

# Meranie latencie v SDN pomocou ryu kontrolera

Patrik Ščensný

Slovenská technická univerzita

Fakulta informatiky

a informačných technológií

Ilkovičova 2, 841 04 Karlova Ves

Email: xscensny@is.stuba.sk

Lam Tuan Anh

Slovenská technická univerzita

Fakulta informatiky

a informačných technológií

Ilkovičova 2, 841 04 Karlova Ves

Email: xlamtuan@is.stuba.sk

**Abstrakt**—Latencia v sieti je dôležitý parameter ktorý je možný použiť v rôznych aplikáciách ktoré sa riadia QoS postupmi. Metódy ktoré sa doteraz používali pri meraní latencie potrebovali veľkú réžiu na vytvorenie presného merania tejto vlastnosti. Do popredia sa ale dostávajú SDN siete ktoré prinášajú nové riešenia predchádzajúcich problémov. Účelom tohto článku je porovnanie dosiahnutých výsledkov práce opísaných v článku [1], kde autori vytvorili spôsob mapovania a merania latencie v sieti s neznámou topológiou. Naša práca ale ukázala že meranie latencie týmto spôsobom je neefektívne, a prináša spomalenie siete a nepresné údaje.

**Keywords**—SDN, Mininet, OpenFlow, Ryu, Latency

## I. ÚVOD

Zistiť latenciu v sieťach je potrebné pre dodržiavanie QoS alebo pre budúci IoT. Na meranie latencie sa používa protokol ICMP(Internet Control Message Protocol), ale utilita ping. Pri SDN sieťach, ktoré fungujú odlišne ako klasické siete je ale možné ale použiť iný prístup pri získavaní informácií o latencii v sieti. Na ovládanie používame Ryu kontrolér.

## II. SDN

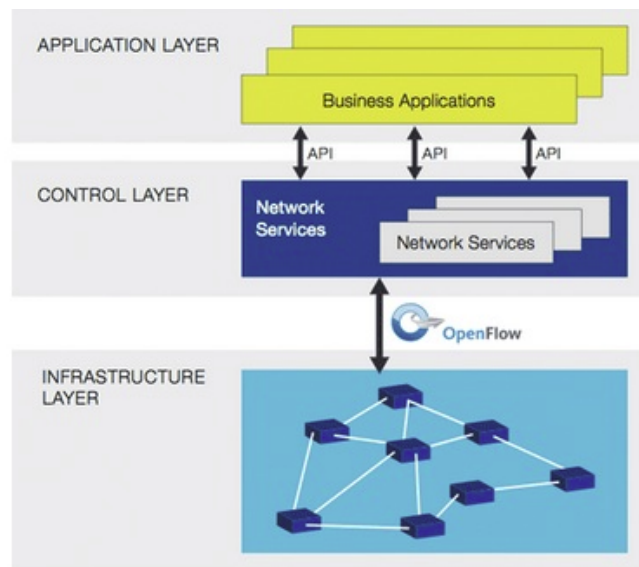
Softvérovo riadené siete majú architektúru ktorá je ovládaná cez jednu centralizovanú ovládaciu konzolu. Táto architektúra je zložená z nasledujúcich vrstiev[2]:

- *Aplikačná vrstva* - nachádzajú sa tu programy ktoré komunikujú s kontrolérom pomocou API
- *Kontrolná vrstva* - v tejto vrstve sú kontroléry ktoré riadia a usmerňujú inštrukcie ktoré boli prijaté z aplikačnej vrstvy.
- *Infraštruktúrna vrstva* - táto vrstva obsahuje samotné zariadenia ktoré preposielajú dáta medzi sebou, čím vytvoria sieť

