



УНИВЕРЗИТЕТ “СВ. КИРИЛ И МЕТОДИЈ” - СКОПЈЕ
Факултет за електротехника и информациски технологии
Институт за телекомуникации



**МОДЕЛИРАЊЕ И АНАЛИЗА НА ПЕРФОРМАНСИ ВО SDN-
БАЗИРАНИ МОБИЛНИ ЈАДРЕНИ МРЕЖИ**
- Докторска дисертација -

Кандидат:
м-р Страхил Панев

Ментор:
Проф. Д-р Перо Латкоски

Јуни 2020
Скопје

Благодарност

Оваа дисертација е резултат на континуирана и напорна работа во период од две години, време исполнето со поддршка, помош и разбирање од многу луѓе. Би сакал да ја изразам мојата голема благодарност кон сите кои придонесоа кон остварување на овој труд:

- Најголема благодарност кон мојата сопруга Ива и синот Илин за поддршката и разбирањето во периодот на докторските студии. Мое извикување за сите отсуства во текот на студиите, вклучувајќи голем дел од викендите, како и доцните часови преку работната недела. Без вашата безрезервна љубов и секојдневна енергија, ќе беше невозможно да се заокружи една ваква сериозна и долготрајна целина.
- Моите родители Благлица и Драги, како и брат ми Кирил, за севкупната поддршка, мотивираност и пожртвуваност во текот на целото мое образование.
- Благодарност кон сите членови на Институтот за телекомуникации. Позитивна енергија, конструктивност и соработка уште од првиот ден, континуирана поддршка и мотивација при секое наше видување, консултации и барање за помош. Посебна благодарност до Доц. д-р Томислав Шуминоски за помошта укажана при надминување на голем дел од препреките кон целта.
- Особена благодарност на мојот ментор Проф. д-р Перо Латкоски, за неговата висока стручност и несебичност, за постојаната помош и нови идеји, за консултациите секој четврток, за достапноста и трпението во било кој дел од денот во текот на целиот период на студирање. Без вашето залагање, принципиелност и добрина, немаше навремено и квалитно да се изработи оваа дисертација, за што засекогаш ќе ви бидам благодарен.

Скопје, 2020

од Авторот

СОДРЖИНА

Листа на Акроними	4
Abstract	7
Апстракт	8
1. Вовед.....	9
2. Софтверски Дефинирано Омрежување (SDN)	16
2.1 Карактеристики на SDN и OpenFlow	17
2.1.1 Засилување на fast-failover со BFD.....	21
2.1.2 Преминување кон корисничка рамнина со локални состојби	21
2.1.3 Бафер механизми.....	23
2.2 Пристапи при анализа на перформанси	24
2.2.1 Теорија на редови.....	26
2.2.2 Quasi-Birth-Death процес	27
3. LTE/5G мобилни јадрени мрежи и примена на SDN	30
3.1 Проблемот на латентност во LTE и зошто латентноста е толку значајна во 5Г	33
3.2 Надградба на LTE пакетски мрежи, воведување на SDN концептот и парцијална виртуелизација	37
3.3 Опоравување од грешки во SDN-базирани јадрени мрежи	39
3.4 OF-базиран хендовер	42
4. Предлог нов алгоритам за брзо опоравување од испад во мобилни јадрени мрежи	45
4.1 Преглед на литературата	46
4.2 Опис на проблемот и научен придонес.....	48
4.2 Предлог за нов механизам – Stateful Fast Failover (SSF)	49
4.3 Анализа на перформанси.....	53
4.4 Заклучок.....	60
5. Аналитичко моделирање на латентноста во SDN-базирани мобилни јадрени мрежи	62
5.1 Преглед на литературата	63
5.2 Опис на проблемот и научен придонес.....	66
5.3 Моделирање на латентноста во SDN	67
5.3.1 Латентност крај-до-крај.....	68
5.3.2 Латентност поради процесирање во контролерот.....	69
5.3.3 Анализа на перформанси.....	70
5.4 Влијанието на најчестите претпоставки користени при моделирање на SDN мрежи.....	74
5.4.1 Неограничен бафер	74
5.4.2 Рата на опслужување независна од оптоварувањето.....	76
5.4.3 Анализа на перформанси.....	77
5.5 Заклучок.....	80
6. Аналитичко моделирање на хендовер во SDN-базирани мобилни јадрени мрежи	82
6.1 Преглед на литературата	83
6.2 Опис на проблемот и научен придонес.....	87
6.3 Моделирање на доцнењето при хендовер со користење единствен контролер.....	88
6.3.1 Аналитички модел	88
6.3.2 Анализа на перформанси.....	92
6.4 Моделирање на доцнењето при хендовер со користење дистрибуирана контролна рамнина	95
6.4.1 Аналитички модел	96
6.4.2 Анализа на перформанси.....	100
6.4.3 Влијание на претпоставките употребени при моделирањето	106
6.5 Моделирање на доцнењето при хендовер со користење комутатор со приоритетни бафери	109
6.5.1 Аналитички модел SFB (Shared Finite Buffering).....	109
6.5.2 Аналитички модел PFB (Priority Finite Buffering).....	114
6.5.3 Анализа на перформанси.....	120
6.6 Заклучок.....	127
7. Заклучок	130
Работна библиографија.....	137

Листа на Акроними

3GPP (3rd Generation Partnership Project)

5G (5th Generation mobile networks)

5G NORMA (5G NOvel Radio Multiservice adaptive network Architecture)

AP (Access Point)

API (Application Programming Interface)

BER (Bit Error Ratio)

BFD (Bidirectional Forwarding Detection)

CAPEX (Capital Expenditure)

CDF (Cumulative Distribution Function)

COTS (commercial-off-the-shelf)

CRC (Cyclic Redundancy Check)

DMM (Distributed Mobility Management)

E2E (End-to-End)

eMBB (enhanced Mobile Broadband)

eNB (evolved NodeB)

EPC (Evolved Packet Core)

FF (Fast-Failover)

FIFO (First In First Out)

GPRS (General Packet Radio Service)

GSM (Global System for Mobile communication)

GTP (GPRS Tunneling Protocol)

HSS (Home Subscriber Server)

HTTP (Hypertext Transfer Protocol)

ICMP (Internet Control Message Protocol)

IMT (International Mobile Telecommunications)

IoT (Internet of Things)

IP (Internet Protocol)

ITU (International Telecommunication Union)

KPI (Key Performance Indicator)

LIFO (Last In First Out)

LOS (Loss of Signal)

LR (Logarithmic Reduction)

LTE (Long Term Evolution)
M2M (Machine-to-Machine)
MAC (Medium Access Controller)
MBB (Mobile Broadband)
MEC (Multi-access Edge Computing)
MME (Mobility Management Entity)
mMTC (massive Machine Type Communication)
MN (Mobile Node)
MPLS (MultiProtocol Label Switching)
MTC (Machine-Type Communications)
MTU (Maximum transmission unit)
NAS (Non-Access Stratum)
NFV (Network Function Virtualization)
NOS (Network Operating System)
NSA (Non-Standalone Architecture)
OF (OpenFlow)
OPEX (Operational Expenditure)
OSI (Open Systems Interconnection)
P2P (Peer-to-Peer)
PCRF (Policy and Charging Rules Function)
PER (Packet Error Ratio)
PFB (Priority Finite Buffering)
PGW (PDN (Packet Data Network) Gateway)
PNPN (Priority Service)
QBD (Quasi-Birth-and-Death)
QoS (Quality of Service)
RAM (Random-Access Memory)
RAN (Radio Access Network)
RR (Regional Router)
RRC (Radio Resource Control)
RTT (Round Trip Time)
SA (Standalone Architecture)
SDMN+O (Software Defined Mobile Networking and Orchestration)
SDN (Software-Defined Networking)

SDNC (SDN Controller)
SFB (Shared Finite Buffering)
SFF (Stateful Fast Failover)
SGW (Serving Gateway)
SLA (Service Level Agreements)
SON (Self-Organising Network)
SPIDER (Stateful Programmable failure DEtection and Recovery)
SPS (Semi Persistent Scheduling)
TCAM (Ternary Content Addressable Memory)
TCP (Transmission Control Protocol)
TTI (Transmission Time Interval)
TTL (Time to Live)
UDP (User Datagram Protocol)
UE (User Equipment)
UMTS (Universal Mobile Telecommunications System)
URLLC (Ultra-Reliable and Low Latency Communications)
vEPC (virtualized Evolved Packet Core)
VM (Virtual Machine)
VNF (Virtual Network Function)
WCDMA (Wideband Code Divided Multiple Access)
XFSM (eXtended Finite State Machines)

Abstract

The Software-Defined Networking (SDN) is a concept that is already widely used in data centers and mobile core networks to provide network-wide abstraction, open interfaces, programmability, and hiding of network complexities. In today's Long-Term Evolution (LTE) implementations, there is no guarantee for a given latency in the network. Furthermore, the LTE architecture does not distinguish between services that require a certain level of latency performance from the services that do not have stringent latency requirements. The upcoming 5th generation (5G) of mobile networks is currently still being standardized to accommodate a large number of services and applications which have very diverse requirements. There are already many use cases defined that require very small guaranteed latency in the region of 1 ms to 100 ms. Fault recovery mechanisms in the 5G core network must support and enable the services for reaching the low latency values. In this thesis, an optimized protocol for fast failure detection and recovery that is implemented directly in the data plane and is suitable for the 5G Evolved Packet Core (EPC) is proposed. The novel protocol uses the OpenFlow's (OF) fast-failure mechanism for local faults, and a more advanced mechanism with stateful user plane recovery protocol that is activated only in the case of remote faults. The simulation results prove that the proposed solution can achieve fast fault recovery times and can be implemented in the future 5G core networks. Furthermore, the major latency contributors in OF-based mobile core and backhauling networks are investigated and mathematically modeled. By using queuing theory, the average packet service time in both switches and controllers are quantified. The results clearly point the major factors that impact the latency in SDN networks: load and number of controllers, arrival rate, mean service rate and probability of Packet-In messages. In this thesis, the aim is to also evaluate the impact of some of the assumptions commonly used when mathematically modelling the performance of SDN-based mobile networks. By applying queuing theory, the impact of the following assumptions is quantified: load independent service rate, and limited buffer size. The results show that considering a limited buffer significantly impacts the accuracy of the modelling, while at load dependent service rate a visible service time deterioration at high loads is noticed. A centralized controller approach in SDN creates problems of network scalability and performance degradation, which is the main reason for introducing an architecture of logically distributed controllers. Ensuring the continuity of a user's ongoing session and reducing the interruption time when performing a handover, is one of the major challenges in today's modern mobile networks. In this dissertation, a novel analytical approach to model the delay introduced by the handover-related OpenFlow signaling messages in SDN networks with multiple controllers is proposed. The results reveal that high probability of Packet-in messages causes rapid degradation of the network performance. Similarly, as the number of controllers increases, the number of synchronization messages intensifies, and this impacts negatively the packet service time. The findings can be used when designing handover delay guarantees in a mobile network with a target system throughput. Finally, two system models with specific OF-switch design are analyzed: 1) single buffer with no prioritization for both control and user plane; 2) two isolated buffers with different priority for handling the control and user plane separately. The two systems are compared by analyzing the total handover delay and the needed buffer size. The conclusion is that using priority buffering in the switch should be the preferred design choice for mobile networks, as it provides smallest handover delay and has the smallest buffer needs in the scenarios of interest.

Key Words: Software Defined Networking, OpenFlow, Performance evaluation, Fault Recovery, Queuing theory, Mobile networks.

Апстракт

Софтверски дефинираното омрежување (на англ. Software Defined Networking (SDN)) денес масовно се користи во податочните центри и во мобилните јадрени мрежи каде обезбедува мрежна апстракција, отворени интерфејси, програмабилност и криење на мрежната комплексност. Во денешните Long-Term Evolution (LTE) имплементации, нема гаранција за дадена латентност во мрежата. Понатаму, LTE архитектурата не прави разлика помеѓу услугите кои побаруваат одредено ниво на латентност, од услугите кои немаат строги побарувања за латентноста. Новата 5-та генерација (5G) на мобилни мрежи во моментот сè уште се стандардизира со цел да се прилагоди за голем број на услуги и апликации кои имаат многу разновидни побарувања. Веќе постојат многу дефинирани кориснички случаи кои бараат екстремно мала гарантирана латентност, во опсег од 1 ms до 100 ms. Механизмите за опоравување од испад во 5G јадрените мрежи мора да ги поддржат новите услуги за постигнување на многу ниски вредности на латентност. Во оваа дисертација, се предлага оптимизиран протокол за брза детекција и опоравување од испад имплементиран директно во корисничката рамнина кој е соодветен за употреба во 5G Evolved Packet Core (EPC). Новиот протокол го користи fast-failover механизмот на OpenFlow (OF) за локални грешки, додека за далечни грешки се користи напреден механизам со корисничка рамнина на состојби, кој се активира само во услови на далечен испад. Резултатите од симулациите докажуваат дека предложеното решение може да постигне брзи опоравувања од испад и може да биде имплементирано во идните 5G јадрени мрежи. Понатаму, се истражуваат и математички моделираат главните контрибутори на латентноста во OF-базираните јадрени мрежи. Со користење на теоријата на редови, се пресметува средното време на пакетско опслужување во комутаторите и контролерите. Резултатите јасно ги посочуваат главните фактори кои влијаат врз латентноста во SDN мрежите: оптоварување и број на контролери, брзина на пристигање на пакети и веројатност на наидување на Packet-In пораките. Во оваа дисертација, исто така се прави евалуација на влијанието на некои од најчестите претпоставки кои се користат при математичко моделирање на SDN-базирани мобилни мрежи. Со користење на теорија на редови, се квантифицира влијанието на следените претпоставки: рата на опслужување која е независна од оптоварувањето и неограничен бафер во комутаторот. Резултатите покажуваат дека воведување на конечен бафер значително влијае на точноста на моделирањето, додека ратата на опслужување која е независна од оптоварувањето покажува видливи нарушувања на времето на опслужување при високи оптеретувања. Пристапот со централизиран контролер во SDN наметнува проблеми на мрежна скалабилност и деградација на перформанси, што се главни причини за воведување на архитектура на логички дистрибуирани контролери. Овозможувањето на континуирана тековна сесија на корисникот и намалувањето на времето на прекин при хендовер се едни од најголемите предизвици во денешните мобилни мрежи. Во оваа теза, се предлага нов аналитички пристап за моделирање на доцнењето кое се јавува при размена на OF сигнални пораки при хендовер во SDN мрежите со повеќе контролери. Резултатите откриваат дека висока веројатност на наидување на Packet-In пораките предизвикува брза деградација на мрежните перформанси. Слично, како што бројот на контролери се зголемува, се засилува бројот на синхронизациски пораки и ова влијае негативно на времето на пакетско опслужување. Заклучоците може да се искористат при дизајнирање на гаранции за доцнењето при хендовер за мобилна мрежа со таргетиран системски проток. Конечно, во оваа дисертација се анализираат два модели на системи со специфичен дизајн на OF-комутаторот: 1) единствен заеднички бафер без приоритизација кој се употребува и за контролната и за корисничката рамнина; 2) два изолирани бафери со различен приоритет кои се користат засебно за контролната и корисничката рамнина. Двата системи се споредуваат преку анализа на вкупното време на доцнење и големината на баферот. Заклучокот е дека користењето на приоритетни бафери во комутаторот, треба да биде претпочитан избор за дизајн во мобилните мрежи, бидејќи овозможува најкратко време на доцнење при хендовер и има потреба од најмал бафер во сите разгледувани сценарија.

Клучни зборови: Софтверски дефинирано омрежување, OpenFlow, Евалуација на перформанси, Опоравување од испад, Теорија на редови, Мобилни мрежи.

1. Вовед

Софтверски дефинираното омрежување (на англ. Software Defined Networking - SDN) претставува технологија во подем која во последно време добива големо внимание од индустријата и академијата. SDN овозможува целосно раздвојување на контролниот и корисничкиот дел од мрежата, обезбедувајќи мрежна апстракција и програмабилност преку отворени стандардни протоколи. Употребата на SDN овозможува една комплексна мрежа која се состои од многу индивидуални уреди, да биде програмирана и оркестрирана како единствен ентитет. Ова значи дека SDN контролерот (на англ. SDN Controller (SDNC)) и неговите интерфејси кои ја сочинуваат контролната рамнина, играат улога на централизирана интелигенција која има глобален преглед на мрежата и авторизација за нејзина конфигурација. Од другата страна, во корисничката рамнина се наоѓаат голем број едноставни комутатори кои се управувани од SDNC. Денес во рамките на SDN, OpenFlow (OF) е најраспространетиот протокол за комуникација контролер-комутатор [1].

Секој OF комутатор содржи една или повеќе така наречени табели на тек (на англ. flow tables) кои се користат за пакетско пребарување (на англ. lookup) и препраќање (на англ. forwarding). При влез на пакет во комутаторот, доколку се најде соодветно запишано правило, се извршуваат специфичните инструкции поврзани со соодветниот тек (на англ. flow). Доколку пакетот на влез не се совпаѓа со ниту една специфична инструкция, тогаш се извршуваат следните активности: (i) првиот пакет од соодветниот тек се праќа од комутаторот кон контролерот (*Packet-In* порака); (ii) контролерот ја пресметува патеката за препраќање (на англ. forwarding path) и праќа инструкции кон соодветните комутатори кои се наоѓаат на податочниот пат со специфична информација за освежување на нивните табели на тек (*Flow-mod* порака); (iii) сите последователни пакети од истиот тек се насочуваат кон наредниот комутатор (на англ. hop) со користење на информацијата која е веќе обезбедена од контролерот и нема потреба од додатна акција во контролната рамнина.

Во рамките на SDN, контролната рамнина може да има еден или повеќе контролери [2]. Единствен контролер не е во можност да управува голем број пакетски текови кои се генерирани од голем број корисници, што е типично за мобилните мрежи. Ова е причината за воведување повеќе контролери во рамки на една SDN мрежа. Во мулти-контролер сценарио, архитектурата може да биде рамна или хиерархиска. Во рамната архитектура сите контролери се позиционирани хоризонтално на исто ниво и секој контролер има само парцијален поглед на својата мрежа, додека во хиерархиската архитектура контролерите се позиционирани вертикално на повеќе нивоа. Во оваа дисертација се разгледуваат сценарија

кои опфаќаат контролна рамнина со единствен контролер, но подоцна концептот се проширува преку воведување на мулти-контролер сценарио со рамна архитектура, со цел воочување на придобивките и ограничувањата кои зависат од бројот на контролери.

Во SDN мрежите, особено е значаен дизајнот на механизмите кои се користат за опоравување од грешки при испад на податочните линкови или комутаторите. Во централизиран и стандарден SDN-базиран механизам за опоравување од грешки, контролерот е оној кој провизионира алтернативни патишта при иницијализација и имплементира механизам за повикување (на англ. *polling*) кон комутаторите [3]. Досегашните истражувања јасно укажуваат на тоа дека доколку се користи OF стандардот точно според спецификацијата, времето на опоравување од испад е афектирано негативно и постигнувањето на т.н. *carrier grade* време на опоравување под 50 ms е скоро невозможно. Проблемот на брзо опоравување од грешки директно во корисничката рамнина без учество на контролерот е во контрадикција со самиот концепт на SDN, во кој целокупната интелигенција е сместена единствено во контролерот, додека комутаторите се обични едноставни уреди. Но, со симнување на дел од “интелигенцијата” од контролерот во комутаторите, преку воведување локални “состојби” дефинирани за секој пакетски тек, контролерот се контактира само при иницијализација, додека во услови на грешка комутаторите можат автономно да го префрлат сообраќајот на алтернативни патишта провизионирани при иницијализацијата. Еден од најпознатите предлози е концептот наречен “OpenState” [4] со кој контролерот при иницијализација ги провизионира сите алтернативни патишта во случај на грешка, а подоцна самите комутатори имаат вграден механизам да ја употребат таа информација во услови на не-локални грешки. Брзото опоравување од грешки е особено значајно во мобилните јадрените мрежи каде операторот е должен да им обезбеди на корисниците минимален прекин на услугата договорен во корисничкиот *Service Level Agreement* (SLA).

Примената на SDN концептот во *Long Term Evolution* (LTE) мобилните мрежи иницијално е воведен во *backhauling* и *Evolved Packet Core* (EPC). SDN е една од основните компоненти на 5G (на англ. 5th Generation mobile networks), каде обезбедувањето многу мала латентност и брзото опоравување од грешки е од примарен интерес за поддршка на идните услуги. Најбрзиот и најефикасен начин за воведување на 5G е преку надградба на постојниот EPC и реискористување на постоечките мрежни јазли од LTE. 3GPP (3rd Generation Partnership Project) ја нарекува оваа алтернатива *Non-Standalone Architecture* (NSA) 5G EPC. Поради различните побарувања за проток и доцнење на контролниот и корисничкиот дел на EPC, популарен е пристапот наречен парцијална виртуелизација, каде се виртуелизираат

само контролните функции, додека во корисничката рамнина се реискористуваат постојните физички SDN комутатори [5].

Во LTE, нема гаранција за одредена латентност во мрежата, уште повеќе, LTE архитектурата не прави разлика помеѓу услуги кои бараат одредено гарантирано ниво на латентност [6]. Меѓутоа, во 5G, одредени кориснички сценарија побаруваат многу ниска латентност и самата мрежа може да гарантира одредено минимално ниво на латентност за различните услуги. Во оваа докторска дисертација, се анализира целиот ланец помеѓу крајните точки во мрежата и се оценува влијанието на сите конституенти на латентноста во LTE и во 5G.

Управувањето со мобилноста на корисниците во целуларните мобилни мрежи е посебен предизвик за операторите. Целта е гарантирање минимален прекин кога корисникот извршува процедура на хендовер. Во основа, латентноста е предизвикана главно поради два фактори: процесирачко доцнење и трансмисионо доцнење [7]. Првото се однесува на средното време на сервисирање пакети во комутаторот/контролерот, додека второто го опишува времето потребно податоците да бидат трансмитирани помеѓу комутаторите и контролерот. Современите мобилни мрежи мораат да обезбедат минимална латентност бидејќи дел од новите апликации бараат гаранција за доцнењето ($< 5\text{ms}$). Мобилните станици при нивното движење извршуваат процедура на хендовер, која со примена на современите SDN концепти може во најголем дел да се изврши во рамките на стандардниот OF. За иницијализација на хендоверот се користи стандардна OF порака *Port-status*, после која доаѓа до размена на повеќе специфични сигнализациони пораки во контролната рамнина. Анализа на перформансите на OF-базирани SDN мобилни мрежи преку аналитичко моделирање со користење на теорија на редови се покажала како ефективна метода за разбирање на факторите кои влијаат на доцнењето при хендовер во мобилните мрежи. Најчесто разгледувани перформансни карактеристики се средното време на процесирање пакети во контролерот и комутаторот, како и влијанието на клучните фактори како што се: број на комутатори, веројатност на наидување на *Packet-In* пораки, брзина на пристигнување пакети, вкупен проток на системот и изнаоѓање оптимален број на контролери. Во оваа дисертација се предлагаат неколку аналитички модели со кои се моделираат главните OF контролни пораки кои се разменуваат во рамките на процедурата за хендовер. Преку дефинирање соодветна метрика за перформанси се евалуира севкупното доцнење кое го доживува мобилната станица во случај на “тврд” хендовер. Во основа, предложените аналитички модели се базираат на теоријата на редови.

Од теоријата на редови, најчесто користени дистрибуции кои се користат за моделирање во SDN се: M/M/1, M/PH/1, M/M/c и M/M/c/K [8]. При користењето на специфичните редови за моделирање некоја особина во SDN, редовно се земаат одредени претпоставки со цел да се добијат равенки во затворена форма (на англ. closed form expressions). Најчестите претпоставки се: Поасонова распределба на пристигнување (на англ. Poisson arrival rate), експоненцијална рата (на англ. rate) на опслужување, неограничен бафер и средна брзина на опслужување (на англ. mean service rate) која не зависи од оптоварувањето. Квантификацијата на влијанието на секоја од претпоставките е битен елемент кој води кон подобрување на точноста на анализите. Посебен предизвик претставува моделирањето комутатори со ограничен бафер во кој се користи приоритизација. Доколку комутаторот има конечен бафер, тогаш се нарушуваат Марковите (на англ. Markovian) особини на индивидуалните редови, што е потребен услов за да се добијат равенки во затворена форма. Стационарното однесување на овој тип мрежи се моделира преку користење равенки на глобална рамнотежа (на англ. global state equations) и примена на *Quasi-Birth-and-Death* (QBD) процесот [9-10].

Предмет на истражувањето во рамките на оваа дисертација е анализа на перформансите на SDN-базирани јадрени мобилни мрежи. **Целта на истражувањето** е да се покаже дека SDN концептите може успешно да се применат во LTE и 5G мобилните јадрени мрежи бидејќи имаат позитивно влијание врз латентноста, времето на опоравување од грешка и намалувањето на времето што мобилната станица го доживува при извршување на процедура на хендовер. **Значењето на ова истражување** е да покаже како аналитичките модели може ефикасно да се искористат за брзо и евтино моделирање на SDN-базирани мобилни мрежи, како прв чекор пред имплементација во тестен систем. Понатаму, целта е да се укаже дека сè уште постојат предлози за усовршување на OF стандардот и дел од нив може и треба да се применат во наредните стандардизирани верзии на OF. Конечно, анализата на латентноста во LTE/5G укажува дека при дизајнирање мобилните јадрени мрежи, концептот на SDN може успешно да се примени и моделира преку употреба на теоријата на редови.

Резултатите од истражувањето треба да помогнат за значително намалување на времето кое е потребно за детекција и опоравување од грешка во мобилните јадрени мрежи. Понатаму, резултатите добиени при моделирање “тврд” хендовер може да се искористат при дизајнирање јадрени пакетски мрежи со цел да се обезбеди гарантирано минимално доцнење при хендовер во зависност од вкупниот системски проток. Конечно, естимацијата на цената

на чинење кога се користат бафери со приоритизација, може да се искористи за добивање оптимална вредност на големина на бафер за дадена пакетска грешка на линкот.

Придонесот на ова истражување може да се сублимира преку главните придобивки од истражувањето:

1. Анализа на 5Г барањата за латентност и идентификација на клучните фактори кои влијаат на различните типови доцнење во мобилни јадрени мрежи базирани на SDN.
2. Интелигентна интеграција на стандардниот OF со “OpenState” со цел минимизирање на времето на опоравување од грешка во случај на локална и далечна грешка.
3. Примена на теоријата на редови при моделирање различни сегменти од SDN мрежи, како што е анализа на системи со повеќе контролери преку користење на M/M/c редот, анализа на ограничен бафер со користење на M/M/c/K редот, како и употреба на сервисно-зависни модели со цел утврдување на грешката при употреба на претпоставката за просечна рата на опслужување која е независна од оптоварувањето.
4. Предлог за нов аналитички модел базиран на M/M/1 и M/PH/1 редот и дефинирање соодветна перформансна метрика со цел квантификација на доцнењето кое го доживува мобилната станица при “тврд” хендовер. Претходно предложените модели се проширени со моделирање на *Port-status* пораки и воведување на дистрибуирана контролна рамнина (мулти-контролер сценарио). Се анализираат главните фактори како што се: влијанието на бројот на контролери и комутатори, брзината на пристигнување на *Packet-In* и синхронизациски интер-контролер пораки, како и целокупниот системски проток.
5. Анализа и предлог на математички пристап за моделирање хендовер при дизајн на комутатор кој користи конечен бафер со приоритизација и споредба на вкупното доцнење при хендовер со сценариото на комутатор со единствен бафер без приоритизација. Дополнително, се прави и евалуација на потребната големина на баферот со цел да се постигне соодветен *Packet Error Ratio* (PER) наметнат од податочниот линк.

Оваа докторска дисертација е организирана на следниот начин: во следната глава се дава вовед во SDN, опис на пристапите при анализа на перформанси, математички модели кои се користат за моделирање во SDN, како и краток осврт на QBD процесот. Во третата

глава се става акцент на примената на SDN во мобилните јадрени мрежи и проблемот на латентноста во LTE/5G. Се дискутира архитектурата на 5G и LTE, со специјален фокус на т.н. парцијална виртуелизација на EPC и како процедурата на хендовер може да се изврши во рамки на OF стандардот без додавање на нови протоколи или OF модификации. Во четвртата глава се анализира проблемот на брзо опоравување од испад и се предлага алгоритам за брза детекција и опоравување од испад имплементиран директно во корисничката рамнина. Петата глава ја разгледува примената на теоријата на редови за моделирање на перформансите во SDN мрежите, со посебен осврт и анализа на латентноста во LTE и како истата да се моделира преку математички пристап. Се разгледува примената на M/M/1, M/M/c и M/M/c/K редовите и се става акцент на естимација на грешката која се внесува при моделирање на неколкуте претпоставки кои редовно се користат при моделирање на SDN: неограничен бафер и средна брзина на опслужување која не зависи од оптоварувањето. Во шестата глава се предлага аналитички пристап кој се користи за моделирање на доцнењето кое го доживува мобилната станица во случај на “тврд” хендовер. Во оваа глава се предлагаат математички модели за обликување на хендовер при три различни сценарија валидирани со помош на симулации: (1) единствен контролер; (2) дистрибуирана контролна рамнина; (3) користење на комутатор со приоритетни бафери. Главата 7 ги сумира заклучоците од истражувачкиот труд кои можат да се искористат од SDN дизајнерите при планирање и оптимизација на SDN-базираните мобилни мрежи.

Во рамки на истражувањата во оваа докторска дисертација, објавени се повеќе трудови во интернационални списанија со фактор на влијание, реномирани регионални списанија, како и домашни и меѓународни конференции:

- [1] S. Panev, P. Latkoski, “Performance Evaluation of SDN-based 5G Core Networks”, *BalkanCom - Third International Balkan Conference on Communications and Networking*, Skopje, May 2019.
- [2] S. Panev, P. Latkoski, “Latency Evaluation and Modelling in Distributed SDN-based Mobile Networks”, *27th Telecommunications Forum, TELFOR*, Belgrade, November 2019.
- [3] S. Panev, P. Latkoski, “The Impact of the Most Common Assumptions When Modelling SDN-based Mobile Networks”, *Microwave Review*, Vol. 25, No. 2, December 2019.
- [4] S. Panev, P. Latkoski, “Delay Modeling in SDN-based Mobile Networks with Single Controller”, *Journal of Electrical Engineering and Information Technologies*, Vol. 4, No. 1-2, December 2019.

- [5] S. Panev, P. Latkoski, “Next generation of Mobile Core Networks – the road to 5G”, *PRESING, Chamber of certified Architects and Certified Engineers of R. Macedonia, year VIII, No. 46*, December 2019.
- [6] S. Panev, P. Latkoski, “Handover Analysis of OpenFlow-based Mobile Networks with Distributed Control Plane”, *Computer and Electrical Engineering, Vol. 81*, January 2020. doi: 10.1016/j.compeleceng.2019.106546, **Impact Factor: 2.189**.
- [7] S. Panev, P. Latkoski, “SDN-based Failure Detection and Recovery Mechanism for 5G Core Networks”, *Emerging Telecommunication Technologies, Vol. 31, Is. 2*, February 2020. doi: 10.1002/ett.3721, **Impact Factor: 1.258**.
- [8] S. Panev, P. Latkoski, "Modeling of OpenFlow-related Handover Messages in Mobile Networks", *Telecommunication Systems*, Springer (Accepted). **Impact Factor: 1.707**.
- [9] S. Panev, P. Latkoski, “Performance Analysis of Handover Delay and Buffer Capacity in Mobile OpenFlow-based Networks”, *International Journal of Communication Systems*, Wiley (Submitted, expected 2020). **Impact Factor: 1.278**.

2. Софтверски Дефинирано Омрежување (SDN)

Мрежните технологии имаа огромен скок во технолошкиот развој во последните години. Индустијата и академијата работеа заеднички со цел да ги направат мрежите отворени и програмабилни. Постојење на отворени мрежи е во спротивност со традиционалниот затворен пристап на вендорите за мрежна опрема кои развиваа сопствени (на англ. proprietary) имплементации на хардвер и софтвер врзани (на англ. bundled) заедно во иста црна кутија (на англ. black box), тешко променливи откако иницијално ќе се имплементираат. Програмабилноста се однесува на давање можност мрежните оператори самите да ги кодираат своите потреби, наместо да се потпираат на бавниот развоен циклус наметнат од вендорите при додавање поддршка за нови протоколи и услуги.

SDN беше дел од епицентарот на оваа револуција. SDN е базиран на едноставна идеја на раздвојување на контролната и корисничката рамнина, што се покажа како ефикасен начин за решавање на проблемот на мрежна програмабилност. Корисничката рамнина е одговорна за препраќање пакети и содржи минимална интелигенција потребна само за преместување или отфрлање пакети помеѓу портите во рамките на еден ист уред. Контролната рамнина е местото каде е имплементирана целокупната мрежна интелигенција, односно тука се донесуваат сите одлуки за тоа низ кои комутатори и порти треба да се насочи пакетот за да се имплементира дадена мрежна полиса (network policy), како што е рутирање или сигурност (security).

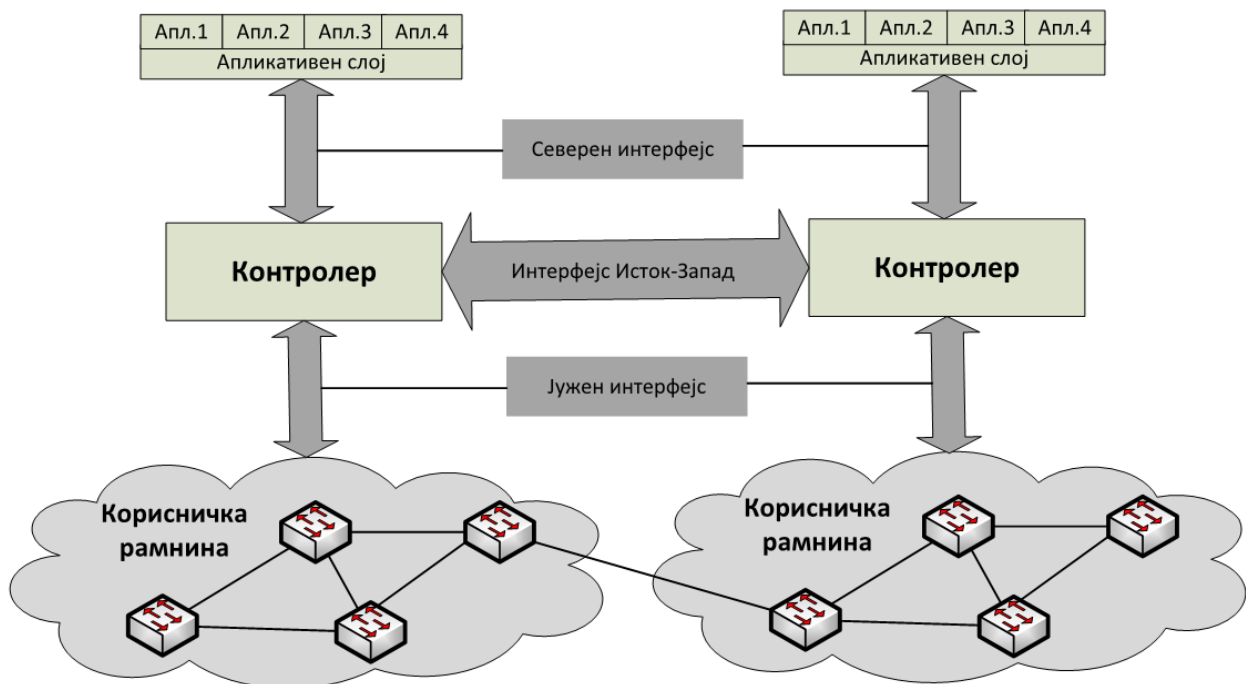
SDN нуди значајни придобивки за операторите, вклучувајќи [11]:

- i) Намалување на трошоците. Од една страна, SDN мрежните уреди (пр. комутатори и рутери) се поевтини. Од друга страна, SDN им овозможува на операторите да ги автоматизираат конфигурациските промени кои се комплексни, се прават уред-по-уред, се временски-побарувачки и се склони на грешки. Со ова се намалуваат оперативните трошоци во мрежата.
- ii) Забрзување на иновацијата. SDN се потпира на користење комерцијален (commodity) хардвер и отворени интерфејси. Двете карактеристики овозможуваат брза иновација и едноставно и забрзано имплементирање на нови мрежни протоколи и функционалности.
- iii) Подобрување на корисничкото искуство. Преку централизирање на мрежната контрола и можноста за предавање на мрежните состојби на апликациите на повисоко ниво, SDN инфраструктурата може подобро да се адаптира на динамичните кориснички потреби.
- iv) Погрануларни полиси на услуги [12].

Во понатамошниот дел од текстот, се разгледуваат основните карактеристики на SDN и најпопуларната имплементација на SDN јужен (south-bound) протокол, OpenFlow.

2.1 Карактеристики на SDN и OpenFlow

SDN е мрежен концепт кој воведува централизирана контрола и консолидиран глобален мрежен преглед, овозможувајќи им на мрежните апликации да се извршуваат на врвот на мрежен оперативен систем (на англ. network operating system). SDN примарно се користи во слоевите 2 – 4 од *Open Systems Interconnection* (OSI) моделот и други протоколи како што е *Multiprotocol Label Switching* (MPLS) [13]. Резултатот од користење на SDN е логички централизирана мрежна интелигенција и глобално управување со состојби на една страна, и посебна инфраструктура која е целосно раздвоена од мрежните апликации, од друга страна. Клучната придобивка на SDN е програмабилноста. Таа овозможува комплексна мрежа која се состои од многу индивидуални уреди да биде програмирана и оркестрирана како единствен ентитет. Ова значи дека SDN контролерот со неговите интерфејси, игра улога на централизирана интелигенција со авторизација за нејзина реконфигурација. На сликата 2.1 е дадена илустрација на SDN интерфејсите.

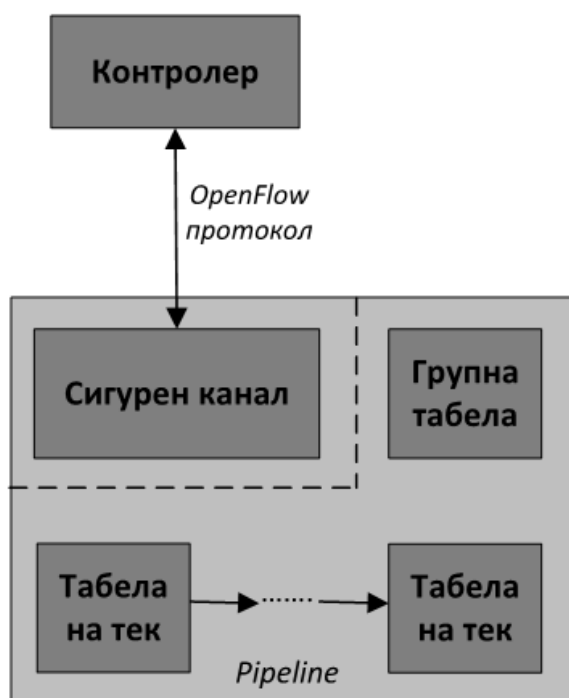


Слика 2.1. SDN интерфејси

Северниот (на англ. northbound) *Application Programming Interface* (API), кој претставува интерфејс помеѓу контролерот и апликациите, се користи од страна на апликациите за оркестрација на корисничката рамнина и извршување комплексни задачи како *Quality of Service* (QoS), балансирање на оптоварувањето (на англ. load balancing),

firewall, инженеринг на сообраќај (на англ. *traffic engineering*), наметнување на полиса (на англ. *policy enforcement*) итн. Во моментот постои вистинска разновидност на северни API кои може грубо да се разделат во три групи: *Representational State Transfer-ful* (REST-ful) API, специјализирани *ad-hoc* API и програмски јазици кои може да се користат за северните интерфејси. Денес се достапни и многу SDN контролери кои поддржуваат различни северни API. Најпопуларните имплементации на контролери се: *NOX*, *Floodlight*, *OpenDayLight*, *Ryu*, *Onix*, *Hyperflow*, итн.

Јужните (на англ. *southbound*) протоколи се користат за контролна комуникација помеѓу контролерот и комутаторите во корисничката рамнина. Јужниот API, кој претставува интерфејс помеѓу контролерот и комутаторите веќе има стандард дефиниран во *OpenFlow*, со многу други мрежни протоколи кои се натпреваруваат за внимание (ForCES, POF, OVSDB, BGP итн.).



Слика 2.2. Основни компоненти на OF комутатор

Секој OF комутатор содржи една или повеќе табели на тек и групна табела (*group table*) кои извршуваат пребарување на пакети и препраќање [1]. При иницијализација, освежување топологијата, грешки итн., контролерот може да ги додава, освежува и брише записите на тековите (на англ. *flow entries*) во табелите на тек и ова може да се прави реактивно (како одговор на пакети пратени од комутаторот) или проактивно. Секој *flow entry* се состои од таканаречени *match fields*, броила (на англ. *counters*) и множество од инструкции (на англ. *set of instructions*) кои се аплицираат при проверката за совпаѓање (на

англ. *matching*) со влезните пакети. Основните компоненти на OF комутаторот се прикажани на сликата 2.2.

Проверката за совпаѓање на пакетите започнува со првата табела на тек и може секвенцијално да продолжи кон останатите табели на тек. Проверката во секоја од табелите на тек се прави со претходно одреден приоритет на разгледување и секогаш се извршува првиот ред при позитивна проверка за совпаѓање. Кога ќе се најде т.н. соодветен запис (на англ. *matching entry*), се извршуваат специфичните инструкции кои се поврзани со разгледуваниот пакетски тек. Во услови кога не може да се најде позитивен исход при проверката за совпаѓање во табелата на тек, соодветната наредна акција зависи од конфигурацијата на записот на тековите во т.н. табела *table-miss*, при што пакетот може да продолжи да биде анализиран во наредната табела на тек, да биде отфрлен (на англ. *dropped*), или да биде пратен кон контролерот преку OF каналот.

