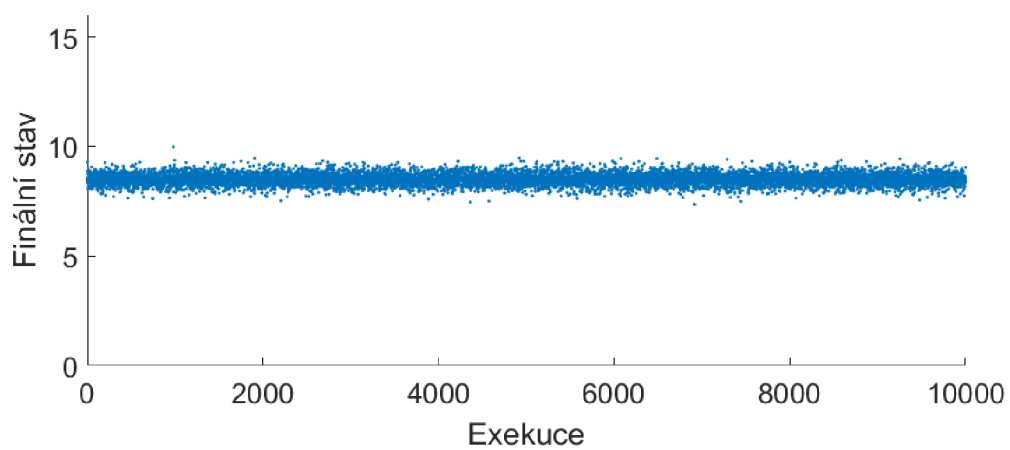


(b) pravděpodobnost 50 %

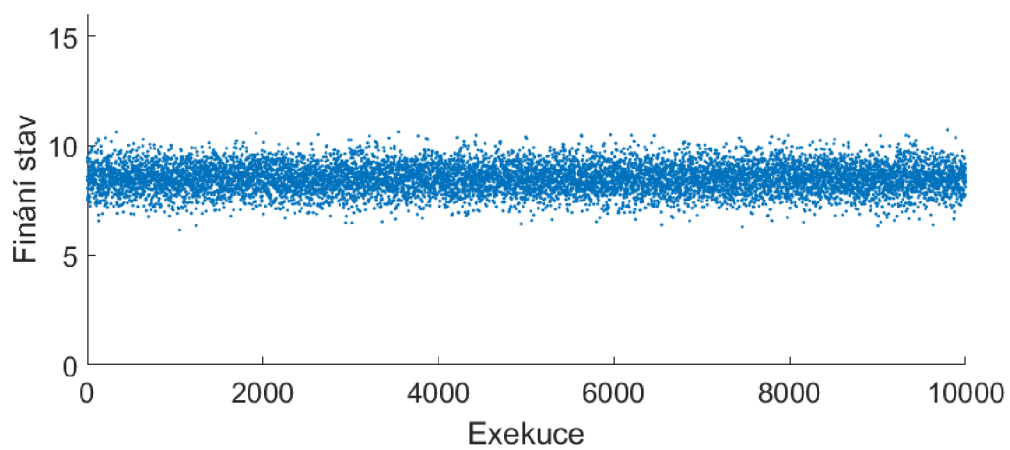


(c) pravděpodobnost 90 %

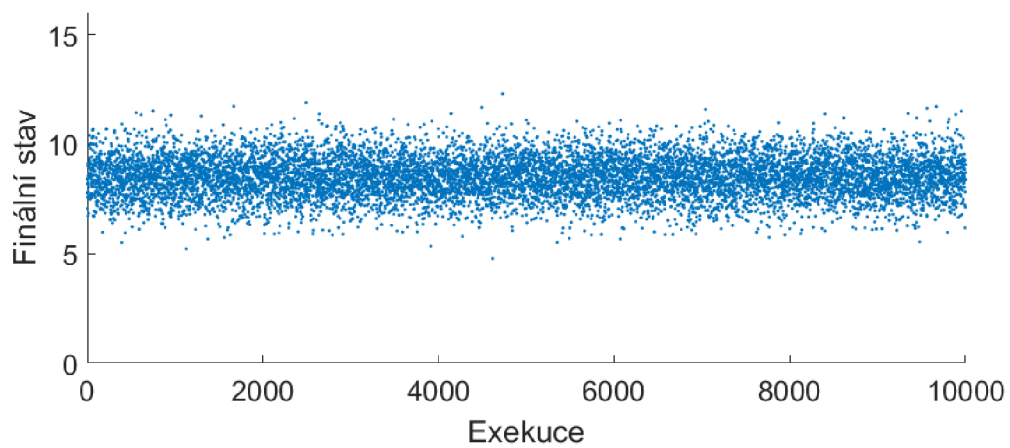
Obr. 4.13: Porovnání vlivu selhání linek ve hvězdicové topologii.



(a) pravděpodobnost 10 %

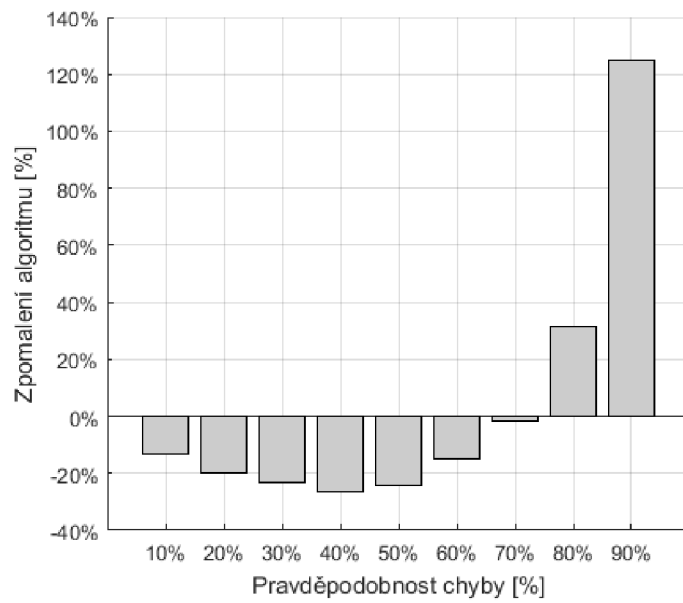


(b) pravděpodobnost 50 %

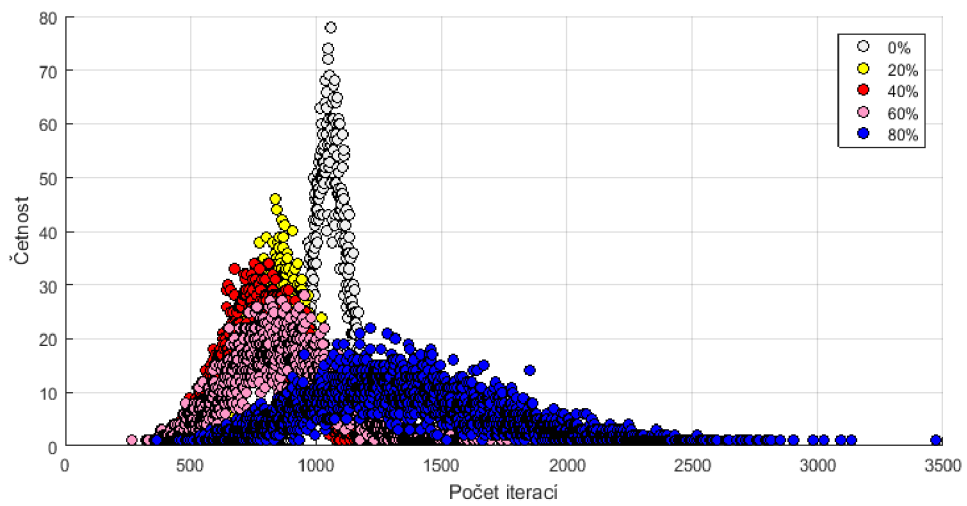


(c) pravděpodobnost 90 %

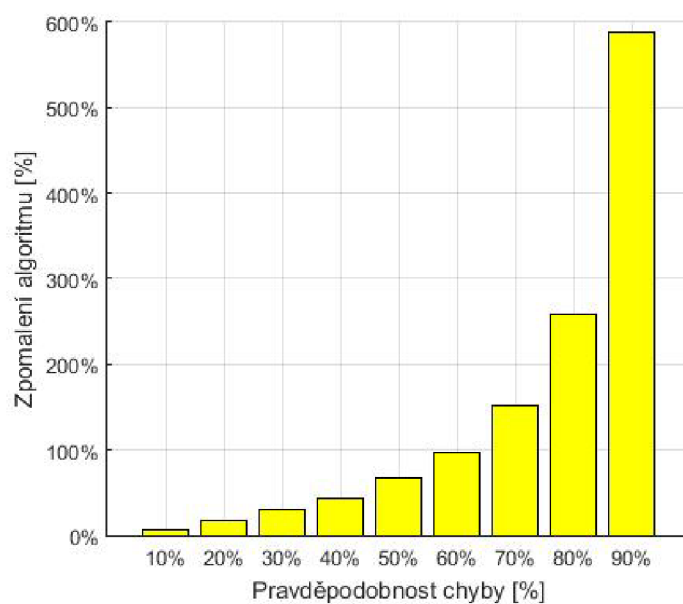
Obr. 4.14: Porovnání vlivu selhání linek v plně konektované mřížce.



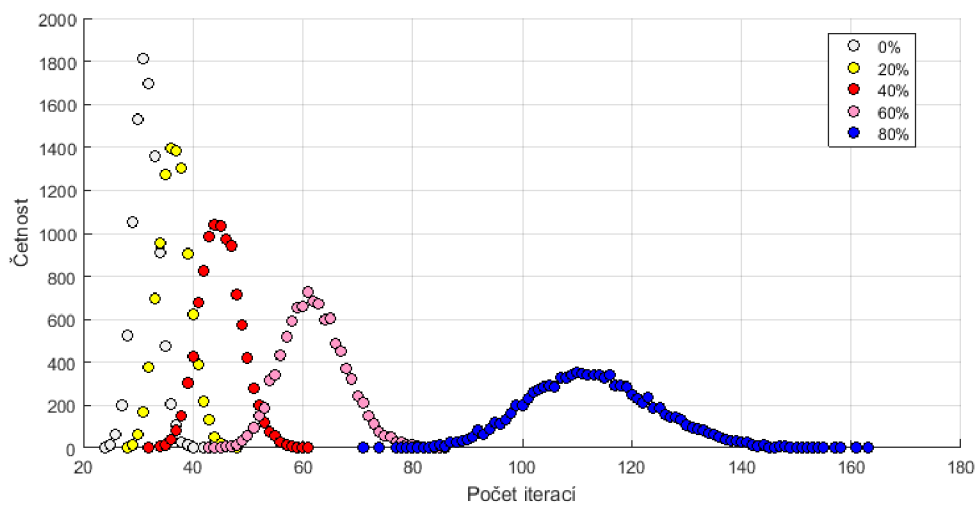
Obr. 4.15: Vliv selhání linek na počet iterací v topologii linky.



Obr. 4.16: Distribuce počtu iterací v topologii linky.