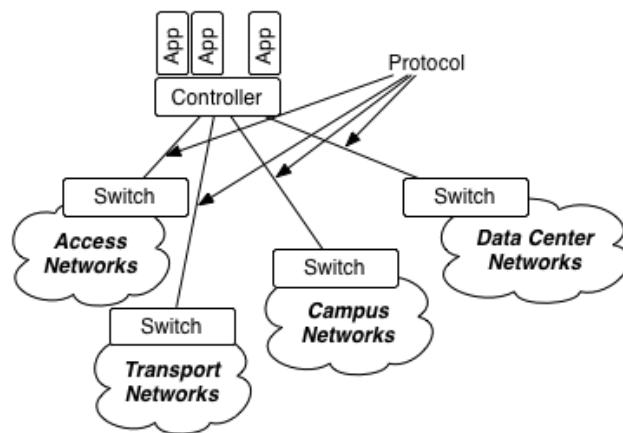
Softvérové sieťovanie má potenciál zmeniť staré dátové centrá tým, že poskytuje flexibilný spôsob ovládania siete, takže môže fungovať skôr ako virtualizovanie moderných sietí ktoré sa používajú v priemysle.

### A. OpenFlow

OpenFlow je súbor špecifikácií. Základom špecifikácií je definícia abstraktného zariadenia na spracovanie paketov nazývaného switch. Switch spracováva pakety pomocou kombinácie obsahu paketu a stavu konfigurácie prepínača. Protokol



Obr. 1: SDN architektúra



Obr. 2: SDN s OpenFlow protokolom

sa definuje na manipuláciu stavov konfigurácie prepínača a na príjem niektorých udalostí switchov. Napokon, kontrolér je prvok, ktorý hovorí a reaguje na správu stavu konfigurácie viacerých prepínačov a na udalosti[3].

### III. RYU

Je to kontrolér pre SDN ktorým definuje správanie switchu, je používaný . Tento kontrolér nám poskytuje nastavenie nasledujúcich funkcionalít switchu.

- Sledovanie eventov
- Generovanie eventov
- Triedy eventov

Každá aplikácia Ryu má zásobník pre udalosti. Tento zásobník je FIFO a zachováva poradie udalostí. Každá aplikácia Ryu má vlákno na spracovanie udalostí. Vlákno stále spracuje zásobník udalostí a volaním obslužného programu udalosti pre typ spracovania udalosti. Keď že obslužný program udalostí sa volá v kontexte vlákna na spracovanie udalostí.[4]

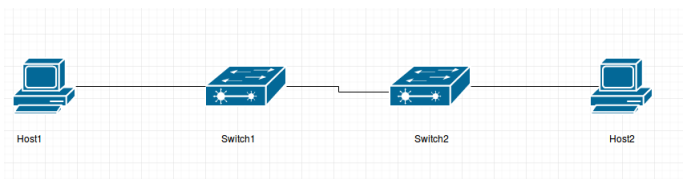
### IV. TESTOVANIE

#### A. Testovacie prostredie

- VirtualBox 5.0.40\_Ubuntu r115130
- Ubuntu-14.04.4
- Mininet - simulátor určený na vytvorenie SDN siete na počítači a vytvárať testovacie topologie bez toho aby sme museli fyzicky zapájať zariadenia
- Ryu - kontrolér ktorým posielame a meriame latenciu v sieti. Je napísaný v jazyku Python, a má internetovú knižnicu a ukážkové programy z ktorých sme vytvorili aplikáciu pre testovanie článku.

#### B. Topológia siete

Pre testovanie sme použili veľmi jednoduchú topológiu ktorej sme ale menili vlastnosti a sledovali sme presnosť výpočtov. V našej topológii sme zapojili dva počítače ktoré sú prepojené dvoma switchmi. Latenciu sme merali iba medzi týmito switchmi. Počas testovania sme menili bandwidth a delay linek.