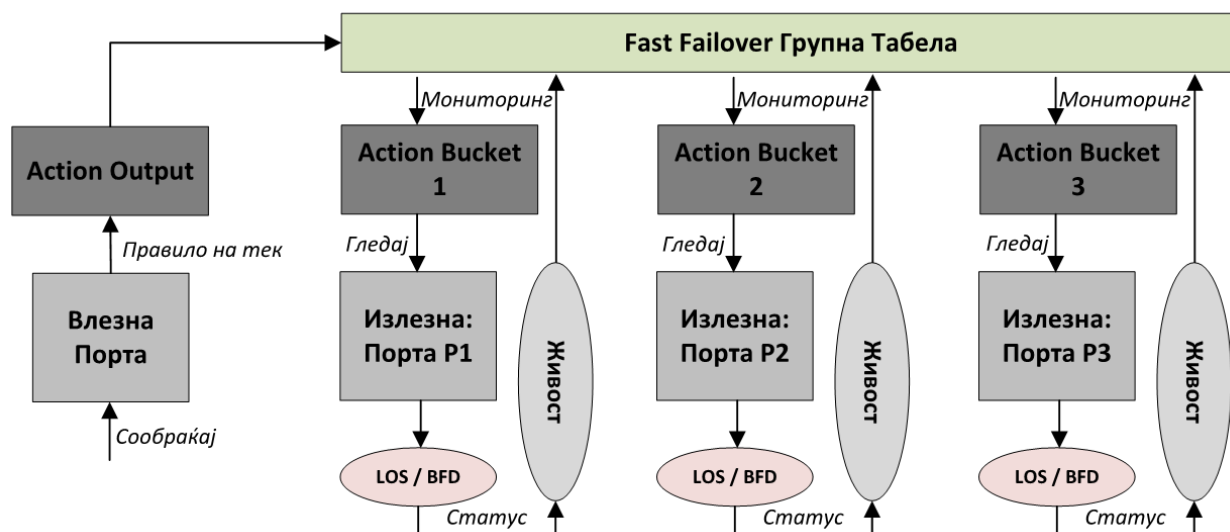
Инструкциите кои се вклучени во множеството инструкции, содржат акции или ги модифицираат таканаречените *pipeline* процесирачки правила. Акцијата го опишува начинот на препраќање на пакетите, модификацијата на пакетите и можното процесирање во групната табела, додека *pipeline* процесирачките инструкции содржат дефиниции за тоа дали и како пакетите треба да се пратат на последователните табели за дополнително процесирање и овозможуваат да се разменува метадата-информација помеѓу различните табели.

Обично записите на тековите за специфичен пакетски тек (пр. дефинирани со изворна и дестинациска IP адреса) посочуваат кон одредена порта (на англ. *port*). Во најголем дел од случаите ова е физичка или логичка порта на комутаторот, но може да биде и резервирана порта, како што е дефинирано во [1]. Резервираните порти се користат за специфицирање на генерички акции како што е праќање пакети кон контролерот, поплавување (на англ. *flooding*) или препраќање со користење на не-OFF методи.

Почнувајќи од OF верзија 1.1, воведена е функцијата на групна табела. Групните табели се екстензија на стандардните OF конфигурациски правила и овозможуваат напредно препраќање (на англ. *port forwarding*) и мониторирање од комутаторот. Групните табели се состојат од записи на групи (на англ. *group entries*), каде секој од нив содржи листа на т.н. *action bucket* кои имаат многу специфична семантика и зависат од типот на групата. Акциите дефинирани во една или повеќе т.н. *buckets* се извршуваат врз пакетите кои се праќаат на групата.

Групната табела *fast-failover* (FF) може да се конфигурира да прави константни проверки на статусот на портите и да комутира сообраќај согласно резултатите на надзор

што е активност независна од самиот контролер. Групата *fast-failover* е опционална и номиналното (на англ. *default*) однесување е секогаш да се изврши првиот жив (*live*) *action bucket*. Секој *action bucket* е асоциран со порта и ја контролира нејзината живост (на англ. *liveness*), поради што оваа групна табела мора да имплементира мониторирање на живост (на англ. *liveness monitoring*) за да го одреди специфичниот *bucket* кој ќе се изврши. Номиналниот механизам за детекција на локални грешки во OF е *Loss of Singal (LOS)* и ова е наједноставниот начин да се детектира испад (на англ. *failure*) на порта на комутаторот. Групната табела FF и нејзиниот механизам за проверка на живост се прикажани на сликата 2.3.



Слика 2.3. OF fast-failover група

Правилата за одредување на живоста на портите се следните:

- Една порта се декларира за жива ако има OFPPS_LIVE знаменце (на англ. *flag*) поставено (на англ. *set*) за нејзината состојба на порта. Живоста на портата може да биде имплементирана и надвор од OF делот на комутаторот, односно може да биде дефинирана надвор од OF спецификациите (пр. *Spanning Tree*, *Bidirectional Forwarding Protocol (BFD)* итн.). Минималниот критериум кој треба да биде задоволен за портата да не биде сметана за жива е доколку битот на конфигурација на портата OFPPC_PORT_DOWN е долу (на англ. *down*) или ако битот на состојба на порта OFPPS_LINK_DOWN укажува дека портата е долу.
- Еден т.н. *bucket* се декларира за жив ако *watch_port* не е OFPP_ANY и мониторираната порта е жива, или ако *watch_group* не е OFPG_ANY и мониторираната група е жива.
- Една група се декларира за жива ако најмалку една од нејзините т.н. *buckets* е жива.

2.1.1 Засилување на fast-failover со BFD

Протоколот BFD имплементира контролен и ехо алгоритам за детекција на живост на линкови помеѓу претходно дефинирани крајни точки [14]. Секој од конфигурираните јазли трансмитира контролни пораки кои ја објавуваат моментната состојба на мониторирањето линк. Кога еден јазол ќе прими контролна порака, тој одговара со ехо порака која дава информација за статусот на сесијата. Типична сесија иницијално се гради преку три-насочно ракување (на англ. three-way handshake), а после тоа регуларните контролни пораки потврдуваат дека нема испад помеѓу крајните точки на сесијата. BFD е дизајниран на таков начин што може да се користи преку секој транспортен протокол со цел подобрена детекција на испад.

Главниот механизам за откривање на испади во BFD се нарекува асинхрон режим (на англ. asynchronous mode) во кој крајните точки на сесијата праќаат BFD пакети со фиксна брзина. Испад се детектира доколку еден од системите престане да прима пакети за дефиниран тајмаут на детекција. Брзината на праќање пакети и тајмаутот на детекција може да бидат конфигурирани на таков начин што може да се постигнат кратки и гарантирани времиња на доцнење при откривање на испад. Во рамките на BFD, постои опција која се нарекува режим на побарување (на англ. demand mode), каде едната крајна точка може експлицитно да побара од другата крајна точка да го активира или деактивира праќањето на контролните пакети. Најважната карактеристика која мора да ја поседува комутаторот за ефективно да го користи BFD, е можноста локалната контролна рамнина да го мери времето кое е поминато помеѓу добиените пакети. Поради ова, во BFD е воведена ехо функција чија намена е да ја тестира корисничката рамнина на комутаторот. Во овој режим на работа на BFD, се праќаат специјални ехо пакети од контролната рамнина на претходно дефинирани интервали и очекуваниот резултат е другата крајна точка да ги “враќа” (на англ. loopback) овие пакети назад кон изворната точка.

Воведувањето на BFD за мониторирање на живоста на FF групната табела треба да го подобри времето на детекција на локална грешка. Во главата 4 се дава предлог за користење на BFD во рамки на FF, со цел детекција на локална грешка. Подоцна, преку симулации се квантификува подобрувањето на времето за детекција на грешка.

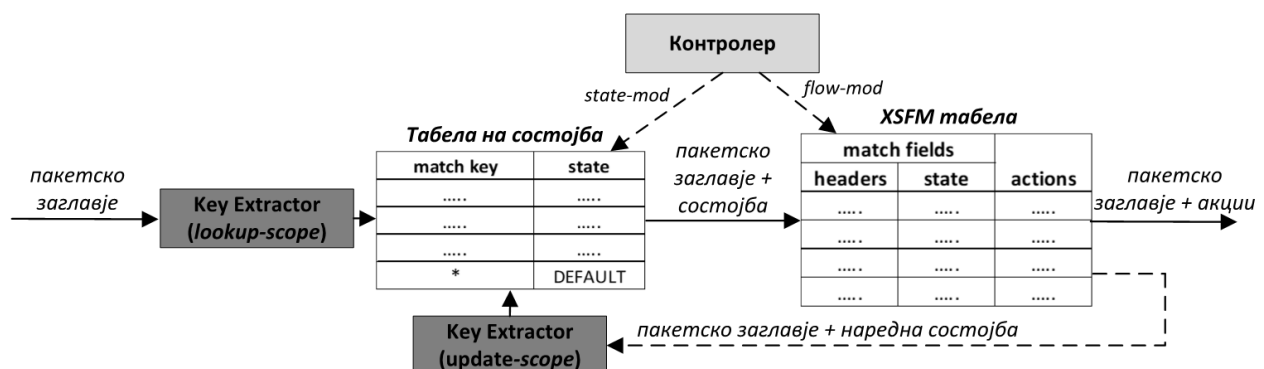
2.1.2 Преминување кон корисничка рамнина со локални состојби

Основната идеја на SDN е да обезбеди целосна поделба на корисничката и контролната рамнина. Ова значи дека целокупната интелигенција е сместена во

контролерот, а комутаторите остануваат максимално едноставни. Во последните години, се водат дискусии за промовирање на идејата за “симнување” на дел од интелигенцијата од контролерот назад кон комутаторите. Постои генерално мислење дека неколку задачи на состојби (на англ. *stateful tasks*) непотребно се централизирани и постојат многу предности доколку се воведат локални состојби во самите комутатори. Потребата за вклучување на контролерот за било кое процесирање на состојба и за секое освежување на *match/action* полињата, води кон непотребна деградација на перформансите на мрежата, процесирачко доцнење и дополнително сигнализациско оптоварување. Дополнително, бавната контролна рамнина го спречува користењето на 5Г мрежните контролни алгоритми кои за дефинираните кориснички случаи бараат конфигурација во корисничката рамнина во реално време (на англ. *realtime*).

Во кратки црти, воведувањето на корисничка рамнина со локални состојби во предлогот наречен *OpenState* [4], имплицира воведување на табела на состојба (на англ. *state table*) пред да се направи проверка за совпаѓање на пакетското заглавие со табелите на тек во OF. Моделирањето на состојби се прави преку користење на *eXtended Finite State Machines* (XFSM) или подобро познати како *Mealey Machine*. Ова е апстрактен модел кој се состои од четирикратно множество параметри (S, I, O, T) и иницијална номинална состојба S_0 каде:

- S е конечно множество на состојби,
- I е конечно множество на влезни симболи (настани),
- O е конечно множество на излезни симболи (акции),
- $T: S \times I \rightarrow S \times O$ е транзициска функција која мапира $\langle \text{состојба}, \text{настан} \rangle$ во $\langle \text{состојба}, \text{акција} \rangle$.



Слика 2.4. OpenState табели

Дизајнот на *OpenState* е прикажан на сликата 2.4. Користи две засебни табели, табела на состојба и XFSM табела. Воведени се следните чекори:

- *State Lookup*: За влезниот пакет, прво се прави проверка за совпаѓање во однос на табелата на состојба и како клуч се користи дел од пакетското заглавие кој го идентификува текот (IP изворна и IP дестинациска адреса), дел кој се нарекува *lookup-scope*. Доколку не се најде соодветна состојба на тек, се користи состојбата *DEFAULT*.
- *XFSM транзиција*: Состојбата најдена во првиот чекор и полето во заглавието користено за проверка за совпаѓање на настан (број на порта), се користат за проверка за совпаѓање со табелата XFSM. Резултат од оваа проверка за совпаѓање е *наредна акција* (на англ. next action) и *ознака на наредната состојба* (на англ. label of the next state),
- *Освежување на состојба* (на англ. State update): Ова е освежување на табелата на состојба со *ознака за следна состојба* (на англ. label of the next state). Овој дел се нарекува *update-scope* и се состои од *следна состојба* + *пакетско заглавие* (на англ. next_state + pkt_header).

Конечно, за да функционира целиот дизајн, мора да се додадат тајмаути кон табелата на состојба, за да може API програмерот експлицитно да специфицира различни тајмаути за секоја посебна транзиција на состојба. Во главата 4 се предлага нов механизам за детекција на далечна грешка базиран на принципот *OpenState*.

2.1.3 Бафер механизми

Разделувањето на контролната и корисничката рамнина воведува предизвици за нови истражувања во однос на побарувањата за баферирање поради различните временски размери во пакетското процесирање и во волуменот на сообраќајот во корисничката и контролната рамнина. Процесирањето на пакети во еден SDN комутатор се класифицира во две групи: i) *брз пат* (на англ. fast path) – се однесува на процесирањето во корисничката рамнина (пр. *checksum* пресметка, декремент на *Time to Live* (TTL)) и ii) *бавен пат* (на англ. slow path) – се однесува на процесирањето во контролната рамнина, функции за управување со комутатори и процесирање на исклучоци (пр. IP пребарување). Пакетите кои поминуваат низ брзиот пат очекуваат ниска латентност, типично со магнитуда од стотици наносекунди, додека пакетите кои поминуваат низ бавниот пат, доживуваат доцнење од десетици милисекунди.

Доцнењата на пакетите во еден комутатор се одредени од капацитетот и брзината на опслужување на излезните бафери затоа што излезните бафери во комутаторот се делат и користат од сите порти. Комутаторите со т.н. делен бафер (на англ. shared buffer) се ценовно

ефективни затоа што повеќе влезни порти делат единствен излезен бафер со што се намалува цената на чинење и комплексноста во комутаторот. Алтернативно, може да се користат два посебни бафери за брзиот и бавниот пат, при што приоритет секогаш има пакетот на бавниот пат. Во Глава 6, се моделираат комутатори кои користат засебни приоритетни бафер и се прави споредба на добивките и цената на чинење при споредба со случајот на единствен делен бафер.

2.2 Пристапи при анализа на перформанси

Со дигиталната револуција современите комуникациски мрежи стануваат сè покомплексни и понапредни. Технолошкиот напредок ги промени различните аспекти на човечкиот живот, од образование, истражување, развој и бизнис [15]. Дизајнерите на телекомуникациските мрежи постојано работат на подобрување на комуникациските системи со помош на моделирање и алатки за анализа на перформанси. Алатките за анализа на перформанси им даваат слобода и флексибилност на мрежните дизајнери да го проучуваат и предвидат системот пред негова имплементација.

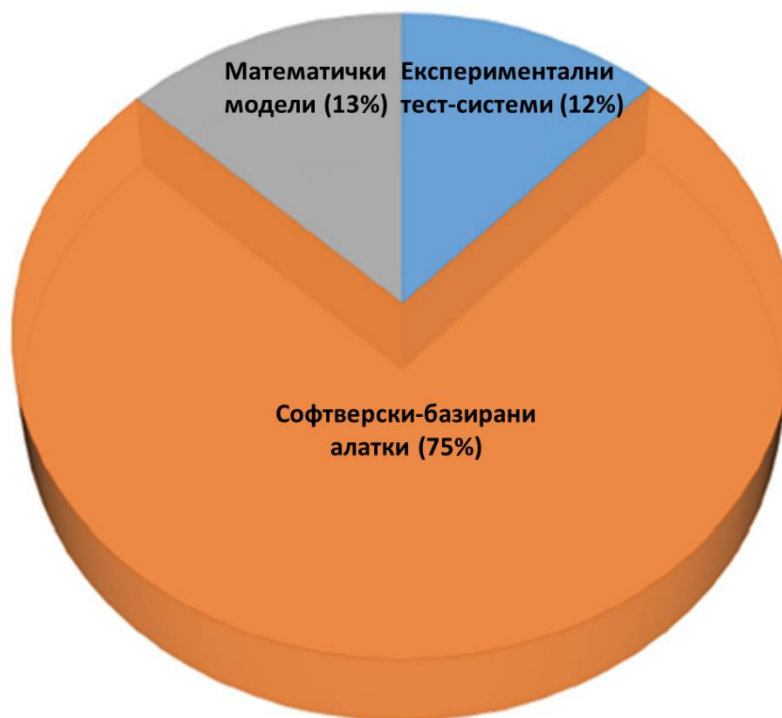
Поради тоа што SDN е нова мрежна архитектура, има потреба од идентификација и анализа на различни дизајн аспекти кои влијаат на севкупните мрежни перформанси, како што се процесирачко доцнење, капацитет на бафер, приоритизацијата на бафери, севкупен проток на системот итн. Ова се причините поради кои треба да се изврши анализа на перформансите со цел идентификација на потенцијални тесни грла (на англ. *bottlenecks*) кои можат негативно да влијаат на севкупните перформанси на SDN мрежата. Трите најчесто користени перформансни метрики за SDN мрежи се:

- Доцнење: Се однесува на времето кое е потребно да се испрати пакет од праќач (на англ. *sender*) кон приемник (на англ. *receiver*). Фокусот најчесто се става на средното време на процесирање на пакети во мрежните елементи и времето на трансмисија на податочниот линк.
- Проток: Ова е брзината на севкупниот број на пакети трансмитирани од праќачот кон примачот.
- Рата на губење пакети (на англ. *packet loss rate*): Ова е број на пакети кои се блокирани или отфрлени од вкупниот број на пакети трансмитирани од праќачот. Се изразува во проценти.

Во комуникациските мрежи, овие перформансни метрики имаат големо влијание врз севкупното корисничко искуство. Доцнењето и протокот ја одредуваат брзината на

мрежата, додека ратата на губење пакети е негативен индикатор кој предизвикува прекин на комуникацијата.

Анализата на перформанси во една SDN мрежа може да се направи на три различни начини: експериментални тестни системи (на англ. test-bed), софтверски-базирани алатки и математички модели [16]. Од истражувањето (на интернет се истражени повеќе стотици трудови кои се фокусираат на SDN) прикажано на сликата 2.5, јасно се гледа нека не постојат многу математички модели и експериментални тестни системи кои може да се искористат за анализа на перформанси во SDN-базирани мрежи.



Слика 2.5. Пристапи при анализа на перформанси во SDN

Експерименталните тестни системи се хардверски имплементации во реално време кои им овозможуваат на истражувачите да ги валидираат и тестираат предложените алгоритми и механизми. Тестните системи овозможуваат флексибилна околина за тестирање, мерење и валидација на нови мрежни технологии и апликации со вистински сообраќај. Тие користат високо-перформансни реални уреди и поради ова се поубедливи од емулаторите и симулаторите [17]. Сепак, експерименталните тестни системи се скапи за имплементација и имаат проблеми на скалабилност поради хардверските ограничувања [18].

Софтверски-базирани алатки се користат за отстранување грешки (дебагирање), верификација и симулација или емулација на SDN апликации и API [19]. Алатките за дебагинг и тестирање го обезбедуваат квалитетот на SDN софтверот преку ослободување од грешки и софтверски багови. Слично, алатките за верификација се користат за

верифицирање и анализа на карактеристиките на SDN комутаторите и контролерите. Симулатор претставува софтвер со кој се поставува потребната околина за симулација на однесувањето на хардверот, додека емулаторот ја емулира секоја страна на однесувањето на хардверот со цел да се направи прецизна копија на вистинскиот хардвер. Некои примери на симулатори кои редовно се користат во SDN се: NS-3, MATLAB, OMNet++ итн. Mininet претставува најпознатиот емулатор кој редовно се користи за проучување на SDN мрежите. Симулаторите и емулаторите се поефтини и пофлексибилни во споредба со тестните системи. Сепак, симулаторите користат вештачки сообраќај, додека емулаторите користат вистински сообраќај. Во оваа дисертација се користат MATLAB и Mininet за верификација на предложените алгоритми и математичките модели.

При дизајнирање нови мрежи или за оптимизација на постојни мрежи, се користи математички-базирана анализа на перформанси која претставува ценовно-ефективно решение за предикција на перформансите метрики [20]. Математичкиот модел го претставува системот во форма на математички равенки, теореми и докази. Моделот може да се користи за поддршка на експериментални или симулациски резултати. Исто така, има предност што е поефтин и побрз од тестните системи и симулаторите. Сепак, бројот на постојни модели за математичка анализа на SDN-базирани мрежи е многу ограничен. Двете основни математички методологии кои се користат за моделирање на SDN мрежи се теоријата на редови и мрежен калкулус (на англ. network calculus). Теоријата на редови ги претставува перформансите на системот во просечни количини и состојба на еквилибриум, додека мрежниот калкулус ги прикажува перформансите на системот со ограничена крива на веројатност (на англ. probabilistic bound curve) во услови на најлошо сценарио. Во оваа дисертација се користи теоријата на редови како основна математичка методологија за моделирање на мобилни јадрени SDN-базирани мрежи.

2.2.1 Теорија на редови

Теоријата на редови е класична математичка алатка за проучување редови. Се користи за анализа на перформанси и моделирање на компјутерски мрежи, каде компјутерската мрежа е претставена како мрежа од редови [21]. Се користи за анализа на мрежно доцнење и мрежен проток во услови на стационарна состојба (на англ. steady state). Во теоријата на редови, за опишување и класификација на редови се користи нотацијата на Kendall [22], т.е. $A/S/c/K/D$, каде

- A претставува дистрибуција на времето на пристигање,
- S претставува дистрибуција на времето на опслужување,

- c го претставува вкупниот број сервери во системот,
- K го претставува максималниот број пакети во системот,
- D ја претставува дисциплината на редот.

Дистрибуцијата на A и S може да биде експоненцијална дистрибуција (M), детерминистичка дистрибуција (D), Ерланг дистрибуција (E_k), генерална дистрибуција (G), фазна дистрибуција (PH) итн. Слично, најчестите дисциплини на редови се FIFO (First In First Out), LIFO (Last In First Out), PNP (Priority Service) итн.

$M/M/1$ претставува наједноставниот модел на редови каде системот има само еден сервер со Поасонова дистрибуција на пристигање, средното време на опслужување следи експоненцијална дистрибуција, редот има бесконечен капацитет и FIFO се користи како номинална дисциплина. Основните перформансни метрики кои се одредуваат со користење на теоријата на редови во компјутерски мрежи се доцнење и проток.

Во рамките на теоријата на редови, QBD процесот масовно се користи за детално моделирање на компјутерски мрежи [20]. Некои примери кои користат QBD за детално проучување на компјутерски мрежи се: распоред на резервација на ресурси, доверливост на компјутерски систем, телекомуникациско моделирање со нестрпливи муштерии (на англ. impatient customers) и моделирање на *Peer-to-Peer* (P2P) системи за споделување на фајлови. За детално да се проучи еден SDN комутатор, во Глава 6 се користи пристапот на моделирање базиран на QBD процеси. Поради ова, во следниот дел се опишува нотацијата и концептите на QBD процесите.

2.2.2 Quasi-Birth-Death процес

Временски континуиран Марков ланец (на англ. Markov chain) со мулти-димензионални простори на состојба (на англ. state space) кои може да се партиционираат во расчленети (на англ. disjoint) нивоа се вика QBD процес [23]. Временски континуиран QBD процес е дводимензионален Марков ланец претставен како:

$$\{(X_t, Y_t), t \geq 0\}, \quad (2.1)$$

со простори на состојба $S = \{(i, j) \in \{0, 1, \dots, K\} \times \{0, \dots, L\}\}$ каде i и j ги означуваат променливите на ниво и фаза (на англ. level and phase variables) на процесот, соодветно. Слично, K и L ги одредуваат капацитетите на редот на променливите на ниво и фаза, и можат да бидат конечни или бесконечни. Во SDN мрежата, контролерот може да биде претставен преку променлива на ниво, а комутаторот преку променлива на фаза.

Во мрежа со ред на чекање, QBD процесите може да бидат мулти-димензионални со една променлива на ниво и повеќе променливи на фаза во зависност од бројот на комутатори или редови во мрежата. За N редови, состојбата на мрежата може да се претстави преку вектор $n = \{n_1, n_2, \dots, n_N\}$, каде n_l е бројот на пакети во редот l . Ако редот 1 е редот од интерес за анализа, тогаш пакетите во редот 1 (т.е. n_1) се претставени преку променливата на ниво, а пакетите во редовите различни од редот 1 се претставени преку променливите на фаза, како вектор $r = (n_2, n_3, \dots, n_N)$.

Во QBD процесите, транзициите помеѓу состојбите се ограничени во рамки на променливата на ниво или помеѓу две соседни нивоа. Ако транзициите на QBD процесот се независни од нивото, тогаш типот на QBD процес е хомоген или независен од нивото (на англ. level independent). Слично, ако транзициите се зависни од нивото, тогаш QBD процесот е не-хомоген или зависен од нивото (на англ. level-dependent).

Користејќи стандардно QBD означување [10], транзициската матрица на хомоген QBD процес е дадена преку матрица на инфинезимален генератор (на англ. infinitesimal generator) со повторувачка три-дијагонална блок структура, како што е прикажано подолу:

$$Q = \begin{pmatrix} B_1 & A_0 & 0 & 0 & \dots \\ A_2 & A_1 & A_0 & 0 & \dots \\ 0 & A_2 & A_1 & A_0 & \dots \\ 0 & 0 & A_2 & A_1 & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}, \quad (2.2)$$

каде B_1 , A_0 , A_1 , и A_2 се стандардни означувања на блок-матрици кои ги претставуваат дистрибуциите на променливите на фаза во хомоген QBD процес. Матриците A_1 and B_1 соодветно ја претставуваат дистрибуцијата на фаза за граничната состојба (т.е. $i = 0$) и не-граничната состојба (т.е. $i \neq 0$) кога променливата на ниво останува непроменета (т.е. $i \rightarrow i$). Слично, A_0 и A_2 ги претставуваат фазните дистрибуции кога променливата на ниво се зголемува (т.е. $i \rightarrow i + 1$) или се намалува (т.е. $i \rightarrow i - 1$, за $i > 0$), соодветно. Во една SDN мрежа, A_1 или B_1 , A_0 и A_2 претставуваат дистрибуција на хомогена состојба на комутаторот кога бројот на пакети во контролерот останува непроменет, се зголемува за 1 и се намалува за 1, соодветно.

Со помош на *Matrix Geometric/Analytic* методот [24], дистрибуцијата на стационарна веројатност на хомогениот QBD процес π , може да се пресмета за да задоволува систем од равенки $\pi Q = 0$ и $\pi e = 0$, каде e вектор-колона со единици. Стационарната веројатност се дефинира како:

$$\pi_{ij} := \lim_{t \rightarrow \infty} P(X_t = i, Y_t = j), \quad (2.3)$$

каде $\pi_i = (\pi_{i0}, \pi_{i1}, \dots, \pi_{iL})$, за $i = 0, 1, \dots, K$ и $\pi_i = (\pi_0, \pi_1, \pi_2, \dots)$. Стационарните веројатности може да се користат за пресметка на мрежните перформанси, како што се проток, доцнење и рата на губење пакети.

Хомогениот QBD процес е рекурентен, позитивен и марковиот ланец е ергодичен ако:

$$\pi_A A_0 < \pi_A A_2 e, \quad (2.4)$$

каде π_A е вектор на стационарна веројатност на матрицата A кој е еднаков на $A_0 + A_1 + A_2$. За ергодичен марков ланец, системот на равенки $\pi Q = 0$ и $\pi e = 0$ е еквивалентен на:

$$\pi_0 B_1 + \pi_1 A_2 = 0, \quad (2.5)$$

$$\pi_{i-1} A_0 + \pi_i A_1 + \pi_{i+1} A_2 = 0, \quad (2.6)$$

каде $\pi_i = \pi_{i-1} R$ за $i \geq 2$, а R е матрица на рата (на англ. rate matrix) на QBD процесот со димензии $(L + 1) \times (L + 1)$ и е не-негативно решение на равенката:

$$A_0 + R A_1 + R^2 A_2 = 0. \quad (2.7)$$

Матрицата R може да се пресмета со користење на различни итеративни алгоритми од кои најпознат е Логаритамска редукција (на англ. Logarithmic Reduction (LR)) [25], кој најчесто се користи при анализа на перформанси на комуникациски системи поради брзата рата на квадратна конвергенција [26]. LR методот всушност ја пресметува матрицата G , наместо директно пресметување на матрицата R . Матрицата G ја претставува дистрибуцијата на веројатност на погодок (на англ. hitting probability distribution) на состојба во дадено ниво и има интуитивна стохастичка интерпретација за ергодичниот QBD процес. Матрицата G има не-негативно решение на следната равенка:

$$A_2 + G A_1 + G^2 A_0 = 0. \quad (2.8)$$

Во оваа дисертација се користи хомоген QBD процес и од примарен интерес е пресметката на стационарната дистрибуцијата π , со чија помош потоа се пресметуваат перформансните метрики како што се проток, средна должина на ред и средно доцнење на редот. Матрицата R се пресметува со помош на LR методот.

3. LTE/5G мобилни јадрени мрежи и примена на SDN

Идните 5G мобилни мрежи се сметаат за следниот чекор во еволуцијата на постојните LTE имплементации. Главната причина за воведување на 5G е ефикасно и ценовно-ефективно адресирање на потребите на вертикалниот пазар кои се појавуваат во рамки на индустриската глобализација [27]. Во моментот, најголемиот дел од мобилните оператори се веќе вклучени во 5G тестирања, ограничени комерцијални имплементации, или се во фаза на набавка на соодветна опрема која ќе поддржува 5G. Сепак, масовна комерцијална имплементација на 5G се очекува најрано во 2020.

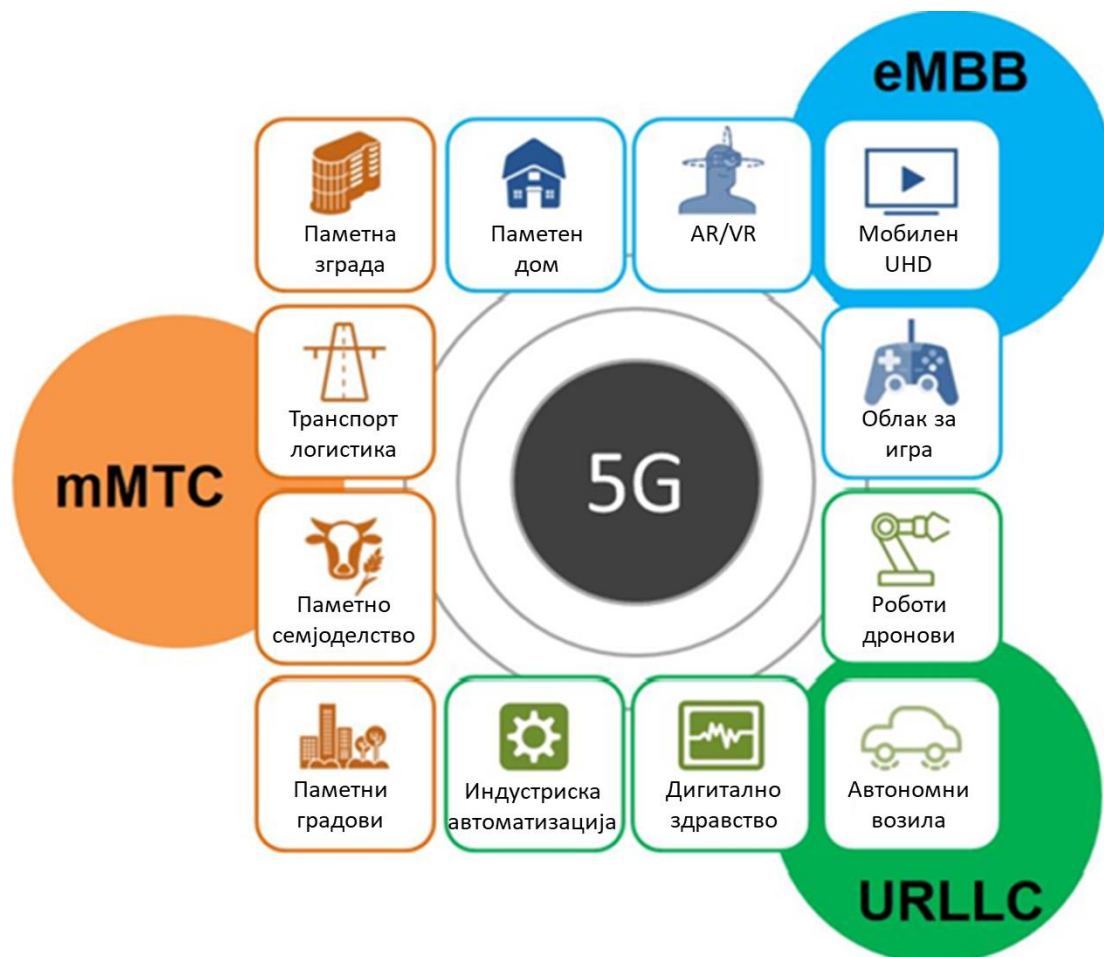
Главните трендови кои влијаат на 5G стандардизацијата и ги мотивираат операторите да инвестираат во мрежна модернизација се следните:

- Неверојатниот пораст на мобилен податочен сообраќај кој се очекува да се зголеми 20000 пати во периодот 2010 до 2030,
- Поврзување на различни типови на “нешта” (на англ. things) и еволуција од *Machine to Machine* (M2M) кон *Internet of Things* (IoT). Бројот на поврзани уреди во овој сегмент се очекува да достигне над 4 милијарди во 2024 [28],
- Појава на нови кориснички случаи како што се: надградена реалност (на англ. augmented reality), допирлив интернет (на англ. tactile internet), мобилно здравство, ултра-брзи паметни решетки (на англ smart grids) итн.

5G како технологија треба да акомодира огромен број на услуги и апликации кои имаат многу разновидни барања. Денес постојат многу кориснички случаи веќе стандардизирани од 3GPP (на англ. 3rd Generation Partnership Project), сепак сите тие генерално може да се класифицираат во три групи (види Слика 3.1):

- eMBB (на англ. Enhanced Mobile Broadband) – опфаќа кориснички случаи поврзани со самите луѓе кои пристапуваат кон мултимедијална содржина, податоци, или друг тип на услуги. Постојните LTE мрежи се дизајнирани на начин да го оптимизираат овој тип на MBB услуги и се смета дека LTE е MBB-ориентирана мрежа. Но сепак, денес постои дури и поголемо побарување за повисоки податочни брзини, во главно поради експлозивниот пораст на мобилни уреди и современите мултимедијални и забавни содржини. Наскоро се очекува дека веќе дизајнираните LTE мрежи нема да можат да ги поддржат новите барања на корисниците. eMBB во основа се однесува на традиционалните MBB услуги, новите апликации и треба да обезбеди непрекинато корисничко поврзување насекаде и во било кое време,

- mMTC (на англ. Massive Machine Type Communications) – се однесува на сценаријата кога уреди комуницираат меѓу себе без човечка интеракција. Ова се типично случаи на голем број поврзани и евтини уреди, кои имаат долг живот на батерија и трансмитираат многу ретко информација која не е осетлива на доцнење. Постојните LTE имплементации не се соодветни за ваков тип апликации и не ја обезбедуваат потребната густина на конекции која се мери во број на поврзани или достапни уреди по ќелија,
- URLLC (на англ. Ultra-Reliable and Low Latency Communications). Ова се специфични кориснички случаи кои имаат многу строги барања во однос на латентноста и расположивоста. Типични примери на URLLC се медицинска операција од далечина и безжична контрола на производствени процеси во различни индустрии. Денешните LTE мрежи едноставно не се дизајнирани да ги поддржат овие нови услуги.



Слика 3.1. Примери на 5Г кориснички случаи

Погоре-наведените групи на услуги наметнуваат дивергентни и конфликтни перформансни ограничувања кои го усложнуваат нивното истовремено постоење во идните 5Г мрежи. Поради ова, нивната интеграција користејќи заедничка инфраструктура, се смета

за еден од најголемите предизвици во 5Г. Базирано на дефинираните кориснички случаи, ITU-R (на англ. Radiocommunication Sector of International Telecommunication Union), има дефинирано клучни перформансни индикатори (на англ. Key Performance Indicators (KPI)) и нивни вредности во 5Г мрежи, што е прикажано на сликата 3.2.



Слика 3.2. Клучни перформансни индикатори за трите главни 5Г групи на услуги

Во Глава 4, се предлага нов механизам за детекција и корекција на грешка соодветен за 5Г мобилните мрежи и во оваа дисертација од интерес е групата на URLLC. Како што може да се види на сликата 3.2, оваа група побарува исклучително ниска латентност, додека големи податочни брзини не се во преден план. Во понатамошниот дел од текстот, се прави анализа на факторите кои влијаат на латентноста во LTE/5Г, но пред тоа потребно е да се разбере кои се основните технологии кои овозможуваат многу мала латентност. Во оваа анализа нема да се дискутира за концептот на мрежно делење (на англ. network slicing), затоа што од интерес е само воведување на 5Г преку едноставна надградба на постојните мобилни пакетски мрежи, при што концептот на мрежно делење не е поддржан. Подоцна во оваа глава, се дискутира специјален тип на надградба на LTE пакетските јадрени мрежи кој овозможува минимална инвестиција од операторите, концепт кој се нарекува парцијална виртуелизација.

Постојните LTE мобилни мрежи се соочуваат со мала флексибилност, заклучување со одреден вендор и високи капитални и оперативни трошоци за операторите. Ова е главно поради користењето на затворени (на англ. proprietary) протоколи, вертикално интегриран

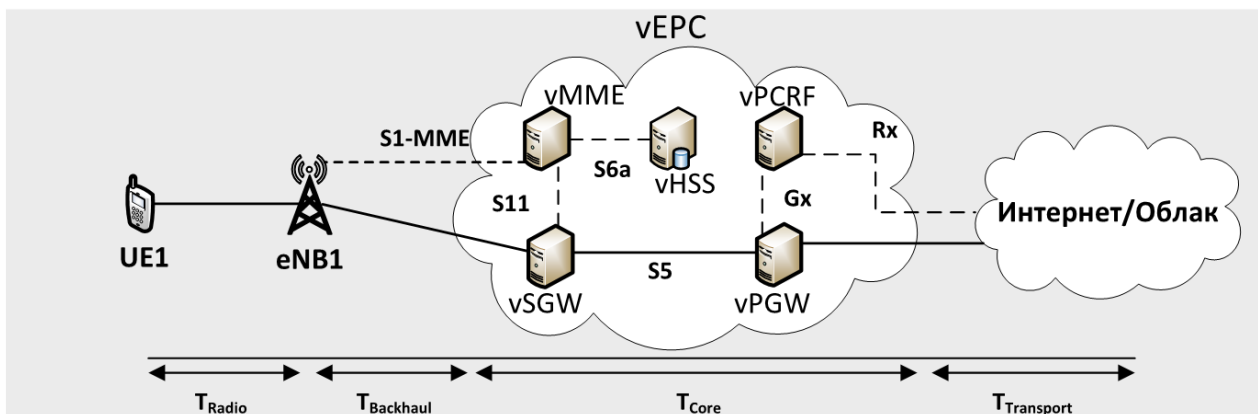
хардвер и немање програмабилни опции за воведување на нови прилагодени услуги. Петата генерација мобилни мрежи ги адресира овие предизвици преку воведување на концепти на мрежна софтверизација, главно базирани на SDN и NFV. Покрај SDN кој е детално опишан во Глава 2, NFV дозволува раздвојување на апликативната функција од затворениот хардвер и дозволува употреба на комерцијален (на англ. off-the-shelf) хардвер. Целта на NFV и SDN е: i) воведување на иновативни услуги, ii) пократко време за маркетинг, iii) намалување на трошоците на операторите и iv) подобрување на корисничкото доживување.

Друга важна технологија која влијае на намалувањето на вкупната латентност претставува *Multi-access Edge Computing* (MEC). Концептот се нарекува пресметка на работ на мрежата (на англ. edge computing), каде ресурсите за процесирање и складирање, како и потребните сервери на содржина (на англ. content servers) мораат физички да бидат лоцирани блиску до корисникот. Причината за ова е обезбедување на минимална латентност, но и ослободување (на англ. offloading) на тешките процесирачки процеси од мобилната станица кон мрежата. Последново е особено важно, бидејќи е единствен начин како мобилните уреди, паметните наочари и слично, да бидат дизајнирани за да се обезбеди нивна соодветна мала големина и модерен изглед што директно влијае на пенетрацијата на уредите и популарноста на 5Г корисничките случаи (пр. виртуелна реалност). Во понатамошниот дел од текстот се дискутира за проблемот на латентност во постојните LTE мрежи и побарувањата за латентност крај-до-крај во 5Г.

3.1 Проблемот на латентност во LTE и зошто латентноста е толку значајна во 5Г

Просечната латентност во постојните LTE мобилни мрежи се мери во десетици милисекунди и најчесто е под 100 милисекунди [29]. Сепак, ова се просечни вредности и тие лесно се афектирани од оптоварувањето во мрежата и географската имплементација. Во LTE нема гаранција за одредена латентност во мрежата, уште повеќе, LTE архитектурата не прави разлика помеѓу услуги кои бараат одредено гарантирано ниво на латентност. Се заклучува, во LTE не постои мрежна услуга која може да искористи специјализирани мрежни ресурси со кои би се гарантирале перформансите барања на корисничката апликација во однос на доцнењето. Во подоцнежните 3GPP објави, постојат обиди за подобрување на LTE латентноста со воведување на нови решенија како: намалување на интервалот на трансмисија (на англ. Transmission Time Interval (TTI)), *Semi Persistent Scheduling* (SPS), латентност при хендовер итн.

За да може да се анализира проблемот на латентност, мора да се анализира целиот ланец помеѓу крајните точки во мрежата. Во LTE може да се зборува за компоненти на латентност кои се прикажани на сликата 3.2.



Слика 3.2. Контрибутори на латентност во вкупното доцнење крај-до-крај при пакетска трансмисија.