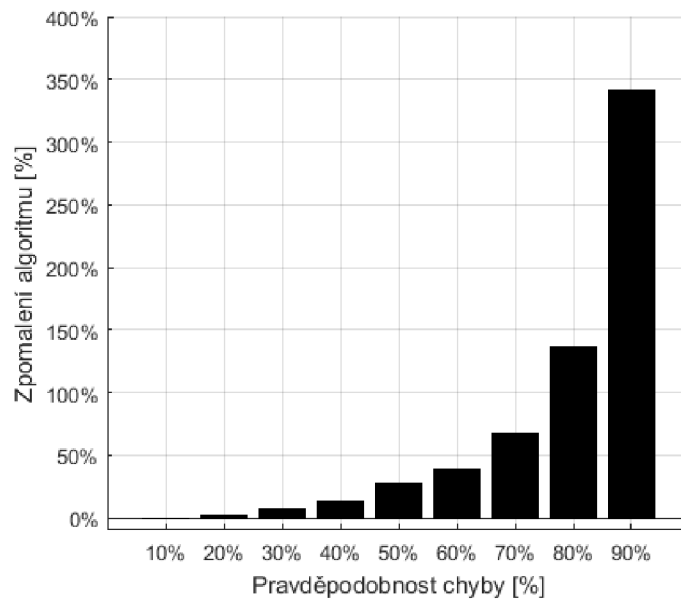
Obr. 4.17: Vliv selhání linek na počet iterací v topologii plně konektované mřížky.



Obr. 4.18: Distribuce počtu iterací v topologii plně konektované mřížky.

4.4.3 Topologie kruhu

Selhání linky v topologii kruhu na zpomalení algoritmu má podobný charakter (obr. 4.19), jako u plně konektované mřížky s tím rozdílem, že u pravděpodobnosti výskytu chyby 10 % dojde ke zanedbatelnému zrychlení algoritmu, jak lze vidět na obr. 4.20. Nejvyššího zpomalení algoritmu téměř až o 350 %, se dosáhlo u pravděpodobnostní chyby 90 %. Pokud porovnáme distribuci počtu iterací s předchozí topologií, tak lze pozorovat, že četnost výsledků a počet iterací je rozmanitější (obr. 4.21), nejvíce však pravděpodobnosti výskytu chyby při 80 %.



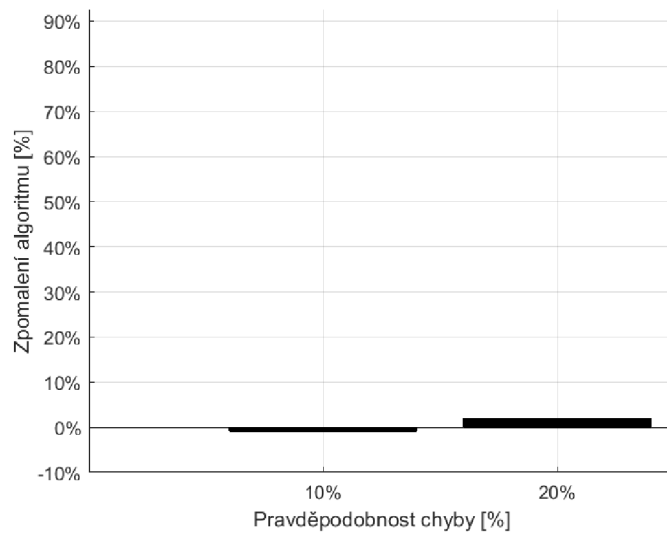
Obr. 4.19: Vliv selhání linek na počet iterací v kruhové topologii.

4.4.4 Topologie hvězdy

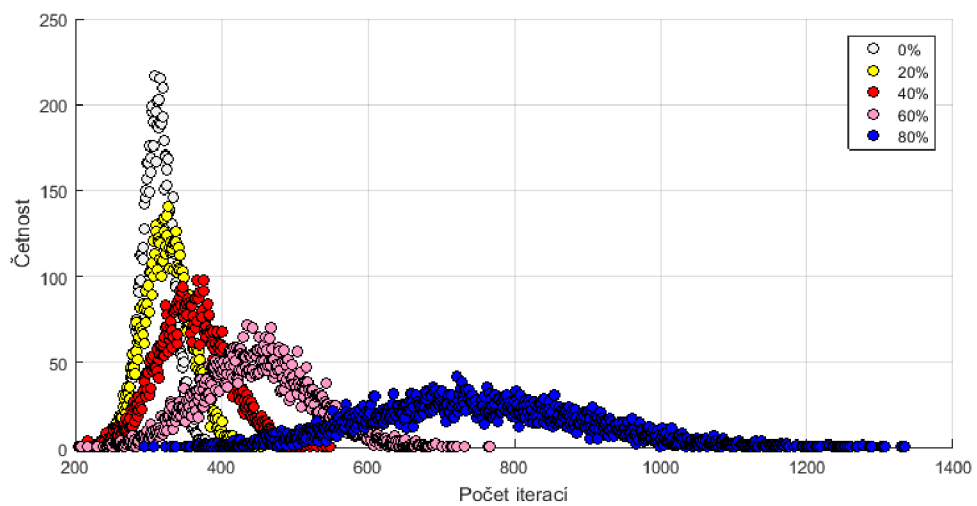
V topologii hvězdy je zpomalení algoritmu z hlediska výskytu chyby rostoucí, jako je znázorněno na obr. 4.22. V této topologii bylo zpomalení algoritmu z hlediska mnohou zkoumaných topologií nejvyšší, a to téměř až o 900 %. Výsledky distribuce počtu iterací je velice podobná jako u předchozí topologie s tím rozdílem, že četnost výsledků se snížila a s ní i počet iterací, lze vidět na obr. 4.23.

4.4.5 Topologie stromu

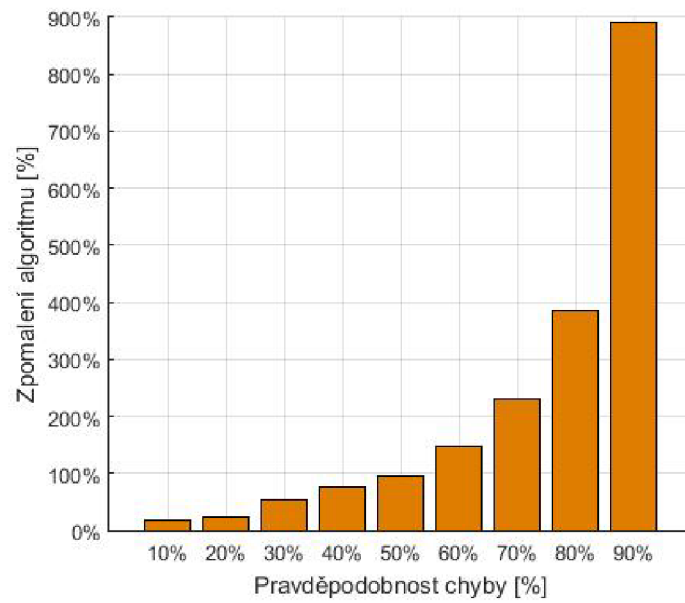
Zpomalení algoritmu v topologii stromu (obr. 4.24) má podobný charakter jako u topologie linky, tedy jde o slabě propojenou strukturu, u které se algoritmus s menší



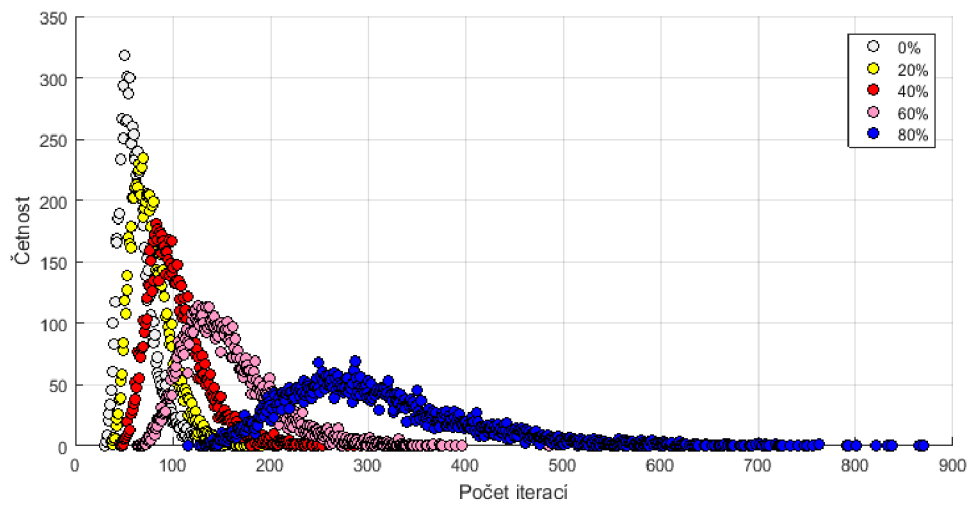
Obr. 4.20: Zoom na zrychlení algoritmu topologie kruhu.



Obr. 4.21: Distribuce počtu iterací v topologii kruhu.

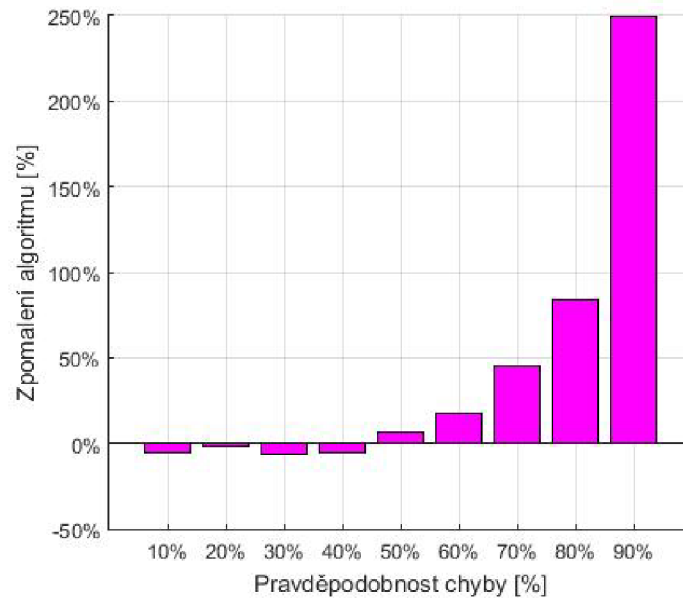


Obr. 4.22: Vliv selhání linek na počet iterací ve hvězdicové topologii.



Obr. 4.23: Distribuce počtu iterací v topologii hvězdy.

pravděpodobnostní chybou zrychluje, a to až do 40 %. Po překonání 50 % se algoritmus z hlediska pravděpodobnostní chyby zpomaluje a to až do 250 %. Distribuce počtu iterací u této topologie je po topologii linky nejvíce rozmanitá, jako na obr. 4.25, což je způsobeno slabší propojeností uzlů.

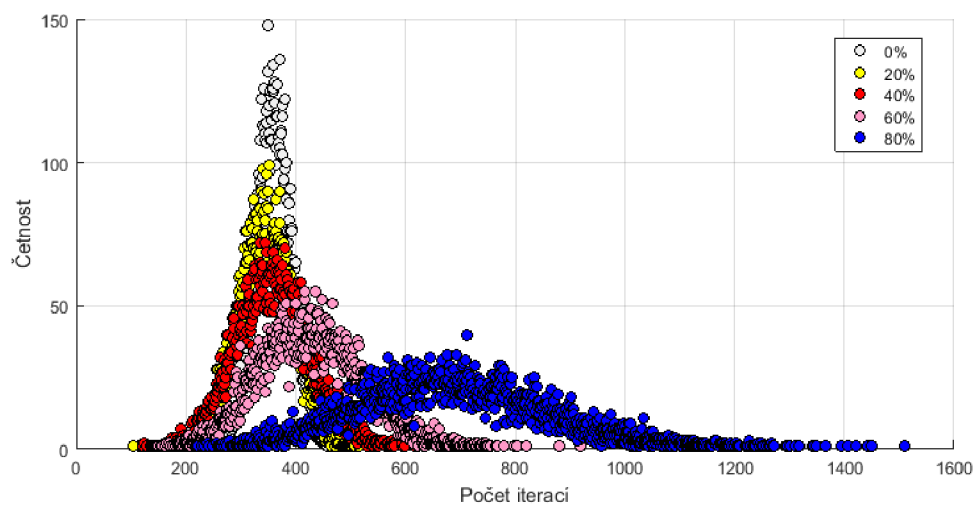


Obr. 4.24: Vliv selhání linek na počet iterací ve stromové topologii.