Obr. 3: Topológia použitá pri testovaní

#### C. Meranie latencie

Meranie a výpočet latencie siete pozostáva s nasledujúcich krokov:

- 1) Zachytenie signálu na začiatok merania latencie
- 2) Odoslanie synchrónnej správy na switch, mi sme použili správu Echo Request/Reply
- 3) Odoslanie packetu na druhý switch
- 4) Aktivovanie eventu na druhom switche
- 5) Odoslanie synchrónnej správy na druhý switch
- 6) Výpočet skutočného času oneskorenia v SDN sieti

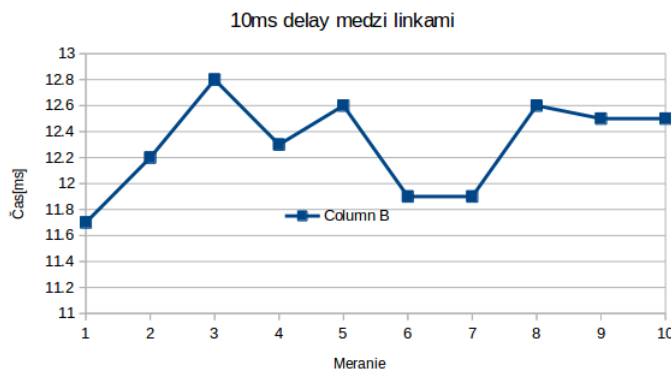
Tabuľka I: Tabuľka meraní pomocou kontroléra

	10ms	20ms	50ms	100ms	200ms	500ms	1000ms
1	11.7	22.1	52.7	101.6	201.4	502.4	1001.6
2	12.2	22.9	53.1	102.3	203.2	503.5	1001.8
3	12.8	22.4	53.1	101.7	202.6	503.2	1002.3
4	12.3	22.3	51.9	102.5	203.7	503.2	1002.5
5	12.3	22.4	53.5	101.7	203.2	502.7	1002.3
6	12.6	22.3	52.6	101.2	202.7	501.8	1003.6
7	11.9	22.2	52.4	101.2	203	503.8	1002.2
8	11.9	21.8	51.4	102.3	202.5	502.7	1003.1
9	12.6	22.1	52	102.7	202.7	502.7	1002.7
10	12.5	22.3	51.6	101.9	202.5	502.1	1002.8

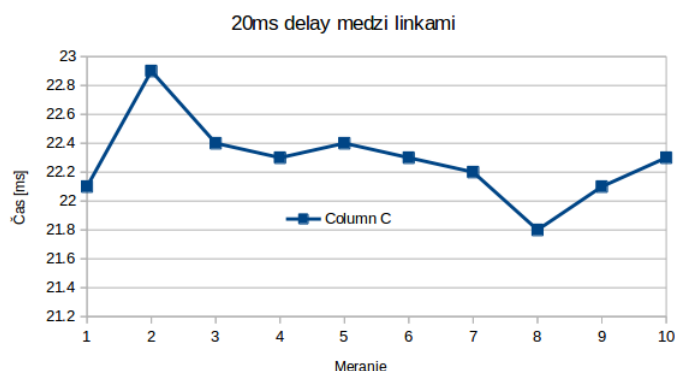
Tabuľka II: Meranie pomocou ping utility

	10ms	20ms	50ms	100ms	200ms	500ms	1000ms
1	148	275	632	1231	disconnect	disconnect	disconnect
2	61.8	121	301	602	1202	3002	6002
3	61.7	121	301	602	1202	3002	6002
4	61.2	121	301	602	1202	3001	6001
5	61.6	121	301	602	1202	3001	6001
6	62.2	121	301	602	1202	3002	6002
7	61.9	121	301	602	1202	3002	6002
8	61.8	121	301	602	1202	3002	6001
9	61.8	121	301	602	1202	3001	6002
10	61.5	121	301	602	1202	3002	6001