Вкупната латентност е сума на следниве поединечни конституенти [30]:

- T_{radio} е времето на трансмисија помеѓу *User Equipment* (UE) и базната станица eNodeB (eNB) и овде најмногу влијае физичкото ниво (на англ. *physical layer*). Се состои од: време на трансмисија, време на процесирање во UE и eNB, пропагациско доцнење и ретрансмисији. Времето на процесирање во UE вклучува сегментација на коден блок, канално кодирање, мултиплексирање, интерливинг итн. Времето на процесирање во eNB е поради канално кодирање, *Cyclic Redundancy Check* (CRC), мапирање на брзината итн.
- $T_{backhaul}$ е латентноста помеѓу eNB и EPC. Овој дел од мрежата може да биде базиран на микробранова мрежа или оптички влакна, каде генерално микробрановата мрежа обезедува помала латентност од оптичката мрежа,
- T_{core} е латентноста воведена од јадрената мрежа, поради различните EPC ентитети и процедурите овде се типично: EPC контрола на носител, управување со мобилноста, сигурност, филтрирање и инспекција на пакети итн.
- $T_{transport}$ е доцнење кое се јавува помеѓу излезниот рутер во јадрената мрежа и интернет. Оваа латентност е афектирана од опсегот (на англ. *bandwidth*), растојанието помеѓу серверот и јадрената мрежа и типот на користени протоколи.

T_{radio} во постојните LTE мрежи изнесува околу 1 милисекунда. За намалување на $T_{backhaul}$ типично се користат напредни техники во подмрежа (на англ. *backhaul*) како кеширање или “магла” (на англ. *fog*) базирани пристапи. За намалување на латентноста T_{core}

веќе се спомена дека најголема улога има воведувањето на концептите на SDN и NFV. За $T_{\text{transport}}$ се користат веќе опишаниот *Multi-access Edge Computing*, интернет кеширање итн.

Некои 5Г кориснички случаи побаруваат многу ниска латентност и многу висока расположивост на комуникациска услуга. Барањата за овие сценарија се дефинирани во 3GPP TS 22.261 V16.5.0 [31] и типично се однесуваат на: контрола на движење, дискретна автоматизација, автоматизација на процеси, интелигентни транспортни системи, допирлива интеракција итн. Латентноста крај-до-крај во 3GPP Rel. 16 се дефинира како време на пренос на парче информација од изворот до дестинацијата, од моментот на испраќање од изворот, до моментот на успешно примање од дестинацијата. Типично, корисничките случаи (пр. виртуелна реалност, далечински операции, паметни решетки) кои бараат латентност помала од 5 ms, мора да користат приоритизација на сообраќај и MEC блиску до крајниот корисник. Другата група на кориснички случаи (пр. далечинско управување со возила или дрoнови, временско-критичен паметен град, колаборативен гејминг итн.) бара латентност крај-до-крај помала од 20 ms. Последната група на кориснички случаи (видео во реално време, масовно поврзани машини, освежување на дигитални мапи итн.) имаат побарување за латентноста помало од 100 милисекунди и се сметаат за не-критични.

Дискусијата погоре е со цел да се разберат строгите барања за латентност во 5Г и генерално ниската толеранција на било каков тип доцнење. Иако не постојат експлицитни 5Г барања за времето на опоравување од грешка во мобилните јадрени мрежи, сепак треба да се заклучи дека се очекуваат многу строги барања ($< 20\text{ms}$). Во Глава 4 се предлага и анализира механизам за детекција и опоравување од грешка во мобилни јадрени мрежи во 5Г и преку симулации се покажува дека понудениот алгоритам може да обезбеди времиња на опоравување со вредности од 5ms до 20ms во зависност од сценариото на грешка и дистрибуцијата на грешка (локална или далечна). Сценариото од интерес во симулациите во Глава 4 се однесува на SDN-базиран EPC со парцијална виртуелизација. Истиот тип надградба на постојните LTE јадрени мрежи е искористен и во Глава 6 и е опишан во наредното поглавје.

Табела 3.1. Минимална латентност за поддршка на примери кориснички случаи од вертикални индустрии (извадено од [31])

Услуги/ Кориснички и случаи	Автомобилска индустрија	Транспорт, логистика и IoT	Здравје, паметни градови	Медиа и забава
Опис	Проширување на опсегот на детекција на капацитетот на вградените сензори преку делење на прегледи или детекција на објекти помеѓу учесниците во сообраќајот, координација на траектории помеѓу возила, делење на груби намери при возење (на англ. coarse driving), далечински операции на возила во реално време	Реално чувствување (на англ. real-time sensing), репортирање, повратна информација, контрола, следење на имот (на англ. asset tracking), мониторирање, услуги со свесност на содржина (на англ. context-aware), препораки за трговски центри, аеродроми	Видео во живо (4K, 8K, 3D) за далечна здравствена нега (консултации, надзор) и асистирана операција, Команди за контрола на медицински уреди во реално време (пр. медикаменти, операција, далечен надзор, надгледување и водење на граѓани и полиција/војска)	Продукција на медиа услуги базирани на агрегација на различни потоци на медиа на серверите, P2P или сервер-клиент делење на податоци (објектна информација) за колаборативен гејминг, стриминг во живо на спортски, концертни или други настани
Латентност	<p>За долгорочно моделирање на околина (динамични дигитални мапи со висока резолуција): Не критични (100 ms крај-до-крај)</p> <p>За краткорочно моделирање на околина (делење на сензори): <20 ms крај-до-крај</p> <p>За кооперативност (координирана контрола): - <3 ms крај-до-крај за возила во група (на англ. platooning), - <10 ms крај-до-крај за кооперативни маневри, - <100 ms крај-до-крај за груби намери при возење,</p> <p>За далечинска операција со возила: 10-30 ms крај-до-крај</p>	<p>За масивна конективност за временско-критично чувствување и повратна информација: <30 ms крај-до-крај.</p> <p>За далечинска операција со дронови и кооперативна машинерија: 10-30 ms крај-до-крај</p> <p>Контрола во реално време за дискретна автоматизација: ≤1 ms крај-до-крај</p>	<p>За видео во реално време/телеприсутност /надградена реалност за далечно здравство и асистирана операција, за мониторирање и водење (паметни градови): 100 ms крај-до-крај</p> <p>Команди и контрола за далечна медикација и операција: 10-100 ms крај-до-крај</p> <p>За паметни решетки (на англ. smart grids): - <5 ms крај-до-крај за трансмисија и јадро на електрична мрежа: - <50 ms крај-до-крај за дистрибуција/електрична подмрежа (на англ. grid backhaul),</p> <p>Временско-критично чувствување и повратна информација за паметни градови: 30 ms крај-до-крај</p>	<p>За стриминг во живо во густо (на англ. crowded) области, услуги за медиа продукција, надградена реалност за колаборативен гејминг итн: 20 ms крај-до-крај</p>

3.2 Надградба на LTE пакетски мрежи, воведување на SDN концептот и парцијална виртуелизација

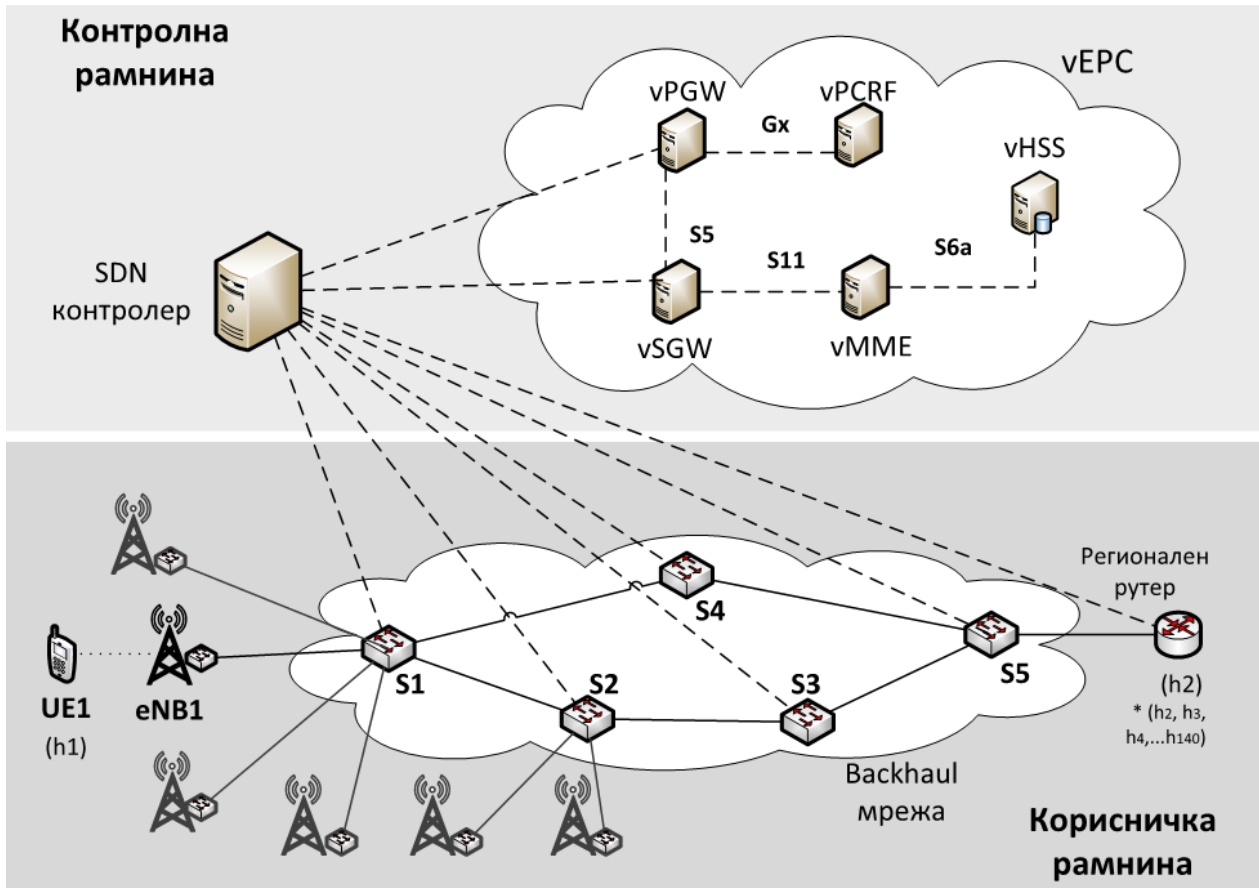
Најголемото ограничување кај постојните LTE мрежи кое влијае на латентноста е фактот што контролната и корисничката рамнина не се целосно поделени. Разделувањето на контролната и корисничката рамнина е неодминлива потреба затоа што двете имаат различни критериуми на мрежен Quality of Service (QoS) кои мора да бидат задоволени. Контролната рамнина има потреба од ниска латентност заради процесирање на сигнални пораки, а корисничката рамнина има потреба од висок проток при процесирањето на информацијата. Ова е една од причините за воведување на SDN преку целосна сепарација на корисничката и контролната рамнина во новата јадрена мрежа наречена 5G EPC. Се воведува SDN контролер со улога на интерфејс помеѓу раздвоените рамнини и овој пристап овозможува независна скалабилност, подобро управување со корисничката мобилност, флексибилност при дистрибуција на тек и исто така има големо влијание на намалување на латентноста. Разделувањето на рамнините е исто така многу важно затоа што овозможува употреба на технологии за пресметка на работ на мрежата (на англ. edge computing) што дополнително ја намалува латентноста.

Во мобилните мрежи, SDN концептот редовно се употребува заедно со NFV што е еден од главните принципи во 5G јадрените мрежи. NFV понатаму ја намалува латентноста крај-до-крај и ги подобрува перформансите на проток. Денес, многу оператори веќе имаат имплементирано NFV-базирани EPC мрежи за LTE и ги користат сите придобивки што може да ги понуди виртуелизацијата.

Најбрзиот и најефикасен начин за воведување на 5G е преку надградба на постоечкиот EPC и реискористување на постојните мрежни јазли од LTE [32]. 3GPP ја нарекува оваа алтернатива на *Non-Standalone* (NSA) архитектурата како 5G EPC. Се очекува најголемиот број оператори да започнат со воведување на 5G EPC и потоа да еволуираат кон 5G *Standalone* (SA) архитектура, или скратено 5G јадрена мрежа (на англ. 5G Core). 5G Core претставува комплетно нова архитектура предложена во 3GPP Rel. 15 и Rel. 16, која содржи нови мрежни јазли, протоколи, интерфејси и генерално нови концепти во споредба со постојните LTE пакетски мрежи. Во оваа дисертација, од интерес е 5G надградбата на постојните LTE јадрени мрежи, или опцијата наречена 5G EPC.

Во постојните EPC мрежи, *Mobility Management Entity* (MME), *Home Subscriber Server* (HSS) и *Policy and Charging Rules function* (PCRF), како чисти контролни функции, имаат мали побарувања во смисла на проток и капацитет, но строги барања во однос на латентност и процесирање. Од друга страна, *Packet Data Network Gateway* (PGW) и *Serving*

Gateway (SGW), мораат да постигнат многу висок проток. Ова се дел од причините за пристапот наречен парцијална виртуелизација [33], каде само контролните функции се виртуелизираат, додека во корисничката рамнина се реискористуваат постојните физички SDN комутатори. Предлогот кој е прикажан на сликата 3.3 овозможува висок степен на реискористување на постоечката LTE јадрена опрема.



Слика 3.3. 5G EPC со парцијална виртуелизација

Предложената архитектура ги раздвојува контролните функции од функциите во корисничката рамнина, делумно имплементирани во SGW и PGW. Ова значи дека контролните функции на SGW и PGW се логички централизирани и се имплементирани над OF SDN контролер, како апликации. Функциите во корисничката рамнина се инсталирани на SDN комутаторите, управувани од SDN контролерот. Оваа дисертација се фокусира само на корисничката рамнина во мобилната јадрена мрежа која содржи:

- Evolved NodeB (eNB): извршува стандардни функции како што се управување со радио ресурси, контрола на радио носители (на англ. bearers), контрола на радио пристап итн. Дополнително, за сценариото од интерес во оваа дисертација, додадена е поддршка за OF протоколот и со тоа е овозможена конфигурација на корисничката рамнина преку SDNC,

- SGW корисничка рамнина (SGW-D): претставува надграден OF комутатор кој може да енкапсулира/декапсулира *GPRS Tunneling Protocol* (GTP) пакети. Основната задача е препраќање на пакети помеѓу eNB и PGW-D,
- PGW корисничка рамнина (PGW-D): претставува надграден OF комутатор кој поддржува GTP енкапсулација/декапсулација. Основна задача е препраќање на пакети и наметнување на QoS.

Опоравување од испад генерално не е проблем на контролната рамнина, бидејќи овие мрежни функции се секогаш имплементирани како јазли со висока расположивост (локална редуванса или гео-редуванса). Од друга страна, корисничката рамнина редовно страда од чести трансмисиjsки грешки и хардверски испади на самите комутатори. Во Глава 4 се става фокус на детекција и опоравување од грешки директно во корисничката рамнина, посебно специфично на SGW-D и PGW-D испади. Во наредното поглавје се опишуваат основните пристапи кои се користат за опоравување од грешки во мобилните комуникациски мрежи.

3.3 Опоравување од грешки во SDN-базирани јадрени мрежи

Модерните комерцијални целуларни мрежи треба да обезбедат висока расположивост и мораат да постигнат екстремно брзи времиња на реставрација (на англ. restoration) во услови на испад на еден или повеќе мрежни елементи [34]. Расположивоста е многу важна особено во следната генерација на 5G мрежи, каде различните кориснички случаи побаруваат висок проток и многу ниска латентност. Ова се причините поради кои мора да се размислува за планови за мрежна реставрација уште во фазата на мрежно планирање пред имплементација, така што во услови на различни типови испади, постои можност за моментално приспособување, најчесто во смисла на ре-рутирање. Мрежната реставрација треба да се погрижи за постоење на алтернативни рути и со ова да се избегне прекин или нарушување на сообраќајот во услови на испад. Постојат неколку фактори кои треба да се земат предвид кога се дизајнираат процедури на реставрација:

- Цена на мрежна инфраструктура,
- Должина на патеките за ре-рутирање,
- Времиња на опоравување потребни за реставрација,
- Капацитет кој треба да се резервира за реставрација.

Целта при дизајнирањето е да се постигне оптимална мрежна продуктивност со користење на што е можно помалку ресурси и цена. Секако, сето ова мора да се направи во најкратко можно време.

Испад во мрежата се дефинира како непредвидлива и присилна привремена модификација на мрежата која влијае на постојниот сообраќаен дизајн и секогаш резултира со капацитивна деградација на афектираните линкови [35]. Постојат неколку категории на мрежни испади:

- Испад на линкови: при испад на линк во мрежата. Најчестиот пристап за справување со испад на линк е препрепраќање на сообраќајот кон моментно исправните линкови кои имаат доволен капацитет да го носат сообраќајот на неисправниот линк. Другиот пристап е преку користење на редундантни линкови како мерка на претпазливост, но ова е многу поскапо решение,
- Испад на јазли: кога има испад на мрежна опрема. Овој тип на испад практично значи дека не функционираат сите линкови кон тој јазол. Додавањето на редундантна опрема која работи во *standby* режим е најчестото решение за овој случај. Сепак, поради високата цена, најчесто редундантни се само главните рутери во корисничката рамнина или активните елементи во контролната рамнина.
- Единечен испад: кога во даден момент се појавува само една грешка. Со претпоставка дека веројатноста на повеќекратни истовремени грешки е мала, постојат различни типови мрежни модели кои се развиваат за заштита од единечен испад [36].
- Повеќекратни испади: кога повеќекратни испади се случуваат скоро истовремено. Ако веројатноста на повеќекратни грешки не е така мала, постојат неколку типови модели на мрежна реставрација, како што е опишано во [37].

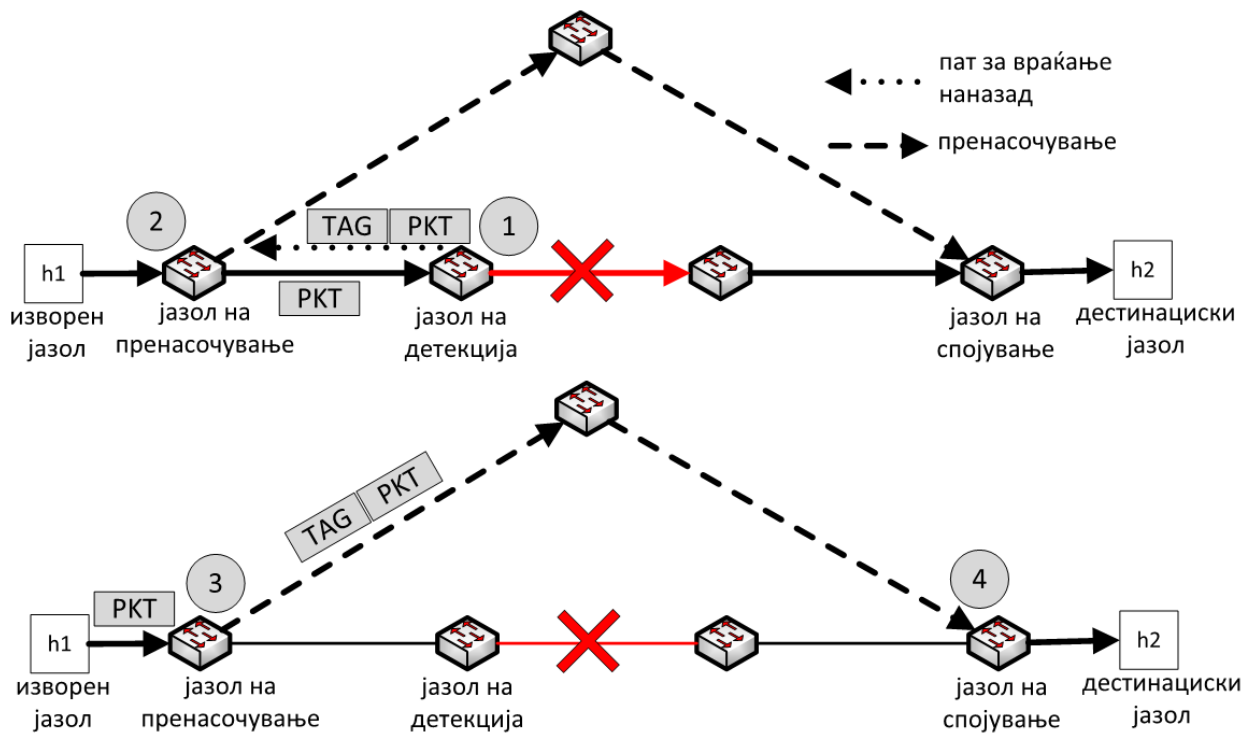
Процедурите за мрежна реставрација може да се класифицираат во неколку категории:

- Реставрација на линк. Примарниот пат кој се користи за рутирање на тек, на кој се случува испад на линк, минимално се модифицира само со замена на линкот зафатен од испад со нова алтернативна рута. За време на овој тип реставрација, најчесто крајните јазли на испаднатиот линк учествуваат во алгоритмот за откривање на нова рута. Реставрацијата најчесто се извршува многу брзо затоа што нема потреба од враќање (на англ. *backtrack*) на пакети на афектираните конекции по истиот пат кон нивните ингрес (на англ. *ingress*) јазли,
- Реставрација на пат: сообраќајот на линкот кој трпи испад се враќа кон неговиот ингрес јазол и се одбираат нови алтернативни рути за сите парови извор-дестинација. Овој тип реставрација вклучува комплетно нов пат кој е сосема различен од

оригиналниот пат. Овие типови механизми имаат дедицирани резервни (на англ. backup) рути кои многу често се чуваат само како резервни и не се користат во услови на нормално сценарио кога нема испад.

- Реактивна реставрација: Алтернативните патишта се пресметуваат само после појавата на испад. Овие се нарекуваат пристапи за реставрација во реално време и се најчести за сценаријата каде сообраќајот во мрежата се менува многу често,
- Проактивна реставрација: Алтернативните рути се однапред провизионирани пред да се случи испад. Инсталацијата на резервниот пат најчесто се прави при иницијализација на контролерот и ова овозможува кратки времиња на опоравување затоа што конекцијата е едноставно пре-рутирана кон претходно дефинирана рута,
- Централизирана реставрација: Постои еден централен контролер кој ги инсталира сите резервни патишта, ги прозива комутаторите и ги извршува сите пресметки на мрежно ниво. Овие типови механизми се многу ефикасни, но исто така претставуваат и единствена точка на испад (на англ. single point of failure) во мрежата,
- Дистрибуирана реставрација: Не постои централна точка на одлука во мрежата, а наместо тоа индивидуални елементи имаат авторизација да носат одлуки за унапредување на мрежната расположивост и обезбедуваат брзо опоравување.

Во Глава 4 од оваа дисертација, се предлага нов механизам за брзо опоравување од испад имплементиран директно во корисничката рамнина. Предлогот се користи само за опоравување од единечни грешки на линк. Понатаму, предмет на анализа е проактивна реставрација на линк со централизиран контролер. За да се кларифицира подрачјето на интерес, сликата 3.4 претставува пример на опоравување од испад на линк и поедноставен тип на механизам кој ќе се користи во главата 4. Во (1), еден од јазлите на примарниот пат детектира испад, го означува (на англ. tag) пакетот и го враќа (на англ. bounce) назад по примарниот пат. Во (2), се наоѓа пат за ре-рутирање (специфичен комутатор кој може да го пренасочи сообраќајот за дадениот пакетски тек), комутаторот извршува транзиција на состојба и го ре-рутира пакетот на алтернативниот пат. Во (3), сите пакети кои припаѓаат на веќе пре-рутираниот пакетски тек се означуваат и насочуваат кон резервниот пат. На крајот, во (4), се отстранува ознаката и пакетот повторно се рутира кон оригиналниот примарен пат.



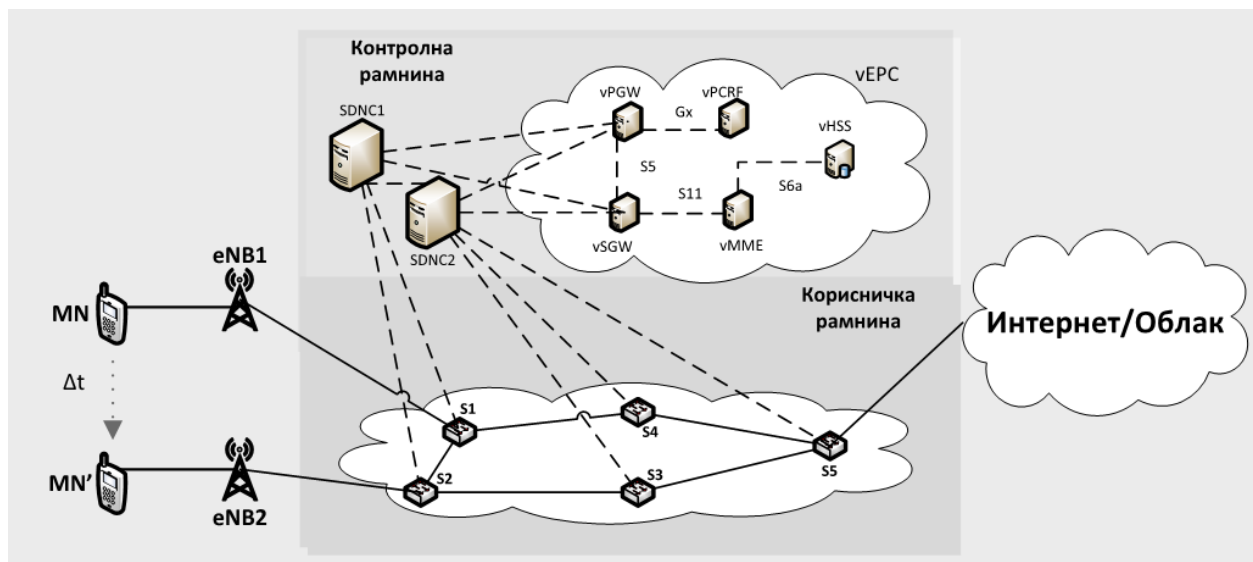
Слика 3.4. Опоравување од испад анализиран во оваа дисертација

3.4 OF-базиран хендовер

Традиционално, операторите секогаш работат на тоа како да обезбедат континуитет на корисничката сесија и како да го минимизираат времето на прекин при извршување на процедурата на хендовер. Доколку мобилниот оператор има имплементирано SDN-базирана мобилна јадрена мрежа, тогаш за време на процедурата на хендовер, помеѓу комутаторите и контролерот се разменува специфично множество на OF сигнални пораки [39]. Посебниот случај на “твррд” хендовер вклучува прекин на тековната сесија и размена на реконфигурациски и пораки за управување помеѓу мрежните ентитети. Во Глава 6, се проучува процедурата на хендовер на MN од еден кон друг OF-комутатор, а целта е математички да се моделира доцнењето при хендовер кое е воведено поради размената на OF контролните пораки.

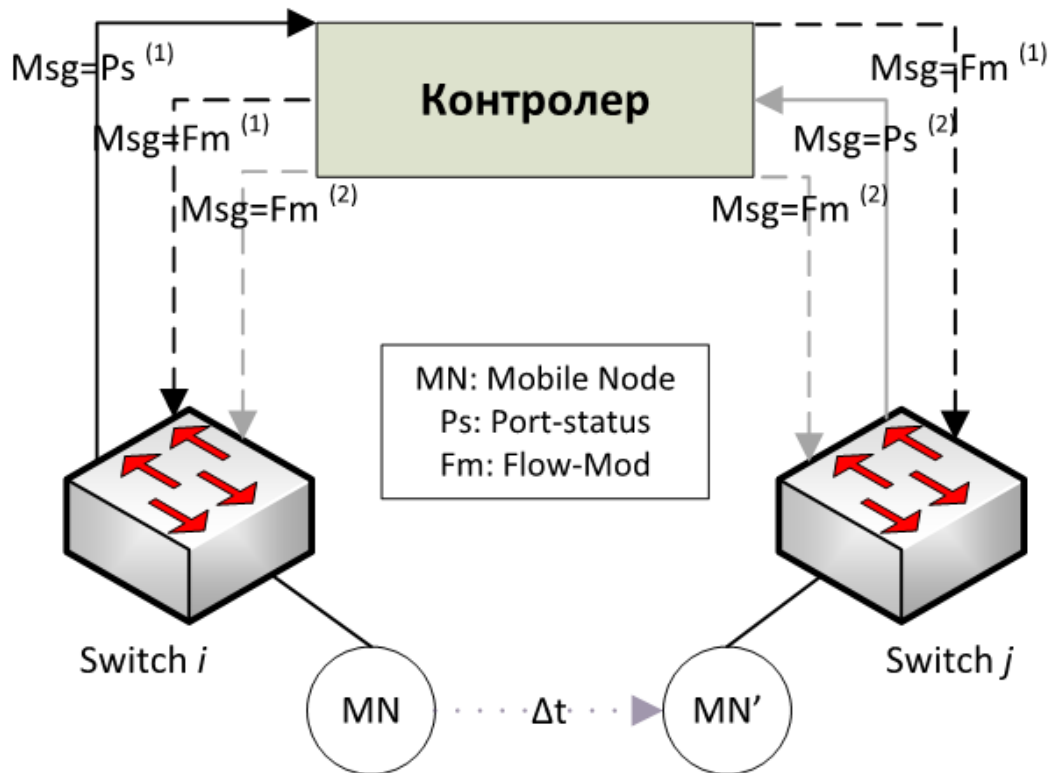
Се претпоставува дека MN се движи наоколу и во еден момент се иницира процедура на хендовер со цел трансфер на тековната сесија од eNB1 кон eNB2. Сценариото од интерес е прикажано на сликата 3.5. Оваа дисертација не се фокусира на влијанието на радио пристапната мрежа (на англ. Radio Access Network (RAN)) или на 3GPP протоколите за мобилност. Наместо тоа, се истражува можноста и се прави квантификација на влијанието што процедурата хендовер го има врз латентноста кога се користи стандарден OF протокол

за сигнализација помеѓу комутаторите и контролерот. Доколку се сака да се инкорпорира придонесот на eNB во латентноста при хендовер, тогаш ова влијание може едноставно да се внесе преку проширување на математичкиот модел со земање предвид на латентноста воведена помеѓу MN и eNB. Конечно, треба да се спомене дека другата разлика на сликата 3.3 од сликата 3.5 е воведувањето на дистрибуирана контролна рамнина со повеќе контролери (SDNC1, SDNC2). Во Глава 6 се моделира хендовер за случајот на единствен контролер, но и за случајот при постоење на мулти-контролери во контролната рамнина.



Слика 3.5. Хендовер во SDN-базирана јадрена мрежа во LTE

Кога еден пакет ќе пристигне на влез на комутаторот, најпрво се прави проверка за совпаѓање со постојните записи на тековите во табелите на тек. Ако проверката за совпаѓање е негативна, пакетот се праќа кон контролерот преку *Packet-in* порака [1]. Другиот тип OF пораки кои се праќаат кога MN ќе се дисконектира или конектира на порта на комутаторот се *Port-status* пораките. Од интерес е анализа на сценариото каде MN се дисконектира од еден комутатор (s_i) и се конектира на друг комутатор (s_j) во мрежа која има m комутатори кои припаѓаат на единствен контролер-домен ($i, j \in \{1, 2, \dots, m\}$). Ова тригерира две *Port-status* (*off-port* и *on-port*) пораки кои се праќаат кон контролерот за да го информираат за мобилноста на MN. Откако ќе ги прими овие пораки, контролерот праќа *Flow-mod* до дел од комутаторите во доменот за да ги променат своите записи на тековите за тој MN. Размената на пораки иницирана од мобилноста на MN е прикажана на сликата 3.6, каде “⁽¹⁾” ја означува *off-port*, а “⁽²⁾” *on-port* иницираната секвенца на пораки.



Слика 3.6 Размена на OF-пораки при хендовер

Поради мобилноста на MN, во одреден момент од времето, се праќа *off-port* порака од комутаторот s_i (Порака= $Ps^{(1)}$) на кој MN е веќе поврзан, укажувајќи на стартот на процедурата за хендовер. Кога оваа порака е примена од SDNC, специфични *Flow-Mod* (Порака= $Fm^{(1)}$) пораки се праќаат кон сите комутатори кои се вклучени во трансмисиониот пат со цел информирање за промената. После оваа прва секвенца пораки, MN мора да го информира дестинацискиот комутатор s_j , со праќање на *on-port* порака (Порака= $Ps^{(2)}$), која веднаш се препраќа кон контролерот. Потоа, SDNC праќа нови *Flow-Mod* (Порака= $Fm^{(2)}$) пораки кон сите инволвирани комутатори со цел информирање за последната мрежна промена. Размената на OF сигнални пораки поради хендовер е завршена кога *Flow-Mod* пораката е примена од комутаторот s_j и соодветните табели на тек се освежени.

4. Предлог нов алгоритам за брзо опоравување од испад во мобилни јадрени мрежи

Во една SDN мрежа, опоравувањето од испад на линк може да се постигне со механизам за реставрација (на англ. *restoration*) или заштита (на англ. *protection*). Механизмите за реставрација најчесто бараат од SDNC повторно да ги воспостави конекциите после детекцијата на испад. Класификацијата на механизмите за опоравување од испад детално се опишани во Глава 3.3. За мобилните мрежи, каде при испад има потреба од опоравување на голем број пакетски текови, овозможувањето брзи времиња на опоравување од испад претставува вистински предизвик. За мали и средни оператори, при секоја комуникација со SDNC, *Round-Trip Time* (RTT) е околу 20 ms до 30 ms, додека за големи мобилни оператори оваа вредност се качува над 50 ms [40]. Во случајот на механизам за заштита, при иницијализација, SDNC треба да ги провизионира алтернативните патеки во сите комутатори за секој пакетски тек. Користење претходно инсталирани патеки за опоравување значајно го намалува времето на опоравување од испад и помага за постигнување на *carrier-grade* време на опоравување на линк. Денес, најголем дел од предложените решенија во оваа група вклучуваат додатен механизам за сигнализација што може да придонесе кон зголемување на времето на опоравување.

Во 5Г мрежите, механизмите за опоравување од испад во SDN корисничката рамнина треба да им овозможат на операторите брзо опоравување од испад (пр. < 20 ms). Денес, OF претставува најшироко применетата апстракција на корисничката рамнина. За далечни (не-локални) испади, OF користи размена на контролни пораки со SDNC со цел да се добијат нови записи на тековите во соодветните OF табели, додека во случај на локален испад, OF може да го користи вградениот механизам на FF, но само доколку постојат дефинирани локални алтернативни патеки. Сепак, локални алтернативни патеки не можат секогаш однапред да се дефинираат. Понатаму, со текот на времето се очекува да се јавуваат и локални и далечни испади. Ова укажува на потребата за нов протокол на податочен линк (на англ. *data-link*) за брзо опоравување од локални и далечни испади.

Во оваа глава се предлага оптимизиран алгоритам за детекција и опоравување од испад кој се нарекува *Stateful Fast Failover* (SFF). Предложениот механизам е базиран на OF FF и на механизмот опишан во [36] кој се нарекува *Stateful Programmable failure DEtection and Recovery* (SPIDER). Колку што им е познато на авторите, овој протокол е првиот кој предлага повторно искористување на тривијалниот OF FF механизам, преку негова интелегентна комбинација со понапреден протокол на состојби, со цел да се постигне *carrier-grade* време на опоравување од испад прифатливо за 5Г.

4.1 Преглед на литературата

Во последните години, екстензии и/или модификации на стандардните OF верзии го привлекуваат вниманието на истражувачите. Во некои случаи, се предлагаат пореволуционерни пристапи, но конечната цел на сите предложени решенија претставува брзо опоравување од испад во SDN мрежите. Во најсвежите научни истражувања, се промовира идејата за симнување на дел од SDNC контролната логика во самите комутатори, така што тие нема да зависат од контролерот за воспоставување на алтернативни патеки во случај на испад. Во основа, ова е во спротивност со генералните принципи на SDN, каде целокупната контролна логика е сместена во SDNC. Сепак, истражувачите анализираат и докажуваат дека префрлање на минимален дел од оваа логика кон комутаторите овозможува големи придобивки за времето на опоравување од испад.

SDN е основна компонента на 5Г архитектурата и постигнување на ултра-ниска латентност и брзо опоравување од испад се од основен интерес на идните услуги. Денес постојат повеќе истражувања за 5Г, вклучувајќи SDN/NFV базирани јадрени мрежи [30], [41]. Дополнително, постојат прегледи (на англ. survey) за анализа на латентност и пристапи за намалување на латентноста во Интернет [42], *Cloud computing* [43], [44], а посебно бројни се истражувања на технологии за намалување на латентноста за 5G. Овие предлози често вклучуваат драстични промени во мрежната архитектура. Во [30] се дискутира за строгите барања за латентност во 5Г и за техниките кои се воведени од 3GPP и истражувачката заедница за адресирање и решавање на посакуваните ниско-латентни комуникации.

Во трудот [29] се воведува нова архитектура наречена *5G NOvel Radio Multiservice adaptive network Architecture (5G NORMA)*. Предложената cloud-базирана архитектура која содржи различни елементи (од *EDGE cloud* до *CENTRAL cloud*), може да вклучи голем број процесирачки задачи кои овозможуваат заедничка оптимизација на мобилната радио и јадрена мрежа, каде *Software Defined Mobile Networking and Orchestration (SDMN+O)* се клучните технологии на кои е заснован предлогот. Во [45] се развива модификација на постојната LTE архитектура преку интеграција на OF во 5Г архитектурата и се прави анализа на придобивките од оваа интеграција преку објаснување на миграцијата на GTP S11 пораките кон OF пораки. Исходот од овој предлог во смисла на намалување на латентноста крај-до-крај е последица на значајното намалување на протоколниот стек (на англ. stack) и користење на ниво 2 (на англ. layer 2) што е во целост комутациска архитектура.

Во централизиран и стандардизиран SDN-базиран механизам за опоравување од испад, контролерот е единствениот ентитет кој провизионира алтернативни патеки при иницијализација, имплементира механизам за повикување кон комутаторите и во услови на

испад ги овозможува алтернативните патеки. Ова најчесто води кон зголемување на оптоварувањето на контролерот, или во случај на грешки на линкот кон контролерот, комуникацијата кон комутаторите не може да биде навремено воспоставена. Во еден друг труд [46], јасно се покажува дека со зголемување на оптоварувањето на контролерот, директно се влијае на времето на опоравување и не може да се постигнат времиња на опоравување помали од 50 ms. [47] докажува дека е посебен предизвик овозможувањето на *carrier grade* време на опоравување во големите комерцијални мрежи доколку SDN стандардот се користи точно според спецификација. Се предлага шема за заштита базирана на BFD *daemon* имплементиран во комутаторот.

Во OF 1.1 и понатамошните OF верзии, се воведува групата FF што претставува исчекор во вистинската насока. Оваа промена се покажува како многу ефикасна, но може да се искористи само во услови на локална грешка кога претходно има провизионирана алтернативна патека во табелите на тек во комутаторот. За случаите во кои нема локална алтернативна патека, комутаторот мора да го контактира контролерот за да добие информација со алтернативни патеки. Во реалните мрежи е невозможно да се очекува дека постои резервна патека за секој пакетски тек и за секоја локална грешка која може да се појави. Научни предлози за намалување на оптоварувањето на контролерот се проучуваат во [48]. Се предлагаат оптимизациски модел и контролер-базиран механизам за надзор со цел да се намали вкупниот број на проверки што контролерот треба да ги прави на линковите. Теоретски пристап базиран на пребарување на графови (на англ. *graph search*) е предложен во [49]. Ова е комплетно нов пристап во кој е имплементиран алгоритам директно врз FF механизмот во OF за случајни обиди (на англ. *random try*) на користење на нови порти за да се стигне до дестинацијата на пакетскиот тек.

Пореволуционерен пристап е изложен во [4]. Се користи корисничка рамнина со локални состојби за опоравување од грешка исклучиво во корисничката рамнина, додека само контролерот има информација за глобалните мрежни состојби. OF е апстракција на корисничката рамнина без состојби (на англ. *stateless*) и мрежната состојба е единствено позната во SDNC, но со пристапот во [4] преку воведувањето на состојби по тек (на англ. *per-flow states*), контролерот се контактира само при иницијализација, додека во услови на грешка, комутаторите се способни за автономно препрепраќање на сообраќајот со користење на алтернативните патеки провизионирани при иницијализација. Овој предлог кој се нарекува OpenState, е проширен во [36] преку додавање на поинтелигентни импулси за синхронизација (на англ. *heartbeat*) и механизам за повикување без воведување на додатна сигнализација. Наместо тоа, предложениот механизам е моделиран само со користење на

веќе постојните пакети на податочната патека или понекогаш преку дуплицирање на оригиналните пораки во услови на повратен одговор (на англ. reply answers). Механизмот во [36] може да се користи само за не-локални (далечни) единечни грешки. Во [50] се оди чекор понатаму, се докажува дека во услови на повеќекратни грешки, *OpenState* покажува подобри резултати од OF.

Во поновите трудови, се предлагаат напредни методи за тоа како да се намали времето на опоравување од испад. Во трудот [51] се користи BFD за тестирање на моментната комуникациска способност помеѓу комутаторите. Кога комутаторот ќе прими пакет и примарниот пат не е достапен, комутаторот го враќа пакетот назад на рекурзивен начин се додека пакетот не најде дефинирана алтернативна патека преку која може да се пренасочи сообраќајот. Сепак, овој пристап бара интеракција со контролерот бидејќи комутаторот кој детектира испад откако пакетот успешно е трансмитиран кон дестинацијата, мора да го информира контролерот да ја конфигурира мрежата. Во [52] се истражува како да се минимизира бројот на записи на тековите во услови на еднократен испад на линк. Се предлага алгоритам кој го одредува значењето на линкот и одбира јазли на резервна патека со цел минимизирање на просечниот број записи на тек на мрежно ниво. Во [53] *OpenState* се усовршува со понатамошно минимизирање на пакетските загуби и оптимизација на користењето на *Ternary Content Addressable Memory* (TCAM). [37] предлага нов дизајн на табелите на препраќање со групирања по тек и методи на агрегација на тек. Алгоритамот ги компресира правилата на тек (на англ. flow rules) за алтернативните патеки во случај на повеќекратен испад на линк. Конечно, [54] креираат концепт на заедничка резерва (на англ. shared backup) преку воведување на мал број резервни комутатори кои се делат од целата мрежа и се користат само во услови на грешка. Овој пристап овозможува брзо опоравување без тоа да го забележат апликациите.