4.4.6 Porovnání dosažených hodnot počtu iterací ve zkoumaných topologiích

Z experimentů bylo zjištěno, čím vyšší hodnota pravděpodobnosti, tím větší zpomalení algoritmu, tedy počet iterací k dokončení konvergence začne dramaticky růst. Dále pak že pro slabě propojené struktury se charakter získaných dat liší od silně propojených a to v tom, že u selhání propojení linky se může výpočetní proces dokonce urychlit, a to zejména při menší pravděpodobnosti výskytu chyby. Tento proces urychlení není žádoucím jevem, jelikož konečné hodnoty, ke kterým uzly konvergují, se liší od skutečného průměru.

Pokud porovnáme četnost počtu iterací ve zkoumaných topologiích, tak nejvíce rozmanitých výsledků bylo u topologie linky a stromu, díky slabšímu propojení struktury. Největší počet iterací proběhl u větší pravděpodobnosti výskytu chyby, zejména největší nárůst byl u topologie linky a kruhu.



Obr. 4.25: Distribuce počtu iterací v topologii stromu.

5 VLIV STOCHASTICKÉHO SELHÁNÍ LINEK PRO VYBRANÉ TOPOLOGIE - ODHAD SUMY

Cílem této části byl praktický experiment zaměřený na dopad selhání spojení v distribuovaném systému pro topologie linky, kruhu, hvězdy, stromu a plně konektované mřížky. V případě odhadu sumy byl zkoumán vliv chyby na odchylku finální hodnoty od reálné hodnoty průměru, rychlosti konvergence algoritmu, distribuce finálních stavů a distribuce rychlostí konvergence.

5.1 Odklon finální hodnoty od skutečné hodnoty ve vybraných topologiích

V této části byl zkoumán odklon finální hodnoty od sumy všech hodnot v důsledku selhávání linek.

5.1.1 Odchylka v topologii linky

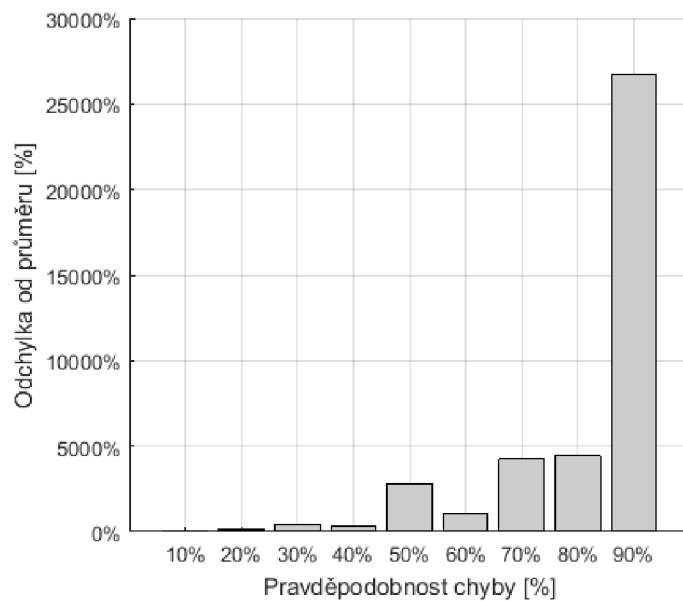
Na obr. 5.1 je znázorněn odklon od sumy všech hodnot, ve kterém lze vidět, že s rostoucí pravděpodobností chyby selhávání linek dochází k nárůstu finální hodnoty od očekávané. Avšak lze pozorovat dvě anomálie u hodnot pravděpodobnosti 40 % a 60 %, kdy je odklon menší, než pro nižší hodnoty. U 90 % pravděpodobnosti chyby nastal výrazný nárůst chyby s porovnáním s ostatními, který dosáhl okolo 27000 %.

5.1.2 Odchylka v topologii plně konektované mřížky

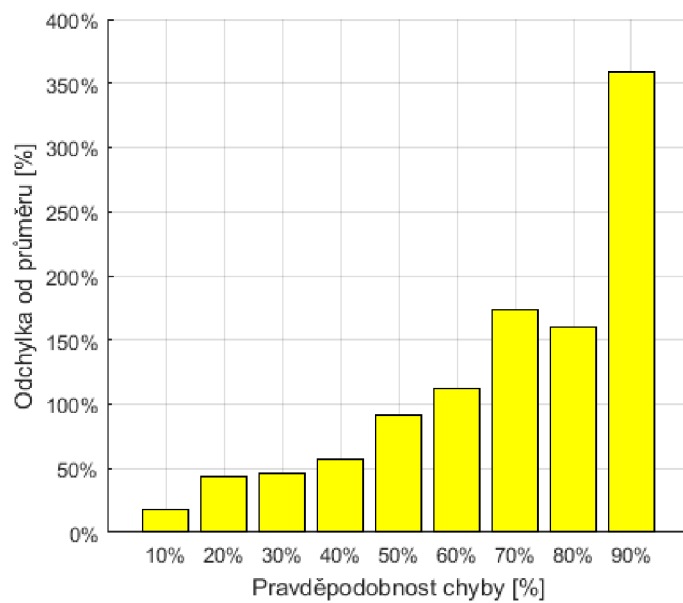
Další experiment byl zaměřen na plně konektovanou mřížku, kde charakter odklonu sumy (obr. 5.2) je podobný jako v předchozím případě u topologie linky, kde vznikla anomálie u hodnoty pravděpodobnosti 80 %. U 90 % pravděpodobnosti chyby dosáhl odklon největšího nárůstu, a to okolo 350 %, avšak tento nárůst není tak velký jako u topologie linky.

5.1.3 Odchylka v kruhové topologii

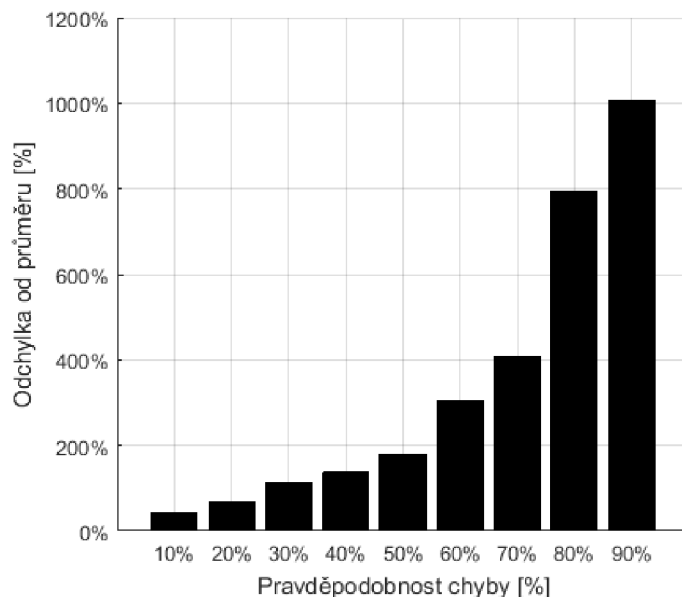
Další zkoumaná topologie byla topologie kruhu, kde charakter průběhu (obr. 5.3) je podobný, jako u předchozí topologie plně konektované mřížky s tím, že nevznikla žádná anomálie. U této topologie však nedošlo k tak prudkému nárůstu chyby u mnou zkoumané pravděpodobnosti chyby. Nejvyšší pozorovaný odklon dosáhl hodnoty více jak 1000 %.



Obr. 5.1: Odklon finální hodnoty v topologii linky.



Obr. 5.2: Odklon finální hodnoty sumy v topologii plně konektované mřížky.



Obr. 5.3: Odklon finální hodnoty v kruhové topologii.

5.1.4 Odchylka v topologii hvězdy

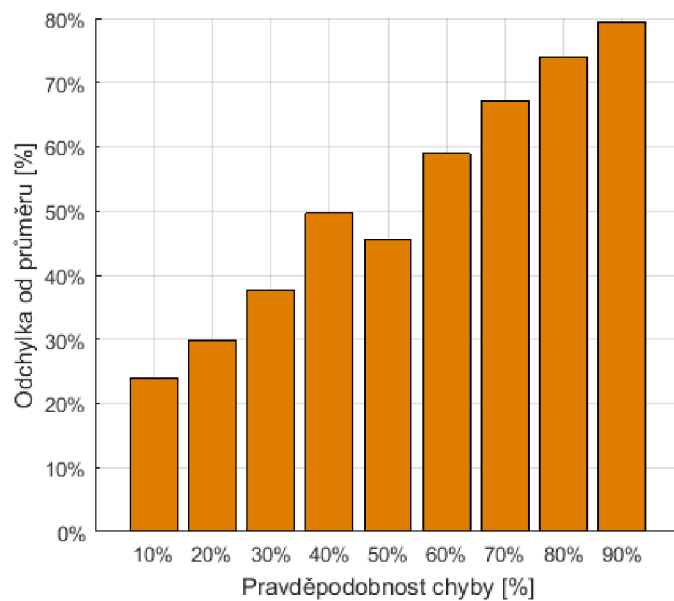
U topologie hvězdy je charakter průběhu (obr. 5.4) podobný, jako u předchozí topologie, kde anomálii lze pozorovat u pravděpodobnosti 50 %. U této topologie nedojde u nejvyšší pravděpodobnosti chyby k tak výraznému odklonu. Maximální chyba nedosahuje ani 80 %.

5.1.5 Odchylka ve stromové topologii

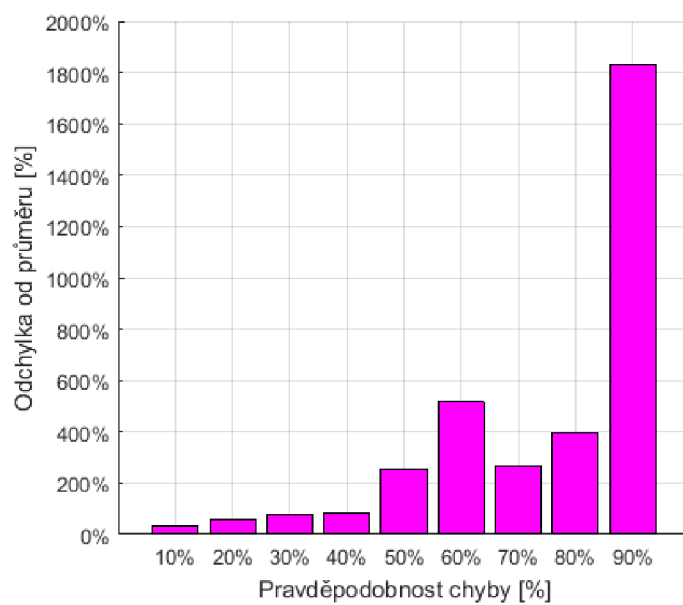
Na obr. 5.5 odklonu finální hodnoty sumy u topologie stromu lze pozorovat, že průběh je podobný jako v případě topologie linky, kde vznikla anomálie u hodnot pravděpodobnosti 70 % a 80 %. Opět lze pozorovat velký nárůst chyby u nejvyšší pravděpodobnosti chyby. Nejvyšší procentuální odklon byl zaznamenán u pravděpodobnosti 90 %, kde hodnota činila okolo 1800 %.