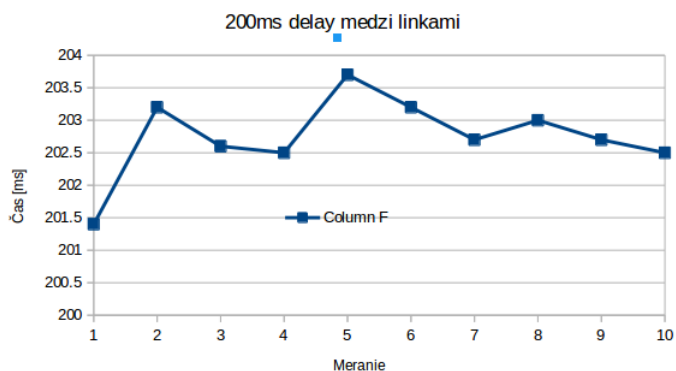
Z tabuliek I kde sa meralo pomocou nášeho programu a II kde sa meralo pomocou ping, sú vidieť rozdiely v oneskoreniach príchodu packetov. Pomocou ping utility sme mali síce veľké hodnoty ale ping meria každý delay, aj cestou späť. Takže ak máme 3 linky ktoré traversujeme 2x tak skutočnú hodnotu získame ak vydelíme 6.



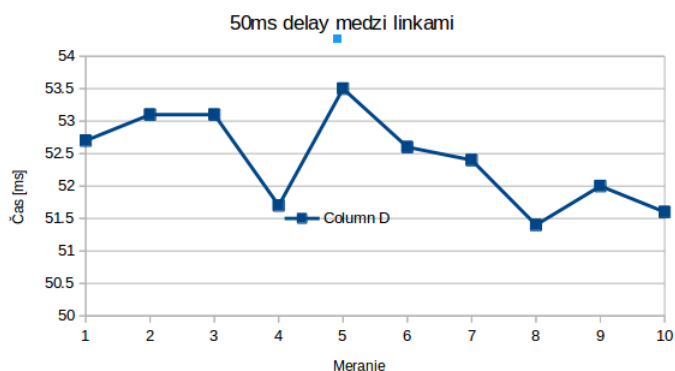
Obr. 4: Meranie s 10ms oneskorením v prepojeniach



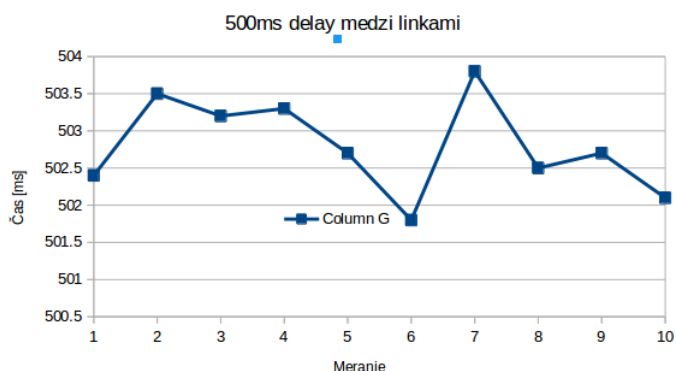
Obr. 5: Meranie s 20ms oneskorením v prepojeniach



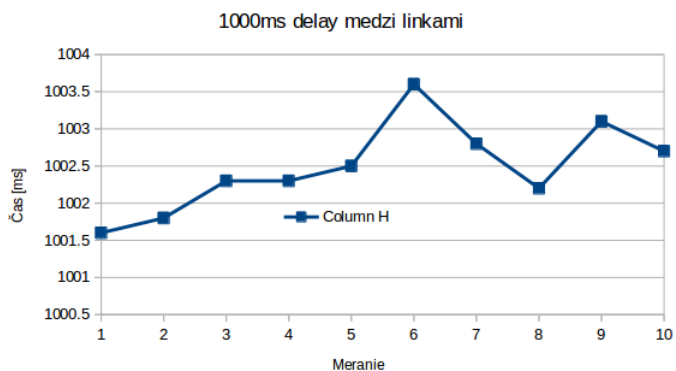
Obr. 8: Meranie s 200ms oneskorením v prepojeniach