4.2 Опис на проблемот и научен придонес

За да се постигне *carrier-grade* квалитет, 5Г мрежата мора да може да се опоравува од испад во рамки на милисекунди. Опоравување на мрежата, како термин, всушност ги опишува активностите кои и овозможуваат на мрежата да се врати во нејзината оперативна состојба после мрежен испад. За да се постигнат *carrier-grade* перформанси, есенцијално е претходно планирање или резервација на алтернативни патеки во услови на испад, за да кога ќе се појават испади, не е потребна додатна сигнализација со цел воспоставување на патеката на заштита и целокупниот сообраќај е веднаш повторно воспоставен. Понатаму, механизмите за детекција и опоравување од испад треба да бидат дизајнирани на тој начин

што не воведуваат дополнителен сообраќај заради сигнализација (на англ. overhead) и не бараат многу записи во меморијата TCAM која е многу скапа.

Во оваа глава се предлага нов алгоритам за опоравување од испад во корисничката рамнина кој е соодветен за употреба во 5Г мобилни јадрени мрежи. Главниот придонес и новитет на предложениот протокол е следниот:

- 1) Квантификација на оптоварувањето на SDNC при иницијализација на комутаторите и во случај на мрежен испад. Се покажува дека алгоритамот е погоден за опоравување од грешка во 5Г јадрени мрежи и воопшто не зависи од SDNC,
- 2) Интелигентна интеграција на OF FF и SPIDER решенијата за да се овозможи брзо опоравување од испад во случај на локални и далечни испади и да се минимизира губењето на пакети при мрежен испад. Ова се прави со модификација на питон (на англ. python) кодот на SPIDER, метадатата на OF (која се користи за да се пренесе информацијата помеѓу табелите во комутаторот) и на други *pipeline* полиња. Дополнително, се оптимизира бројот на кодни линии на SPIDER (преку заобиколување (на англ. bypass) на деловите кои оригинално се однесуваат на справување со локални грешки). Конечно, се оптимизира имплементацијата со додавање на можност за синхронизација и континуитет во моментите на префрлање помеѓу SPIDER и OF FF механизмите во текот на нормалното извршување (на англ. at runtime) на операциите.
- 3) Имплементација на протоколот BFD за мониторирање живост на порта кога се користи OF FF во случај на локален испад.

Во наредното поглавје се опишува предлогот за механизам за опоравување од испад имплементиран директно во корисничката рамнина. Решението не бара додатна сигнализација и нема високи побарувања во однос на меморија во комутаторите. Дополнително, предложениот механизам вклучува проактивна реставрација на линк со централизиран контролер каде се имплементира опоравување од испад при единечни грешки.

4.2 Предлог за нов механизам – Stateful Fast Failover (SSF)

Предложениот механизам е базиран на OF 1.3.1 како што е специфицирано во [1]. Секој OF комутатор има една или повеќе табели на тек кои се состојат од записи на тек: 1) *полиња на проверка за совпаѓање* (на англ. match fields) кои го дефинираат секој тек, 2) *бројачи* кои снимаат статистика и 3) *инструкции* кои дефинираат како се управува со пакетот (насочен кон порта, отфрлен, пратен кон наредната табела). Доколку проверката за

совпаѓање на влезниот пакет е позитивна, се извршуваат дефинираните инструкции, во спротивно се тригерира табелата на промашување (на англ. table-miss) и во најголем број од случаите се праќа OFP_PACKET_IN порака кон контролерот. OF механизмот ја користи групната табела наречена *fast-failover* која имплементира *механизам на живост* за одредување на специфичниот *bucket* кој треба да се изврши.

Идејата за ослободување на дел од контролните функции од контролерот кон комутаторите, при што контролерот е постојано информиран и има целосна мрежна контрола, за прв пат е иницирана во трудот [4]. Како што претходно е опишано во поглавјето 2.1.2, со овој предлог се прави екстензија на OF табелите на тек преку воведување на состојби на тек. Со овој пристап, однесувањето при препраќање на пакети не зависи само од информацијата во заглавието на пакетот, но исто така зависи и од состојбата на текот. Иницијално предложените кориснички случаи во [4], подоцна се проширени во [55] преку воведување на нов модел на однесување подобен за отпорност (на англ. resilience) од мрежни испади, каде шемата за опоравување од испад овозможува справување со не-локални (далечни) грешки без оглед на достапноста на контролерот.

Предложениот механизам ги користи предностите на OF-базираниот *pipeline* дизајн наречен SPIDER [36] кој е базиран на: 1) механизам за детекција базиран на периодично пробување (на англ. probing) на линкот; 2) брзо препрепраќање на сообраќајот во случај на единствен далечен испад. SPIDER не воведува додатни сигнализациски пораки, наместо тоа различните потреби за препраќање се извршуваат во корисничката рамнина преку лабелирање (на англ. labeling) на пакетите со ознаки (на англ. tags) кои се вметнуваат во постојните полиња на пакетско заглавие (MPLS лабела).

Комуникацијата со контролерот се прави само при бутирање (на англ. booting) кога се додаваат сите резервни патеки за секој пакетски тек. Во нормални услови, пакетите се лабелираат со $tag=0$ и се рутираат на нивната примарна патека. Детекција на испад се врши преку воведување на многу едноставна шема на детекција базирана на размена на двонасочни *пакети на импулси за синхронизација*. Ако пакетот не се прими во рамки на специфицираниот интервал, се побарува *пакет на импулс за синхронизација* преку лабелирање на пакетот со $tag=HB_req$ (*HB_reply* се враќа назад). Во случај на испад на порта/линк, пакетот се лабелира со $tag=Fi$, состојба во која комутаторот i не е достапен. Конечно, се воведува механизам за *пробување*, така што примарната патека може повторно да се воспостави кога ќе се отклони грешката ($tag=Pi$).

Дизајнот SPIDER ги проширува оригиналните OF табели на тек и воведува нови табели (означени како Табела 0 до Табела 3). Табела 0 и Табела 1 се користат за

процесирање на пакети без состојби и имплементираат потиснување (на англ. push) на MPLS лабелата и вметнување на специјална ознака како што е опишано погоре. Табела 1, дополнително ја освежува Табела 3 при прогласување на порта како исправна (*UP*) и Табела 2 само за пакети на пробување при промена на состојбата од *Fi* во *Normal*. Табела 2 се користи во случај на далечен испад и воведува неколку состојби: *Normal* (номинална состојба), *Сигнализирана грешка* (на англ. Fault signaled), *Овозможено препрепраќање* (на англ. Detour enabled), *Потреба од проба* (на англ. Need probe) и *Отстранета грешка* (на англ. Fault resolved). Табела 3 се користи само во случај на локален испад и ги воведува следните состојби: *UP: побаран импулс за синхронизација* (на англ. heartbeat requested), *DOWN: треба пробување* (на англ. need probe) и *DOWN: пратено пробување* (на англ. probe sent). Конечно, се дефинираат *idle* и/или *тврди* (на англ. hard) тајмаути за барањата и одговорите на импулсите за синхронизација, пакети на пробување и свесност на тек (на англ. flowlet-awareness) со кој се адресира проблемот на пакетско прередување (на англ. reordering).

Алгоритам 1: Stateful Fast Failover (SFF)

```

1: # при иницијализација, SDNC ги дефинира
2: # алтернативните патеки
3: for i in range (1, N):                                # за N комутатори
4:   SPIDER.install_flow_entries                          # SPIDER Табели 0-3
5:   OF.group_table.install_flow_entries                  # OF FF овозможување
6:   OF.group_table.start_liveness_monitoring
7:   select_active = SPIDER                               # SPIDER е номинален алг.
8:
9 # при извршување (на англ. at runtime), комутатор Si
10: # користи OF FF само во случај на локална грешка
11: for p in range (1, n):                                # за n порти во комутатор Si
12:   if port_state != OFPPS_LIVE:                       # провери живост на порта
13:     if watch_port != OFPP_ANY:
14:       if watch_group != OFP_ANY:
15:         select_active = OF
16:
17: # синхронизација и континуитет при извршување
18: for i in range (1, N):                                # за N комутатори
19:   if faultless_state = YES:                          # кога е отстранета грешка

```

Во предложениот алгоритам се користи SPIDER методот за опоравување само во случај на далечен испад, а се задржува оригиналниот механизам на OF FF за справување со локални грешки. На овој начин, се користат предностите на двата механизми (SPIDER и OF FF), како што е покажано во наредното поглавје. Според Алгоритам 1, при иницијализација контролерот ги пополнува SPIDER табелите со примарни и резервни патеки, но исто така го активира и опциониот FF механизам во OF групната табела и вметнува *action buckets* со соодветна селекција на порта во услови кога портата на примарната патека има испад. При извршување, опишаниот процес на означување и импулси за синхронизација во SPIDER ја презема контролата за препраќање на пакетскиот тек. Во паралела, механизмот за живост ја проверува состојбата на портата и во случај на правилно функционирање на портата на примарната патека за дадениот тек, не се преземаат додатни акции. Ова е работната ситуација во случај на далечни грешки кога SPIDER овозможува опоравување од испад. Во случај на локална грешка, SPIDER механизмот е заобиколен (на англ. *bypassed*) и OF FF ја превзема контролата за опоравување од испад. За да се овозможи OF FF, покрај проверка на порта, се прават додатни проверки на живост на самиот *bucket*. После иницијализација, од витално значење е способноста за синхронизација и континуитет при префрлањето помеѓу SPIDER и OF FF механизмите. Поради оваа причина, требаше да се креира скрипта која постојано ги мониторира двата механизми на ниво на комутатор, чита настани од OF метадатата и во услови на грешка кога е дефинирана локална резервна патека во групната табела FF, инстантно реагира и го декларира за активен OF FF механизмот, додека SPIDER се поставува како пасивен. Откако грешката е отстранета (ситуација без грешка), скриптата го поставува знаменцето (на англ. *flag*) на номиналната состојба при која активен е SPIDER. Дополнително, овој механизам е засилен со способноста да собира оперативни статистики по сесија, што овозможува прибирање на битни информации за деталите при извршување. Овие статистики беа дополнително искористени за понатамошна оптимизација на кодот и за детекција и корекција на некои грешки пријавени од OF. Важно е да се спомене дека беше успеано значително да се намали бројот на кодни редови на SPIDER кои оригинално се справуваа со локални грешки. Со оваа оптимизација на код, се успеа дополнително да се намали времето на опоравување (за околу 0.5 ms). Конечно, во напорите да се подобри времето на опоравување од испад, исто така се анализира и сценариото каде наместо LOS за мониторирање на порта, се имплементира BFD. Подоцна, во симулациите се мери влијанието на оваа промена врз времето на опоравување.

4.3 Анализа на перформанси

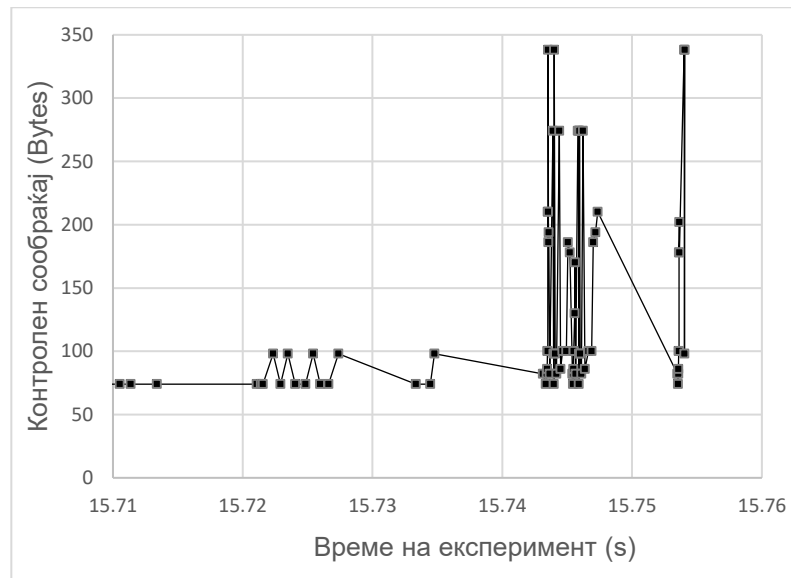
За евалуацијата на перформанси, се користи *python scripting* (Python 2.7) и *Mininet* 2.2.1 [56]. Се користи модифицирана верзија на *OF Ryu Controller* [57] и изворниот код на SPIDER кој е достапен на [58]. На сликата 3.3 е прикажано симулациското сценарио. За SDN комутаторите S1 – S5 се користи *CPqD OF 1.3. softswitch* [59], проширен со поддршка за *OpenState*.

За да се симулира 5Г околина, се претпоставува дека капацитетот на линкот UE – eNB е 100 Mbps, додека сите други линкови имаат опсег од 2,5 Gbps. Доцнењето на линк во *backhaul* мрежата е поставено на 0,25 ms. *Idle* и тврдите тајмаути на импулсите за синхронизација во SPIDER се со вредност 0,1 ms, додека тајмаутите на тек се поставени на најниска можна вредност и не се од интерес на анализата. Во сите сценарија, h1 (на англ. host 1) се имплементира во eNB и овој корисник го пинга регионалниот рутер (на англ. regional router (RR)) каде што е лоциран h2. За *ping* командата, се користи фиксна пакетска брзина од 1000 pkt/s и номинален *Internet Control Message Protocol* (ICMP) од 56 Bytes.

При иницијализација на SPIDER, контролерот ја инсталира примарната патека на дефинираниот тек како eNB (h1) » S1 » S2 » S3 » S5 » RR (h2). Резервната патека е пре-конфигурирана како eNB » S1 » S4 » S5 » RR. За OF FF, контролерот ја пополнува групната табела FF и соодветниот *action bucket* во S1. За симулација на далечна грешка, се вметнува грешка помеѓу S2 и S3, а за симулација на локална грешка, грешката се вметнува на линкот помеѓу S1 и S2.

На почетокот, се анализира и квантифицира оптоварувањето на SDNC при случајот на симулација на далечен испад и се прави обид да се даде одговор на прашањето зошто механизмите за опоравување од грешка во *carrier grade* мрежите се имплементираат само во корисничката рамнина. При иницијализација, задачата на контролерот е да ги провизионира сите табели на тек за сите комутатори за сите побарувања (на англ. demand) во мрежата. Сликата 4.1 го прикажува интензитетот на OF сообраќајот при иницијалната конфигурација на комутаторите во случајот на само едно побарување (h1 кон h2). X-оската е времето на симулација во милисекунди, а Y-оската го прикажува контролниот сообраќај во бајти. 15,71 ms после почетокот на симулацијата почнуваат да се разменуваа OF контролни пораки помеѓу контролерот и комутаторите. Со *Wireshark*, се снимаат сите OF 1.3 контролни пораки. Најголем дел од пораките се од типот OFPT_FEATURES_REQUEST (REPLY) и OFPT_FLOW_MOD. Контролерот праќа OFPT_FEATURES_REQUEST порака кон комутаторот откако ќе се воспостави сесијата. Комутаторот одговара на оваа порака со OFPT_FEATURES_REPLY порака. OFPT_FEATURES_REPLY пораката има неколку

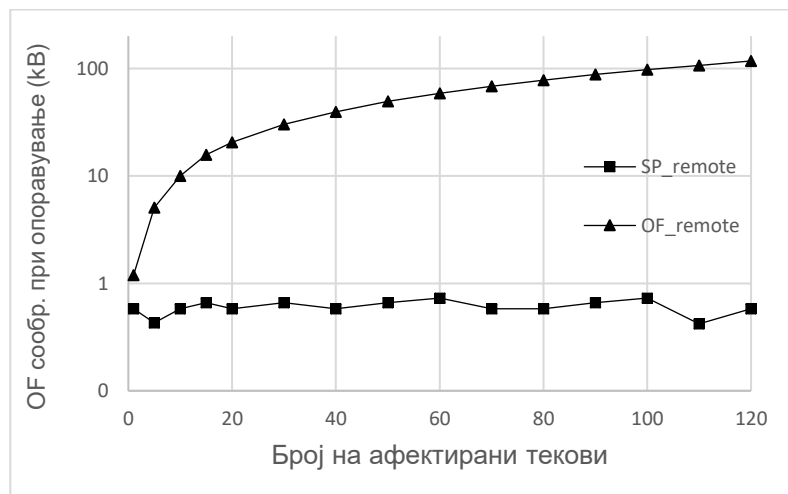
полиња, вклучувајќи поле на можности (на англ. capabilities field), кое ги идентификува можностите поддржани од податочната патека на комутаторот. OFPT_FLOW_MOD се пораки од SDNC кон комутаторите за вметнување на записи во OF табелите. Заклучокот од добиените резултати е дека дури и за мала мрежа и за само едно побарување дефинирано со симулацијата, сообраќајот кој се разменува помеѓу комутаторите и SDNC е значаен. Ова може да придонесе кон сериозно оптоварување на SDNC во сценарија со повеќе пакетски текови и комплексни мрежни топологии.



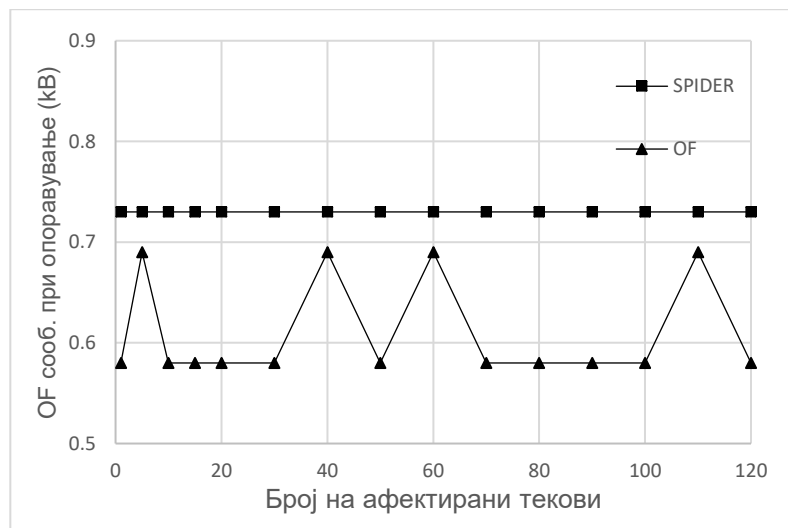
Слика 4.1. Интензитет на OF контролен сообраќај при иницијалната конфигурација на комутаторот.

На сликата 4.2 и сликата 4.3 прикажан е OF сообраќајот при опоравување за стандарден OF и SPIDER за случајот на далечни и локални грешки, соодветно. Во случај на далечна грешка (слика 4.2), генерираниот контролен сообраќај во SDNC се мери од моментот на испад, до моментот кога сите текови се пренасочени. X-оската го претставува бројот на текови кои се афектирани од единствен далечен испад, а Y-оската е големината на OF сообраќајот генериран за препрепраќање после испадот на линкот. Резултатите покажуваат дека OF сообраќајот драматично се зголемува со зголемување на бројот на тековите кои треба да се пренасочат. Спротивно, сообраќајот за SPIDER е скоро ист без разлика на бројот на текови. Поради тоа што во случајот на SPIDER нема потреба од контактирање на SDNC заради провизионирање на табели на тек во ситуација на грешка, генерираниот сообраќај се состои само од нотификациски пораки за падната порта кои се праќаат кон контролерот (OFPT_PORT_STATUS). За случајот на OF, најголемиот број на пораки се всушност пораки за модификација на тек (OFPT_FLOW_MOD) кои се користат за модификација на табелите на тек во комутаторите после детекција на далечната грешка.

За локална грешка (Слика 4.3), нема потреба од комуникација со контролерот заради модификација на табелите на тек, па за OF и SPIDER пораките се само од типот OFPT_PORT_STATUS. Резултатите добиени на сликата 4.2 и сликата 4.3 водат кон заклучок дека опоравувањето од грешка мора да се извршува исклучиво во корисничката рамнина затоа што размената на пораки со SDNC во овие ситуации лесно може да го загуши линкот кон контролерот. Ако се сумира, ова значи дека стандардниот OF 1.3 алгоритам за детекција и опоравување од испади, не може да се употребува во мобилни мрежи за опоравување од далечни испади.



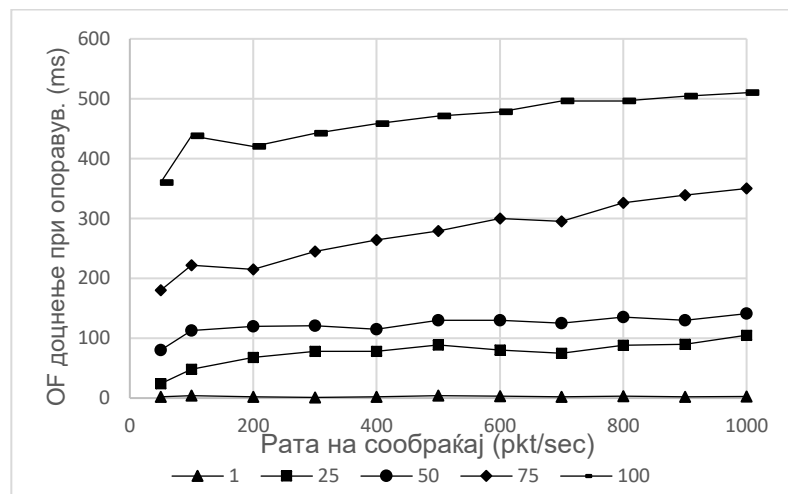
Слика 4.2. OF сообраќај на опоравување во случај на далечен испад



Слика 4.3. OF сообраќај во случај на локален испад

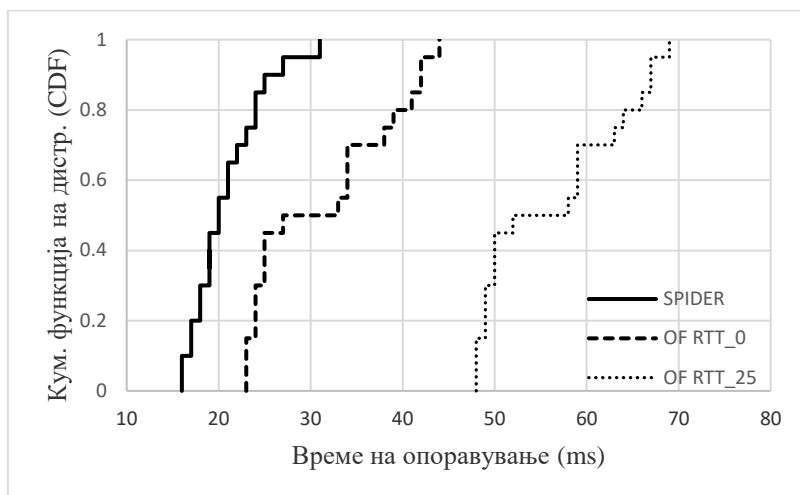
Понатаму, се мери времето на OF опоравување во случај на далечен испад (мора да се контактира SDNC заради провизионирање на табелите на тек) и на сликата 4.4 се презентирани резултатите. Се прикажува зависноста на доцнењето при OF опоравување (ms) од сообраќајниот проток (на англ. traffic rate) (pkt/sec). Секоја точка на графикот е времето измерено помеѓу OFPT_PORT_STATUS пораката и конечната OFPT_FLOW_MOD

порака. Се заклучува дека за OF, доцнењето при опоравување се зголемува како што расте сообраќајниот проток. Дополнително, во анализата е вклучен и број на пакетски текови (1, 25, 50, 75, 100) за да се види како ова влијае на времето на опоравување. Резултатите покажуваат дека постои стрмен пораст на доцнењето при опоравувањето кое е пропорционално со бројот на текови. Поради тоа што во мобилните мрежи се очекува бројот на пакетски текови да биде многу голем, повторно може да се заклучи дека во услови на грешка, не смее да се контактира SDNC и целокупната интелигенција за опоравување од грешка мора да биде целосно имплементирана во корисничката рамнина.

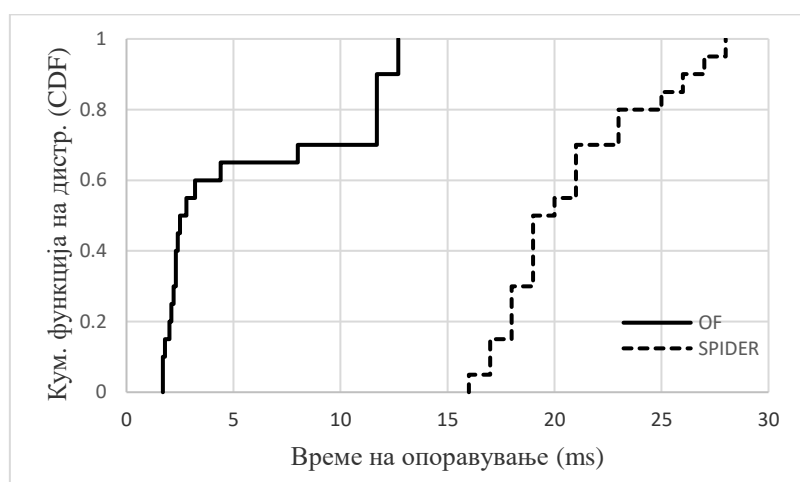


Слика 4.4. OF доцнење при опоравување од далечна грешка

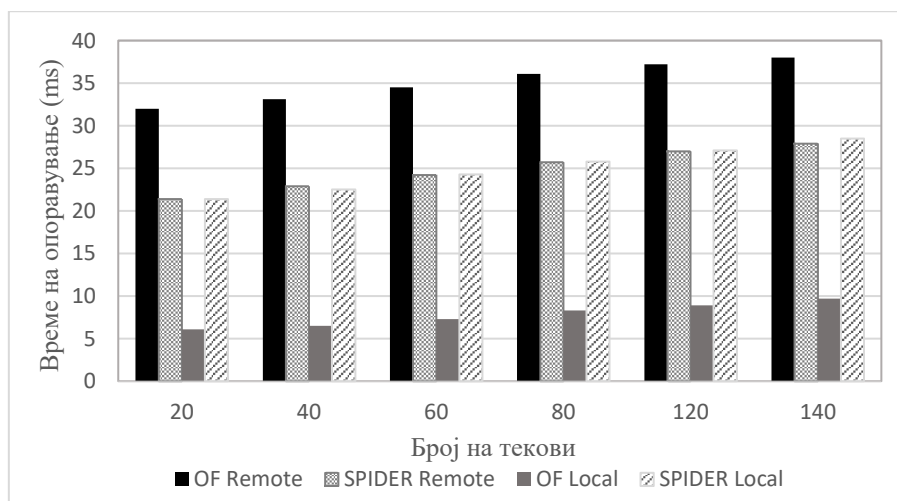
Анализата продолжува со мерење на времето на опоравување во случај на далечна и локална грешка засебно, за OF и SPIDER поединечно. Времето на опоравување од грешка се пресметува како временски интервал помеѓу моментот на внесување на грешка и времето кога комутаторот за препрепаќање (на англ. detour node) (S1) го пренасочи сообраќајот на резервниот пат. На сликата 4.5 е прикажана функцијата на кумулативна дистрибуција (на англ. Cumulative Distribution Function (CDF)) на времето на опоравување. Во случај на далечна грешка, SPIDER овозможува побрзо опоравување со просечно време на опоравување од 20.85 ms. Дури и за случајот кога RTT = 0 ms (кон SDNC), постои значајно процесирачко доцнење во контролерот кое предизвикува просечно време на опоравување за OF FF од 31.2 ms. Испрекинатата линија го прикажува реалниот случај за мал или среден мобилен оператор, каде RTT при контактирање со контролерот е 25 ms. Сликата 4.6 го прикажува случајот на локален испад, каде традиционалниот OF користи FF и контролерот не се контактира. Резултатите демонстрираат дека OF FF овозможува подобро време за опоравување со просечна вредност од 5.81 ms, додека просечната вредност за SPIDER е 20.75 ms.



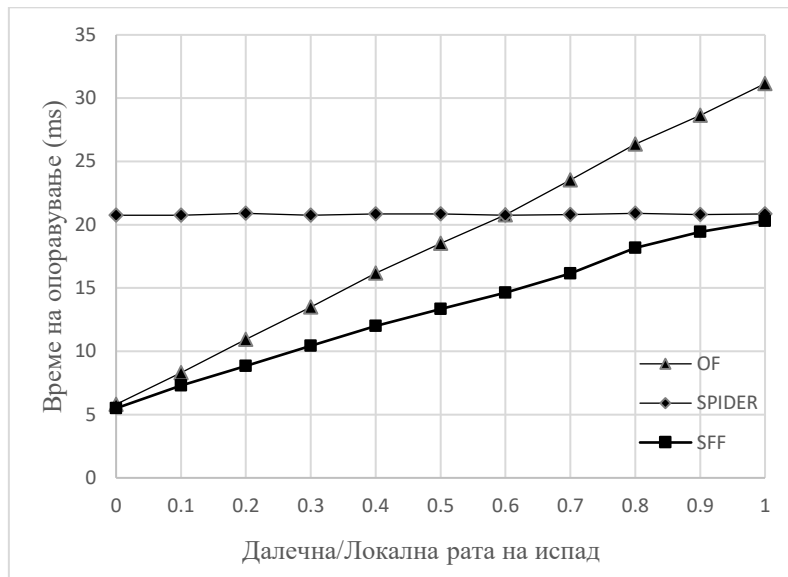
Слика 4.5. CDF на времето на опоравување (далечна грешка)



Слика 4.6. CDF на времето на опоравување (локална грешка)



Слика 4.7. Време на опоравување наспроти број на текови



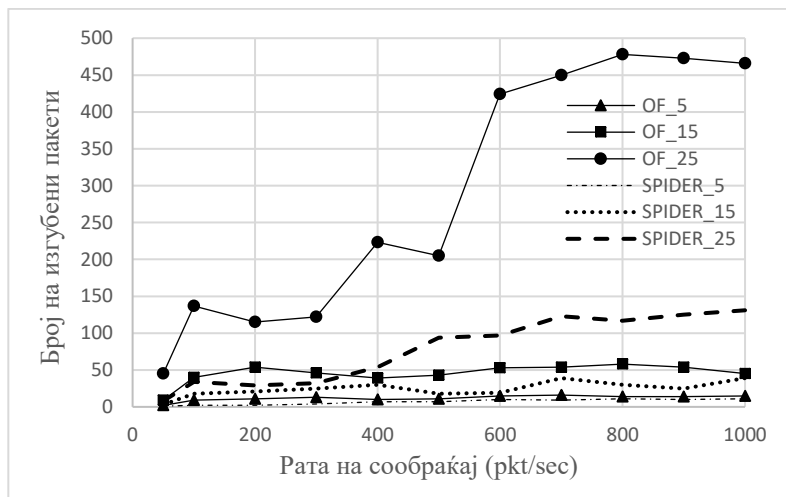
Слика 4.8. SSF – Време на опоравување наспроти рата на далечна/локална грешка

Сликата 4.7 прикажува резултати добиени кога истиот h1 иницира неколку пакетски текови во паралела кон корисниците (на англ. hosts) поврзани на S5. Целта е да се истражи потенцијалната зависност од бројот на текови. Резултатите покажуваат дека времето на опоравување благо се зголемува како што расте бројот на текови, затоа што е потребно дополнително време комутаторот да ги пренасочи сите текови на резервната патека. Последното е причината поради која се продолжува со анализа на сценарио со единствен пакетски тек и се воведуваат случајни грешки (далечни или локални) при што се мери времето на опоравување за OF FF, SPIDER и SSF.

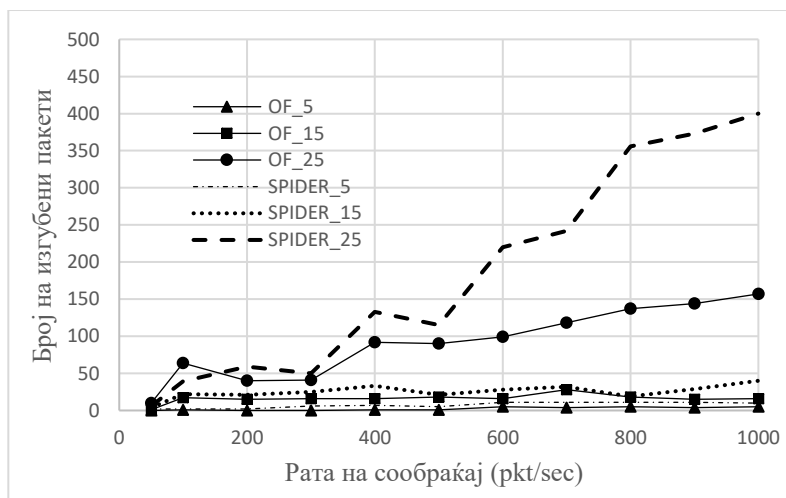
На сликата 4.8 се прикажани различни далечни-локални (L-R) рати на грешка (пр. рата од 0,2 значи дека 80% од грешките кои се генерирани се локални грешки, а 20% се далечни грешки). Во предложениот SSF алгоритам, интелигентно се комбинираат најдобрите особини на OF FF и SPIDER според претходно прикажаните откритија. Резултатот од ова комбинирање е тоа што SSF ги надминува по перформанси и OF FF и SPIDER и овозможува најбрзо време на опоравување од испад. Понатаму, резултатите докажуваат дека работењето во паралела на OF FF и SPIDER, не влијае на нивните засебни перформанси поради тоа што се добиваат слични резултати со оние кои се добиени преку изолирано тестирање на протоколите.

На сликата 4.9 и сликата 4.10 се прикажува бројот на изгубени пакети после пад на линк за двата случаи на далечна и локална грешка. X-оската е брзината на сообраќај (pkt/sec) и како што се очекува со зголемување на брзината се зголемува и бројот на изгубени пакети. Исто така, се покажува дека бројот на изгубени пакети зависи скоро пропорционално од бројот на текови (5, 15, 25). Во случај на далечен испад, SPIDER покажува подобри резултати, додека за локални испади, OF има помалку изгубени пакети. Интересно е да се

нагласи дека во случај на локална грешка, за кривата која го означува случајот при 25 пакетски тека за OF, се очекуваше помал број на отфрлени пакети. Се заклучува дека со комбинирање на OF FF за локални и SPIDER за далечни грешки, исто така се минимизира и бројот на изгубени пакети од моментот на внесување на грешка до моментот кога целокупниот сообраќај е нормално пренасочен на алтернативната патека.



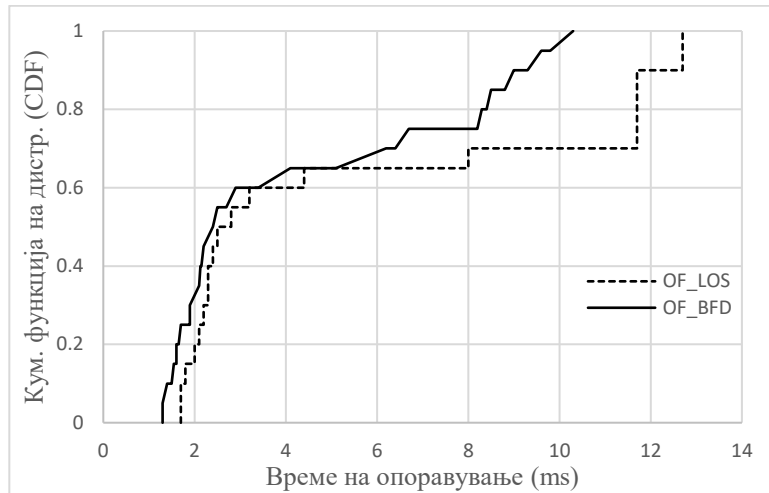
Слика 4.9. Број на изгубени пакети наспроти брзина на сообраќај за случајот на далечна грешка



Слика 4.10. Број на изгубени пакети наспроти брзина на сообраќај за случајот на локална грешка

Како последен случај, се анализира сценариото кога наместо LOS се користи BFD како механизам на FF групната табела за проверка на живост. Целта е да се утврди можно подобрување во времето на опоравување за случајот на единствена грешка. Симулациите се извршуваат и повторуваат доволен број пати за опишаното сценарио. Резултатите покажуваат дека за локално опоравување од грешка, има просечно подобрување на времето на опоравување од 1.39 ms кога се користи BFD. Просечното време на опоравување за OF FF кога се користи LOS е 5,67 ms, додека кога е имплементиран BFD, тоа изнесува 4,28 ms.

Функцијата на кумулативна дистрибуција за двата анализирани случаи е претставена на сликата 4.11.



Слика 4.11. CDF на времето на опоравување од испад (локална грешка) за OF_LOS и OF_BFD

На крајот, во табелата 4.1 се споредуваат резултатите добиени со симулација на предложениот SFF, со слични пристапи кои го прилагодуваат OF протоколот и ја користат OF FF групната табела со цел брзо опоравување од испад. Анализата покажува дека алгоритмот SFF овозможува побрзо опоравување во споредба со останатите предложени OF-базирани пристапи.

Табела 4.1: Време на опоравување на слични OF-базирани предлози

Референца	Просечно време на опоравување (ms)	Механизам за опоравување
[60]	42 - 48	FF група
[51]	40	FF група
[52]	28.2	Прилагодување на OF
[53]	32.74	Прилагодување на OF
[37]	20.18 - 21.83	FF група
SFF	4.28 - 20.85	Прилагодување на OF и FF група

4.4 Заклучок

Овозможување на ниска латентност е една од најважните насоки во следната генерација на 5G мрежи и примарен фокус за истражувачката заедница претставува имплементацијата на технологии кои го поддржуваат намалувањето на латентноста. За да се контролира латентноста крај-до-крај, операторите мора да имплементираат различни техники во делот на радио мрежа, *backhaul* и јадрената мрежа и конечно помеѓу излезниот рутер EPG и интернет. Брзо опоравување од испад во корисничката рамнина е главен

приоритет за мобилните оператори, а SDN е вистинската технологија која поддржува разновидни имплементации за детекција и препрепраќање во услови на грешка.

Во оваа Глава, се предложи оптимизиран *Stateful Fast Failover* протокол на податочен линк за SDN-базирани 5G EPC мобилни мрежи со парцијална виртуелизација. Предложеното решение интелигентно ги комбинира најдобрите страни на SPIDER и OF FF и е дизајнирано да го оптимизира времето на опоравување во случај на единствена далечна или локална грешка. Резултатите од симулациите ги покажуваат перформансите на протоколот и потврдуваат дека времето на опоравување од грешка може да се задржи под 20 ms, што е во согласност со *carrier-grade* мрежните барања. Дополнително, преку симулации се потврдува дека решенијата кои не вклучуваат контакт со SDNC мора да бидат имплементирани директно во корисничката рамнина бидејќи волуменот на OF контролниот сообраќај и времето на опоравување од испад не се во согласност со 5G очекувањата. Идните екстензии на оваа Глава може да бидат во насока на проширување на предложениот протокол со цел користење на истиот за опоравување од повеќекратни грешки. Понатаму, може да се анализира вкупното доцнење воведено од мобилната јадрена мрежа и преку симулации во Mininet да се направи обид да се пресмета максималниот број на јазли и максималните растојанија кои се дозволени за да се задоволат 5G барањата кои беа презентирани.

5. Аналитичко моделирање на латентноста во SDN-базирани мобилни јадрени мрежи

Проблемот на латентност во LTE и значењето на латентноста во 5G беа анализирани во Глава 3.1. Детално се анализираа сите конституенти на латентност во LTE, што беше прикажано на сликата 3.2. Во ова поглавје се става акцент на мрежната латентност која е воведена помеѓу комутаторите и контролерот, к оја има и најголемо и најкритично влијание на целокупните SDN перформанси. Моделирањето на овој дел од латентноста е во фокусот на истражувачката заедница [6], [61]. Доколку контролерот е преоптеретен или линковите се загушени (или имаат ограничен опсег), временски-осетливите апликации жестоко се деградираат и сите останати услуги може значајно да се успорат. Во една OF-базирана мрежа, може да има еден или повеќе контролери.

Покрај предностите, постојат и некои интринсични недостатоци кога се централизира мрежната интелигенција во единствен ентитет, како што се перформансни и проблеми со скалабилност, единствена точка за испад и неможност за справување со голем број пакетски текови [62]. Во мобилните мрежи постојат голем број корисници и единствен контролер не е во можност да ги опслужи во разумно време сите барања кои доаѓаат од комутаторите. Ова е причината за имплементирање на повеќе контролери во контролната рамнина, каде комутаторите припаѓаат на домени и секој комутатор е управуван само од еден контролер кој е одговорен за соодветниот домен. Оваа дистрибуирана мрежа креирана од контролерите нема хиерархиска структура, што значи дека не постои единствен глобален мрежен преглед и контролерите мораат да разменуваат мрежна информација помеѓу себе [63]. Во дистрибуирана SDN архитектура, еден контролер разменува пораки со апликациите (*Northbound*), со другите контролери (*West and East bound*) и со корисничката рамнина (*Southbound*). Последното беше детално дискутирано во главата 2.1. Иако повеќе контролери може значително да придонесат кон подобрувањето на мрежните перформанси, сепак пристапот го зголемува сообраќајот заради дополнителна сигнализација, бара додатни ресурси за пресметка и воведува повисока комплексност при дизајнирање и одржување на мрежата. Може да се заклучи дека со додавање на повеќе контролери во мрежата, сообраќајот кој се управува од секој контролер е помал, но расте размената на сигнализационски пораки помеѓу контролерите. Во оваа Глава 5 (и во Глава 6) од интерес е архитектурата и со единствен и со повеќе контролери.

Математичкото моделирање масовно се користи од истражувачката заедница поради особините на брза и евтина анализа и валидација на модели пред преминување кон реални хардверски имплементации. Во оваа глава, се става акцент на идентификација и аналитичко

моделирање на севкупните фактори кои влијаат на латентноста во SDN мрежи со дистрибуирана контролна рамнина. За предложените аналитички модели, се користи теоријата на редови. При математичко моделирање на SDN-базирани мрежи, скоро секогаш се земаат во предвид неколку претпоставки: рата на опслужување независна од оптоварувањето и неограничен бафер. Сепак, реалните SDN хардверски имплементации имаат средна рата на опслужување која се влошува со пораст на оптоварувањето на контролерот, а комутаторите и контролерите имаат конечен бафер. Целта во оваа глава е нумеричка анализа и квантификација на грешката која се прави при користењето на двете хипотези.