5.1.6 Porovnání výsledků odklonu zkoumaných topologií

V případě odklonu finální hodnoty lze pozorovat jeho nárůst se zvyšující se pravděpodobností nastalé chyby. Toto chování je stejné pro všechny typy topologií. Nejnižší odchylka byla pozorována v topologii hvězdy, kde nejvyšší odchylka představuje téměř 80 %. Protokol byl druhý nejrobustnější v topologii plně konektované mřížky, kde odchylka dosahuje téměř 350 %. Třetí nejrobustnější byl v topologii kruhu, kde



Obr. 5.4: Odklon finální hodnoty v topologii hvězda.



Obr. 5.5: Odklon finální hodnoty ve stromové topologii.

nejvyšší odchylka byla více jak 1000 %. Čtvrtý nejrobustnější byl ve stromové topologii, kde nejvyšší odchylka činila více jak 1800 %. Nejméně robustním byl protokol v topologii linky, kde nejvyšší odklon byl více jak 2500 %.

5.2 Vliv selhání vazeb na rozmanitost konečných výsledků

Následující část je zaměřena na zkoumání vlivu selhání spojení na rozmanitost finálních stavů, ke kterým uzly konvergují. Experiment byl prováděn pro hodnoty pravděpodobnosti 10 %, 50 % a 90 % pro výše zmíněné topologie, tedy topologie linky (obr. 4.10), kruhu (obr. 4.11), stromu (obr. 4.12), hvězdy (obr. 4.13) a plně konektované mřížky (obr. 4.14).

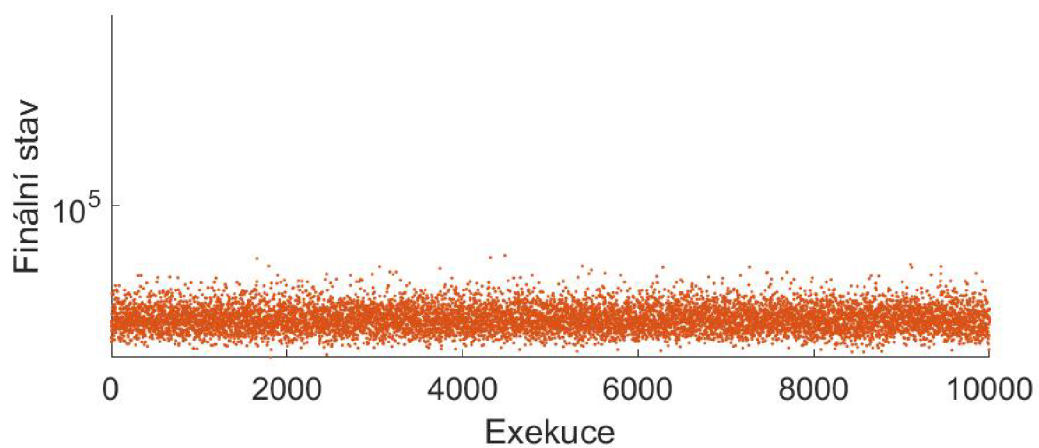
V případě rozmanitosti konečných výsledků je možné vidět, že se zvyšující pravděpodobností nastalé chyby se zvětšuje rozptyl ve všech topologiích. Ve slabě propojených strukturách byl tento rozptyl výraznější.

5.3 Vliv selhání linek na počet iterací

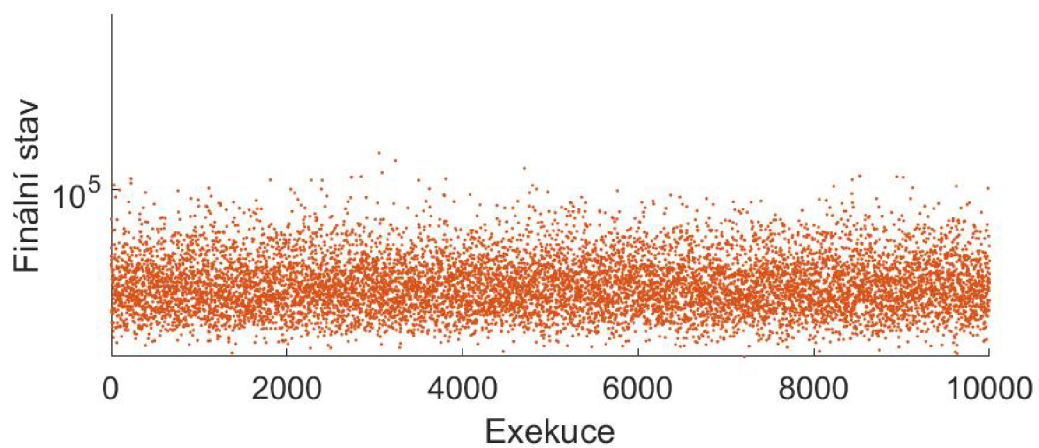
V následující části byl zkoumán vliv selhání linek na výše uvedených topologiích pro hodnoty pravděpodobnosti 10 %, 20 %, 30 %, 40 %, 50 %, 60 %, 70 %, 80 % a 90 % na počet provedených iterací, tedy zdali dojde ke zpomalení algoritmu, nebo naopak k jeho zrychlení. Dále pak byla posuzována distribuce rychlostí konvergence u 10 000 pokusech opakování pro pravděpodobnosti výskytu chyby 0 %, 20 %, 60 %, 80 %.

5.3.1 Topologie linky

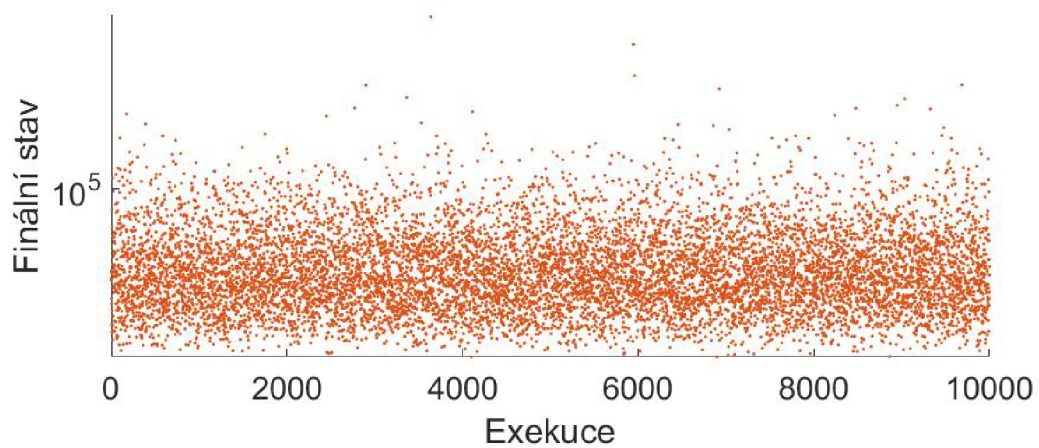
Na obr. 5.11 lze pozorovat, že u hodnot pravděpodobnosti nastalé chyby 10 % až 60 % nastalo zrychlení algoritmu. Nejnižší možné zrychlení nastalo u hodnoty pravděpodobnosti 10 %. Dále bylo zjištěno, že zrychlení roste s vyšší pravděpodobností nastalé chyby. U hodnoty pravděpodobnosti 40 % dosáhne maxima, poté začne klesat, kde u pravděpodobnosti 70 % lze pozorovat zpomalení algoritmu. Nejvyšší zpomalení nastalo u hodnoty pravděpodobnosti 90 %, kde tato hodnota činí okolo 150 %. Na obr. 5.12 lze vidět rozložení rychlosti konvergence, která má normální rozdělení a s rostoucí pravděpodobností nastalé chyby se zvyšuje rozptyl. U hodnot 20 %, 40 % a 60 % lze pozorovat zrychlení algoritmu i u distribucí rychlostí konvergence.



(a) pravděpodobnost 10 %

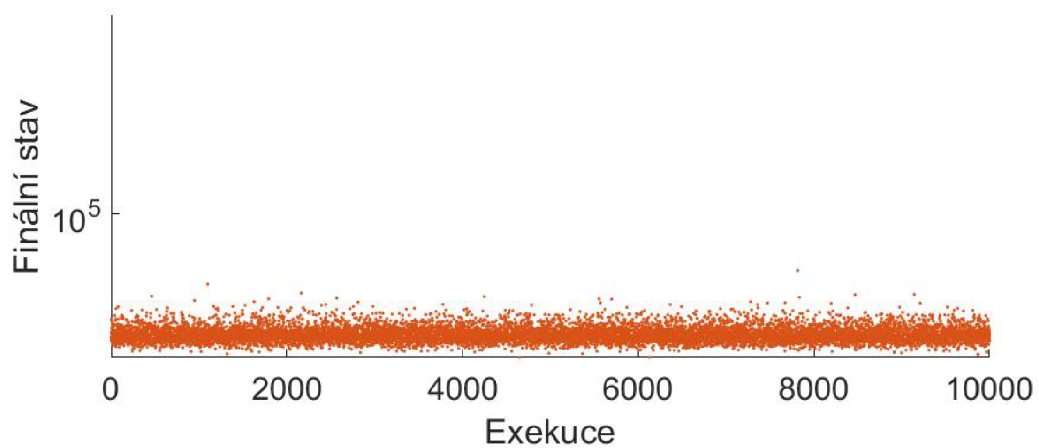


(b) pravděpodobnost 50 %

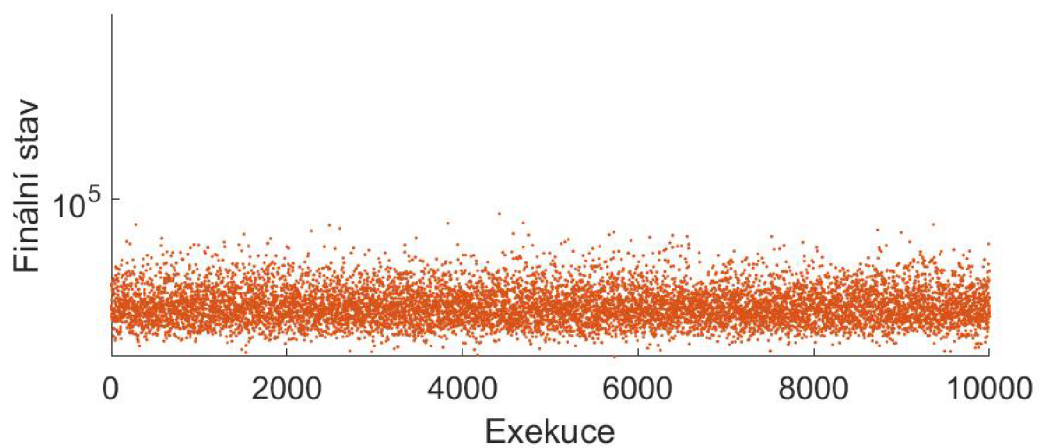


(c) pravděpodobnost 90 %

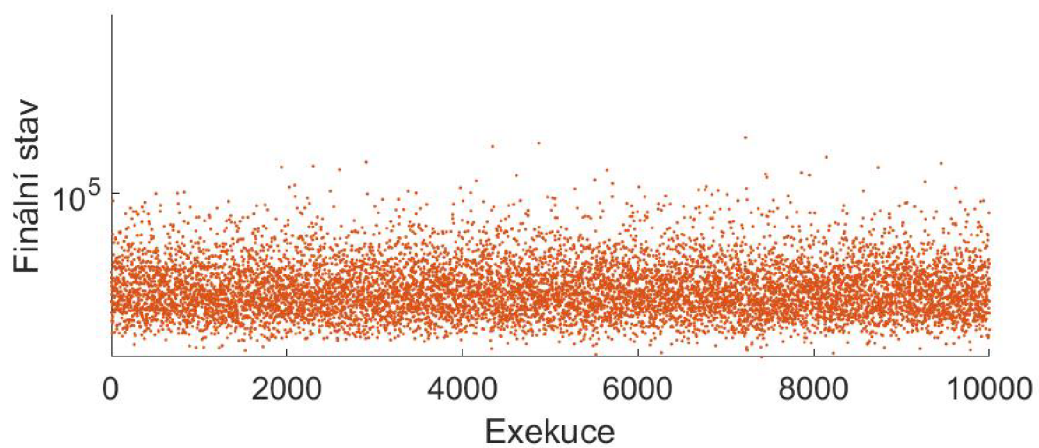
Obr. 5.6: Porovnání vlivu selhání linek v linkové topologii.