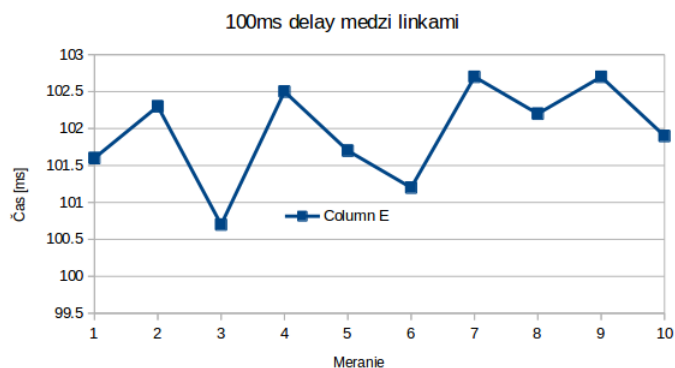
Obr. 6: Meranie s 50ms oneskorením v prepojeniach



Obr. 9: Meranie s 500ms oneskorením v prepojeniach



Obr. 10: Meranie s 1000ms oneskorením v prepojeniach



Obr. 7: Meranie s 100ms oneskorením v prepojeniach

V tabuľke si môžeme všimnúť že pri niektorých nastaveniach siete je oneskorenie skoro dvojnásobné. Toto je spôsobené spôsobom akým sú riešené switche v OpenFlow protokoli. Ak toto aplikujeme na všetky hodnoty tak zistíme že ping je rýchlejší ako keď meriame pomocou kontroléra.

Dôvod je že pri meraní pomocou kontroléra musíme zahodiť flow tabuľky switchov, čo spôsobí že switch sa pri každom packetu spýta kontroléru čo s ním má spraviť[5]. Preto prvé meranie pomocou ping museli obidva switche zistiť kto sa nachádza na portoch. Toto trvá čas a ak sú oneskorenia

na linkách veľmi vysoké môže to spôsobiť stratu packetu a aplikácia môže hlásiť odpojenie od siete. A toto není dobré ak chceme dodržiavať QoS.

## V. CONCLUSION

V tejto práci som porovnával meranie latencie na jednoduchej sieti s rôznymi oneskoreniami. Rozdieli sme hľadali čo je efektívnejšie a presnejšie. Zistili sme že pre jednu linku nám viacej vyhovuje meranie pomocou kontroléra, lebo pri meraní pomocou pingu odmeriame celú cestu tam aj späť. Ale komunikácia kontroléra a switchu nám znižuje presnosť tohto merania, ako aj vykonanie programu na presmerovanie packetu na port kde chceme vykonávať meranie. Takisto switch stratí svoju flow tabuľku pre vykonanie merania čo spomalí sieť ako celok aby si mohol switch znovu vytvoriť novú flow tabuľku.

Nedokázali sme dosiahnuť rovnaké výsledky ako autori originálneho článku. Jeden z problémov bola neskúsenosť s SDN sieťami a ryu kontrolérom, ale problémy nastali aj neurčitosti článku a nie všetky podstatné údaje boli z neho známe. Na čo sme ale prišli bolo, že táto metóda merania latencie není úplne vyhovujúca a nemá dostačujúce výstupy. Najväčší problém prišiel vo forme manuálneho smerovania packetov cez switche čo spomalilo celé meranie, a v prípade často používanej siete viditeľne spomalilo tok dát sieti.

## LITERATÚRA

- [1] Allakany, Alaa M. and Okamura, Koji, Latency Monitoring in Software-Defined Networks. In: Proceedings of the 12th International Conference on Future Internet Technologies. ACM, 2017. p. 5:1–5:4.
- [2] Understanding the SDN Architecture <https://www.sdxcentral.com/sdn/definitions/inside-sdn-architecture/> Accessed: 9-12-2017
- [3] SDN/OpenFlow <http://flowgrammable.org/sdn/openflow/> Accessed: 9-12-2017
- [4] Ryu docs <http://ryu.readthedocs.io/en/latest/index.html> Accessed: 9-12-2017
- [5] Openflow switch specification <http://www.cs.yale.edu/homes/yu-minlan/teach/csci599-fall12/papers/openflow-spec-v1.3.0.pdf> Accessed: 9-12-2017