5.1 Преглед на литературата

Еден од првите трудови за моделирање на OF-базирани интеракција помеѓу комутатори и контролер е претставен во [64]. Се дизајнира OF тестен систем и се одбираат перформансни параметри со цел да се конструира аналитички модел. Искористен е модел на редови со повратна врска кој има систем од ред за препраќање (комутатор) и систем од ред со повратна врска (контролер). Предложениот модел го користи M/M редот (неограничена должина на ред) за комутаторот и M/M – S (ограничена должина на ред) за контролерот. Најголемото ограничување на овој пристап е моделирањето на единствен комутатор кој е во интеракција со единствен контролер.

Во трудот [65], се користи Џексонова (на англ. Jackson) мрежа за моделирање на корисничката рамнина, но се имплементираат модификации заради приближување кон природата на тековите на сообраќајот во SDN мрежите. Повторно се зема модел со единствен јазол и се имплементира повратна интеракција помеѓу комутаторите и контролерот. Се користи Поасонова дистрибуција за ратата на пристигнување и експоненцијална рата на опслужување и во комутаторите и во контролерот. Во евалуацијата на перформанси, се мери просечното време на процесирање на пакети и дистрибуцијата на времето кое секој пакет го поминува. Сепак, се анализира само еден контролер/комутатор и се зема во предвид само M/M/1 моделот. Во трудот [66], се предлага аналитички модел базиран на теоријата на мрежен калкулус (на англ. network calculus) како надградба на користењето на теоријата на редови за SDN моделирање. Целта е евалуација на должината на редот во комутаторот и контролерот и пресметување на горната граница на доцнењето при пакетско процесирање во комутаторот/контролерот. Недостаток на оваа научна работа е фактот што математичкиот модел не се валидира со симулациско сценарио. Од друга страна, трудот [67] истражува различни аспекти и предизвици преку користење на теоријата

на редови за анализа на перформанси во SDN. Акцентот се става на латентностите кои се појавуваат кога се префрлаат податоци помеѓу различните кеш мемории (на англ. caches), хардверски и софтверски тригерирани прекини, дизајн на интерфејсите како редови итн. Битен предмет на анализа е конечниот капацитет на системот и лимитацијата на големината на бафер, сепак, како што е објаснето, генерално најголемиот дел од истражувачката работа во оваа област комплетно ги има занемарено последните две хипотези кои редовно се земаат да бидат бесконечни.

Трудот [68] е од огромно значење затоа што за прв пат се земаат во предвид повеќе комутатори во корисничката рамнина. Во [69] е направена екстензија со моделирање на *Packet-in* пораки. Се предлага аналитичен пристап со цел квантификација на средното време на опслужување на контролерот и отпорноста (на англ. solidity) на пакетите кои се вбригуваат во мрежата. Моделот зема во предвид конечни и бесконечни бафери, но не се анализира пропагациското доцнење. Слично, во трудот [70] се моделираат *Packet-in* пораките и како тие влијаат на перформансите на средното време на процесирање на контролерот. Се користи M/H2 (2-фазна хипер-експоненцијална дистрибуција) ред за моделирање на комутаторите, додека контролерот се моделира како M/M. Резултатите јасно покажуваат дека ако веројатноста на наидување на *Packet-in* пораките е висока, мрежните перформанси остро се деградираат. Најголем недостаток на трудот претставува фактот што не е направена валидација на математичкиот модел со симулации.

Во последно време, детално се истражени лимитациите на единствен контролер поради ограничениот капацитет и неможноста да се управува со голем број текови. Трудот [71] анализира архитектура на мулти-контролери во SDN-базирани мрежи. Во детали се анализираат разликите помеѓу повеќе типови архитектури преку опишување на нивниот дизајн, комуникациски процес и нивните перформансни резултати. Генерално, архитектурата која содржи повеќе контролери следи рамна или хиерархиска архитектура. Во рамната архитектура, сите контролери се позиционирани хоризонтално на едно ниво, што значи дека секој контролер има само сопствен парцијален преглед на својот домен. Во хиерархиската архитектура, контролерите се позиционирани вертикално во повеќе нивоа. Дополнително, се врши анализа на постојните логички централизираните архитектури (*HyperFlow*, *ONOS*, *DISCO*), и логички дистрибуираните архитектури (*KANDOO*, *ORION*). Во [72] понатаму се анализира логичката централизација во SDN, посспецифично дистрибуираната контролна рамнина. Се опишуваат главните особини на дистрибуираните SDN контролери, како што се: мрежни домени, локални/глобални состојби, *East/Westbound* API и детално се анализираат аспекти поврзани со скалабилност, испад, конзистентност и

приватност. Конечно, авторите даваат детална класификација на постојните проекти со дистрибуирани SDN контролери, при тоа земајќи ги во предвид сите можности за дизајн. Во оваа глава и во Главата 6, се става фокус на рамна мулти-контролер архитектура во логички централизирана мрежа со дистрибуирана SDN контролна рамнина.

Во [73] се опишува зависноста на барањата за QoS, бројот на контролери и минимизација на цената на чинење на ресурсите и одржување на мрежата. Се моделира систем како M/M/m ред и се предлага солиден математички модел кој овозможува минимизација на времето на опслужување на контролерот, како и естимација на влијанието врз перформансите на следните параметри: број на контролери, број на комутатори и цена на ресурси. Конечно, се предлага и дистрибуирана шема на одлука за распоред на оптоварувањето (на англ. load balancing), преку проучување на економската врска помеѓу ратата на пристигнување на тек, ресурсите и цената. Истиот математички пристап е проширен и ставен во друг контекст во [74]. Екстензивно се проучени контрибуторите на латентност во дистрибуирани SDN мрежи, а целта е да се минимизира латентноста крај-до-крај преку предлагање на концепт за мрежно партиционирање, наречено *Clustering-based Network Partition Algorithm* (CNPA). Предложената шема се валидира преку симулации со користење на две реални мрежни топологии. Во трудот [75] се анализира пристап со хиерархиски мулти-контролери со главен контролер (на англ. root controller) кој се моделира како M/M/1 и локални контролери кои се моделираат како M/M/1/m систем. Се користи моделот на Клајнрок (на англ. Kleinrock) за да се поврзат во каскада двата типа на контролери и се предлага сценарио во кое главниот контролер се контактира само доколку локалниот контролер не успее да изврши проверка за совпаѓање на пакетскиот тек. Со користење на детални симулации, се докажува валидноста на предложениот математички модел преку анализа на перформансите на контролерот во два случаи: без помош од главниот контролер и кога локалниот контролер мора да ги насочи сите пакети кон главниот контролер. [76] го анализира проблемот на единствена точка на испад (на англ. single point of failure) во контролата рамнина и предлага нов механизам за засилување на отпорноста на SDN. Предложената организација на компоненти се извршува независно на врвот на мрежниот оперативен систем (на англ. network operating system (NOS)) и овозможува отпорност на неколку типови испади и незабележлива транзиција помеѓу контролерот во состојба на испад и резервниот контролер. Во различните сценарија и симулации на топологии во Mininet, се докажува ефикасноста на предложената компонента, дека може да постигне отпорност без значителна модификација на OF протоколот и без значајно зголемување на дополнителниот сообраќај заради сигнализација. Слично, во [77] се

истражува дизајн со *master-slave* OF контролер кој е соодветен за софтверски дефинирани еластични оптички мрежи (на англ. software defined elastic optical networks (SD-EON)). Се развива комуникациски протокол за контролери кој овозможува *master* контролерот да го синхронизира статусот со *slave* контролерот во реално време, затоа што *slave* контролерот може брзо да детектира испад на *master* контролерот и да ја превземе контролата, со што се одбегнува прекин на услугата. Предложената конструкција (на англ. framework) е имплементирана во контролната рамнина на SD-EON тестен систем и се состои од сервери со високи перформанси кои се користат за евалуација на ефективност на решението во различни сценарија при мрежен испад.

Во трудот [78] се предлага архитектура која применува кластер од повеќе контролери и се фокусира на сигналниот сообраќај кој се разменува помеѓу контролерите. Се развива аналитички модел за квантификација на процесирачкото доцнење во комутаторите настанато поради комуникацијата помеѓу контролерите. Преку проучување на некои реални топологии, се анализира проблемот на поставување на контролер (на англ. controller placement problem) и можните зависимости при доцнењето. Исто така, во [79] се адресира проблемот на поставување на контролер со цел да се максимизира расположивоста на контролната мрежа. Се естимира влијанието на бројот на контролери во контролната рамнина преку користење реални мрежни топологии. Се предлагаат неколку алгоритми за поставување контролери и заклучокот е дека поставување на преголем или премал број контролери сериозно влијае на расположивоста. Во трудот [80] се воведува ефикасен алгоритам кој го зема во предвид оптоварувањето на контролерот. Направената евалуација покажува дека предложената шема може значајно да го намали бројот на контролери и да го снижи оптоварувањето на најоптеретениот контролер во мрежата. Конечно, авторите во [63] истражуваат како да се намали времето на процесирање на пакети во контролерот преку имплементација на повеќе контролери. Главната цел е да се евалуира времето на процесирање на пакети и да се оптимизира бројот на контролери. Се предлага аналитички модел и се имплементира прототип со мулти-контролери. Преку мерење на времето на опслужување на контролерот, се прилагодува хипер-Ерланг (на англ. Hyper-Erlang) дистрибуција кон времето на одговор, која подоцна се користи во математичкиот модел за одредување на оптималниот број контролери.

5.2 Опис на проблемот и научен придонес

Во оваа Глава 5, се анализира целокупната латентност во SDN-базирани мобилни мрежи, со специјален интерес ставен на моделирање на доцнењето помеѓу комутаторите и

контролерите. Добиените резултати може да се искористат од SDN дизајнерите за да ги идентификуваат и квантифициваат критичните фактори кои влијаат на латентноста и да се адресира едно важно прашање во денешните SDN мрежи, проблемот на поставување на контролер.

Понатаму, целта е да се анализира и евалуира влијанието што најчестите хипотези го имаат на прецизноста при моделирање на перформансите во SDN-базирани мобилни мрежи. Со користење на теоријата на редови, се предлага аналитички модел со цел да се пресметаат ефектите на ограничен бафер и средна рата на опслужување која зависи од оптоварувањето. Потоа, предложениот модел се споредува со постојните аналитички пристапи кои не ги земаат во предвид овие особини. Најбитниот придонес во оваа Глава може да се сумира преку следните точки:

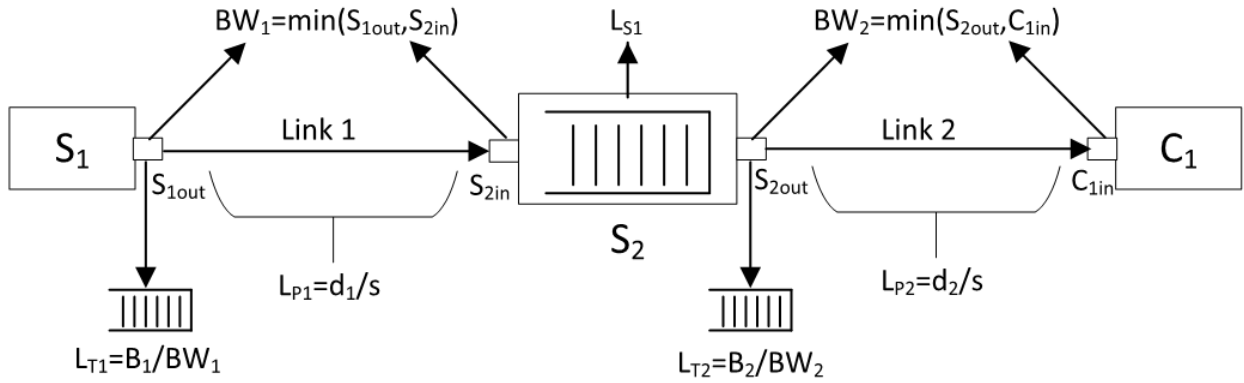
1. Детално се истражуваат сите можни конституенти на латентноста од комутаторите до контролерите. Свкупната SDN латентност се дели на две групи: латентност крај-до-крај и латентност поради процесирање во контролерот.
2. Се предлага математички модел базиран на теоријата на редови за моделирање на средното време на опслужување на комутаторот и контролерот преку користење на M/M/1 и M/M/c моделите на редови, соодветно. Се заклучува дека оптоварувањето и бројот на контролери, ратата на пристигнување на пакети, средната рата на опслужување и веројатноста на наидување на *Packet-in* пораките се главни фактори кои влијаат на латентноста и перформансите во SDN мрежи.
3. Се предлага аналитички модел базиран на M/M/m/K и M/M/1 (со рата на опслужување зависна од состојбата) редови за да се квантифицива ефектот на ограничен бафер и ратата на опслужување која зависи од оптоварувањето.
4. Преку користење на MATLAB, нумерички се евалуираат предложените аналитички модели и се прави споредба на добиените резултатите со традиционалните аналитички пристапи.

5.3 Моделирање на латентноста во SDN

Латентноста во LTE беше детално опишана во Глава 3.1. Во ова поглавје се става фокус на факторите кои влијаат на латентноста помеѓу комутаторите и контролерот и се врши аналитичко моделирање на воведеното доцнење користејќи теорија на редови.

5.3.1 Латентност крај-до-крај

Се претпоставува SDN мрежа со n комутатори, $S = \{s_1, s_2, \dots, s_n\}$, и c контролери $C = \{c_1, c_2, \dots, c_c\}$. Со $h(s_n, c_c)$ се означува бројот на делницата (на англ. hop) од комутаторот s_n до контролер c_c . На сликата 5.1 е прикажана севкупната латентност кога еден пакет се трансмитира од комутатор s_1 до контролерот c_1 .



Слика 5.1. SDN латентност крај-до-крај

Кога еден пакет се трансмитира од комутаторот s_n кон контролерот c_c , постојат три главни компоненти кои ја сочинуваат латентноста крај-до-крај: трансмисијска латентност (L_{Ti}), пропагацијска латентност (L_{Pi}) и латентност поради процесирање во комутаторот (L_{Si}). Трансмисијската латентност (време за вбризување пакетски бити на линкот) е дадено со $L_{Ti} = B_i/BW_i$, каде B_i е количината на пакетски бити на линкот i и BW_i е опсегот на линкот. На сликата 5.1, опсегот на Линкот 1 е $BW_1 = \min(S_{1out}, S_{2in})$, што значи помалата трансмисијска брзина од двата мрежни интерфејси. Пропагацијската латентност (времето кое е потребно пакетот да стигне до дестинацијата) е дадена со $L_{Pi} = d_i/s$, каде d_i е растојанието на линкот i , s е брзината на податоците кои патуваат низ линкот. Понатаму, се продолжува кон истражување на процесирачката латентност на комутаторот (L_{Si}).

Се користи пристапот од [64], [70] и комутаторот се моделира со користење на M/M/1 ред, каде μ_i е средната рата на опслужување на комутаторот i , а λ_i е ратата на пристигнување на пакети во истиот комутатор. Се претпоставува дека влезните пакети следат Поасонова дистрибуција и секој комутатор се моделира како единствен ред, а не по интерфејс. Веројатноста да нема пакети во редот е означена како $p_0 = 1 - \lambda_i/\mu_i$, а веројатноста дека системот има k текови кои чекаат во редот е $p_k = p_0 (\lambda_i/\mu_i)^k$. За системот да биде стабилен, за оптоварувањето на i -тиот комутатор, мора да се задоволи $\rho_i = \lambda_i/\mu_i < 1$, кое овозможува $p_0 > 0$. Просечниот број текови $E(N_i)$ кои чекаат да бидат опслужени во системот е даден со:

$$E(N_i) = \sum_{k=0}^{\infty} k p_k = \sum_{k=0}^{\infty} k \left(\frac{\lambda_i}{\mu_i}\right)^k \left(1 - \frac{\lambda_i}{\mu_i}\right) = \frac{\lambda_i}{\mu_i - \lambda_i}. \quad (5.1)$$

Со користење на законот на Литл (на англ. Little's law) [10], просечното време кое текот го поминува во комутаторот i е:

$$E(T_i) = \frac{E(N_i)}{\Lambda_i} = \frac{1}{\mu_i - \Lambda_i} = \frac{1/\mu_i}{1 - \rho_i}. \quad (5.2)$$

Конечно, средното време на процесирање на комутаторите $E(T_s) = L_{S_i}$ е дадено со:

$$E(T_s) = \sum_{i=1}^n \frac{\Lambda_i}{\sum_{i=0}^n \Lambda_i} E(T_i) = \sum_{i=1}^n \frac{\Lambda_i}{\sum_{i=0}^n \Lambda_i} * \frac{1}{1 - \rho_i}. \quad (5.3)$$

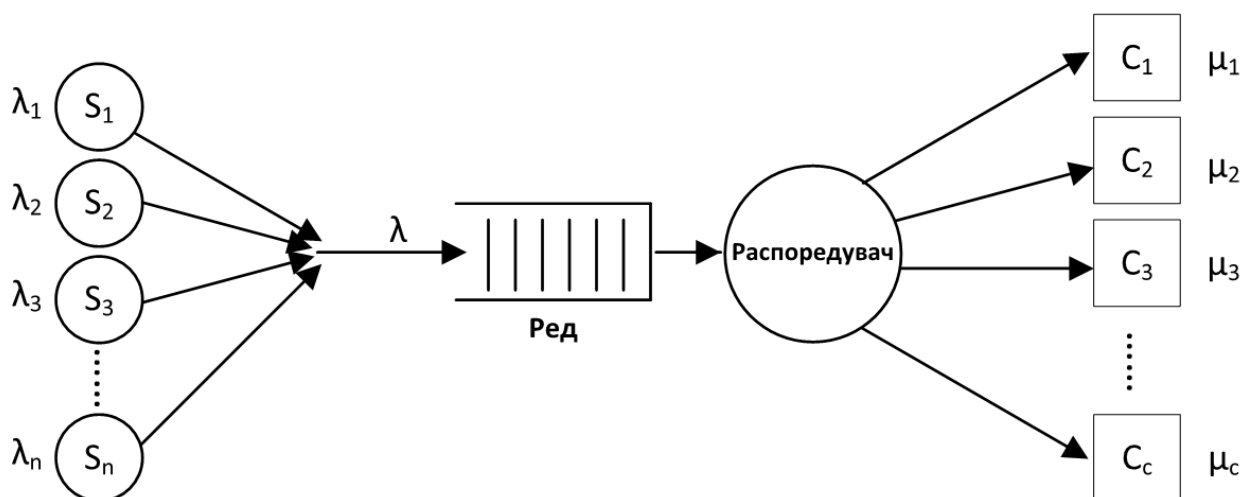
Враќајќи се наназад, севкупната латентност која се однесува на пакет што патува од комутаторот s_n кон контролерот c_c е даден со следниот израз:

$$L_{e2e(s_n, c_c)} = \sum_{i=1}^{h(s_n, c_c)} \left(\frac{B_i}{BW_i} + \frac{d_i}{s} + L_{S_{i-1}} \right). \quad (5.4)$$

За мобилна *backhaul* и јадрена мрежа која најчесто се простира низ стотици километри и е опремена со 10 Gbit/s комутатори, трансмисијската латентност може да се смета за занемарлива, додека пропагацијската латентност може да биде значајна. Во наредното поглавје се прави евалуација на процесирачката латентност во контролерот.

5.3.2 Латентност поради процесирање во контролерот

Кога влезниот пакет нема да се совпадне со постојните табели на тек во комутаторот, пакетот се праќа кон контролерот преку *Packet-In* порака. Потоа, контролерот ги инсталира потребните нови записи во табелите на тек кои се потребни за препраќање на пакетот кон дестинацијата. Моделот на опслужување помеѓу комутаторите и контролерот е прикажан на сликата 5.2.



Слика 5.2. Модел на опслужување помеѓу комутаторите и контролерот

За да се анализира средното време на одзив на контролерот, се користи M/M/c ред, слично како во [73]. Секој контролер има независна средна рата на опслужување μ_i која

слиди експоненцијална дистрибуција, а влезните пакети следат Поасонова дистрибуција на пристигнување. Системот има неограничен бафер и ратата на пристигнување λ е константа дадена со:

$$\lambda = \sum_{i=1}^n q_i^p \Lambda_i, \quad (5.5)$$

, каде q_i^p е веројатноста на наидување на *Packet-in* пораките и Λ_i е ратата на пристигнување на пакети во i -тиот комутор. Системот има c контролери, па:

$$\mu_k = \begin{cases} k\mu & 0 \leq k \leq c \\ c\mu & c \leq k \end{cases}. \quad (5.6)$$

Веројатноста на пакети кои чекаат во редот е дадена со:

$$p_k = \begin{cases} p_0 \frac{(c\rho)^k}{k!} & k \leq c \\ p_0 \frac{(\rho)^k c^c}{c!} & k \geq c \end{cases}, \quad (5.7)$$

каде $\rho = \lambda / c\mu \leq 1$. За да се реши p_0 , се користи [26]:

$$1 = \sum_{k=0}^{\infty} p_k = p_0 + \sum_{k=1}^{\infty} p_k = p_0 \left[1 + \sum_{k=1}^{c-1} \frac{(c\rho)^k}{k!} + \sum_{k=c}^{\infty} \frac{(\rho)^k c^c}{c!} \right]. \quad (5.8)$$

Дел од равенката (5.8) се заменува со следниот израз:

$$\sum_{k=c}^{\infty} \frac{\rho^k c^c}{c!} = \frac{1}{c!} \sum_{k=c}^{\infty} \rho^k c^c = \frac{(c\rho)^c}{c!} \sum_{k=c}^{\infty} \rho^{k-c} = \frac{(c\rho)^c}{c!} \frac{1}{1-\rho}, \quad (5.9)$$

и за p_0 се добива:

$$p_0 = \left[1 + \sum_{k=1}^{c-1} \frac{(c\rho)^k}{k!} + \frac{(c\rho)^c}{c!} \frac{1}{1-\rho} \right]^{-1} \quad (5.10)$$

Веројатноста дека нов пакетски тек треба да чека во редот е дадена со:

$$Prob(q.) = \sum_{k=c}^{\infty} p_k = p_0 \sum_{k=c}^{\infty} \frac{c^c}{c!} \rho^k = p_0 \frac{(c\rho)^c}{c! (1-\rho)} \quad (5.11)$$

Средното време кое еден пакетски тек го поминува во системот се пресметува со следниот израз:

$$E(t_c) = \frac{1}{\lambda} \left(c\rho + \rho \frac{(c\rho)^c}{c!} \frac{p_0}{(1-\rho)^2} \right) \quad (5.12)$$

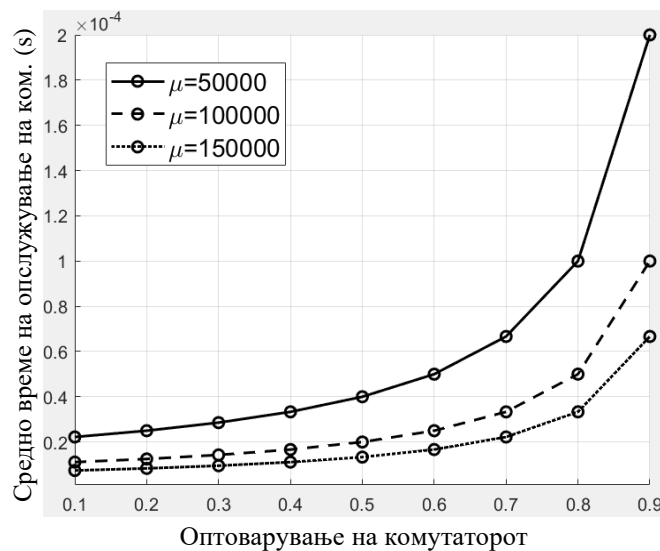
$$= \frac{\left(\frac{\lambda}{\mu}\right)^c \mu}{(c-1)!(c\mu - \lambda)^2} p_0 + \frac{1}{\mu} \quad (5.13)$$

5.3.3 Анализа на перформанси

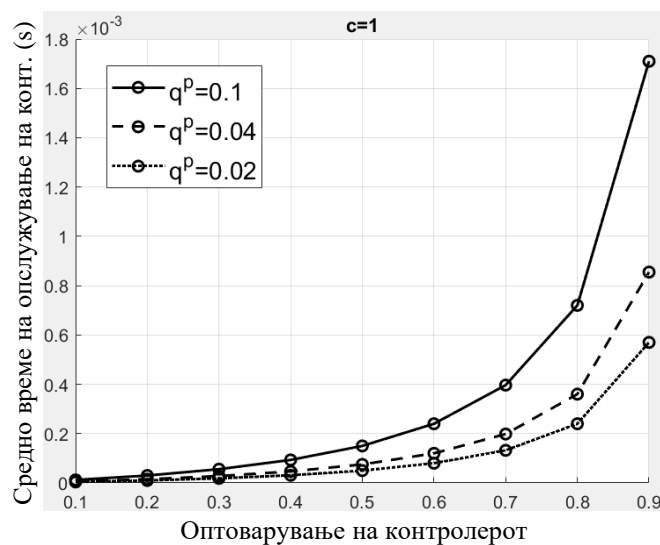
Во ова поглавје, се презентираат перформансните резултати на предложениот аналитички модел. Евалуацијата се врши во MATLAB. Генерално, круцијален фактор кој влијае на севкупната латентноста е всушност латентноста при процесирање во контролерот, додека другите типови латентност имаат помало влијание. Ова е причината поради која од

интерес се главните фактори кои влијаат на средното време на опслужување на контролерот, како и анализа на тоа како бројот на контролери влијае на латентноста.

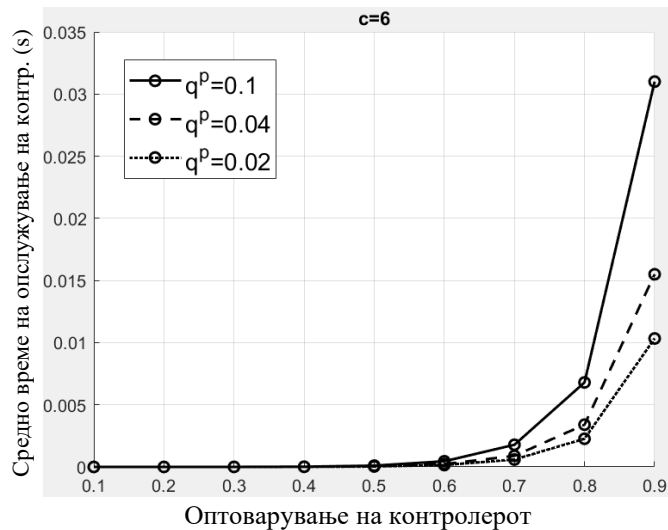
На сликата 5.3 прикажано е средното време на опслужување на комутаторот како функција од оптоварувањето. Се користи равенката (5.3), каде ρ варира за дадено μ . За пониски вредности на оптоварувањето, се забележува помало средно време на опслужување, но како ρ се приближува кон повисоки вредности, времето на опслужување има тенденција да расне експоненцијално. Другиот коментар е дека времето на процесирање на комутаторот зависи од средната рата на опслужување на комутаторот, доколку μ е повисоко, средното време на опслужување е пониско. Денешните SDN комутатори се доста напредни и за комутатор со 10 Gbit/s (или 100 Gbit/s), латентноста при процесирање (L_{Si}) може да се занемари при пресметка на вкупната латентност.



Слика 5.3. Рата на опслужување на комутаторот наспроти оптоварувањето на комутаторот.

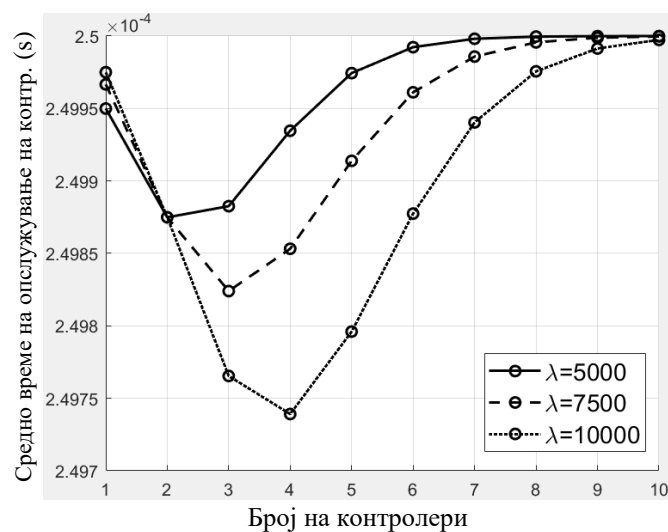


Слика 5.4а. Рата на опслужување на контролерот наспроти оптоварувањето на контролерот ($c=1$).



Слика 5.4b. Рата на опслужување на контролерот наспроти оптоварувањето на контролерот ($c=6$).

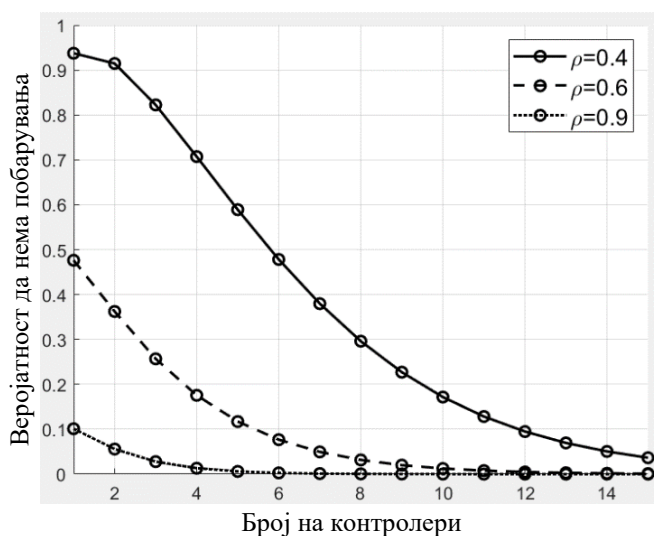
На сликата 5.4a и 5.4b претставена е зависноста на средното време на опслужување на контролерот од оптоварувањето на контролерот. Средното време на опслужување се стреми кон бесконечност за оптоварување блиску до 1. Како што расне веројатноста на наидување на *Packet-in* пораките, времето на опслужување на контролерот се деградира побрзо, особено при високо оптоварување. За сликата 5.4a се користи $c=1$, а за сликата 5.4b се користи $c=6$. Целта е да се истражи влијанието на бројот на контролери врз средната рата на опслужување. Како што се очекуваше, додавање на повеќе контролери во мрежата влијае позитивно на перформансите. Дополнително, резултатите покажуваат дека преку додавање на повеќе контролери во системот, не се афектира времето на опслужување при повисоки вредности на оптоварувањето, во споредба со случајот кога се користи помал број контролери.



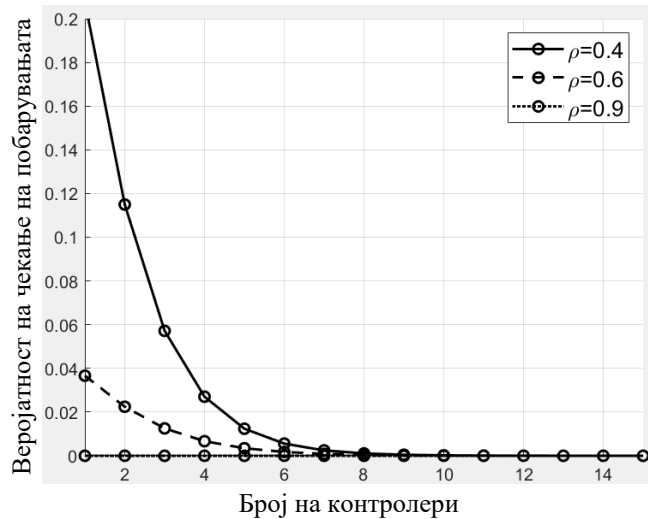
Слика 5.5. Рата на опслужување на контролерот наспроти број на контролери

Понатаму, целта е да се види дали предложениот модел може да се искористи за проценка на оптималниот број на контролери. Се користи равенката (5.13) со цел да се истражи зависноста на времето на опслужување од бројот на контролери, при променлива брзина на пристигнување на пакети. Ова е прикажано на сликата 5.5. Генерално, се забележува дека за секоја брзина на пристигнување на пакети (при фиксна средна рата на опслужување), постои оптимален број на контролери кој овозможува минимално време на опслужување. За $\lambda=5000$, $c_{min}=2$, за $\lambda=7500$, $c_{min}=3$ и за $\lambda=10000$, $c_{min}=4$. Сепак, заклучокот е дека добивката од оваа анализа е минимална, со оглед дека времето на опслужување за оптималниот број контролери минимално се намалува. Како што е прикажано на сликата 5.5, ова намалување е во ранг на микросекунди.

На крајот, се анализираат сликите 5.6 и 5.7. За сликата 5.6 се користи равенката (5.10) за да се докаже дека веројатноста на бројот на текови во редот варира како што варира и оптоварувањето на контролерот. Веројатноста да нема пакети во системот е ниска ако оптоварувањето е високо и како што расне бројот на контролери. Се користи равенката (5.11) за сликата 5.7, каде се заклучува дека со зголемување на бројот на контролери, веројатноста на пакетите кои чекаат во системот да бидат опслужени, значително се намалува. Анализата ги покажува очекуваните резултати, сепак мора да се нагласи дека зголемувањето на бројот на контролери води кон зголемени трошоци за операторите и мора да биде предмет на внимателна оптимизација. Од друга страна, кога станува збор за виртуелни контролери, тогаш додавање и одземање на контролерите е всушност додавање и бришење на виртуелни машини (на англ. virtual machines (VM)), што претставува многу корисна техника за оптимизација на ресурси.



Слика 5.6. Врска помеѓу ρ и бројот на контролери, за дадено оптоварување



Слика 5.7. Врска помеѓу $Prob(q)$ и бројот на контролери, за дадено оптоварување

5.4 Влијанието на најчестите претпоставки користени при моделирање на SDN мрежи

Во ова поглавје се дискутираат и математички моделираат едни од најчестите хипотези кои се користат при моделирањето на SDN мрежите: неограничен бафер и рата на опслужување која е независна од оптоварувањето. Целта е да се квантифицива грешката која се прави кога се земаат во предвид овие претпоставки при математичко моделирање на SDN мрежите.

5.4.1 Неограничен бафер

Со цел да се добијат равенки во затворена форма, претпоставката за неограничен бафер можеби е една од најчестите хипотези. Во реален OF-комутатор, постојат повеќе влезни и излезни порти и секоја влезна порта има фиксна големина на бафер. Пакетите на влез од секој бафер понатаму се процесираат според истите табели на тек и нема приоритизација на контролниот сообраќај (секој пакет се процесира како FIFO). Мора да се нагласи дека некои OF-комутатори имаат вградено способност за приоритизација на сообраќај, но тие не се предмет на оваа анализа (ќе бидат разгледани во Глава 6). Користејќи ја теоријата на редови, целта е математички да се моделира времето на опслужување на контролерот во случај на бесконечен и конечен бафер и да се анализираат разликите.

Се претпоставува SDN мрежа со n комутатори, $S = \{s_1, s_2, \dots, s_n\}$, и c контролери, $C = \{c_1, c_2, \dots, c_c\}$. Се претпоставува дека влезните пакети следат Поасонова распределба и секој контролер се моделира како единствен ред, а на по интерфејс. Се користи пристапот од [73] и контролерот се моделира како M/M/c/K (мулти-сервер, систем со конечен капацитет) ред. Моделот на опслужување помеѓу комутаторите и контролерот е прикажан на сликата 5.2.

При негативна проверка за совпаѓање на влезниот пакет со постојните записи во табелата на тек во комутаторот, пакетот се праќа кон контролерот преку *Packet-in* порака. Тогаш контролерот ги инсталира сите потребни записи во табелата на тек во сите комутатори на пакетската патека. Секој контролер има независна рата на опслужување μ_i која следи експоненцијална дистрибуција, додека влезните пакети следат Поасонова дистрибуција на пристигнување. Системот на контролери има ограничен ред од K пакети, бројот на контролери е c , а ратата на пристигнување е константна и изнесува λ . За М/М/с/К ред, вредностите за ратата на пристигнување и средната рата на опслужување се:

$$\lambda_n = \begin{cases} \lambda, & 0 \leq n < K \\ 0, & n \geq K \end{cases} \quad (5.14)$$

$$\mu_n = \begin{cases} n\mu, & 1 \leq n \leq c \\ c\mu, & c \leq n \leq K \end{cases} \quad (5.15)$$

Веројатноста на пакетите кои чекаат во редот е дадена со:

$$p_n = \begin{cases} p_0 \frac{1}{n!} \left(\frac{\lambda}{\mu}\right)^n & 0 \leq n \leq c, \\ p_0 \frac{1}{c^{n-c} c!} \left(\frac{\lambda}{\mu}\right)^n & c \leq n \leq K, \end{cases} \quad (5.16)$$

каде p_0 е веројатноста да нема пакети во редот. Со користење на следната равенка:

$$\sum_{i=0}^K p_i = 1, \quad (5.17)$$

за p_0 се добива:

$$p_0 = \left[\sum_{n=0}^{c-1} \frac{1}{n!} (\rho)^n + \sum_{n=c}^K \frac{1}{c^{n-c} c!} (\rho)^n \right]^{-1}, \quad (5.18)$$

каде $\rho = \lambda / c\mu \leq 1$. Средното време кое текот го поминува во системот, $E(t_c)$ може да се претстави со:

$$E(t_c) = \frac{E(N)}{\lambda'} + \frac{1}{\mu} = \frac{E(N)}{\lambda(1-p_K)} + \frac{1}{\mu}, \quad (5.19)$$

каде $E(N)$ е просечниот број баферирани пакети во системот, а $\lambda' = \lambda(1-p_K)$ е ефективната рата на пристигнување. После користење на неколку замени во равенката (5.19), за $E(N)$ се добива:

$$E(N) = p_0 \frac{(c\rho)^c \rho}{c! (1-\rho)^2} (1 - \rho^{K-c+1} - (1-\rho)(K-c+1)\rho^{K-c}). \quad (5.20)$$

Целта е да се споредат $E(t_c)$ и p_0 од равенката (5.19) и равенката (5.18) со истите параметри калкулирани во случајот на М/М/с ред (неограничен бафер). Соодветните равенки за М/М/с редот се дадени со следните изрази:

$$E(t_c) = \frac{1}{\lambda} \left(c\rho + \rho \frac{(c\rho)^c}{(c-1)! (1-\rho)^2} p_0 \right), \quad (5.21)$$

$$p_o = \left[1 + \sum_{n=1}^{c-1} \frac{(c\rho)^n}{n!} + \frac{(c\rho)^c}{c!} \frac{1}{1-\rho} \right]^{-1}. \quad (5.22)$$

5.4.2 Рата на опслужување независна од оптоварувањето

Во ова поглавје се користи аналитичко моделирање со цел да се земе во предвид фактот дека реалните имплементации на контролери имаат деградација на средната рата на опслужување со порастот на оптоварувањето. Се користи M/M/1 ред со рата на опслужување зависна од состојбата (на англ. state dependent service) [26], што значи дека се разгледува систем во кој серверот работи со средна рата на опслужување μ_1 се додека нема l пакети во системот, момент во кој средната рата на опслужување се менува во μ_2 . Изразите за ратите на пристигнување и опслужување се дадени со:

$$\lambda_n = \lambda, \quad \text{за сите } n, \quad (5.23)$$

$$\mu_n = \begin{cases} \mu_1, & 1 \leq n \leq l \\ \mu_2, & n \geq l \end{cases} \quad (5.24)$$

Веројатноста на пакети кои чекаат во редот е:

$$p_n = \begin{cases} p_o \left(\frac{\lambda}{\mu_1} \right)^n & 0 \leq n \leq l, \\ p_o \frac{\lambda^n}{\mu_1^{l-1} \mu_2^{n-l+1}} & n \geq l. \end{cases} \quad (5.25)$$

Веројатноста да нема пакети во редот, p_o и ратата на опслужување на контролерот $E(t_c)$ се дадени со следните равенки:

$$p_o = \left[\frac{1 - \rho_1^l}{1 - \rho_1} + \frac{\rho_2 \rho_1^{l-1}}{1 - \rho_2} \right]^{-1}, \quad (5.26)$$

$$\begin{aligned} E(t_c) &= \frac{E(N)}{\lambda} = \frac{\sum_{n=0}^{\infty} n p_n}{\lambda} = \\ &= \frac{1}{\lambda} p_o \left(\sum_{n=0}^{l-1} n \rho_1^n + \sum_{n=l}^{\infty} n \rho_1^{l-1} \rho_2^{n-l+1} \right) = \\ &= \frac{p_o}{\lambda} \left(\frac{\rho_1 (1 + (l-1)\rho_1^l - l\rho_1^{l-1})}{(1-\rho_1)^2} + \frac{\rho_2 \rho_1^{l-1} (l - (l-1)\rho_2)}{(1-\rho_2)^2} \right). \end{aligned} \quad (5.27)$$

каде $\rho_1 = \lambda/\mu_1$ и $\rho_2 = \lambda/\mu_2$. Целта е да се споредат равенката (5.26) и равенката (5.27) со равенките на општ M/M/1 ред кој има една средна рата на опслужување, μ . Равенките од интерес се следните:

$$E(t_c) = \frac{E(N)}{\lambda} = \frac{1/\mu}{1-\rho} = \frac{1}{\mu - \lambda} \quad (5.28)$$

$$p_o = 1 - \rho. \quad (5.29)$$

На крајот, се истражува зависноста на средната рата на опслужување на контролерот од бројот на комутатори. Со модификација на равенката (5.29) користејќи го трудот [81], се добива:

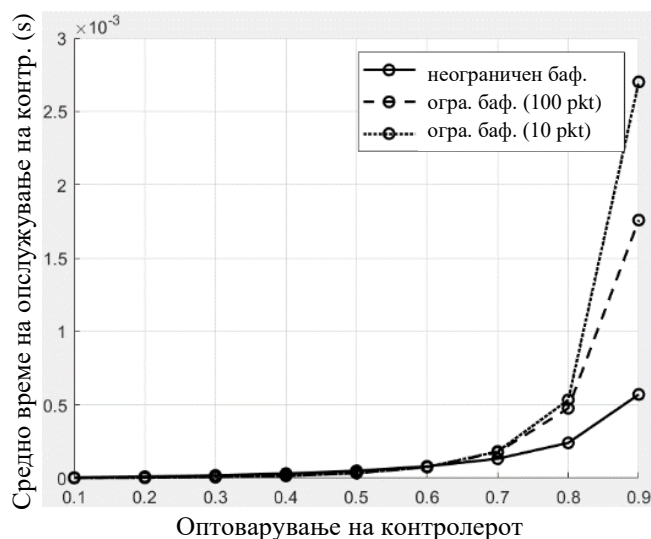
$$E(t_c) = \frac{s + 1}{2(\mu - \lambda s)}, \quad (5.30)$$

каде s е бројот на комутаторите.