(a) pravděpodobnost 10 %

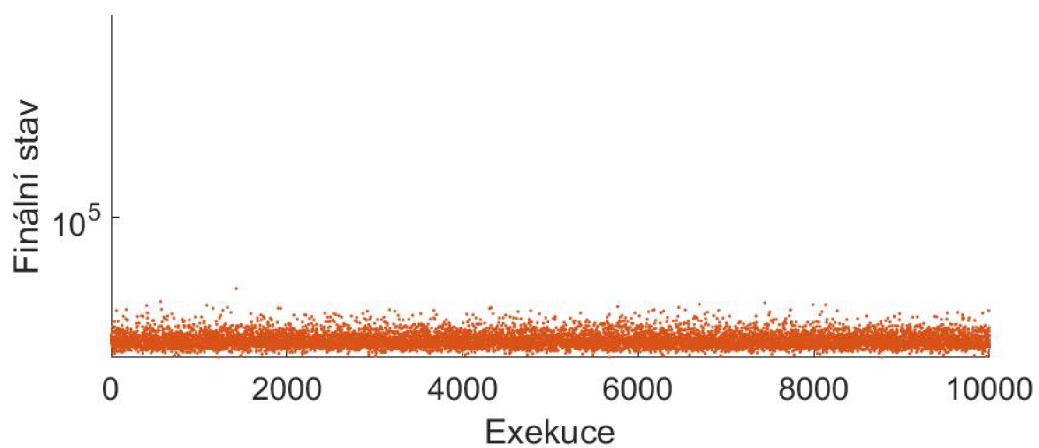


(b) pravděpodobnost 50 %

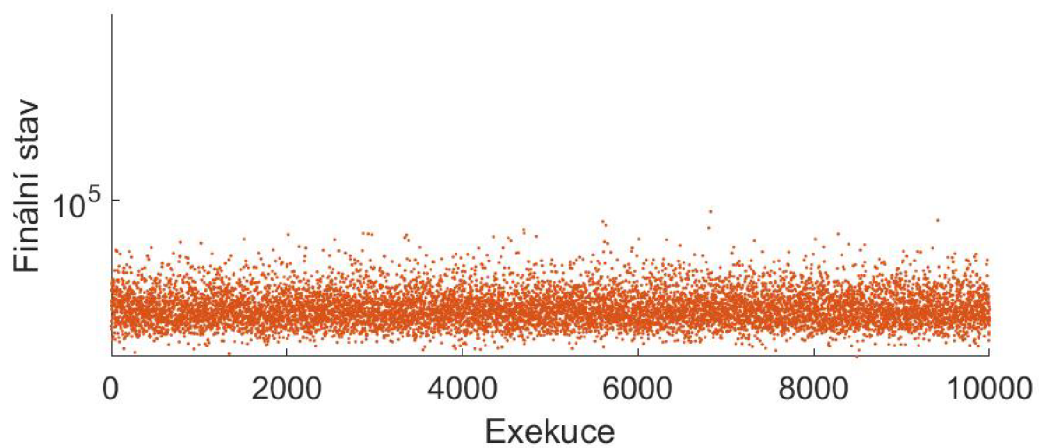


(c) pravděpodobnost 90 %

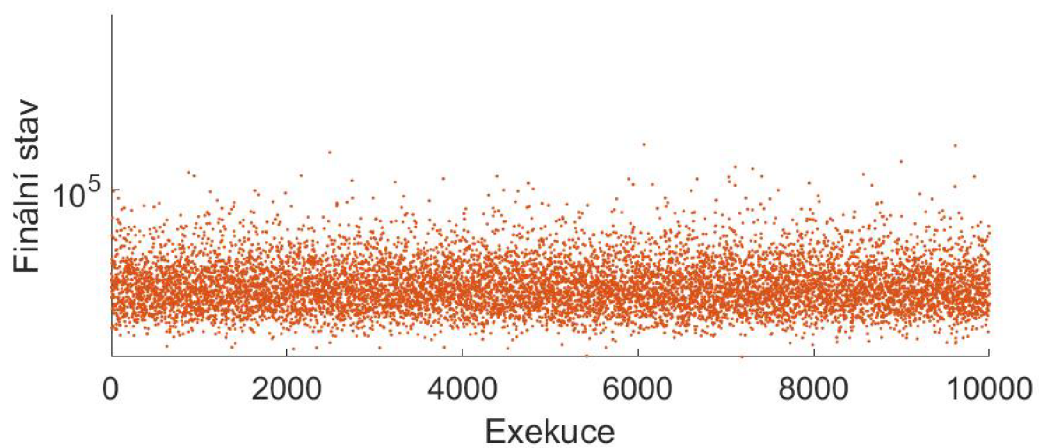
Obr. 5.7: Porovnání vlivu selhání linek v kruhové topologii.



(a) pravděpodobnost 10 %

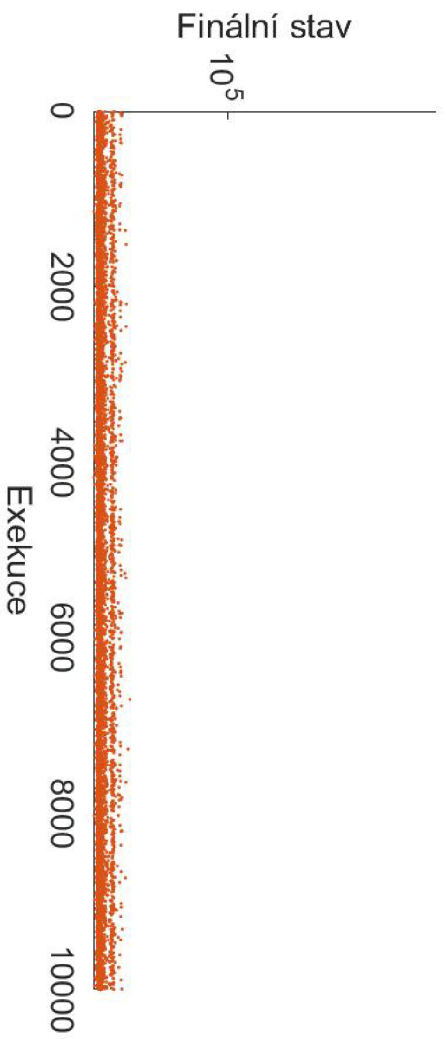


(b) pravděpodobnost 50 %

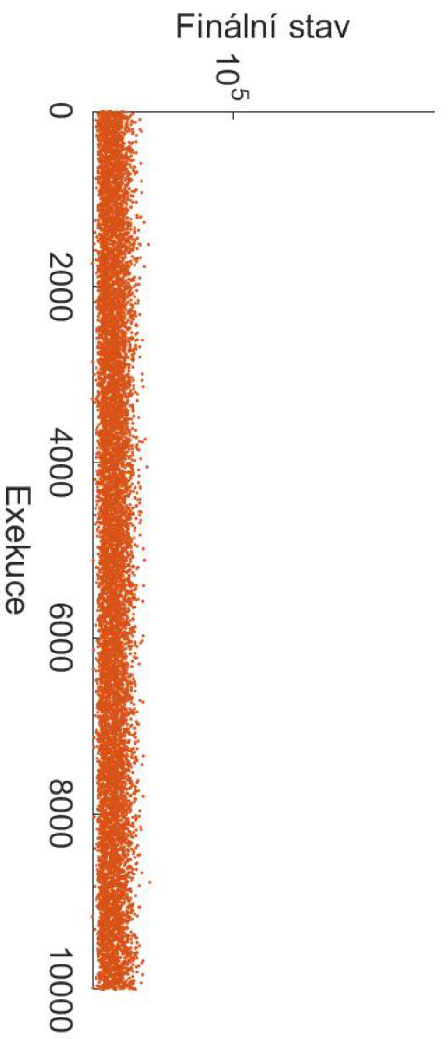


(c) pravděpodobnost 90 %

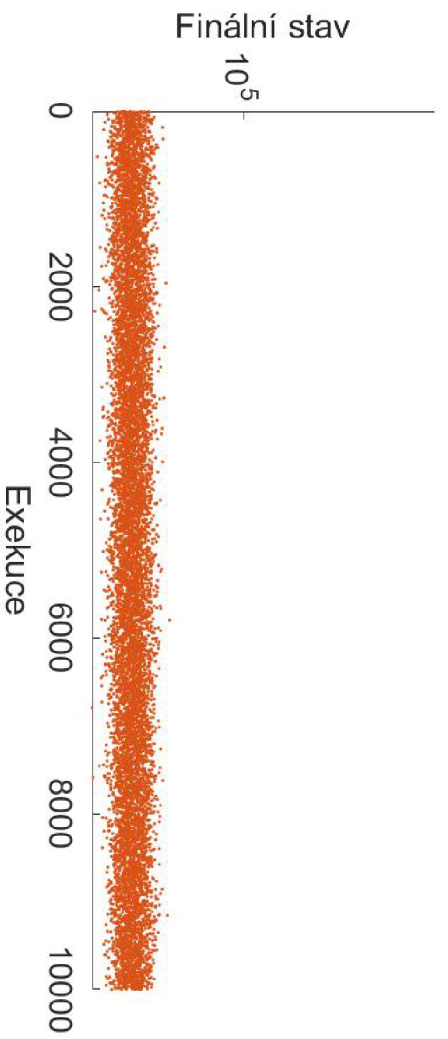
Obr. 5.8: Porovnání vlivu selhání linek ve stromové topologii.