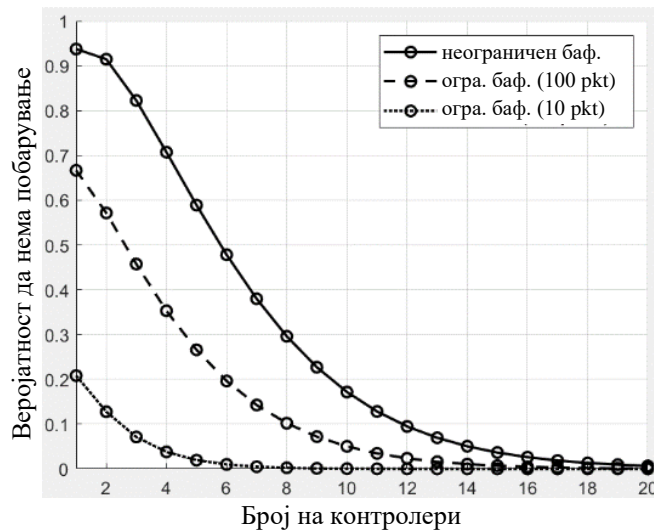
5.4.3 Анализа на перформанси

Во овој дел се презентираат перформансните резултати од предложениот аналитички модел. Евалуацијата се врши со користење на MATLAB. Акцент се става на анализата на времето на опслужување на контролерот поради тоа што претставува најкритичен фактор кој влијае на севкупната латентност во SDN-базирани мобилни мрежи.

Во трудот [64], се користи тестен систем за евалуација на средната рата на опслужување на контролерот, каде за овој параметар измерена е средна вредност од 4166. Поради последното, во оваа анализа за μ се користат вредности кои се во регионот на оваа референтна вредност. Просечната големина на пакет се зема да биде 128 В. На сликата 5.8 може да се види зависноста на ратата на опслужување на контролерот од просечното оптоварување на контролерот. Равенката (5.20) се заменува во (5.19) и се споредува со (5.21). Се зема во предвид дека $c=2$ и $\mu=5000$. Општата анализа покажува дека ако оптоварувањето е пониско, ратата на опслужување е исто така ниска, но како што оптоварувањето расне, особено при високи оптеретувања > 0.7 , времето на опслужување расне експоненцијално. Може да се заклучи дека за случајот на неограничен бафер, средното време на опслужување е пониско, споредено со случајот кога баферот има конечна вредност, средното време на опслужување се влошува побрзо. Ова јасно се забележува за оптоварување > 0.8 . Се заклучува дека при високи оптеретувања, постои девијација во аналитичкиот модел кога се споредуваат двата случаи на бесконечен и конечен бафер во контролерот.



Слика 5.8. Време на опслужување наспроти оптоварување на контролерот (ограничен бафер)

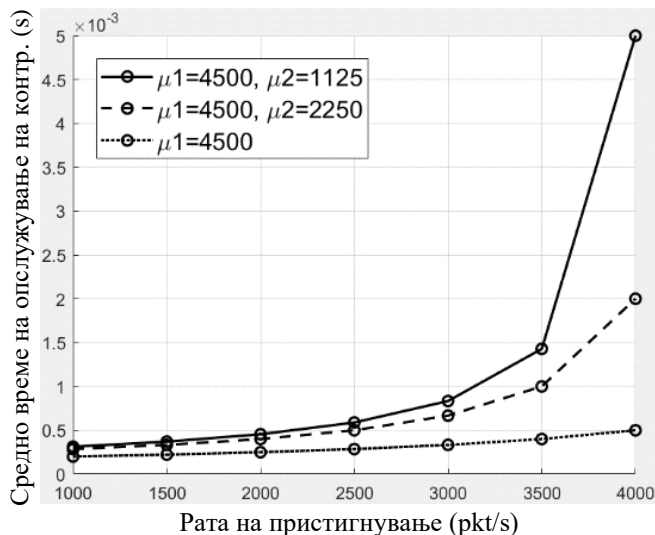


Слика 5.9. Врска помеѓу p_0 и бројот на контролери (ограничен бафер)

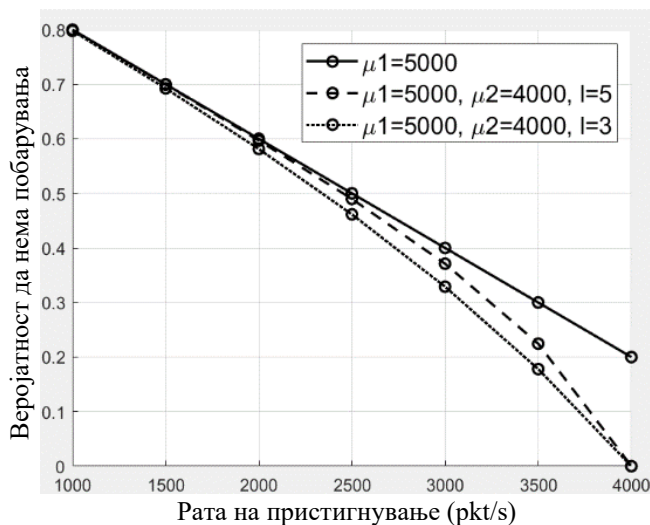
На сликата 5.9 се споредуваат резултатите од равенките (5.18) и (5.22). Се прави обид за анализа на веројатноста да нема пакети во редот, за случаите на ограничен и неограничен бафер. Ако баферот е неограничен, тогаш за мал број контролери, веројатноста да нема пакети во баферот е многу висока. Како што се зголемува бројот на контролери, оваа веројатност се движи кон помали вредности. Ако баферот е конечен, тогаш веројатноста да нема пакети при низок број контролери е пониска отколку во случајот на неограничен бафер. Ако бројот на контролери е висок, тогаш оваа веројатност се намалува. Заклучокот е според очекувањата, а тоа е дека ограничениот бафер влијае на прецизноста на моделирањето и претставува битен параметар кој мора да се земе во предвид.

Понатаму, се продолжува со анализа на ратата на опслужување која зависи од оптоварувањето. Се користат равенките (5.27) и (5.28) за да се анализира влијанието на оваа хипотеза. На сликата 5.10, се споредува случајот на единствена средна рата на опслужување од $\mu_1=5000$ со два случаи каде $\mu_1=5000$, но откако баферот ќе се исполни со $l=4$ пакети, ратата на опслужување се деградира и има вредност од μ_2 . Се одбира вредноста за $l=4$ пакети, бидејќи користењето на оваа вредност креира јасен визуелен заклучок за ефектот кога μ_1 се менува кон μ_2 . За ратите на пристигнување кои се користат, би се добиле слични заклучоци и за $l=2$ до $l=5$, но зголемувањето на l кон повисоки вредности би ги замаглиле заклучоците кои треба да се потенцираат. Како што ратата на пристигнување се зголемува, ратата на опслужување на контролерот исто така се зголемува, сепак за случајот на единствена средна рата на опслужување, ова зголемување е многу умерено. Ако μ_2 е пониско, тогаш за случајот на рата на опслужување зависна од состојбата, влошувањето на времето на опслужување на контролерот е многу повеќе видливо. Кога l варира, може лесно да се заклучи дека ако l е помало, времето на опслужување на контролерот се влошува

побрзо дури и при помали рати на пристигнување. Како што l се зголемува, деградацијата е помалку видлива. Конечно, може да се коментира дека влијанието на варијабилната средна рата на опслужување не е занемарливо и претставува битен фактор кој треба да се земе во предвид при моделирање на SDN мрежи.



Слика 5.10. Време на опслужување на контролерот наспроти оптоварувањето (рата на опслужување зависна од оптоварувањето)

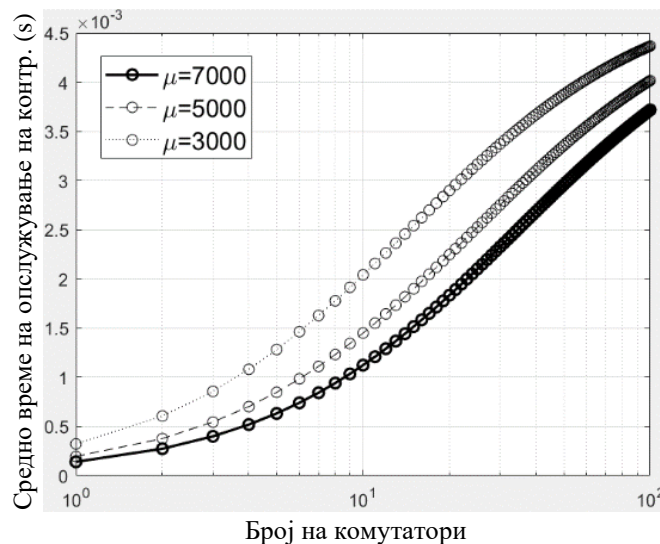


Слика 5.11. Врска помеѓу ρ_0 и ратата на пристигнување (рата на опслужување зависна од оптоварувањето)

Сликата 5.11 прикажува слична анализа како сликата 5.9, но овој пат за X-оската се користи ратата на пристигнување. Се споредуваат равенките (5.26) со (5.29). Со зголемување на ратата на пристигнување, веројатноста да нема побарувања се намалува, како што и се очекуваше. Сепак, во случај на варијабилна средна рата на опслужување, оваа деградација е многу повидлива, особено ако l е пониско, како што е прикажано на сликата 5.11. Се анализираат две вредности за l , $l=3$ и $l=5$. Дури и ако се земат во предвид повисоки вредности за l , резултатите на сликата 5.11 нема многу да се променат. Се заклучува дека

бројот на пакети во системот, при кој се менува ратата на опслужување, претставува битен параметар кој влијае на моделирањето на перформансите.

Конечно, се прави анализа на времето на опслужување на контролерот од бројот на комутатори, прикажано на сликата 5.12. Се користи равенката (5.30) и се анализира горе споменатата зависност за три различни вредности на средното време на опслужување на контролерот. Како што се зголемува бројот на комутаторите, времето на опслужување на контролерот се зголемува и како што се намалува средната рата на опслужување μ , ова зголемување во времето на опслужување е уште повидливо. Последниот заклучок е во согласност со очекувањата и може да се констатира дека во едно вакво сценарио на голем број комутатори, треба да се размислува за примена на дистрибуирана контролна рамнина (наместо единствен контролер) бидејќи ратата на опслужување навистина брзо се деградира кога мрежата се состои од многу комутатори.



Слика 5.12. Време на опслужување на контролерот наспроти број на комутатори

5.5 Заклучок

Во оваа глава се анализираше латентноста во мобилни SDN-базирани мрежи. Математичкото моделирање редовно се користи од истражувачката заедница поради тоа што овозможува брза и евтина анализа и валидација на моделите во дизајн фаза, пред реални хардверски имплементации. Се предложи математички пристап за моделирање на латентноста крај-до-крај и латентноста поради процесирање во контролерот. Моделот беше нумерички евалуиран во MATLAB. Се заклучи дека доминантен фактор кој влијае на латентноста во SDN делот на јадрената мрежа претставува времето на процесирање на пакетите во контролерот. Резултатите покажуваат дека преку додавање на повеќе контролери во системот, не се афектира времето на опслужување при повисоки вредности на оптоварувањето, во споредба со случајот кога се користи помал број контролери.

Генерално, како и што се очекуваше, додавањето на повеќе контролери во мрежата влијае позитивно на перформансите.

Понатаму, се анализираа и најчестите хипотези кои редовно се користат при моделирање на SDN мрежи: неограничен бафер и рата на опслужување која е независна од оптоварувањето. Се даде предлог за нов аналитички модел за рата на опслужување која зависи од состојбата и ограничен бафер и се направи обид да се измери влијанието на овие претпоставки кога се прави споредба со стандарден математички модел кој се користи за моделирање на SDN мрежи кој не ги зема во предвид овие хипотези. За случајот на неограничен бафер, средното време на опслужување е пониско, споредено со случајот кога баферот има конечна вредност, кога средното време на опслужување се влошува побрзо. За случајот на варијабилна средна рата на опслужување, деградацијата е многу повидлива, особено ако l е пониско. Конечно, се заклучи дека и двете хипотези влијаат на точноста на моделирањето и доколку се сака да се креираат веродостојни SDN модели, мора да се земат во предвид и двете претпоставки.

Како предлог за понатамошна екстензија на предложените модели претставува насоката која се однесува на проблемот на поставување на контролер со цел минимизирање на вкупната латентност. Друга насока може да биде експанзија на анализата со користење на теоријата на мрежен калкулус или можеби вметнување на реални хардверски влијанија во аналитичкото моделирање.

6. Аналитичко моделирање на хендовер во SDN-базирани мобилни јадрени мрежи

Поради неверојатниот пораст на податочниот мобилен сообраќај, постои сериозна загриженост за тоа како да се обезбеди висок квалитет на сервис и да се задоволат барањата на корисниците. Управувањето со мобилноста на корисниците во целуларните мобилни мрежи е посебен предизвик за операторите. Целта е гарантирање на минимален прекин кога корисникот извршува процедура на хендовер. Во случај на SDN-базирани јадрени мрежи, успешната хендовер процедура се состои од размена на OF сигнални пораки помеѓу контролната и корисничката рамнина. Специфичниот случај на “тврд“ хендовер, вклучува прекин на тековната сесија и размена на реконфигурациски и пораки за управување помеѓу комутаторите и контролерот. Фокусот во оваа глава од дисертацијата, се става на хендоверот на мобилен корисник помеѓу различни комутатори, а целта е да се моделира доцнењето при хендовер поради OF контролните пораки преку користење на нов математички пристап. Од интерес е единствен контролер, но и сценарио со мулти-контролери во контролната рамнина.

Пристапот со централизиран SDN-контролер, наметнува потенцијални проблеми (како мрежна скалабилност и деградација на перформанси), што е причината за воведување на архитектурата на логички дистрибуирани контролери. Особините на различните мулти-контролер архитектури беа детално дискутирани во главата 5.

Генерално, средното време на опслужување на OF-комутаторот значително зависи од капацитетот на излезниот бафер. Овој бафер се користи за привремено складирање на влезните пакети. Пред да може комутаторот да ги опслужува пакетите (има слободен капацитет за процесирање), прво ги складира во баферот, а подоцна ги насочува кон коректната дестинација. Денес, најчестиот случај при реални имплементации на OF-комутатори е користење на единствен заеднички бафер, што значи дека истиот бафер се дели подеднакво од сите излезни порти. Овој тип бафер е евтин и значително ја намалува комплексноста во системот. Сепак, постојат предлози за користење на два посебни бафери, еден за корисничката рамнина, а другиот за контролната рамнина. Во оваа глава се дискутираат позитивните и негативните страни при споредба на комутатори со единствен делен бафер и комутатори со два посебни бафери кои користат приоритизација, во контекст на квантификација на вкупното доцнење при хендовер.

Како што веќе беше дискутирано, теоријата на редови веќе од поодамна се користи за моделирање на перформанси на системи базирани на OF архитектура бидејќи преставува брз и евтин метод на донесување заклучоци за зависности на параметри и верификација на

предлози за алгоритми или мрежни протоколи пред нивна имплементација во тестен систем. Иако постојат многу аналитички модели кои опишуваат SDN мрежи, сепак има многу малку трудови за контролна рамнина со повеќе контролери и за моделирање на комутатори кои користат конечни бафери со приоритизација.

Во оваа глава се предлагаат неколку нови аналитички пристапи кои може да се искористат за моделирање на доцнењето при хендовер поради размената на OF сигнализационски пораки во мрежи со еден или повеќе контролери. Се разгледува времето на процесирање на пакети и влијанието на клучните фактори како што се: број на комутатори, веројатност на наидување на *Packet-In* пораките, брзина на пристигнување на пакети, вкупен проток на системот итн. Резултатите покажуваат дека со растење на веројатноста на наидување на *Packet-in* пораките, перформансите во мрежата се намалуваат. Слично, како што расне бројот на контролери, бројот на синхронизационски пораки станува висок и ова влијае негативно на средното време на опслужување на контролерот. Понатаму, се моделираат и споредуваат два дизајна на бафер во комутатор: (i) заеднички бафер без приоритизација; (ii) два различни бафери со користење на приоритизација. Двата системи се споредуваат во однос на доцнењето воведено при хендовер и во однос на потребите за големина на бафер за постигнување на соодветна веројатност на губење на пакети наметната од линкот. Резултатите покажуваат дека користењето на комутатор со приоритетни бафери покажува подобри резултати во сите споредби во однос на единствен делен бафер и треба да биде претпочитаниот дизајн за бафери во комутаторите во мобилните мрежи.

6.1 Преглед на литературата

Во ова поглавје се прави краток осврт на трудови кои истражуваат предизвици во врска со управување на мобилноста во комерцијални целуларни мрежи и различните нови предлози и/или модификации/екстензии на стандардниот OF со цел намалување на вкупната латентност, со посебен осврт на намалување на доцнењето при хендовер. Во вториот дел од ова поглавје, се става акцент на преглед на придонесот на истражувачката заедница во областа на моделирање на конечни бафери во OF-комутаторите.

Во трудот [6] се анализираат главните фактори кои влијаат на целокупната латентност во SDN-базирани мобилни мрежи. Се дискутира дека латентноста може да се подели во две главни групи: доцнење при процесирање и трансмисионо доцнење. Доцнењето при процесирање се однесува на средното време на опслужување во рамки на комутаторот и контролерот, додека трансмисионото доцнење го опишува времето кое е

потребно за податоците да бидат трансмитирани помеѓу комутаторите и контролерот. Се истражува влијанието на бројот на делници, оптоварувањето на контролерот и баферирањето. Во [82] се проучува средното доцнење на пакети и се дефинираат два типа на доцнење: (i) доцнење воведено од комутаторот во случај на потреба за препраќање на пакет кон контролерот; (ii) доцнење кое се појавува после одлуката за хендовер поради тоа што и двата инволвирани комутатори мораат да бидат во комуникација со контролерот за успешен хендовер. Се предлага детална аналитичка рамка за моделирање на пакетското доцнење во OF-базирани мобилни мрежи со единствен контролер. Сепак, од интерес е само моделот на единствен јазол (на англ. *single-node model*) и размената на сигнални пораки помеѓу комутаторите и контролерот не е во фокусот на истражувањето.

Понатаму, се разгледуваат неодамна предложени OF-базирани процедури за овозможување на неосетен хендовер во мобилни мрежи со различни радио пристапни делови. Во трудот [7], се анализира влијанието на SDN врз управувањето со мобилноста и како мобилниот уред може да се движи и да ја користи SDN мрежата. Се дизајнира решение базирано на OF протоколот која има неколку предности: тоа е решение за мобилност без партиципација на мобилниот уред, лесно се имплементира и користи, ја избегнува потребата за тунелирање и дополнителен сообраќај заради сигнализација и го минимизира губењето на пакети и доцнењето при хендовер. Од друга страна, во [83] се предлага 5G оптимизациска рамка на процесот на мобилност каде SDN достигнува до крајниот уред. Се прави модификација и екстензија на стандардниот OF протокол со цел да се додаде можност за управување со мобилноста за да може мобилниот терминал да обезбеди информација кон мрежата за радио условите, додека контролерот е засилен со можноста да ги наметне потребните пораки назад кон мобилниот уред. MN користи *Packet-in* порака за да индицира кон контролерот пад на квалитетот на сигналот и кога оваа порака ќе стигне до контролерот, мрежата реагира на примената информација и носи одлука за хендовер, што значи префрлање на MN од *Access Point 1* (AP1) кон AP2 преку праќање на *Flow-mod* порака. Пристапот всушност претставува рамка за “креирање-пред-распуштање” (на англ. *make-before-break*) што се прави преку првата *Flow-mod* порака преку превентивно имплементирање на правило кое му овозможува на MN да го насочи пакетот кон AP2, додека втората *Flow-mod* порака ги менува изворната и дестинациската *Medium Access Control* (MAC) адреса за да се комплетира процедурата. Предложениот концепт се валидира со прототип имплементиран на физички тестен систем, при што се докажува дека процедурата на хендовер може целосно да се направи во рамки на OF и без користење на било какви додатни протоколи. Во трудот [84] се комбинираат скоро воведениот концепт

на *Distributed Mobility Management* (DMM) со SDN, поради тоа што имаат слични принципи. Се предлага ново решение кое овозможува независност од основните технологии на подолните нивоа (на англ. *underlying technologies*), поддршка на мобилност по тек и нема потреба за развивање на било каков модул за мобилност во пристапниот рутер. Интелигенцијата за управување со мобилноста е всушност апликација која се извршува на северниот интерфејс на SDNC и со овој пристап се добиваат експериментални резултати од реални имплементации со исти перформанси како за DMM, но со високо намалена комплексност. Сличен пристап е користен и во [85] каде се користат DMM концептите за имплементација на брза, скалабилна и расположива шема за управување со мобилноста. Предлогот се валидира со помош на лабораториски прототип преку имплементација на механизмот на OF-контролирана IEEE 802.11 мрежа.

Трудот [86] се интересира за квантификација на доцнењето при хендовер поради размена на OF сигнализацијата пораки. Се моделираат *Packet-in* и *Port-status* пораки и се дефинира соодветна перформансна метрика за да се евалуира вкупното доцнење што го доживува MN во случај на “тврдо” хендовер. Понатаму, се предлага LTE архитектура и преку симулации се врши споредба со постојните решенија. Најголемиот недостаток на овој предлог е фактот што се моделира само единствен контролер без да се истражи влијанието што потенцијално би го имала дистрибуираната контролна рамнина. Уште повеќе, во нумеричките резултати се презентираат заклучоци кои всушност се добиени од симулациите и се во врска со предложеното LTE решение, а не се става фокус на детална валидација на предложениот аналитички модел. Најголемиот придонес на [87] е да отвори дискусија за тоа како SDN технологијата може ефикасно да се искористи во мобилните мрежи за управување со мобилноста. Се анализираат најважните фактори поврзани со управување на мобилноста, како што се зачувување на континуитетот на сесијата и скалабилност при хендовер во густо (на англ. *dense*) мобилни мрежи. Презентираните концепти за управување со мобилноста може да се искористат за вклучување на функционалноста на LTE *Self-Organizing Network* (SON), но за да се забрза имплементацијата на предложените концепти, сепак мора да се дефинираат додатни SDN протоколи.

Најголемиот број на референци дискутирани во оваа глава и во главата 5, земаат во предвид единствен делен бафер во комутаторот кој се користи и за контролната и за корисничката рамнина. Иако моделот на заеднички бафер ја поедноставува анализата, воопшто не е реален бидејќи постои имплицитна претпоставка дека сообраќајот насочен кон контролерот се враќа назад кон комутаторот и се третира на ист начин како и новите

пакети кои доаѓаат на влез на комутаторот. Ако комутаторот има конечни бафери [88], тогаш постојат сериозни забелешки околу зачувувањето на Марковите особини на индивидуалните редови што е потребен предуслов за добивање на анализа во продуктна форма (на англ. *product form analysis*) [89]. Стационарното однесување на ваква мрежа не може да се моделира преку користење на анализа во продуктна форма, туку стационарното однесување во овој случај може да се моделира со користење на глобални равенки за рамнотежа [90].

Аналитичките модели предложени во [68] го моделираат комутаторот како $M/M/1$, а контролерот како $M/M/1/K$ ред, каде комутаторот и контролерот се моделирани да оперираат независно. Поради последната претпоставка и фактот што баферите во комутаторот се бесконечни, задоволени се Марковите особини на редот и ова резултира во анализа во продуктна форма. Во трудот [64] не се зема во предвид дека контролерот и комутаторите се независни. Наместо тоа, сообраќајот од контролерот се меша со влезните пакети и ова подобро го моделира однесувањето во SDN, но по цена на зголемена комплексност. Во [91] се моделира тандемска мрежа во два стадиуми, каде комутаторот е моделиран како $M/M/1$ ред, но моделот не зема во предвид раздвојување помеѓу контролниот и корисничкиот сообраќај. Во [92] преку експерименти јасно се покажува дека $M/M/1$ редот не е добар модел за OF-комутатори, поради што се користи $M/G/1$ модел. Многу слично, во [93] и контролерот и комутаторите се моделираат како $MMPP/1/1$, а во трудот [94] се користи $M/Geo/1$ модел, каде времето на опслужување е дистрибуирано геометриски, но повторно не се зема во предвид засебно разгледување на различните типови на пакети.

Од друга страна, во трудот [95], се предлага аналитички модел за комутаторите преку користење на бесконечни бафери и преемтивни редови со приоритизација. Ова можеби е првата анализа која прави споредба на различни бафер модели (приоритетен наспроти заеднички баферинг) и резултатите на анализата се во корист на бафер моделите кои користат приоритетизација. Димензионирањето на баферот воопшто не е предмет на анализа, што е најголем пропуст на ова истражување. Моделот е проширен во [96], каде се претпоставува сценарио со *bursty* мултимедијален сообраќај и се користат конечни бафери за ниско-приоритетниот ред и бесконечни бафери за високо-приоритетниот ред. Во трудовите [97] и [98] се користат редови со приоритет со конечни бафери за двата реда, преку имплементирање на $GI/M/1/K$ ред и преку користење на QBD процесот. Се предлага модел со не-преемтивен ред, што значи дека пакетите во ниско-приоритетниот бафер се опслужуваат од комутаторот само тогаш кога нема пакети во високо-приоритетниот ред.

Овој тип на моделирање е многу пореалистичен, но значително ја зголемува комплексноста и не дозволува решенија кои се продуктно ориентирани [99]. За среќа, стационарното однесување на овие мрежи може да биде моделирано со користење на равенките за глобална рамнотежа [100].

6.2 Опис на проблемот и научен придонес

Во оваа глава, се предлагаат неколку нови аналитички модели со кои се моделира размената на OF сигнални пораки при процедурата на хендовер, кога MN се движи од еден кон друг комутатор во SDN-базирана мобилна мрежа. Основната цел е одредување на доцнењето кое го доживува MN, но и идентификација на клучните фактори кои влијаат на ова доцнење, како и димензионирање на баферот во комутаторот. Во сите разгледувани сценарија, се претпоставува дека MN извршува “тврдо” хендовер. Научниот придонес во оваа глава може да се сумира во следните точки:

- 1) Се предлага аналитички пристап за моделирање на перформансите на комутаторите и единствен контролер базиран на теоријата на редови. Комутаторите и контролерот се моделираат како M/M/1 ред и се моделираат само *Packet-in* пораките. Врз основа на понудениот математички модел, се врши нумеричка валидација со помош на Wolfram Mathematica. Добиените резултати укажуваат дека бројот на комутатори, ратата на пристигнување, веројатноста на наидување на *Packet-in* пораките и вкупниот системски проток, се критични фактори при моделирањето на SDN-базирани мрежи.
- 2) Се предлага нов математички пристап за моделирање на перформансите во SDN мрежи со повеќе комутатори, но и повеќе контролери во контролната рамнина. Комутаторите се моделирани како M/M/1 ред, додека контролерите се моделирани како M/PH/1 ред. Се прави експанзија на претходните трудови со тоа што не се моделираат само *Packet-in*, туку и *Ports-status* пораките. *Port-status* пораките се од клучен интерес кога MN изведува “тврдо” хендовер и претставуваат први пораки кои се разменуваат во контролната рамнина при иницијализација на процедурата на хендовер. Предложениот модел се валидира со споредба на нумеричките резултати добиени со помош на Wolfram Mathematica и симулациските резултати добиени со користење на MATLAB. Заклучоците може да се искористат при дизајнирање на гаранции за доцнењето при хендовер во мобилна мрежа за одреден сообраќај кој се насочува кон

комутаторот. Дополнително, се анализираат и другите стандардни фактори: број на комутатори и контролери, рати на пристигање на *Packet-in* и синхронизациските пораки, системски проток итн.

- 3) Се предлага аналитички модел кој ги моделира комутаторите користејќи GI/M/1/K ред и M/M/1 за контролерот. Повторно се моделираат и *Packet-in* и *Port-status* пораки. Се конструираат и се споредуваат математички модели за два типа на комутатори: (i) кога се користи единствен делен бафер без приоритизација и (ii) кога се користат два посебни бафери со не-преемтивна приоритизација. Се прави валидација на предложените модели, преку споредба на нумеричките резултати генерирани во MATLAB и симулациските резултати добиени со помош на Monte Carlo. Двата системи се споредуваат во однос на доцнењето при хендовер, како и за потребата на големина на бафер за да се постигне соодветна рата на пакетска загуба на линк.
- 4) Во контраст на скорешните трудови кои се базираат на OF модификации/експанзии заради овозможување механизам за управување со мобилноста, во оваа глава се предлага чист математички пристап кој може многу лесно да се примени и прошири, а обезбедува брз и многу корисен увид во есенцијалните параметарски зависности.

6.3 Моделирање на доцнењето при хендовер со користење единствен контролер

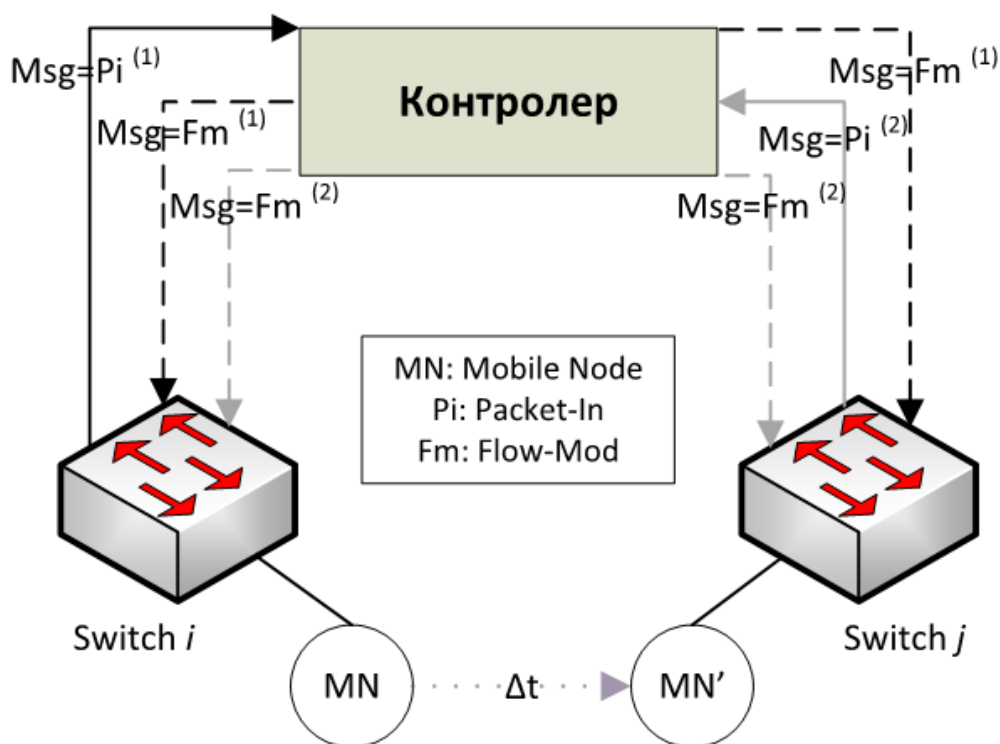
Во ова поглавје се предлага нов аналитички модел за SDN-базирана јадрена мрежа која има само еден контролер. Најпрво се дава опис на системот и препоставките кои се користат, а потоа се преминува на математичко моделирање со цел наоѓање на зависноста помеѓу времето на процесирање во контролерот и вкупниот системски проток.

6.3.1 Аналитички модел

Од интерес е моделирање на времето на процесирање во контролерот во мрежа која се состои од n комутатори, контролирани од единствен контролер. Генерално, кога пакетот ќе стигне на влезот на интерфејсот во комутаторот, се прави проверка за совпаѓање со постојните записи на табелата на тек, доколку не се најде постоен запис, пакетот се препраќа кон контролерот. Слично, кога MN се конектира или дисконектира од комутатор, праќа *Packet-in* порака. OF вклучува и други асинхрониски пораки, како *Port-status*, *Flow-*

mod/removed и *Error*, но во ова поглавје од интерес се само *Packet-in* пораките бидејќи тие се доминантните пораки во една мобилна мрежа.

Комутаторите и контролерот се моделираат како единствен ред, а не по интерфејс. Се претпоставува дека сите комутатори имаат иста средна рата на опслужување и сите елементи на системот имаат неограничен бафер и рати на услуги кои се независни од оптоварувањето. Пристигањето на пакетите следи Поасонова распределба, а и комутаторите и контролерот се моделирани како M/M/1 ред. Дестинациите на пакетски тек се униформно дистрибуирани и има отворена мрежа на редови, каде секој пакет може да влезе или излезе од системот од било кој јазол. Дисциплина на опслужување е FIFO. Се претпоставува дека при секоја *Packet-in* порака, кон контролерот се праќа само заглавието на првиот пакет од текот, а после тоа сите следни пакети се насочуваат директно без интервенција на контролерот. Исто така, се претпоставува дека контролерот одговара со *Flow-mod* (инсталирај нов тек) порака кон комутаторот. Конечно, само контролерот има целосен увид на мрежната топологија и секогаш ги конфигурира комутаторите на најоптимален начин.



Слика 6.1. Размена на OF пораки при хендовер (моделирање само на *Packet-in* пораки)

Сценариото на OF хендовер, беше детално опишано во главата 3.4. На сликата 6.1 е прикажана размената на OF пораките при хендовер на MN. Кога MN ќе сака да се дисконектира од комутаторот *i* и да се конектира на комутаторот *j*, ова тригерира две *Packet-in* пораки кои се праќаат кон контролерот за да го информираат за мобилноста на MN. Кога

ќе ги прими овие пораки, контролерот праќа *Flow-mod* кон некои од комутаторите со цел да ги променат записите во табелите на тек за тој MN. Размената на пораки иницирана од мобилноста на MN е прикажана на сликата 6.1, каде “(1)” ја означува пораката иницирана од комутаторот i , а “(2)” пораката иницирана од комутаторот j .

Се дефинира генерална рата на пристигање на пакети во комутаторот i како λ_i , q_i^{pc} е веројатноста на наидување на *Packet-in* пораките кон контролерот, c_{ij} е веројатноста за рутирање помеѓу комутаторите i и j , а $b_{ji} \in \{0,1\}$ е индикатор кој опишува дали освежувањето од комутаторот j го вклучува, или не го вклучува комутаторот i на пакетската патека. Вкупната рата на пристигнување на пакети Λ_i може да се претстави како:

$$\Lambda_i = \lambda_i + \sum_{j=1, j \neq i}^n \lambda_j \times c_{ij} + q_i^{pc} \lambda_i + \sum_{j=1, j \neq i}^n (q_j^{pc} \times b_{ji}) \lambda_j. \quad (6.1)$$

Ратата на услуга на комутаторот i е μ_i и се користи теоремата на Џексон (на англ. Jackson theorem) за отворени мрежи на редови. Оптоварувањето на комутаторот i е дадено како:

$$\rho_i = \frac{\Lambda_i}{\mu_i}. \quad (6.2)$$

Бројот на пакети и времето на процесирање во комутаторот i се означуваат со N_i и T_i , соодветно. Средната вредност $E[.]$ за M/M/1 редот е дадена во [101] со:

$$E(N_i) = \frac{\rho_i}{1 - \rho_i}, \quad (6.3)$$

$$E(T_i) = \frac{E(N_i)}{\Lambda_i}. \quad (6.4)$$

Ако се заменат равенките (6.2) и (6.3) во (6.4), се добива:

$$E(T_i) = \frac{1}{\mu_i - \Lambda_i}. \quad (6.5)$$

Времето на процесирање во комутаторот може да се претстави како:

$$\begin{aligned} E(T_s) &= \sum_{i=1}^n \frac{\Lambda_i}{\sum_{i=0}^n \Lambda_i} E(T_i) \\ &= \sum_{i=1}^n \frac{\Lambda_i}{\sum_{i=0}^n \Lambda_i} * \frac{1}{\mu_i - \Lambda_i}. \end{aligned} \quad (6.6)$$

При позитивна проверка за совпаѓање на пакетот во табелата на тек, комутаторот ги насочува пакетите на патеката на тек, инаку мора да се контактира контролерот. Веројатноста на наидување на *Packet-in* пораки кон контролерот е q_i^{pc} . Вкупната рата на *Packet-in* пораки пратени кон контролерот од вкупно n комутатори е:

$$\Lambda_c = \sum_{i=1}^n q_i^{pc} \lambda_i. \quad (6.7)$$

Како што претходно беше нагласено, процесирањето на *Packet-in* пораките во контролерот може да се окарактеризира како M/M/1 модел на ред. Ако p_i е веројатноста за i *Packet-in* пораки во контролерот, тогаш средната вредност на овие пораки е:

$$\begin{aligned} E(N_c) &= \sum_{i=0}^{\infty} i p_i \\ &= \sum_{i=0}^{\infty} i \left(\frac{\Lambda_c}{\mu_c}\right)^i \left(1 - \frac{\Lambda_c}{\mu_c}\right) \\ &= \frac{\Lambda_c}{\mu_c - \Lambda_c}. \end{aligned} \quad (6.8)$$

Времето на процесирање во контролерот може да се претстави како:

$$E(T_c) = \frac{E(N_c)}{\Lambda_c}, \quad (6.9)$$

$$E(T_c) = \frac{1}{\mu_c - \Lambda_c}. \quad (6.10)$$

Ако се замени равенката (6.7) во равенката (6.10), се добива:

$$E(T_c) = \frac{1}{\mu_c - \sum_{i=1}^n q_i^{pc} \lambda_i}. \quad (6.11)$$

Во однос на изборот на перформансна метрика, целта е да се квантификува доцнењето при хендовер кое настанува поради размената на OF пораки при самата процедура. За *Packet-in* настан, средното време на интеракција помеѓу комутаторот i и контролерот c може да се претстави како:

$$E[T_i^{tot}] = E[T_i] + E[T_c] + E[T_{i_prop}^c] + E[T_{i_prop}^{c_max}], \quad (6.12)$$

каде T_i and T_c се соодветните времиња на опслужување на комутаторот i и контролерот во доменот, $T_{i_prop}^c$ е пропагациското време на *Packet-in* пораката од комутаторот i кон контролерот и $T_{i_prop}^{c_max}$ е максималното пропагациско доцнење (потребно да се достигне до најдалечниот комутатор) за *Flow-mod* пораката пратена кон одреден број комутатори како реакција на контролерот кон примената *Packet-in* порака. Вкупното доцнење при хендовер T_{ho} кое го доживува MN е сума од доцнењата воведени од двете *Packet-in* пораки пратени од комутаторот i и комутаторот j :

$$E[T_{ho}] = E[T_i^{tot}] + E[T_j^{tot}], \quad (6.13)$$

$$E[T_{ho}] = 2 \left(\sum_{i=1}^n \frac{\Lambda_i}{\sum_{i=0}^n \Lambda_i} * \frac{1}{\mu_i - \Lambda_i} + \frac{1}{\mu_c - \sum_{i=1}^n q_i^{pc} \lambda_i} \right) + E[T_{prop}^{tot}], \quad (6.14)$$

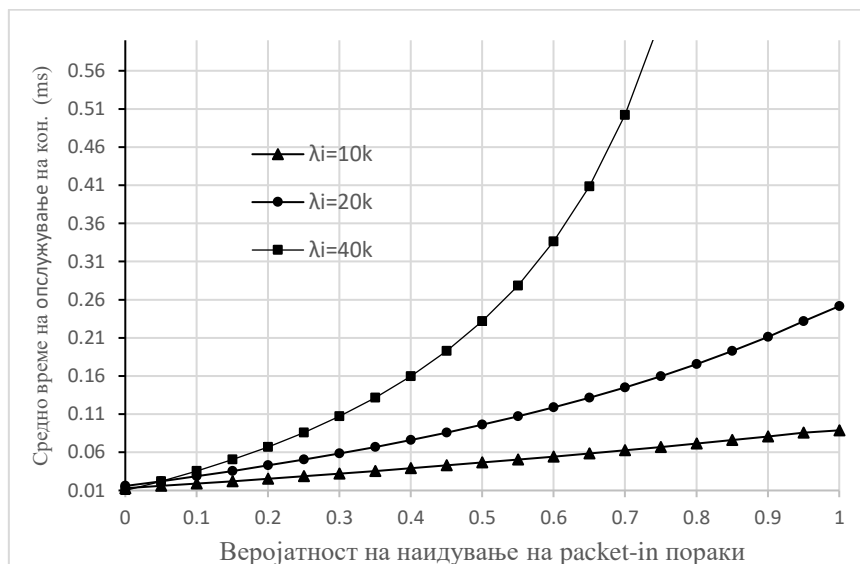
каде

$$E[T_{prop}^{tot}] = E[T_{i_prop}^c] + E[T_{i_prop}^{c_max}] + E[T_{j_prop}^c] + E[T_{j_prop}^{c_max}]. \quad (6.15)$$

Во пракса, процесирачкото доцнење во комутаторот е многу помало од процесирачкото доцнење во контролерот, како што се покажува во следното поглавје. Пропагациското време на пораките треба да се земе во предвид за мобилни мрежи поради големите растојанија.

6.3.2 Анализа на перформанси

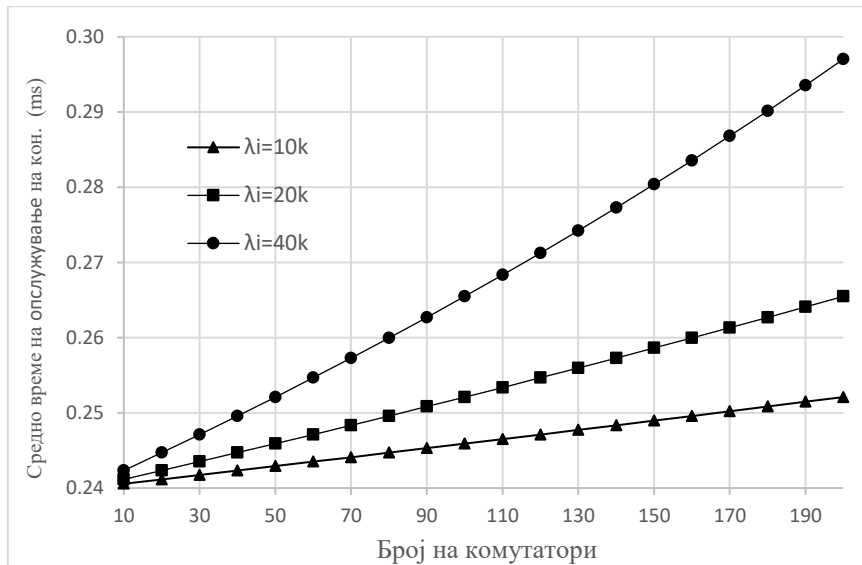
Во ова поглавје се валидира предложениот аналитички модел преку нумеричка анализа направена во Wolfram Mathematica 12.0. За q^{pc} се користи вредност од 4%, $\mu_s=9.8 \mu s$, $\mu_c=240 \mu s$ [64] и се претпоставува дека пакетите пристигнуваат во сите комутатори со иста рата од λ_i . Просечното растојание помеѓу комутаторите и контролерот е 100 km (се моделира мал мобилен оператор), а линковите помеѓу комутаторите и контролерот се 1 Gbit/s. Бројот на комутатори е $n = 10$, а просечната големина на пакет е 128 B.



Слика 6.2. Средно време на опслужување во комутаторите наспроти веројатноста на наидување на *Packet-in* пораките

Сликата 6.2 го прикажува средното време на опслужување во комутаторите како функција од q^{pc} . Вредноста $q^{pc} = 0$ означува дека комутаторот може да ги препраќа сите влезни пакети без да го контактира контролерот, додека $q^{pc} = 1$ означува дека целиот сообраќај мора да помине низ контролерот. Се забележува дека средното време на опслужување се зголемува пропорционално со зголемување на веројатноста на наидување на *Packet-in* пораките. Доколку пакетите пристигаат со пониска рата, порастот на времето на опслужување е поспор, за $\lambda_i=10k$ и $\lambda_i=20k$ порастот е скоро линеарен. Но, за рати на пристигање на пакети од $\lambda_i=40k$, времето на опслужување на комутаторот се зголемува експоненцијално и станува многу високо при високи вредности на q^{pc} . Може да се забележи дека за $\lambda_i=40k$, зголемувањето на средното време на опслужување е околу 10 пати кога се

споредува случајот на $q^{pc} = 0$ со $q^{pc} = 0.3$. Како што е докажано во [64], во нормална SDN мрежа се очекува веројатноста на наидување на *Packet-in* пораки да биде 0.04. Ова значи дека во нормални услови, OF нема да го деградира времето на опслужување на комутаторите.

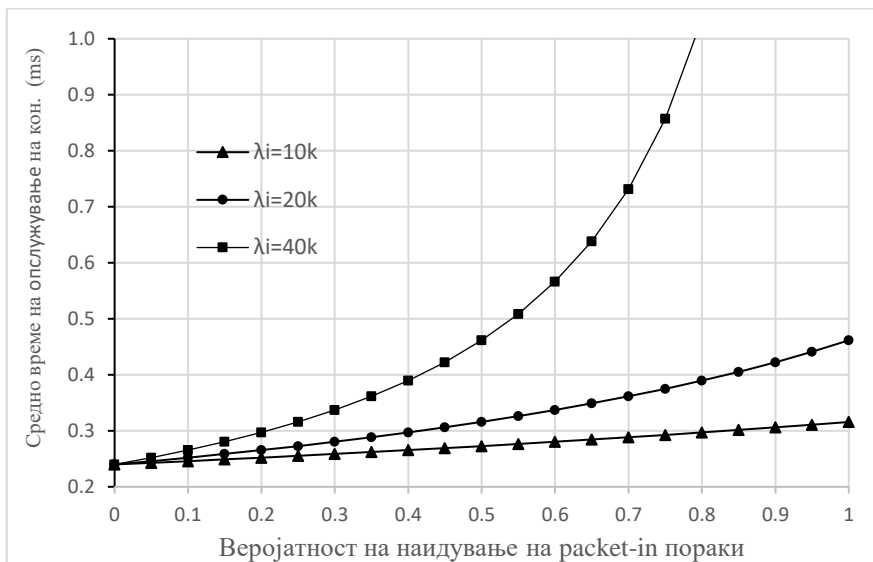


Слика 6.3. Средно време на опслужување на контролерот наспроти број на комутатори

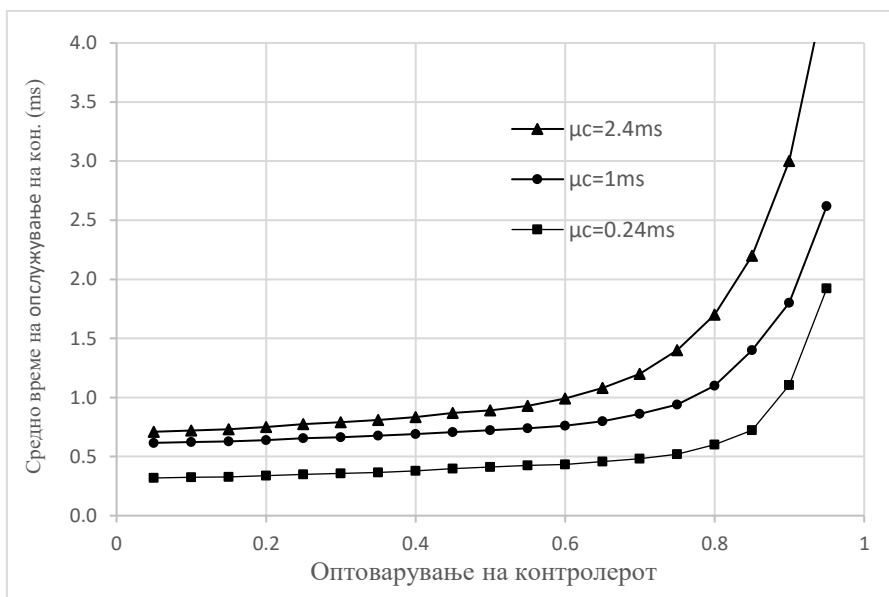
Следно, се истражува дали додавањето на комутатори во мрежата ќе предизвика деградација на времето на опслужување на контролерот. Додавањето на комутатори во мрежата би значело повеќе барања пратени кон контролерот, поради тоа што новите комутатори немаат инсталирано записи во табелите на тек. На сликата 6.3 се анализира зависноста на времето на опслужување на контролерот од бројот на комутатори, при промена на ратата на пристигнување на пакети во комутаторите. За веројатноста на наидување на *Packet-in* пораките се зема вредност од 0.04. Како што бројот на комутатори се зголемува, времето на опслужување на контролерот исто така расне. Сепак, може да се забележи дека времето на опслужување се зголемува многу бавно како што бројот на комутатори станува повисок. Кога пакетите пристигнуваат со рата од 40 Kpkt/s, времето на опслужување на контролер кој е поврзан со 100 комутатори е само околу 0.22 ms повисок споредено со случајот кога контролерот е поврзан со 10 комутатори. Дополнително, се забележува дека времето на опслужување се влошува побрзо како што расне ратата на пристигање на пакети.

Сликата 6.4 го прикажува средното време на опслужување на контролерот и зависноста од веројатноста на наидување на *Packet-in* пораките. Се зема $n=10$ и сите комутатори примаат λ_i пакети во секунда. Начинот на кој q^{pc} влијае на перформансите е многу сличен на начинот на кој истиот влијае на комутаторите. Кога пакетите пристигнуваат со пониски рати ($\lambda_i=10k$ и $\lambda_i=20k$), веројатноста на наидување на *Packet-in*

пораките не е фактор кој влијае на перформансите во мрежата. Но, кога пакетите пристигаат со повисока рата ($\lambda_i=40k$), тогаш времето на опслужување се зголемува многу брзо за повисоките вредности на q^{pc} .



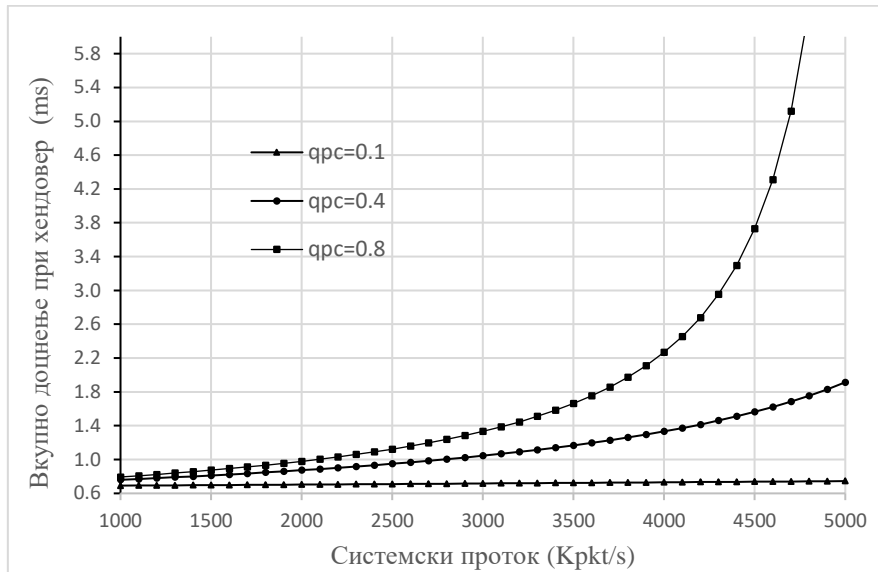
Слика 6.4. Средно време на опслужување на контролерот наспроти веројатност на наидување на Packet-in пораки



Слика 6.5. Средно време на опслужување на контролерот наспроти оптоварувањето, ефект на μ_c за фиксна вредност на q^{pc}

Понатаму, се истражува влијанието на средното време на опслужување на контролерот врз просечното време на опслужување на контролерот, при константно $q^{pc} = 0.04$. Резултатите се прикажани на сликата 6.5. Како што се очекуваше, за повисоки вредности на μ_c , времето на опслужување на контролерот се зголемува. Графикот може да се искористи за поставување на гаранции на времето на опслужување на контролерот кога се дизајнира мрежа со контролер кој има фиксно просечно време на опслужување и има

ограничено оптоварување. При оптеретувања од 0.7 до 0.8, времето на опслужување нагло се зголемува и станува неприфатливо високо за високите рангови на оптоварувањето.



Слика 6.6. Вкупно доцнење при хендовер наспроти системскиот проток

На крајот, сликата 6.6 го прикажува вкупното доцнење при хендовер како функција од системскиот проток. Системскиот проток е вкупниот сообраќај кој го носи мрежата. Параметарот q^{pc} има три различни вредности, поради што на графикот има три криви. Графикот покажува дека за $q^{pc} = 0.1$ и $q^{pc} = 0.4$, иницијално латентноста при хендовер се зголемува скоро линеарно како што се зголемува протокот на системот. Но, за $q^{pc} = 0.8$, доцнењето при хендовер прво умерено се зголемува се до вредности на системскиот проток од околу 3500 Kpkt/s, но потоа наеднаш нагло се зголемува иако протокот расне незначително. Оваа анализа може да биде корисна при дизајнирање на гаранции за доцнење при хендовер во мобилна мрежа со одредена вредност на сообраќај кој се насочува кон контролерот. На пример, за $q^{pc}=0.4$ и мрежен проток од 4250 Kpkt/s, доцнењето при хендовер поради OF сигнализацијата е 1,45 ms.

6.4 Моделирање на доцнењето при хендовер со користење дистрибуирана контролна рамнина

Во ова поглавје се предлага аналитички модел кој може да се користи за квантификација на доцнењето при хендовер во SDN-базирани мобилни мрежи. За разлика од моделот предложен во главата 6.3, овде се прави проширување преку моделирање и на *Port-status* пораките, како и користење на дистрибуирана контролна рамнина во која сите контролери се на исто ниво и го опслужуваат само соодветниот домен од комутатори. Повторно од интерес е вкупното време на доцнење при хендовер, а главен конституент на

ова доцнење е т.н. *packet sojourn time* на контролерот, што всушност претставува вкупното време што пакетот го поминува во контролерот, вклучувајќи го и времето додека е услужен.

6.4.1 Аналитички модел

При моделирањето, се користат истите претпоставки од глава 6.3.1, со основна разлика што сега, кога MN се конектира или дисконектира од комутатор, праќа *Port-status* порака. Значи, овој пат се моделираат *Packet-in* и *Port-status* пораки што е во ред бидејќи овие се апсолутно доминантни во однос на бројноста споредено со сите други OF пораки.

Секој контролер е одговорен за одлуките за препраќање на сите комутатори кои припаѓаат во доменот на контролерот. Не постои хиерархиска структура, што значи секој контролер нема информација и пристап кон сите комутатори, наместо тоа разменува информација на промените во мрежата со другите контролери (синхронизациски или пораки на инвоцирање (на англ. *invocation*)). Како што расне бројот на контролери, вкупниот сообраќај за кој е одговорен секој контролер се намалува, но сигнализацијата помеѓу контролери станува повисока. Кога се инсталира нов тек во комутаторот, ако сите комутатори на патот на пакетот припаѓаат во доменот на ист контролер, тогаш може само еден контролер да ги инсталира сите записи во табелите на тек во сите комутатори. Во спротивниот случај, еден контролер може да инсталира записи на тек само во комутаторите кои тој ги контролира и дополнително мора да ги инвоцира останатите контролери за да ги инсталираат записите на тек во своите домени. Поради ова, се предлага модел на контролер кој има две различни типови на задачи (на англ. *jobs*) кои контролерот ги опслужува со различни рати (единствен контролер наспроти мулти-контролери).

Во однос на претпоставките од главата 6.3.1, дополнително се претпоставува дека бројот на *off-port* пораки е ист со бројот на *on-port* пораки (MN мора да се дисконектира од еден комутатор и вендаш да се конектира на другиот комутатор, со цел хендовер). Оваа претпоставка е разумна затоа што бројот на нови MN кои се приклучуваат на мрежата е значајно помал од бројот на активни MN кои се обидуваат да направат хендовер поради нивната мобилност.

Сценариото од интерес е детално опишано во главата 3.4, а размената на OF пораките при хендовер е прикажана на сликата 3.6. Секвенцата на размена на пораки беше опишана таму и нема да се повторува овде. Најпрво се разгледува само група од n комутатори кои се целосно под контрола на еден контролер, а подоцна моделот се проширува преку воведување на влијанието на другите контролери од останатите домени. Се моделираат *Port-status* и *Packet-in* пораки.

Се започнува со моделирање на *Port-status* пораките. Ако Λ_{is} е вкупната рата на пристигнување на модификации на порта од мрежата кон комутаторот i , тогаш може да биде моделирана како:

$$\Lambda_{is} = \lambda_{is} + \sum_{j=1, j \neq i}^n \lambda_{js} \times a_{ij}, \quad (6.16)$$

каде λ_{is} и λ_{js} се ратите на пристигнување на *Port-status* пораките иницирани од комутаторот i и j , соодветно, $a_{ij} = 1$ во случајот кога табелата на тек на s_i треба да се освежи поради s_j *Port-status* пораката, инаку $a_{ij} = 0$. За контролерот, вкупната рата на пристигнување на *Port-status* пораките е следната:

$$\Lambda_{cs} = \sum_{i=1}^n \lambda_{is}. \quad (6.17)$$

Од друга страна, *Packet-in* пораките се третираат различно од *Port-status* пораките, па предложената мрежа со редови содржи две *job* класи и станува збор за мулти-класна мрежа. Се дефинира генерална рата на пристигнување на пакети во комутаторот i како λ_{ip} , q_i^{pc} е веројатноста на наидување на *Packet-in* пораките, c_{ij} е веројатноста на рутирање помеѓу комутаторите i и j , а $b_{ji} \in \{0,1\}$ е индикатор кој опишува дали освежувањето од комутаторот j го вклучува комутаторот i на патеката на тек, или не. Вкупната рата на пристигнување на пакети може да се претстави како:

$$\Lambda_{ip} = \lambda_{ip} + \sum_{j=1, j \neq i}^n \lambda_{jp} \times c_{ij} + q_i^{pc} \lambda_{ip} + \sum_{j=1, j \neq i}^n (q_j^{pc} \times b_{ji}) \lambda_{jp}, \quad (6.18)$$

а ратата на пристигнување на *Packet-in* пораките во контролерот е следната:

$$\Lambda_{cp} = \sum_{i=1}^n q_i^{pc} \lambda_{ip}. \quad (6.19)$$

Се земаат ратите на пристигнување дефинирани во равенките (6.16) до (6.19) и се заменуваат во следните равенки:

$$\rho_{is} = \frac{\Lambda_{is} + \Lambda_{ip}}{\mu_{is}}, \quad \rho_c = \frac{\Lambda_{cs} + \Lambda_{cp}}{\mu_c}, \quad (6.20)$$

каде ρ_{is} and ρ_c се оптоварувањето, а μ_{is} и μ_c се средните рати на опслужување на комутаторот и контролерот. Конечно, се израмнуваат ρ_{is} и ρ_c при оптоварување ~ 1 и се пресметуваат вредностите на λ_{is} and λ_{ip} . Во пракса, λ_{is} се очекува да биде многу помало од λ_{ip} .

Понатаму, моделот се проширува преку воведување на влијанието на другите контролери од останатите домени. Се зема во предвид нов модел на ред за контролерот со цел да се пресмета времето на опслужување. За апроксимација на времето на опслужување,

се користи *Phase-Type* (PH) дистрибуција [102], која се опишува со помош на вектор-матрица (α, T) , каде:

$$\alpha = (\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_n), \quad T = \begin{pmatrix} t_{11} & \dots & t_{1c} \\ \vdots & \ddots & \vdots \\ t_{c1} & \dots & t_{cc} \end{pmatrix}. \quad (6.21)$$

Квадратната матрица T има не-негативни не-дијагонални елементи и негативни дијагонални елементи. Средната вредност на PH дистрибуцијата е дадена со:

$$E[X] = \alpha(-T)^{-1}e = \frac{1}{\mu_c}, \quad (6.22)$$

каде e е вектор колона од единици со соодветна големина, μ_c е просечната рата на опслужување на контролерот. Искористувањето е $\rho = \lambda_c / \mu_c$, каде λ_c е вкупната рата на пристигнување во контролерот.

Вкупната рата на пристигнување на сите *Packet-in* пораки кон контролерот е λ_p , со c контролери во мрежата, *Packet-in* пораките кај секој контролер пристигнуваат со рата $\lambda_{cp} = \lambda_p/c$. Ратата на пристигнување на инвокации е λ_{ci} и за единствен контролер постојат $(c-1)$ контролери кои праќаат инвокациски пораки со рата од $\lambda_p/(c-1)$. Се добива:

$$\lambda_{ci} = \frac{(c-1)\lambda_p}{c^2}. \quad (6.23)$$

Ратата на пристигнување на синхронизациските пораки во еден контролер се означува со λ_e , така да вкупната рата на размена на пораки λ_{ce} е:

$$\lambda_{ce} = (c-1)\lambda_e. \quad (6.24)$$

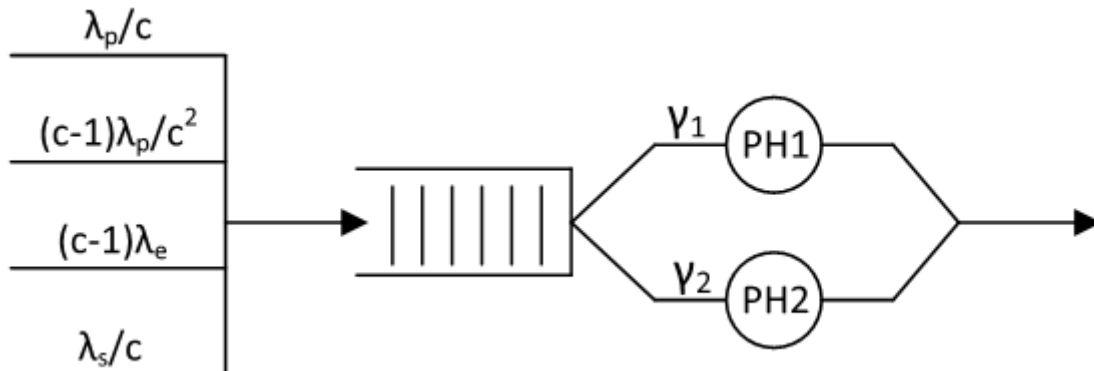
Со λ_{cs} се означува вкупната рата на пристигнување на *Port-status* пораките, за единствен контролер се добива:

$$\lambda_{cs} = \frac{\lambda_s}{c}. \quad (6.25)$$

Конечно, вкупната рата на пристигнување во контролерот може да биде дадена со следната равенка.

$$\lambda_c = \lambda_{ce} + \lambda_{ci} + \lambda_{cp} + \lambda_{cs}. \quad (6.26)$$

Системот на редови е презентираан на сликата 6.7.



Слика 6.7. Модел на редот во контролерот

Ако контролерот може сам да го опслужи барањето, тогаш на него поминува време $t_1 \sim PH(\alpha_1, T_1)$, инаку поминува време $t_2 \sim PH(\alpha_2, T_2)$. Се дефинира γ_1 како веројатност дека контролерот поминува време t_1 на еден *job*, а $\gamma_2 = 1 - \gamma_1$ е веројатноста контролерот да поминува време t_2 на *job*, каде:

$$\gamma_1 = \frac{c^2(n-1)\lambda_e + c(\lambda_p + \lambda_s)}{c^2(n-1)\lambda_e + c(\lambda_p + \lambda_s) + (c-1)(\lambda_p + \lambda_s)}. \quad (6.27)$$

Времето на опслужување се означува со нотацијата $PH(\alpha, T)$. Дистрибуцијата на времето на опслужување на контролерот е комбинација од $PH(\alpha_1, T_1)$ и $PH(\alpha_2, T_2)$, каде:

$$\alpha = (\gamma_1 \alpha_1, \gamma_2 \alpha_2), \quad T = \begin{pmatrix} T_1 & 0 \\ 0 & T_2 \end{pmatrix}. \quad (6.28)$$

M/PH/1 редот може да се проучува како QBD процес со простори на состојба $E = \{0, (i, j), i \geq 1, 1 \leq j \leq v\}$. Состојбата 0 го претставува случајот на празен ред, состојбата (i, j) да има i муштерии во системот и процесот на опслужување да биде во фаза j . Генераторот Q на тој Марков процес е следниот:

$$Q = \begin{pmatrix} -\lambda & \lambda\alpha & 0 & 0 & 0 \\ \tau & T - \lambda I & \lambda I & 0 & 0 \\ 0 & \tau\alpha & T - \lambda I & \lambda I & 0 \\ 0 & 0 & \tau\alpha & T - \lambda I & \lambda I \\ 0 & 0 & \ddots & \ddots & \ddots \end{pmatrix}, \quad (6.29)$$

каде I претставува квадратна матрица со единици како дијагонални елементи и $Te = -\tau$.

Генераторот Q има вектор на стационарна веројатност даден како $x = (x_0, x_1, x_2, \dots)$.

Равенките на рамнотежа се дадени во [8], во теоремата 1, како:

$$-\lambda x_0 + x_1 \tau = 0, \quad (6.30)$$

$$\lambda x_0 \alpha + x_1 (T - \lambda I) + x_2 \tau \alpha = 0, \quad (6.31)$$

$$\lambda x_{i-1} + x_i (T - \lambda I) + x_{i+1} \tau \alpha = 0, \quad i \geq 2. \quad (6.32)$$

Ако се помножат равенките (6.31) и (6.32) со векторот колона e од десно, се добива:

$$x_{i+1} \tau = \lambda x_i e, \quad i \geq 1. \quad (6.33)$$

Равенката (6.33) се множи со α на десно и се комбинира со равенката (6.32), се добива:

$$x_i (\lambda I - \lambda e \alpha - T) = \lambda x_{i-1}, \quad i \geq 2, \quad (6.34)$$

$$x_1 (\lambda I - \lambda e \alpha - T) = \lambda x_0 \alpha. \quad (6.35)$$

Поради ова, следи:

$$x_i = x_0 \alpha R^i, \quad (6.36)$$

$$R = \lambda (\lambda I - \lambda e \alpha - T)^{-1}, \quad (6.37)$$

каде R е матрица на рата дефинирана со равенката (6.37). Поради тоа што x е векторот на стационарната веројатност, мора да важи:

$$\sum_{i=0}^{\infty} x_i = 1, \quad (6.38)$$

па се добива:

$$\begin{aligned}
\sum_{i=0}^{\infty} x_i &= x_0 + x_0 \alpha \sum_{i=1}^{\infty} R^i e \\
&= x_0 + x_0 \alpha R (I - R)^{-1} e \\
&= x_0 - \lambda x_0 \alpha (\lambda e \alpha + T)^{-1} e \\
&= x_0 - \lambda x_0 \alpha T^{-1} (I + \lambda e \alpha T^{-1})^{-1} e \\
&= x_0 - \lambda x_0 \alpha T^{-1} \sum_{k=0}^{\infty} (-1)^k \lambda^k (e \alpha T^{-1})^k e \\
&= x_0 - \lambda x_0 \alpha T^{-1} (I - \lambda (1 - \rho)^{-1} e \alpha T^{-1}) e \\
&= x_0 + x_0 \rho + x_0 \rho^2 (1 - \rho)^{-1},
\end{aligned} \tag{6.39}$$

така да $x_0 = 1 - \rho$.

Просечниот број на *jobs* во редот може да се одреди со користење на:

$$\begin{aligned}
E(k_c) &= \sum_{i=1}^{\infty} i x_i e = \sum_{i=1}^{\infty} i x_1 R^{i-1} e \\
&= x_1 \sum_{i=1}^{\infty} \frac{d}{dR} R^i e = x_1 \frac{d}{dR} \left(\sum_{i=1}^{\infty} R^i \right) e \\
&= x_1 \frac{d}{dR} ((I - R)^{-1} - I) e = x_1 (I - R)^{-2} e
\end{aligned} \tag{6.40}$$

Конечно, просечното време на опслужување на контролерот $E(t_c)$ може да се претстави како:

$$E(t_c) = \frac{E(k_c)}{\lambda_c} = \frac{x_1 (I - R)^{-2} e}{\lambda_c}. \tag{6.41}$$

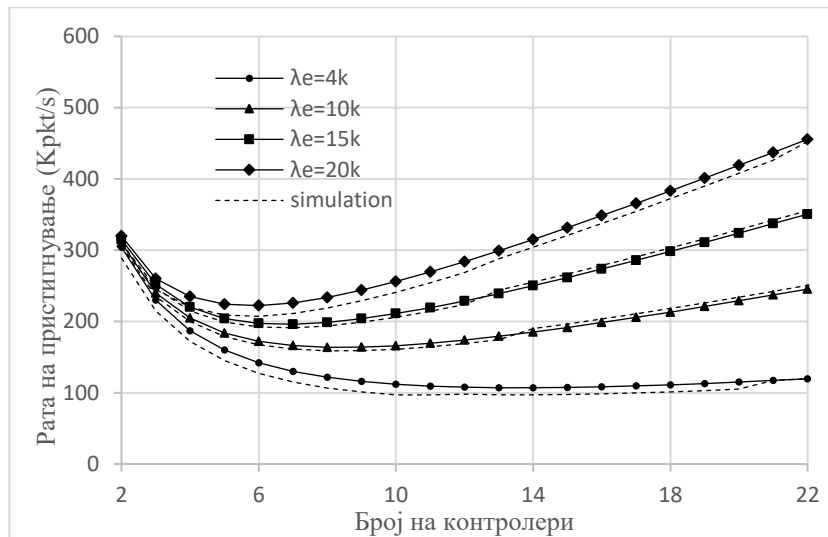
Во однос на перформансната матрица, повторно се користи истиот пристап како во поглавјето 6.3.1, каде доцнењето при хендовер е дефинирано со равенките (6.12), (6.13) и (6.15). Разликата овој пат е што наместо *Packet-in* пораките, се разменуваат *Port-status* пораки.

6.4.2 Анализа на перформанси

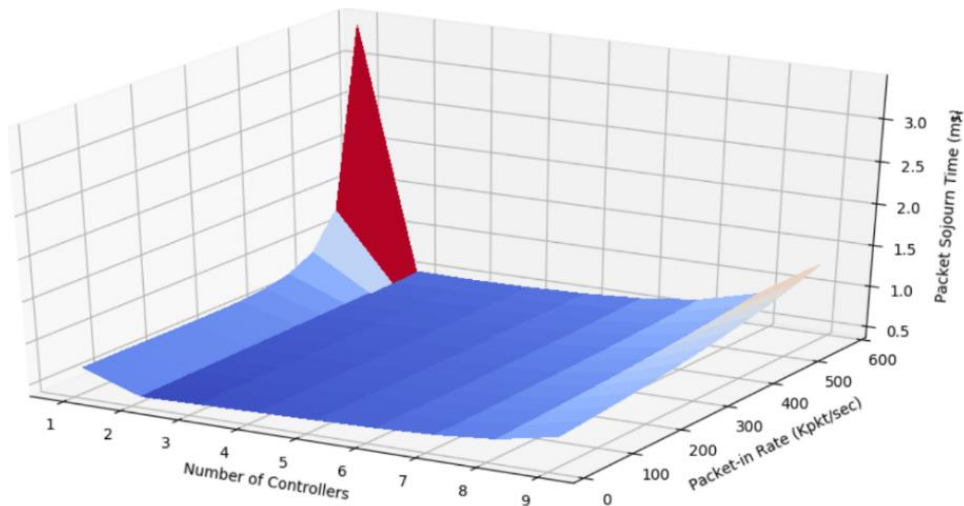
Во ова поглавје се валидира предложениот аналитички модел преку споредба на нумеричките резултати со симулациски резултати добиени со дискретен симулатор на настани, развиен во MATLAB. Се користи последната верзија на MATLAB R2019a, а симулациите се извршуваат на лаптоп со *Intel Core i5-8350U 1.9 GHz* со 16GB RAM. Основното симулациско сценарио се состои од 2 контролери и 10 комутатори по домен на контролер (лесно може да се додаваат/одземаат контролери/комутатори). Постојат 10 корисници конектирани на секој комутатор и вкупната рата на пристигнување на пакети кај секој комутатор е еднаква на вкупниот системски проток поделен со бројот на комутатори.

Мануалено може да се контролира количината на сообраќај која се насочува кон контролерите (односно q^{pc} може да се менува). За време на симулациите, целиот генериран сообраќај помеѓу комутаторите и контролерите се снима во датотеки, после што се прави пост-процесинг, повторно во MATLAB. Секоја симулација се извршува по 20 пати (со траење од 5 секунди) и во фазата на пост-процесинг се пресметува средната вредност на секоја точка од графовите. Доминантните пораки во фазата на иницијализација на контролерот се OFPT_FEATURES_REQUEST (REPLY) и OFPT_FLOW_MOD, додека времето на опслужување на контролерот се одредува преку мерење на временската разлика помеѓу примање на OFPT_PORT_STATUS и праќање на OFPT_FLOW_MOD порака. Се моделира реален мал мобилен оператор со вкупен системски проток од 5 Gbps – 20 Gbps. Врз основа на реална мобилна мрежа со сличен мрежен проток, се зема дека вкупниот број мрежни барања за *attach* е 40 барања/s. За q^{pc} се користи 4%, $\mu_{is}=9.8 \mu s$, $\mu_c=240 \mu s$ и големината на *Packet-in*, *Port-status* и *Flow-mod* пораките е 128 B. Растојанието помеѓу комутаторите и контролерот е 50 km – 200 km а линкот е 1 Gbit/s. Се зема реален број на комутатори по домен, $n = 10$. За нумеричките резултати, за времето на одговор на контролерот, се користат мерењата во [63] и се прилагодува добиената Hyper-Erlang дистрибуција.

На сликата 6.8 се анализира влијанието на вкупната рата на пристигнување на пакети кај еден контролер, како функција од бројот на контролери. Контролерите разменуваат синхронизациски пораки во чекори, со вкупна рата од 4k – 20k во секунда, а λ_p е фиксирано на 400k во секунда. Може да се забележи дека ратата на пристигнување се намалува кога бројот на контролери се зголемува се до одредена точка (пр. за $\lambda_e = 20k$, до $c=6$), но потоа се зголемува после оваа точка. Кога бројот на контролери е мал, контролерите може да делат меѓу себе голем број *Packet-in* пораки и ова ќе тригерира само неколку синхронизациски пораки. Во спротивниот случај, секој контролер прима помал број *Packet-in* пораки, но ова имплицира поголем број синхронизациски пораки. Исто така, може да се заклучи дека за повисоки вредности на λ_e , ратите на пристигнување се исто така повисоки, што е за очекување. Конечно, може да се донесе заклучок дека за помал λ_e , за да се постигне минимална рата на пристигнување, бројот на контролери мора да биде повисок за да се постигне истото споредено со случајот кога има повисоки вредности за λ_e .



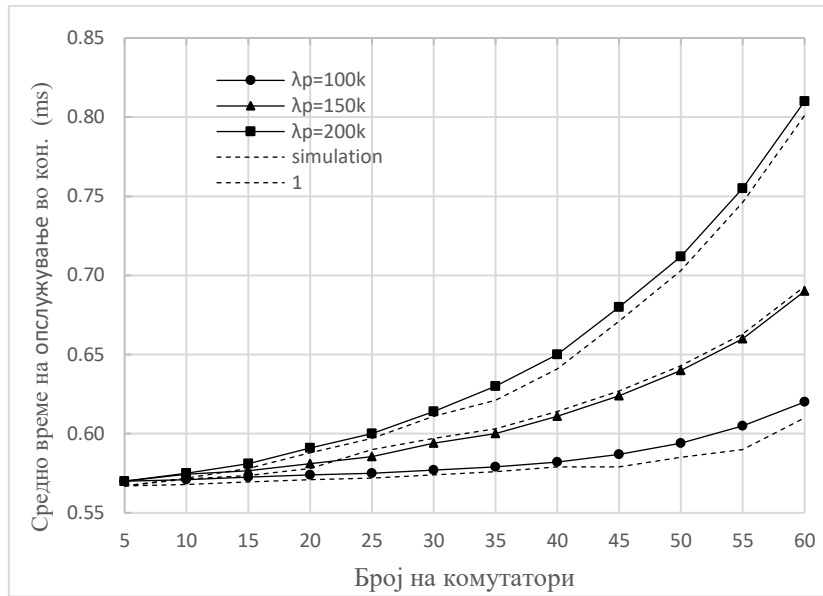
Слика 6.8. Рата на пристигнување наспроти број на контролерите



Слика 6.9. Време на опслужување наспроти рата на *Packet-in* пораки и број на контролери

Сликата 6.9 го претставува времето на опслужување на контролерот при различен број контролери и променлива рата на *Packet-in* пораките. Синхронизациските пораки се фиксирани на 20k во секунда, додека ратата на пристигнување на *Packet-in* пораки се менува во рангот 100k - 600k во секунда. Времето на опслужување на контролерот се намалува како што бројот на контролери се зголемува, но само до $c=2$ контролери, што претставува точка на прекршување, после што доаѓа до линеарно зголемување се до $c=7$. Заклучокот е дека иако се додаваат повеќе контролери со цел да може да се примаат повеќе *Packet-in* пораки, сепак екстра контролерите влијаат негативно на времето на опслужување поради дополнителниот сообраќај заради сигнализација. Исто така, треба да се спомене и огромната деградација на времето на опслужување во случај на еден контролер, за што се повторно се дискутира понатаму во текстот. Понатаму, се приметувва дека после $c=7$, времето на опслужување се деградира многу побрзо, поради тоа што ова е границата после

која бројот на синхронизациски пораки станува толку висок што драстично влијае на влошување на перформансите.

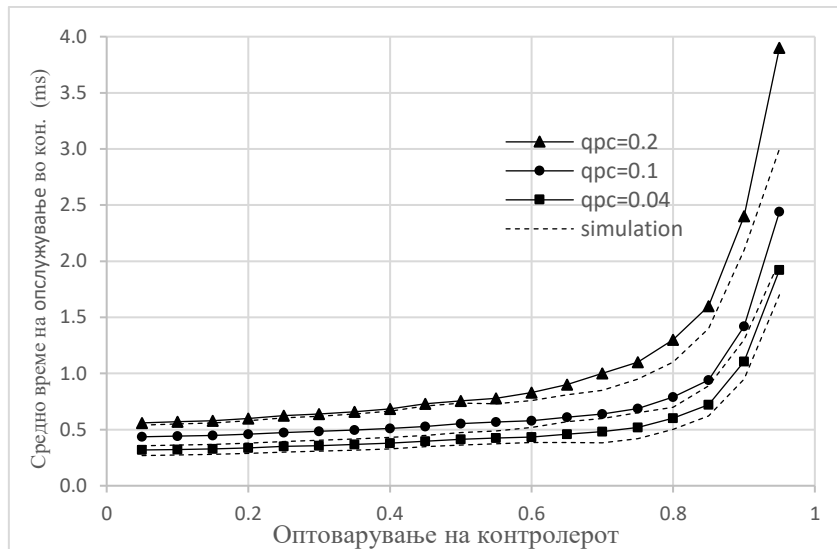


Слика 6.10. Време на опслужување наспроти број на комутатори

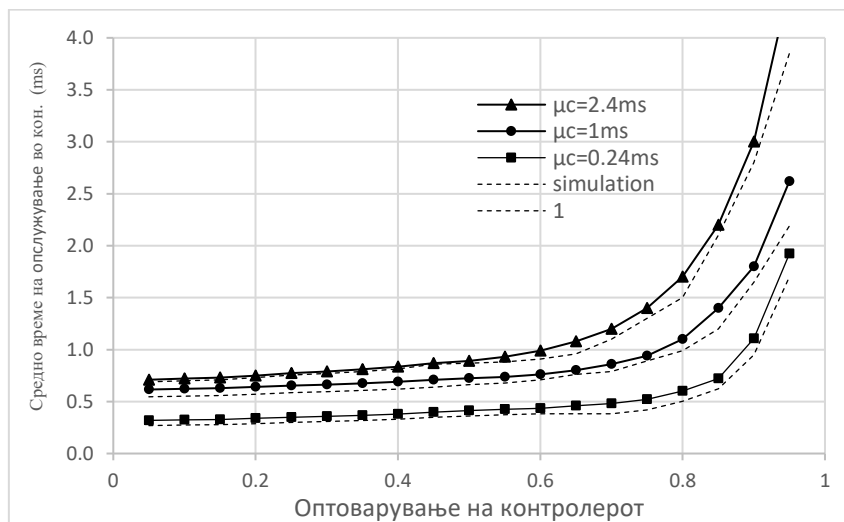
На сликата 6.10 се анализира зависноста на времето на опслужување на контролерот од бројот на комутатори, при променлива рата на пристигнување на *Packet-in* пораките. Како што расне бројот на комутаторите, се зголемува времето на опслужување на контролерот. Сепак, може да се забележи дека времето на опслужување расне многу споро како што се зголемува бројот на комутаторите. Кога *Packet-in* пораките пристигаат со рата од 100k во секунда, времето на опслужување на контролер кој е поврзан со 50 комутатори е само околу 0.04 ms повисоко од случајот кога контролерот е поврзан само со еден комутатор. Дополнително, се забележува дека времето на опслужување забрзано се деградира како што расне веројатноста на наидување на *Packet-in* пораките. Симулациските резултати го следат блиску аналитичкиот модел, освен за случајот $\lambda_p=100k$ (кога бројот на комутаторите е над 50).

Сликата 6.11 го прикажува времето на опслужување како функција од оптоварувањето на контролерот. Како што оптоварувањето на контролерот се приближува кон 1, доцнењето се стреми кон бесконечност и како што q^{pc} расне, времето на опслужување пропорционално се зголемува. Исто така, се забележува дека аналитичкиот модел ја следи блиску симулацијата за пониските вредности на оптоварувањето и за пониските вредности за q^{pc} , додека за повисоки вредности на q^{pc} , аналитичкиот модел станува помалку точен. Во трудот [64] се покажува дека веројатноста за нови текови е околу 4%, па може да се заклучи дека предложениот модел може да се користи со доволна точност. Понатаму, се истражува влијанието на средното време на опслужување врз просечното време на опслужување,

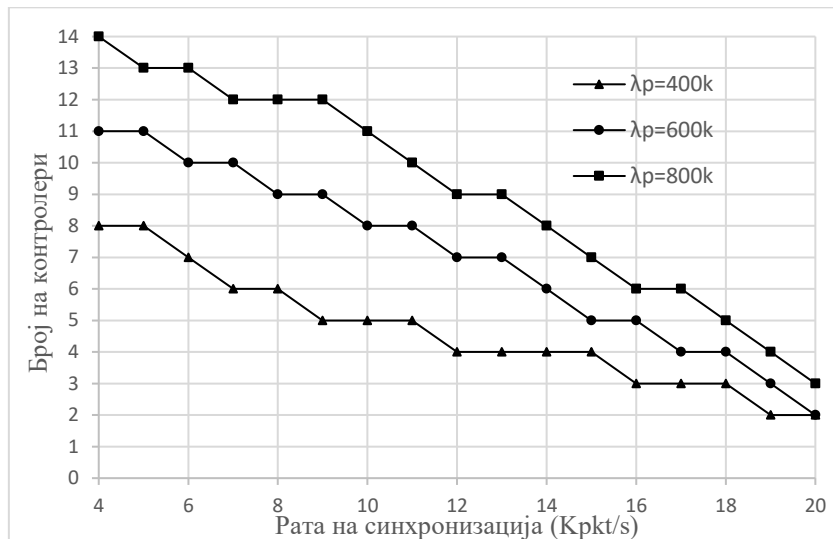
додека q^{pc} се држи на фиксна вредност од 0.04. Резултатите се прикажани на сликата 6.12. Како што се очекуваше, за повисоки вредности на μ_c , времето на опслужување се намалува. Овој график може да се искористи за поставување гаранции за времето на опслужување на контролерот кога се дизајнира мрежа со контролер кој има фиксно средно време на опслужување и има ограничено оптоварување на контролерот. При оптеретувања од околу 0,7 до 0,8, времето на опслужување нагло се зголемува и станува неприфатливо високо за највисокиот ранг на вредности на оптоварувањето.