(a) pravděpodobnost 10 %

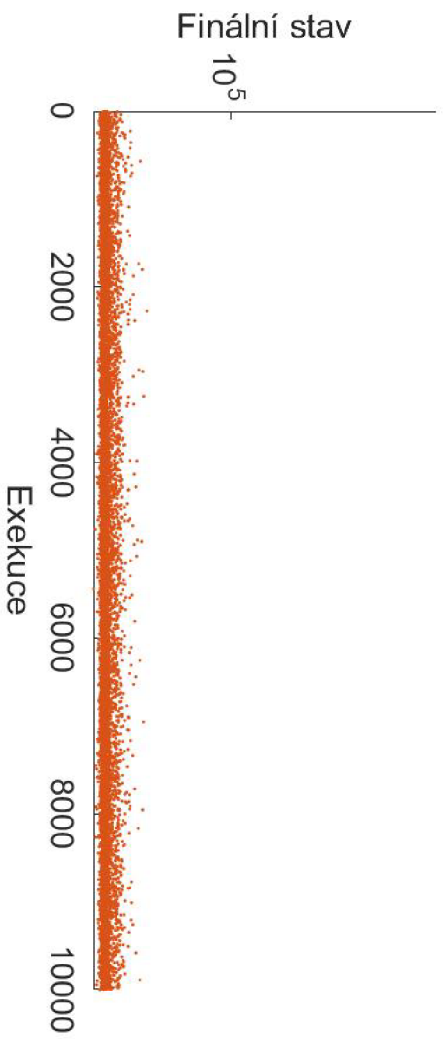


(b) pravděpodobnost 50 %

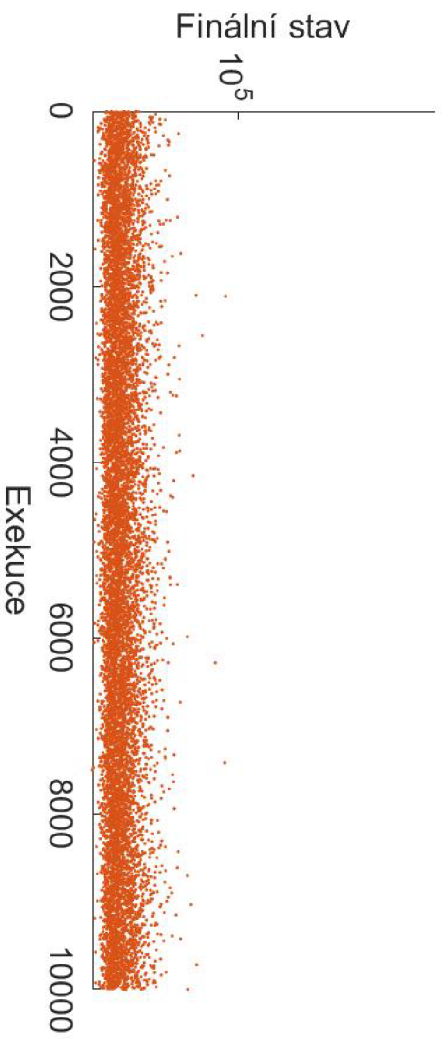


(c) pravděpodobnost 90 %

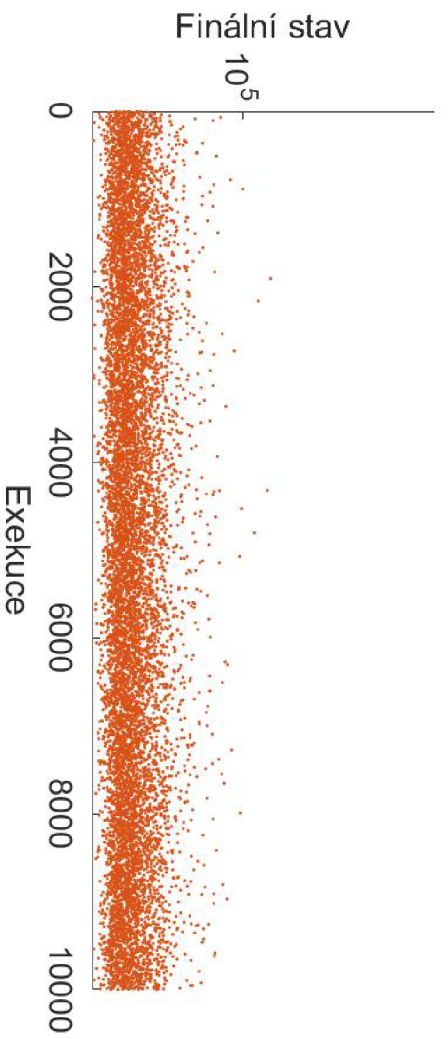
Obr. 5.9: Porovnání vlivu selhání linek ve hvězdicové topologii.



(a) pravděpodobnost 10 %

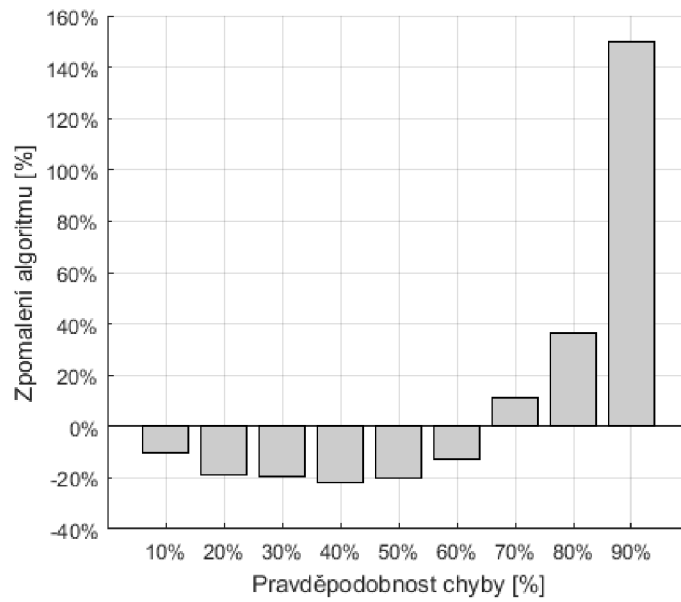


(b) pravděpodobnost 50 %

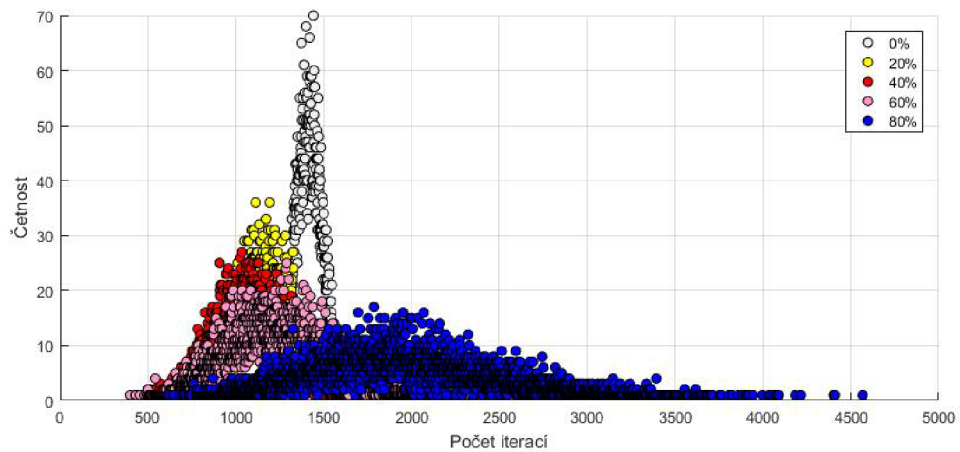


(c) pravděpodobnost 90 %

Obr. 5.10: Porovnání vlivu selhání linek v plně konektované mřížce.



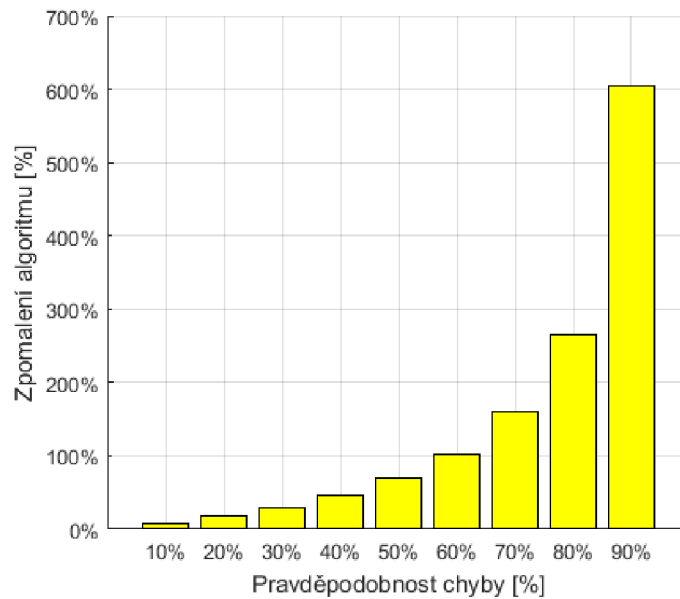
Obr. 5.11: Vliv selhání linek na počet iterací v topologii linky.



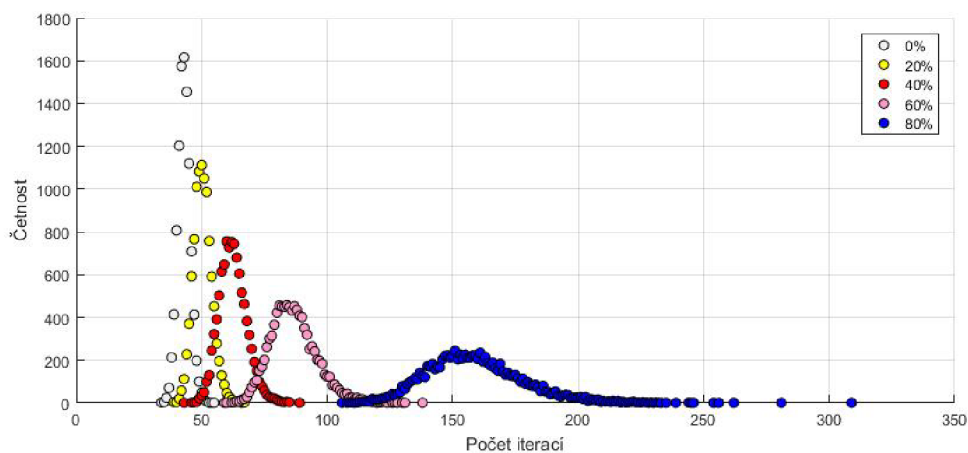
Obr. 5.12: Distribuce počtu iterací v topologii linky.

5.3.2 Topologie plně konektované mřížky

V topologii plně konektované mřížky ke zrychlení algoritmu nedošlo. Nejnižší zpomalení bylo zaznamenáno u hodnoty pravděpodobnosti 10 %, zpomalení narůstalo s postupným vyšším nárůstem nastalé pravděpodobné chyby. U 90 % chyby bylo zpomalení vyšší jak o 600 %, což můžeme vidět na obr. 5.13. Na obr. 5.14 můžeme vidět porovnání četnosti výsledků na počet iterací, kde zjistíme, že vyšší pravděpodobnost nastalé chyby způsobí větší rozptyl.



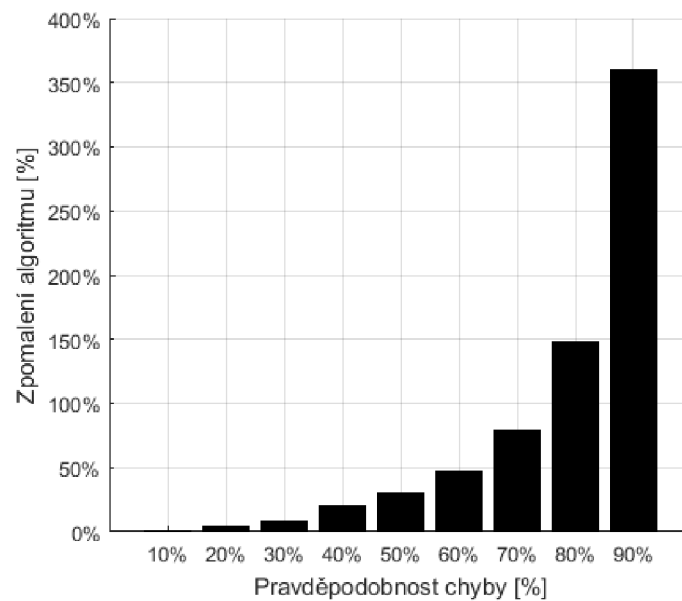
Obr. 5.13: Vliv selhání linek na počet iterací v topologii plně konektované mřížky.



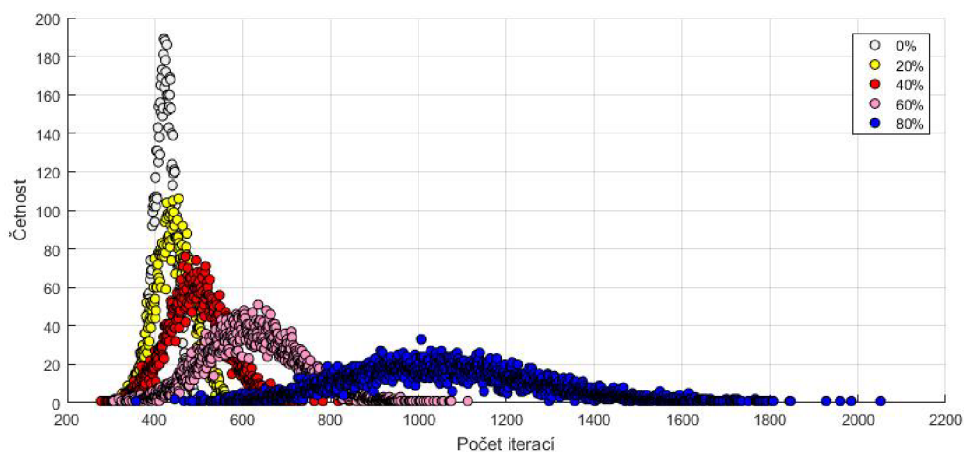
Obr. 5.14: Distribuce počtu iterací v topologii plně konektované mřížky.

5.3.3 Topologie kruhu

U topologie kruhu (obr. 4.19) je charakter podobný jako u topologie plně konektované mřížky. Speciálně pro nízké hodnoty nastalé pravděpodobné chyby je zpomalení téměř zanedbatelné. Nejvyšší hodnoty nastalé pravděpodobné chyby byly zaznamenány u pravděpodobnosti chyby 90% a to téměř o více jak 350%. U distribuce počtu iterací je tomu též obdobné jako v případě topologie plně konektované mřížky, s tím rozdílem, že se rapidně zvýšil počet iterací a snížila četnost výsledků, můžeme vidět na obr. 4.21.



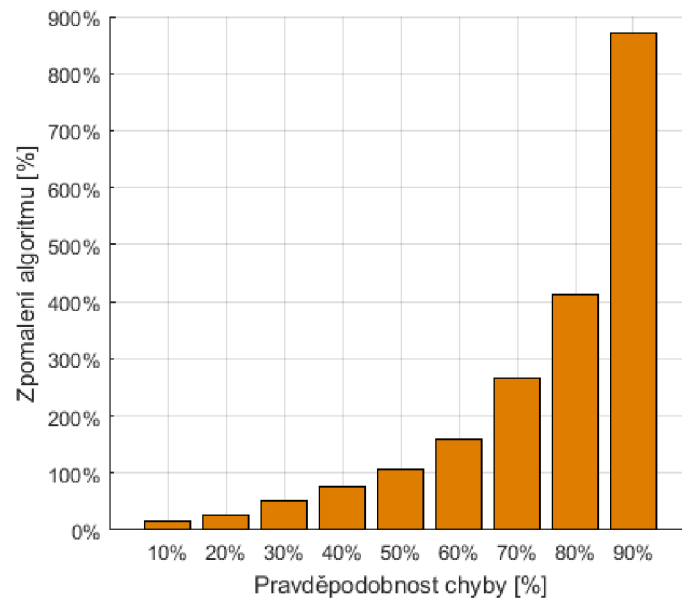
Obr. 5.15: Vliv selhání linek na počet iterací v kruhové topologii.



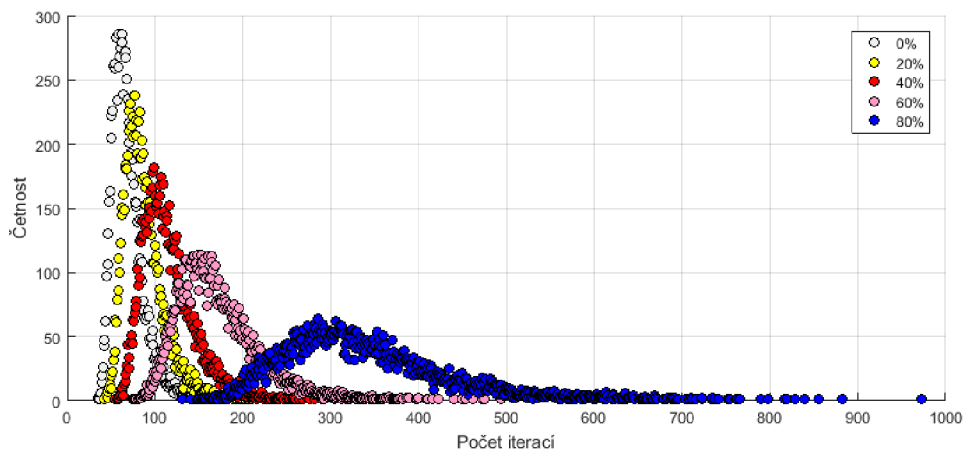
Obr. 5.16: Distribuce počtu iterací v topologii kruhu.

5.3.4 Topologie hvězdy

V topologii hvězdy je zpomalení algoritmu z hlediska výskytu chyby rostoucí (obr. 5.17), podobně jako u předchozí topologie. V této topologii bylo zpomalení algoritmu z hlediska mnou zkoumaných topologií nejvyšší, a to téměř až o 900%. Výsledky distribuce počtu iterací (obr. 5.18) jsou velice podobné jako u předchozí topologie s tím rozdílem, že četnost výsledků se zvýšila a počet iterací se snížil.



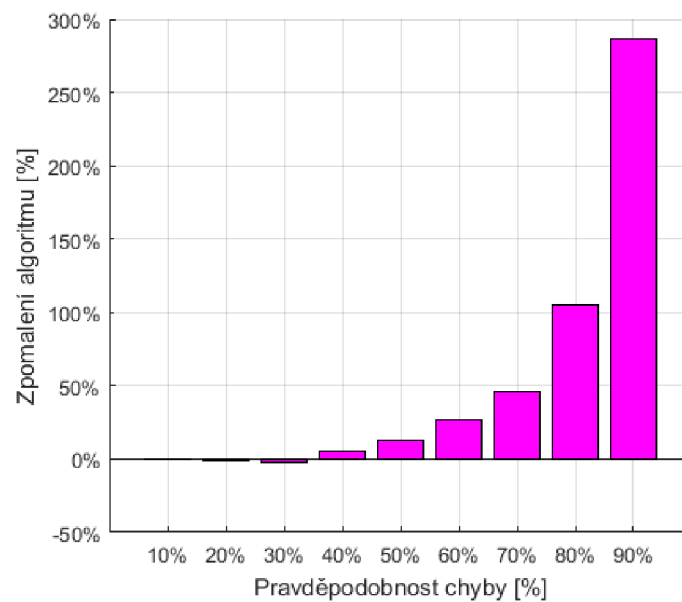
Obr. 5.17: Vliv selhání linek na počet iterací ve hvězdicové topologii.



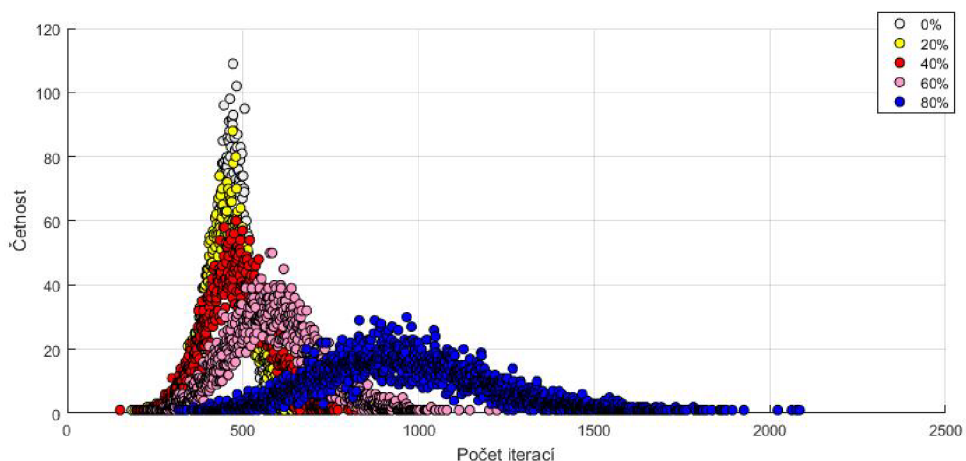
Obr. 5.18: Distribuce počtu iterací v topologii hvězdy.

5.3.5 Topologie stromu

U topologie stromu (obr. 5.19) lze pozorovat zrychlení algoritmu pro hodnoty pravděpodobnosti 10 %, 20 % a 30 % (pro 30 % je maximální zrychlení). Od 40 % až do 90 % je algoritmus zpomalený. S narůstající hodnotou pravděpodobné chyby se algoritmus zpomaluje. Největší zpomalení bylo zaznamenáno opět u hodnoty 90 %, kde zpomalení dosáhlo hodnoty téměř 300 %. U distribuce počtu iterací (obr. 5.20) lze vidět, že u hodnoty 20 % nastalo zrychlení algoritmu, jinak je charakter podobný jako u předchozí topologie.



Obr. 5.19: Vliv selhání linek na počet iterací ve stromové topologii.



Obr. 5.20: Distribuce počtu iterací v topologii stromu.

5.3.6 Porovnání dosažených hodnot počtu iterací ve zkoumaných topologiích.

Ohledem zpomalení algoritmu bylo zjištěno, že nejmenší zpomalení je v linkové topologii, kde pro pravděpodobnost 10 % až 60 % přišlo dokonce zrychlení algoritmu, nejvyšší zpomalení je přibližně okolo 150 %. Druhé nejmenší zpomalení bylo možné pozorovat v topologii stromu, které dosahuje téměř 300 %. Pro pravděpodobnosti 10 % až 40 % bylo zaznamenáno zrychlení algoritmu podobně jako v linkové topologii. Třetí nejmenší zpomalení bylo zjištěno v kruhové topologii, které dosahovalo více jak 350 %. U této topologie nedošlo na rozdíl od předcházejících dvou případů ke zrychlení algoritmu. Protokol čtvrtý nejvíce zpomalený v topologii plně konektované mřížky, kde maximální zpomalení dosahovalo 600 %. Nejvíce zpomalený byl protokol v topologii hvězdy, u které maximální zpomalení dosahovalo téměř 900 %. U distribuce počtu iterací bylo pozorováno, že u všech mnou zkoumaných topologií mimo hvězdy, mají rychlosti konvergence normální rozdělení. Dále ve všech topologiích zvyšování pravděpodobnosti nastalé chyby, zvětšuje rozptyl dat. V silně propojených strukturách byl algoritmus odolnější vůči chybě, avšak zato více zpomalený.

5.4 Vzájemné porovnání výsledků pro odhad průměru a sumy.

5.4.1 Odklon finální hodnoty od skutečné hodnoty

U zkoumání odklonu v topologii linky byl zaznamenán signifikantní nárůst odklonu pro odhad sumy s porovnáním odhadu průměru. V případě odhadu průměru, nedosahuje odklon ani 30 %, zatímco v případě sumy je to o více jak 2500 %. Jedná se o více jak 83 násobné zvýšení odklonu. V případě odhadu sumy došlo k výraznému nárůstu odklonu pro pravděpodobnost nastalé chyby rovné 90 %.

V případě topologie plně konektované mřížky činila nejvyšší odchylka u odhadu průměru něco přes 8 %, zatímco u odhadu sumy je to více jak o 350 %, což je přibližně 44 násobný nárůst odklonu. V případě sumy došlo k vysokému nárůstu odchylky u hodnoty pravděpodobnosti rovné 90 %. Nejedná se o až tak dramatický nárůst, jako tomu bylo v linkové topologii.

V topologie kruhu byla zjištěna maximální odchylka u odhadu průměru přibližně 20 % a více jak 1000 % u odhadu sumy, což je přibližně 50 násobek nárůstu odklonu. V případě topologie hvězdy je maximální odklon méně jak 8 % u odhadu průměru a téměř 80 % v případě odhadu sumy, což je přibližně 10 násobný nárůst odchylky. U topologie stromu v případě odhadu průměru byl maximální odklon okolo 15 %

a více jak 1800 % u odhadu sumy, což je přibližně 120 násobný nárůst odklonu. U této topologie byl nárůst odchyly nejvíce znatelný. V případě distribuce finálních stavů je charakter stejný pro odhad průměru i sumy s tím rozdílem, že v případě odhadu sumy lze pozorovat výrazně větší rozptyly.

5.4.2 Vliv selhání linek na rychlost konvergence a její distribuci

V případě zpomalení algoritmu se charakter pro odhad průměru a sumy od sebe výrazně neliší. Jediným rozdílem je, že v případě odhadu sumy bylo zpomalení algoritmu maximálně o 30 % vyšší, avšak tento fenomén nebyl pozorován v topologii hvězdy. U distribuce rychlosti konvergence nebyl zaznamenán výrazný rozdíl mezi odhadem sumy a průměru.

6 ZÁVĚR

Cílem práce bylo otestovat vliv chyby, tedy selhávání linek protokolu Push-sum na přesnost a očekávané zpomalování algoritmu pro různé pravděpodobnosti vzniku chyb ve vybraných topologiích.

V úvodu je popsána problematika distribuovaných systémů, přehledu zaměřeného na distribuované algoritmy a potenciálních chyb v distribuovaných systémech. Dále pak matematický aparát sloužící k modelování distribuovaných systémů v simulačním prostředí Matlab.

V experimentální části práce byl analyzován vliv chyby na přesnost a rychlost konvergence u vybraného protokolu Push-sum ve vybraných topologiích, jako je plně konektovaná mřížka, linková, kruhová, hvězdicová a stromová topologie. Byl zkoumán vliv odklonu finální hodnoty od očekávané, vliv na rychlost konvergence, distribuce počtu iterací, distribuce vnitřních stavů a dále pak vliv chyby na vnitřní stavy uzlů v topologii linky. Aby byl experiment věrohodnější, tak byl pro porovnání v každé topologii desetisíckrát opakován.

Z výsledků experimentů lze pozorovat, že protokol Push-sum není schopen bezchybného výpočtu průměru při nenulové pravděpodobnosti chyby, tedy bylo zjištěno, že čím vyšší odchylka pravděpodobnosti výskytu chyby, tím vyšší odchylka hodnot od očekávaných. Především u slabě propojených struktur je hodnota odchylky konvergence od očekávané vyšší. Dále bylo zjištěno, že v silně propojených strukturách vyšší hodnota pravděpodobnosti výskytu chyby zpomaluje protokol stejně, jako způsobuje vyšší odchylku získaných dat. Zatímco ve slabě propojených strukturách může selhání linky urychlit celý výpočetní proces, což se děje především u menších hodnot pravděpodobnosti. Dále bylo zjištěno, že rozmanitost konečných výsledků, ke kterým uzly konvergují jsou tím vyšší, čím vyšší je hodnota pravděpodobnosti výskytu chyby, avšak vyšší rozmanitost byla výrazněji pozorována ve slabě propojených strukturách.

Získané výsledky ukázali, že pokud chceme předejít větším ztrátám dat při selhávání linek, je lepší použít lépe propojené struktury, které jsou sice více zpomalené v důsledku chyby od očekávaných (bylo pozorováno několik případů, kdy to neplatilo), avšak odklon finálních hodnot je menší.

Z hlediska vzájemného porovnání odhadu sumy a průměru bylo zjištěno, že chování obou protokolů je podobné s tím rozdílem, že v případě sumy došlo k výraznějšímu nárůstu odklonu finálních stavů.

LITERATURA

- [1] ZHAO, O., GUBIAS L. J. *Wireless Sensor Networks: An Information Processing Approach*, Amsterdam, The Netherlands: Morgan Kaufmann, 2004.
- [2] HLINKA, O., SLUČIAK, O., HLAWATSCH, F., DJURIČ, P., RUPP, M. *Likelihood Consensus and Its Application to Distributed Particle Filtering*, IEEE Trans. Signal Process., vol. 60, no. 8, August 2012, pp. 4334–4349.
- [3] ECLER, T. *Přítomnost mrtvého uzlu v distribuovaném systému*, [online]. [cit. 2017-10-29]. 2016. Dostupné na URL: <https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=129532>.
- [4] LEDVINA, J. *Přednášky z distribuovaných systémů* Plzeň: Západočeská univerzita [online]. [cit. 2016-2-22]. 2002. Dostupné na URL: <<http://zcu.arcao.com/kiv/ds/ds.pdf>>.
- [5] TANENBAUM, A. S. *Distributed systems - Principles and paradigms* Prentice Hall, 2002, 803 s. ISBN 0-13-088893-1.
- [6] BIŇAS, M. *Využitie distribuovaných systémov pre archiváciu videokonferenčných prenosov* Košice: FEI TU Košice, 2006.
- [7] ŽITKOVIČ, G. *Introduction to Stochastic Processes* Department of Mathematics The University of Texas at Austin [online]. [cit. 2017-10-5]. 2010. Dostupné na URL: <https://www.ma.utexas.edu/users/gordanz/notes/introduction_to_stochastic_processes.pdf>.
- [8] KENYERES, M. *Optimalization of Distributed Classification of the Convergence Event* Sborník příspěvků z 21. konference STUDENT EEICT 2015. Brno: Vysoké učení technické v Brně, 2015.
- [9] JIA, W., ZHOU, W. *Distributed network systems - from concepts to implementations* Springer Science, 2005. ISBN 0-387-23840-9.
- [10] LIN, M. J., MARZULLO, K., MASINI, S. *Gossip versus deterministically constrained flooding on small networks* Berlín: Heidelberg, 2000. ISBN:3-540-41143-7.
- [11] BLASA, F., CAFIERO, S., FORTINO G. *Symmetric Push-Sum Protocol for Decentralised Aggregation* Italy: University of Calabria [online]. [cit. 2017-10-10]. Dostupné na URL: <<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.1.678.8543&rep=rep1&type=pdf>>.

- [12] KEMPE, D., DOBRA, A., GEHRKE, J. *Gossip-based computation of aggregate information* Foundations of Computer Science. 44th Annual IEEE Symposium on (pp. 482-491). 2003.
- [13] JESUS, P., BAQUERO, C., ALMEIDA, S. *Fault-tolerant aggregation by flow updating* Distributed Applications and Interoperable Systems (pp. 73-86). Berlin: Heidelberg, 2009.
- [14] KEMPE, D., DOBRA, A., GEHRKE, J. *Gossip-based computation of aggregate information*, in: FOCS '03: Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science, IEEE Computer Society, 2003, pp. 482-491.
- [15] KENYERES, J., KENYERES, M., RUPP, M. *Experimental Node Failure Analysis in WSNs* Vienna: University of Technology [online]. [cit. 2017-10-12]. Dostupné na URL: <http://publik.tuwien.ac.at/files/PubDat_200361.pdf>.
- [16] HLINĚNÝ, P. *Základy teorie grafů - Masarykova Univerzita* [online]. [cit. 2017-10-12]. 2010. Dostupné na URL: <<http://is.muni.cz/do/1499/el/estud/fi/js10/grafy/Grafy-text10.pdf>>.
- [17] HLINĚNÝ, P. *Základy teorie grafů - Souvislost grafů - Masarykova Univerzita* [online]. [cit. 2017-10-12]. 2010. Dostupné na URL: <<http://www.fi.muni.cz/~hlineny/Vyuka/GT/Grafy-lect--2.pdf>>.
- [18] HOANG, T. *ON THE COMPLEXITY OF SOME PROBLEMS IN LINEAR ALGEBRA - Univerzita Ulm* [online]. [cit. 2017-10-12]. 2003. Dostupné na URL: <https://www.uni-ulm.de/fileadmin/website_uni_ulm/iui.inst.190/Mitarbeiter/hoang/dissertation.pdf>.
- [19] *Linear Algebra - MIT OpenCourseWare* [online]. [cit. 2017-10-13]. 2011. Dostupné na URL: <https://ocw.mit.edu/courses/mathematics/18-06sc-linear-algebra-fall-2011/ax-b-and-the-four-subspaces/graphs-networks-incidence-matrices/MIT18_06SCF11_Ses1.12sum.pdf>.
- [20] *Identity Matrix: Definition - High School Algebra II: Homework Help Resource / Math Courses* [online]. [cit. 2017-10-18]. 2017. Dostupné na URL: <<https://study.com/academy/lesson/identity-matrix-definition-properties.html>>.
- [21] PASEKA, J. *SOUSTAVY LINEÁRNÍCH ROVNIC - Masarykova Univerzita Brno* [online]. [cit. 2017-10-20]. Dostupné na URL: <<http://www.math.muni.cz/~paseka/ftp/lectures/LA/bwslid3.pdf>>.

- [22] KENYERES, J., KENYERES, M., ŠKORPIL, V. *The Distributed Convergence Classifier Using the Finite Difference*, Brno University of Technology [online]. 2015 [cit. 2017-10-25]. Dostupné na URL: <https://dspace.vutbr.cz/bitstream/handle/11012/57921/16_01_0148_0155.pdf?sequence=1>.
- [23] FEIBEL, W. *Encyklopedie počítačových sítí* 1.vyd. Praha: Computer Press, a.s. 1996. 1230s. ISBN 80-8589-667-2.

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

A	Matice sousednosti
E	Vliv hodnot přilehlých sousedů
f	Kmitočet
G	Graf
p	Hodnota pravděpodobnosti
V	Množina procesů
WSN	Bezdrátové senzorové sítě

SEZNAM PŘÍLOH

A Obsah přiloženého DVD

79

A OBSAH PŘILOŽENÉHO DVD

Přiložené DVD obsahuje elektronickou verzi práce ve formátu PDF a soubory pro práci v prostředí Matlab, testované ve verzi 2014a. Kompletní obsah přiloženého média je uveden níže:

- *VLIV STOCHASTICKÝCH SELHÁVÁNÍ LINEK NA PROTOKOL PUSH-SUM*
- *ZDROJOVÉ KÓDY M-FILE PRO ODHAD PRŮMĚRU*
 - *Linkova_topologie.m*
 - *Plne_konektovana_mrızka.m*
 - *Kruhova_topologie.m*
 - *Hvezdicova_topologie.m*
 - *Stromova_topologie.m*
- *ZDROJOVÉ KÓDY M-FILE PRO ODHAD SUMY*
 - *Linkova_topologie.m*
 - *Plne_konektovana_mrızka.m*
 - *Kruhova_topologie.m*
 - *Hvezdicova_topologie.m*
 - *Stromova_topologie.m*