Слика 6.11. Време на опслужување наспроти оптоварувањето, ефект на q^{pc}

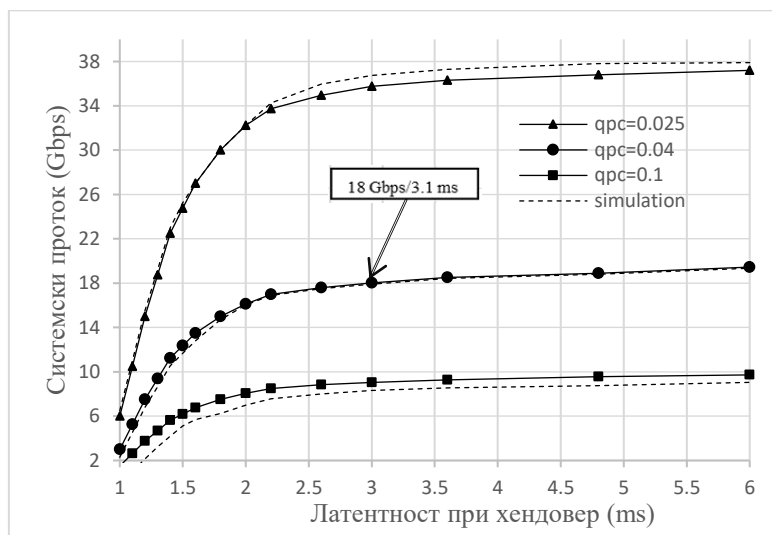


Слика 6.12. Време на опслужување наспроти оптоварувањето, ефект на μ_c за фиксна вредност на q^{pc}



Слика 6.13. Оптималниот број на контролери

Со примена на оптимизациски алгоритам на равенката (6.41), се прави обид да се добие оптималниот број на контролери за да се постигне минимално време на опслужување, во услови кога варираат ратите на синхронизациски и *Packet-in* пораки. Резултатите прикажани на сликата 6.13 покажуваат дека кога ратата на синхронизација се зголемува, оптималниот број на контролери се намалува. За *Packet-in* пораките, ефектот е спротивен, како што расне λ_p , бројот на контролери потребен за да се постигне минимално време на опслужување исто така се зголемува. Ако има многу контролери во мрежата, синхронизациските пораки предизвикуваат дополнителен сообраќај заради сигнализација и ова влијае негативно на времето на опслужување, додека потребни се поголем број контролери за да се има ефикасно справување со голем број текови/комутатори. Резултатите на сликата 6.13 се во согласност со оние од 6.9, каде може лесно да се забележи дека за $\lambda_p=20k$, оптималниот број на контролери 2. Ако се направи повторен осврт на сликата 6.9, исто така може да се заклучи дека во случајот на единствен контролер, времето на опслужување е сериозно деградирано. Ова би значело дека еден SDN дизајнер никогаш не смее да имплементира единствен контролер за мал мрежен оператор. За подетално да се истражи овој феномен, направени се неколку анализи на аналитичкиот модел проследени со симулации во услови на многу високи рати на синхронизација ($> 100k$). Од резултатите се заклучува дека само во овој случај на екстремно висок дополнителен сообраќај заради сигнализација, единствен контролер е оптималниот избор, но сепак мора да се нагласи дека вакво сценарио е многу неверојатно.



Слика 6.14. Системски проток наспроти латентноста при хендовер

Конечно, сликата 6.14 го прикажува целниот системски проток како функција од вкупното доцнење при хендовер што го доживува MN кога извршува “тврд” хендовер. Системскиот проток е всушност вкупниот сообраќај кој се носи од мрежата. Бројот на контролери се зема да биде $c=2$, бројот на синхронизациски пораки е фиксиран на 20k во секунда, додека q^{pc} има три различни вредности. Графикот покажува дека иницијално хендовер латентноста расте умерено, но потоа стига до точка кога мрежата се заситува (на англ. saturates). После оваа точка на заситување, латентноста при хендовер нагло се зголемува додека протокот има незначаен раст. Оваа анализа ја покажува границата на еластичност на мрежата и може да биде корисна кога се дизајнира гарантирано максимално време на хендовер во мобилна мрежа со даден % на сообраќај кој се насочува кон контролерот. На пример, за $q^{pc}=0.04$ и за мрежен проток од 18 Gbps, доцнењето при хендовер поради OF сигнализација е 3.1 ms. Дополнително, се забележува дека кога веројатноста на наидување на *Packet-in* пораките е висока, латентноста при хендовер нагло се зголемува дури и при мали вредности на системскиот проток. Со други зборови, дизајнерите мора да работат на намалување на очекуваниот q^{pc} доколку сакаат да им понудат на корисниците високо ниво на квалитет на опслужување преку задржување на доцнењето при хендовер во прифатливи граници.

6.4.3 Влијание на претпоставките употребени при моделирањето

Со цел да се добијат равенки во затворена форма, во претходните секции се употребија неколку претпоставки, како што се неограничен бафер, моделирање на редот како единствен ред (не по интерфејс), Поасанова рата на пристигнување и рата на опслужување која е независна од оптоварувањето. Првата и последната претпоставка веќе

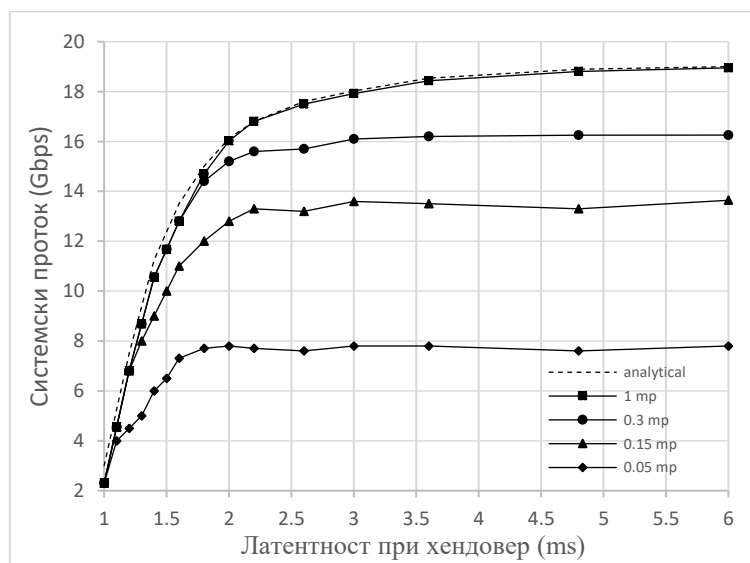
беа детално дискутирани и моделирани во главата 5, сепак во ова поглавје се прави додатна анализа и споредба со симулациите со цел мерење на влијанието врз доцнењето при хендовер поради овие претпоставки.

Во вистински OF-комутатор, постојат повеќе ингрес и излезни порти кои имаат фиксна големина на бафер. Влезните пакети од секој бафер се процесираат од истите табели на тек и не се користи приоретизација. Во иницијалните симулации, за комутаторите и контролерот претходно се користеа бафери со многу висока вредност (1 милион пакети (mp)), но овој пат вкупната големина на бафер варира во рангот 0.05 до 1 mp. Дополнително, за секој комутатор се користат повеќе редови (2 и 6), а ратата на пристигнување по комутатор е 100Kpkt/s – 500 Kpkt/s. Големината на пакетите останува 128 B, $q^{pc}=0.04$ и симулациите се извршуваат 20 пати со траење од 5 секунди.

Првите заклучоци кои се добиваат од симулациските резултати е дека за фиксна големина на бафер по комутатор (и фиксна средна рата на опслужување), максималната рата на пристигнување на пакети пред да дојде до пакетски загуби (поради недоволен простор во баферот), останува иста. Ова значи дека дури и со зголемување на бројот на бафери (интерфејси), иако максималната рата на пакети пред да дојде до пакетски загуби се намалува, на ниво на комутатор оваа рата останува константна. Се забележува дека како што се зголемува ратата на пристигнување, минималната големина на бафер со цел да се избегне губење на пакети исто така треба да се зголеми, што е за очекување. Во предложениот аналитички модел, оправдано се претпостави неограничен бафер за комутаторот, затоа што типично баферот е многу голем во хардверските имплементации на OF комутаторите. Дополнително, времето на процесирање на комутаторот придонесува многу малку во вкупната латентност при хендовер и доминантен фактор претставува времето на процесирање во контролерот. Ова значи дека дури и значително влошени перформанси во комутаторот поради премала големина на бафер, нема премногу да променат во вкупното време на доцнење при хендовер.

На сликата 6.15 се истражува влијанието на ограничениот бафер во контролерот и се прикажуваат симулациските резултати за зависноста на системскиот проток од латентноста при хендовер. Генерално, целта е да се евалуира точноста на сликата 6.14 каде беше искористен доволно голем бафер за да се избегне било каква загуба на пакети. Се земаат во предвид истите симулациски параметри како и претходно, се задржува $q^{pc} = 0.04$ и големината баферот се менува за три вредности: 0.05, 0.15 и 0.3 mp. Се забележува посериозна деградација на перформансите за големина на бафер од 0.05 mp, додека разликата во времето на хендовер од аналитичкиот модел за големина на бафер од 0.3 mp е

многу мало. Поради пакетските загуби и механизмите за избегнување на TCP загушување, латентноста при хендовер се зголемува значително кога се намалува баферот. Се доаѓа до сличен заклучок и се забележува истиот ефект на ограничен бафер за резултатите прикажани на сликите 6.8, 6.10, 6.11 и 6.12. Како што се зголемува бројот на *Packet-in* пораки во контролерот, за помали големини на баферот, деградацијата на времето на опслужување е повидлива. Се заклучува дека за да се подобри точноста на предложениот модел, мора да се земе во предвид ефектот на ограничен бафер, додека ефектот на моделирање на OF-комутатор со единствен бафер не е претпоставка која изразито влијае на перформансите во услови кога не се користи приоритизација за сообраќајот.



Слика 6.15. Влијание на ограничениот бафер на контролерот врз латентноста при хендовер

Во аналитичките модели предложени за моделирање на OF мрежи, најчестата хипотеза која се зема во предвид претставува Поасоновата рата на пристигнување на пакети. Во најголемиот број на трудови кои беа анализирани во оваа дисертација, искористена беше оваа претпоставка, но сепак општо познато е дека во вистинска мрежа, реалниот сообраќај покажува себеслични карактеристики. Себесличниот сообраќајот е широко застапен феномен во модерните мрежи и неговата главна карактеристика е: сегменти на процесот имаат многу слични статистички особини при различни скалирања (на англ. *at different scales*). Најважните дефиниции и особини на себесличност се дадени во трудот [103], каде се дефинира еден битен параметар за квантификација на степенот на себесличност, наречен *Hurst* параметар. Симулирањето на себесличноста исто така го има привлечено вниманието на истражувачката заедница, во трудот [104] се споредуваат неколку модели за генерирање на сообраќај имплементирани во MATLAB. Во оваа дисертација нема практично да се анализира ефектот на претпоставката Поасонова рата на пристигнување и овој феномен може да биде дел од идна научно-истражувачка работа.

Конечно, ефектот од претпоставката за ратата на опслужување кое е независна од оптоварувањето, веќе беше детално дискутирана во главата 5. Сега само би се напоминало дека оваа претпоставка видливо би ги афектирала графовите на сликата 6.11 и сликата 6.12, каде би се очекувала повидлива деградација при високи оптеретувања. Сепак, ефектот на ратата на опслужување која е зависна од оптоварувањето може ефективно да се ублажи со внимателно SDN дизајнирање, преку одбирање на контролер со доволно висока рата на средна опслужување.

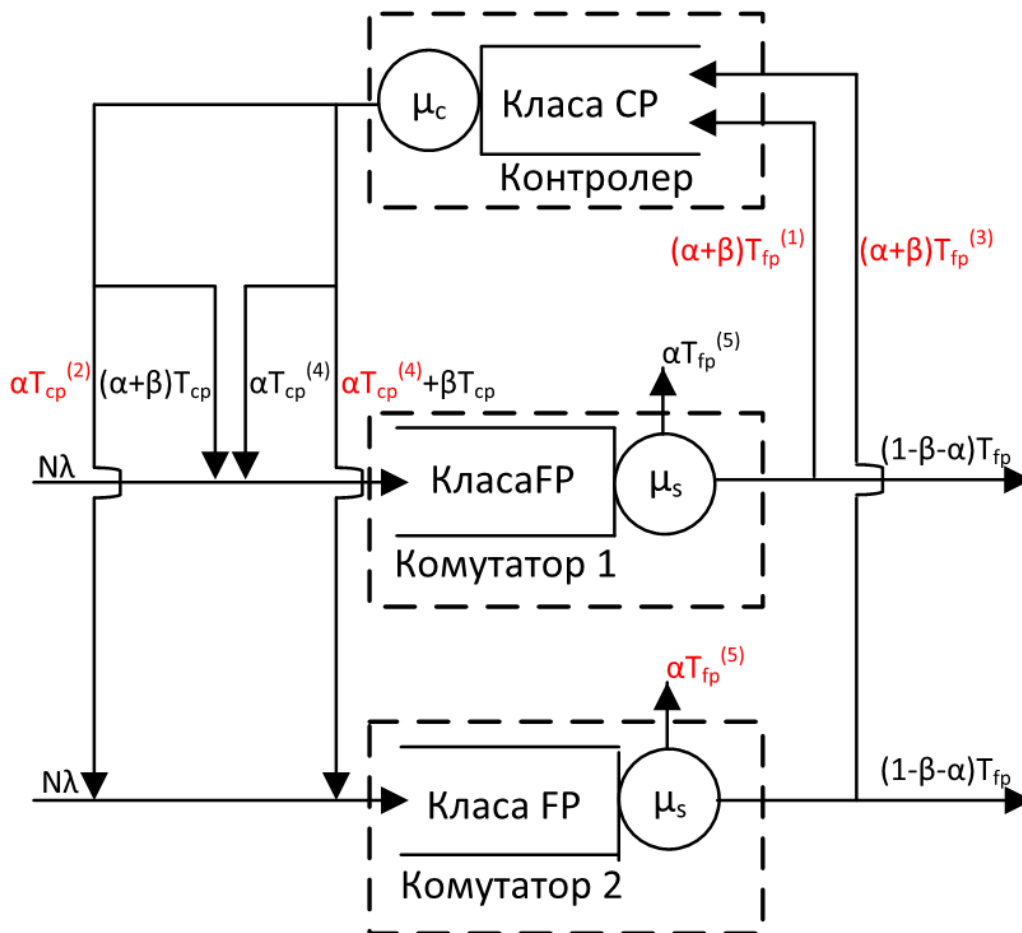
6.5 Моделирање на доцнењето при хендовер со користење комутатор со приоритетни бафери

Во ова поглавје се предлага нов математички пристап за моделирање на доцнењето при хендовер, при што се прави споредба на два типа комутатори во однос на типот на бафер кој се користи: (i) еден бафер без користење на приоритизација; (ii) два засебни бафери кои користат не-преемтивна приоритизација. Двата предложени системи се споредуваат во однос на доцнењето при “тврди” хендовер и во однос на минималниот капацитет на баферот кој е потребен за да се задоволи соодветната веројатност на пакетска грешка на излезниот линк.

При моделирањето, се користат истите претпоставки од поглавјето 6.4.1, што значи дека и овој пат се моделираат *Packet-in* и *Port-status* пораки. Сценариото од интерес е детално опишано во поглавјето 3.4, а размената на OF пораките при хендовер е прикажана на сликата 3.6. Секвенцата на размена на пораки при хендовер беше опишана таму и нема да се повторува овде.

6.5.1 Аналитички модел SFB (Shared Finite Buffering)

Моделот на системот на заеднички конечен баферинг (Модел SFB) е прикажан на сликата 6.16. Комутаторот користи единствен заеднички и конечен бафер за контролната и корисничката рамнина и вкупниот сообраќај на влез од комутаторот е сума од сообраќајот кој влегува однадвор и сообраќајот кој го враќа контролерот. Овој тип на ред е опишан како GI/M/1/K ред (конечен бафер), поради тоа што ратата на пристигнување следи генерална дистрибуција поради мешањето на два потока на сообраќај кои имаат независни рати на пристигнување.



Слика 6.16. Моделирање на хендовер при користење на комутатор со заеднички бафер без приоритизација (модел SFB)

Се користи β за да се означи веројатноста на наидување на *Packet-In* пораките пратени кон контролерот при негативна проверка за совпаѓање на влезниот пакет со табелата на тек на комутаторот, а со α се означува веројатноста на *Port-status* пораките кои се праќаат од комутаторот за да го означат почетокот на хендовер процедурата. Системот има две класи: *Fast Path* (FP), кој се користи од комутаторот за процесирање на сите пакети и *Controller Path* (CP) за сите пакети кои се процесирани од контролерот. Класата CP во овој случај се моделира како M/M/1 ред (бесконечен ред) и има средна рата на опслужување μ_c , додека класата FP е моделирана со средна рата на опслужување μ_s . N е бројот на MN кои се конектирани на еден комутатор, а K претставува конечната големина на баферот во комутаторот. Целта е да се конструира систем кој ќе овозможи моделирање и квантификација на доцнењето при хендовер. На сликата 6.16 со црвено е означена секвенцата на пораки која се однесува на хендовер процедурата: (1) *Port-Status (off-port)* порака е примена во комутаторот 1, се процесира од заедничката класа Class FP во комутаторот 1 и се праќа кон контролерот ($\alpha T_{fp}^{(1)}$, каде T_{fp} е протокот на класата FP); (2) *Port-Status* пораката потоа се процесира од заедничката класа CP, се насочува кон

комутаторот 1, а се генерира и екстра пакет со цел информирање на комутаторот 2 за *off-port* пораката за MN во комутаторот 1 ($\alpha T_{cp}^{(2)}$, каде T_{cp} е проток на класата CP); (3) се претпоставува максимално синхронизиран систем, што значи дека веднаш после примањето на *off-port* пораката, комутаторот 2 знае дека *on-port* порака е примена во класата FP во комутаторот 2 и оваа порака слично се препраќа од комутаторот 2 кон контролерот ($\alpha T_{fp}^{(3)}$); (4) овој пакет потоа се процесира во контролерот и се генерираат пакети со кои се информираат и двата комутатори за промената во мрежата ($\alpha T_{cp}^{(4)}$); (5) откако пакетот кој е од интерес се процесира во комутаторот 2 (и комутаторот 1), се отфрла ($\alpha T_{fp}^{(5)}$). Целта е да се измери вкупното доцнење воведено поради различното процесирање на релевантните пакети кои учествуваат во секвенцата на опишаната процедура при хендовер поради OF сигнализацијата.

Системот е креиран како континуиран Марков ланец $\{n_{cp}(t), n_{fp1}(t), n_{fp2}(t), \text{ каде } t > 0\}$, каде $n_{cp}(t)$, $n_{fp1}(t)$, и $n_{fp2}(t)$ го претставуваат бројот на пакети во контролерот, комутаторот 1 и комутаторот 2, соодветно. Со x, y, z се означува множество на вредности за $n_{cp}(t)$, $n_{fp1}(t)$, и $n_{fp2}(t)$, каде $x \in Z_+$, $y \in Z_+^{(\leq K)}$ и $z \in Z_+^{(\leq K)}$. Дозволените транзиции на Марковиот процес се прикажани во табелата 6.1.

Табела 6.1. Дозволените транзиции за моделот SFB.

	Од	Кон	Рата
Пакет пристигнува во ком. 1	(x, y, z)	$(x, y+1, z)$	$N\lambda$
Пакет пристигнува во ком. 2	(x, y, z)	$(x, y, z+1)$	$N\lambda$
Пакет се препраќа од ком. 1 кон контролер ⁽¹⁾	$(x, y>0, z)$	$(x+1, y-1, z)$	$(\alpha+\beta)\mu_s$
Пакет е сервисиран од контролер кон ком. 1	$(x>0, y, z)$	$(x-1, y+1, z)$	μ_c
Пакет е сервисиран од контролер кон ком. 2 ⁽²⁾	$(x>0, y, z)$	$(x-1, y, z+1)$	$\alpha\mu_c$
Пакет се препраќа од ком. 2 кон контролер ⁽³⁾	$(x, y, z>0)$	$(x+1, y, z-1)$	$(\alpha+\beta)\mu_s$
Пакет е сервисиран од контролер кон ком. 1 ⁽⁴⁾	$(x>0, y, z)$	$(x-1, y+1, z)$	$\alpha\mu_c$
Пакет е сервисиран од контролер кон ком. 2 ⁽⁴⁾	$(x, y, z>0)$	$(x-1, y, z+1)$	μ_c
Пакет е отфрлен во ком. 1 ⁽⁵⁾	$(x, y>0, z)$	$(x, y-1, z)$	$\alpha\mu_s$
Пакет е отфрлен во ком. 2 ⁽⁵⁾	$(x, y, z>0)$	$(x, y, z-1)$	$\alpha\mu_s$
Пакет заминува од ком. 1	$(x, y>0, z)$	$(x, y-1, z)$	$(1-\beta-\alpha)\mu_s$
Пакет заминува од ком. 2	$(x, y, z>0)$	$(x, y, z-1)$	$(1-\beta-\alpha)\mu_s$

За мал мобилен оператор (со ~ 20 Gbit/s вкупен мрежен проток), бројот на барања за хендовер е околу 40 pkt/s на ниво на мрежа (податок земен од статистика на вистински оператор) за околу 200k истовремени активни корисници (со активна пакетска сесија). Ако се земе во предвид дури и многу мал проток за еден MN, од околу ~ 1 Mbit/s (за MTU од 1500 B, ова е 83 pkt/s), за најмалата вредност на $\beta = 0.01$ и само еден MN, ова резултира со 0.83 pkt/s. Значи за еден MN, кога се споредува најлошиот случај од 0.83 pkt/s за *Packet-in*

пораците и просек од 0.0002 пакети/s (40 пакети/s поделено со 200k корисници) за *Port-Status* пораките, станува јасно дека $\alpha \ll \beta$ и $\alpha \ll 1$. Ова е причината поради која се смета дека $(\alpha+\beta)\mu_s \sim \beta\mu_s$, $(1-\beta-\alpha)\mu_s \sim (1-\beta)\mu_s$ и $(1+\alpha)\mu_c \sim \mu_c$. Овие апроксимации се користат при пресметка на под-матриците на QBD процесот (матрицата на генераторот на транзициската рата Q) и при тоа се користат дозволените состојби дефинирани во табелата 6.1. Овие матрици потоа се влез во МММ со цел да се пресметаат веројатностите на стационарната дистрибуција.

Под-матрицата A_0 ја дефинира фазната дистрибуција на класата FP во комутаторот 1 и комутаторот 2, соодветно, кога бројот на пакети во контролерот се зголемува за 1:

$$A_0(y, y') = \begin{cases} A_{01}^{(y)}, & y' = y - 1, \\ A_{02}^{(y)}, & y' = y, \\ 0, & \text{поинаку,} \end{cases}$$

$$A_{01}(z, z') = \begin{cases} \beta\mu_s, & z' = z, \\ 0, & \text{поинаку,} \end{cases}$$

$$A_{02}(z, z') = \begin{cases} \beta\mu_s, & z' = z - 1, \\ 0, & \text{поинаку.} \end{cases}$$

Во таа насока, под-матрицата A_2 ја дефинира фазната дистрибуција кога бројот на пакети во контролерот се намалува за 1:

$$A_2(y, y') = \begin{cases} A_{21}^{(y)}, & y' = y + 1, \\ A_{22}^{(y)}, & y' = y, \\ 0, & \text{поинаку,} \end{cases}$$

$$A_{21}(z, z') = \begin{cases} \mu_c, & z' = z, \\ 0, & \text{поинаку,} \end{cases}$$

$$A_{21}(z, z') = \begin{cases} \mu_c, & z' = z + 1, \\ 0, & \text{поинаку.} \end{cases}$$

Подматрицата A_1 ја претставува фазната дистрибуција кога бројот на пакети во контролерот не се менува и бројот на пакети не е нула:

$$A_1(y, y') = \begin{cases} A_{11}, & y' = y + 1, \\ A_{12}, & y' = y, \\ A_{13}, & y' = y - 1, \\ 0, & \text{поинаку,} \end{cases}$$

$$A_{11}(z, z') = \begin{cases} N\lambda, & z' = z, \\ 0, & \text{поинаку,} \end{cases}$$

$$A_{12}(z, z') = \begin{cases} N\lambda, & z' = z + 1, \\ (1 - \beta)\mu_s, & z' = z - 1, \\ 0, & \text{поинаку,} \end{cases}$$

$$A_{13}(z, z') = \begin{cases} (1 - \beta)\mu_s, & z' = z, \\ 0, & \text{поинаку.} \end{cases}$$

Дијагоналните елементи на A_{12} каде $z'=z$, се дефинирани како:

$$A_{12}(z, z') = \begin{cases} -2N\lambda - \mu_c, & y = 0, \quad z = 0, \\ -2N\lambda - \mu_c - \mu_s, & y = 0, \quad 0 < z < K, \\ -N\lambda - \mu_c - \mu_s, & y = 0, \quad z = K, \\ -2N\lambda - \mu_c - \mu_s, & 0 < y < K, \quad z = 0, \\ -2N\lambda - \mu_c - 2\mu_s, & 0 < y < K, \quad 0 < z < K, \\ -N\lambda - 2\mu_s - \mu_c, & 0 < y < K, \quad z = K, \\ -N\lambda - \mu_s - \mu_c, & y = K, \quad z = 0, \\ -N\lambda - 2\mu_s - \mu_c, & y = K, \quad 0 < z < K, \\ -2\mu_s, & y = K, \quad z = K, \\ 0, & \text{поинаку.} \end{cases}$$

Конечно, подматрицата B_1 ја претставува фазната дистрибуција кога бројот на пакети во контролерот не се менува и бројот на пакети во контролерот е еднаков на нула:

$$B_1 = A_1 (\mu_c = 0).$$

Целта е да се пресметаат веројатностите на стационарната дистрибуција ($\pi_{x,y,z}$), за x пакети во контролерот, y пакети во комутаторот 1 и z пакети во баферот на комутаторот 2. Протокот на класата FP (комутатор 1) е сума од веројатностите дека класата FP има барем еден пакет за препраќање со средна рата на опслужување μ_s :

$$T_{fp1} = \mu_s \sum_{x=0}^{\infty} \sum_{y=1}^K \sum_{z=0}^K \pi_{x,y,z}. \quad (6.42)$$

Слично, протокот на класите FP и CP во комутаторот 2, се дадени со:

$$\begin{aligned} T_{fp2} &= \mu_s \sum_{x=0}^{\infty} \sum_{y=0}^K \sum_{z=1}^K \pi_{x,y,z}, \\ T_{cp} &= \mu_c \sum_{x=1}^{\infty} \sum_{y=0}^K \sum_{z=0}^K \pi_{x,y,z}. \end{aligned} \quad (6.43)$$

Просечната должина на ред на класата FP во комутаторот 1 е:

$$E(N)_{fp1} = \sum_{y=1}^K \pi_y \times y, \quad (6.44)$$

каде, π_y е маргиналната веројатност за просечен број на пакети во класата FP (комутатор 1) и е дефинирана како:

$$\pi_y = \sum_{x=0}^{\infty} \sum_{z=0}^K \pi(x, y, z). \quad (6.45)$$

На сличен начин, се одредуваат и просечните должини на редот во класата FP ($E(N)_{fp2}$) во комутаторот 2 и класата CP ($E(N)_{cp}$) како:

$$\begin{aligned} E(N)_{fp2} &= \sum_{z=1}^K \pi_z \times z, \\ \pi_z &= \sum_{x=0}^{\infty} \sum_{y=0}^K \pi(x, y, z), \\ E(N)_{cp} &= \sum_{x=1}^{\infty} \pi_x \times x, \end{aligned}$$

$$\pi_x = \sum_{y=0}^K \sum_{z=0}^K \pi(x, y, z). \quad (6.46)$$

Се користи теоремата на Литл [10] за да се пресмета просечното доцнење при процесирање на пакетот за секој од трите класи (t_{fp1} , t_{fp2} , t_{cp}):

$$\begin{aligned} t_{fp1} &= \frac{E(N)_{fp1}}{T_{fp1}}, \\ t_{fp2} &= \frac{E(N)_{fp2}}{T_{fp2}}, \\ t_{cp} &= \frac{E(N)_{cp}}{T_{cp}}. \end{aligned} \quad (6.47)$$

Вкупното доцнење при хендовер кое го доживува MN, t_{tot}^{sfb} , е сума од следните компоненти:

$$t_{tot}^{sfb} = t_{fp1} + 2t_{cp} + 2t_{fp2}. \quad (6.48)$$

Во паралела, целта е да се истражи проблемот на димензионирање на бафер. Намерата е да се одреди минималната големина на бафер која е потребна за апсорпција на флукуациите на излезниот линк кој има одредена веројатност на загуби и да се осигура дека загубите предизвикани од редот во баферот се под веројатноста на загуби на линкот.

Во M/M/1 ред, веројатноста дека должината на постојниот ред ја надминува вредноста K_{min} на баферот, може да биде дадена со $P(L > K_{min}) = \rho^{K_{min}}$, каде $\rho^{K_{min}}$ е искористувањето на комутаторот при K_{min} пакети во баферот. Се користи M/M/1 модел на редови за моделирање на комутаторите и минималната големина на бафер се дефинира како:

$$K_{min}^{sfb} \geq \frac{\log[PER]}{\log[\rho_{fp}]} \times 100\%, \quad (6.49)$$

каде PER е *Packet Error Ratio* кој може да се изрази преку *Bit Error Ratio* (BER) со користење на:

$$PER = 1 - (1 - BER)^n, \quad (6.50)$$

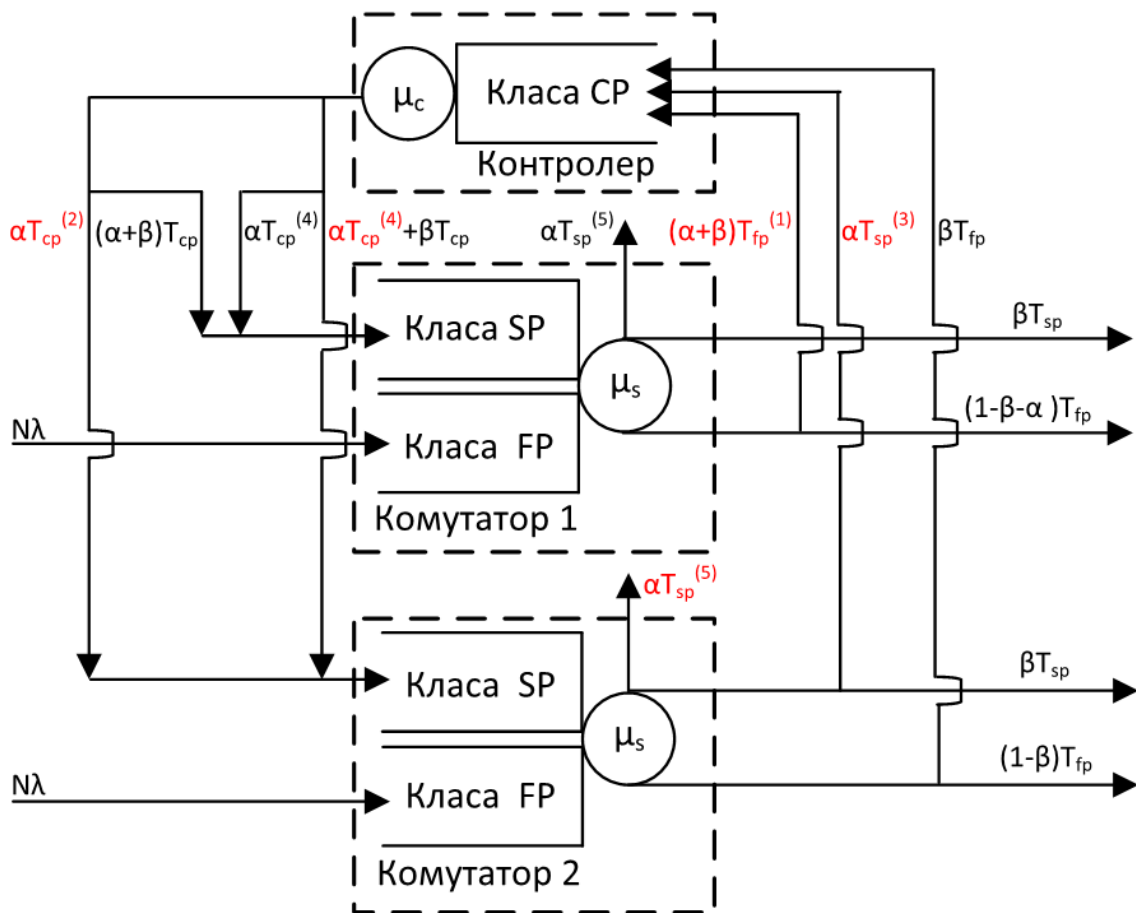
а n е бројот на бити во пакетот. ρ_{fp} е искористувањето на комутаторот кое е дефинирано како:

$$\rho_{fp} = \frac{(1 + \beta)N\lambda}{\mu_s}. \quad (6.51)$$

6.5.2 Аналитички модел PFB (Priority Finite Buffering)

На сличен начин како во поглавјето 6.5.1, сега се моделира систем во кој комутаторот користи бафери со приоритизација (модел PFB). Предложениот систем е

претставен на сликата 6.17. Овој пат, во моделот се инкорпорира приоритизација во редовите на чекање, која што води кон изолација помеѓу пакетите во корисничката и контролната рамнина. Секој комутатор има две дефинирани класи: класа *Slow Path* (SP), која се користи за примање на пакети од контролерот и препраќање на *Port-status* пораките и класа *Fast Path* (FP), каде пакетот се процесира само доколку нема пакети во класата SP. μ_s е средната рата на опслужување на комутаторот и истата се дели помеѓу двете класи. Се користи не-преемтивен ред со приоритизација, што значи дека класата FP се опслужува само доколку нема пакети во класата SP. Вкупниот капацитет на баферот во комутаторот е K , класата SP има капацитет K_1 , додека класата FP има K_2 , така да $K=K_1+K_2$.



Слика 6.17. Моделирање на хендовер при користење на комутатор со приоритетни бафери (модел PFB)

Во моделот PFB, главната разлика во процедурата на хендовер од онаа опишана во претходното поглавје, е таа што веднаш откако *Port-status* пораката ќе се процесира во контролерот, сега се праќа кон специјализирана приоритетна класа SP. На сликата 6.17 со црвено е прикажана релевантната секвенца која се однесува на процедурата хендовер: (1) *Port-Status* (*off-port*) пораката е примена во класата FP на комутаторот 1 и се праќа кон контролерот ($\alpha T_{fp}^{(1)}$, каде T_{fp} е протокот на класата FP); (2) *Port-Status* пораката се процесира од заедничката класа CP, се насочува кон комутаторот 1 (Class SP) и се генерира додатен

пакет за да се информира комутаторот 2 (класа SP) за *off-port* пораката на MN во комутаторот 1 ($\alpha T_{cp}^{(2)}$, каде T_{cp} е протокот на класата CP); (3) се претпоставува максимално синхронизиран систем, што значи дека комутаторот 2 веднаш знае дека *on-port* порака е примена во класата SP и оваа порака повторно се насочува од комутаторот 2 кон контролерот ($\alpha T_{sp}^{(3)}$, каде T_{sp} е проток на класата SP); (4) пакетот е процесира во контролерот и се праќа со цел да се информира комутаторот 1 (класа SP) и комутаторот 2 (класа SP) за промената во мрежата ($\alpha T_{cp}^{(4)}$); (5) откако пакетот ќе се процесира во комутаторот 2 (класа SP), се отфрла ($\alpha T_{sp}^{(5)}$).

Системот е конструиран како континуиран Марков ланец $\{n_{cp}(t), n_{sp1}(t), n_{fp1}(t), n_{sp2}(t), n_{fp2}(t) \text{ каде } t > 0\}$, каде $n_{cp}(t)$, $n_{sp1}(t)$, $n_{fp1}(t)$, $n_{sp2}(t)$, и $n_{fp2}(t)$ го претставуваат бројот на пакети во контролерот, комутаторот 1 (класа SP), комутаторот 1 (класа FP), комутаторот 2 (класа SP), и комутаторот 2 (класа FP), соодветно. Се зема дека v, w, x, y, z претставуваат множество вредности за $n_{cp}(t)$, $n_{sp1}(t)$, $n_{fp1}(t)$, $n_{sp2}(t)$, и $n_{fp2}(t)$, каде $v \in Z_+$, $w \in Z_+^{(\leq K1)}$, $x \in Z_+^{(\leq K2)}$, $y \in Z_+^{(\leq K1)}$, $z \in Z_+^{(\leq K2)}$. Дозволените транзиции на Марковиот процес се дадени во табелата 6.2.

Табела 6.2. Дозволените транзиции за моделот PFB

	Од	Кон	Рата
Пакет пристигнува во ком. 1, класа FP	(v, w, x, y, z)	(v, w, x+1, y, z)	$N\lambda$
Пакет пристигнува во ком. 2, класа FP	(v, w, x, y, z)	(v, w, x, y, z+1)	$N\lambda$
Пакет се препраќа од ком. 1 кон контролерот, класа FP ⁽¹⁾	(v, 0, x>0, y, z)	(v+1, 0, x-1, y, z)	$(\alpha+\beta)\mu_s$
Пакет се сервисира од контролерот кон ком. 1, класа SP	(v>0, w, x, y, z)	(v-1, w+1, x, y, z)	μ_c
Пакет се сервисира од контролерот кон ком. 2, класа SP	(v>0, w, x, y, z)	(v-1, w, x, y+1, z)	$\alpha\mu_c$
Пакет се препраќа од ком. 2 кон контролерот, класа SP	(v, w, x, y>0, z)	(v+1, w, x, y-1, z)	μ_s
Пакет се препраќа од ком. 2 кон контролерот, класа FP	(v, w, x, 0, z>0)	(v+1, w, x, 0, z-1)	$\beta\mu_s$
Пакет се сервисира од контролерот кон ком. 1, класа SP ⁽⁴⁾	(v>0, w, x, y, z)	(v-1, w+1, x, y, z)	$\alpha\mu_c$
Пакет се сервисира од контролерот кон ком. 2, класа SP ⁽⁴⁾	(v>0, w, x, y, z)	(v-1, w, x, y+1, z)	$(1+\alpha)\mu_c$
Пакет се отфрла од ком. 1, класа SP ⁽⁵⁾	(v, w>0, x, y, z)	(v, w-1, x, y, z)	$\alpha\mu_s$
Пакет се отфрла од ком. 2, класа SP ⁽⁵⁾	(v, w, x, y>0, z)	(v, w, x, y-1, z)	$\alpha\mu_s$
Пакет заминува од ком. 1, класа FP	(v, 0, x>0, y, z)	(v, 0, x-1, y, z)	$(1-\beta-\alpha)\mu_s$
Пакет заминува од ком. 2, класа FP	(v, w, x, 0, z>0)	(v, w, x, 0, z-1)	$(1-\beta)\mu_s$
Пакет заминува од ком. 1, класа SP	(v, w>0, x, y, z)	(v, w-1, x, y, z)	μ_s
Пакет заминува од ком. 2, класа SP	(v, w, x, y>0, z)	(v, w, x, y-1, z)	μ_s

Повторно се зема претпоставката дефинирана во 6.5.1 и се смета дека $(\alpha+\beta)\mu_s \sim \beta\mu_s$, $(1-\beta-\alpha)\mu_s \sim (1-\beta)\mu_s$, и $(1+\alpha)\mu_c \sim \mu_c$. Овие апроксимации се користат при пресметка на подматриците на QBD процесот (матрицата на генераторот на транзициската рата Q) и при тоа се користат дозволените состојби дефинирани во табелата 6.2. Овие матрици потоа се влез во МАМ со цел да се пресметаат веројатностите на стационарната дистрибуција.

Подматрицата A_0 ја дефинира фазната дистрибуција кога бројот на пакети во контролерот се зголемува за 1:

$$\begin{aligned}
 A_0(w, w') &= \begin{cases} A_{01}, & w' = 0, \\ A_{02}, & w' = w, \\ 0, & \text{поинаку,} \end{cases} \\
 A_{01}(x, x') &= \begin{cases} A_{011}, & x' = x - 1, \\ 0, & \text{поинаку,} \end{cases} \\
 A_{011}(y, y') &= \begin{cases} A_{0111}, & y' = y, \\ 0, & \text{поинаку,} \end{cases} \\
 A_{0111}(z, z') &= \begin{cases} \beta\mu_s, & z' = z, \\ 0, & \text{поинаку,} \end{cases} \\
 A_{02}(x, x') &= \begin{cases} A_{021}, & x' = x, \\ 0, & \text{поинаку,} \end{cases} \\
 A_{021}(y, y') &= \begin{cases} A_{0211}, & y' = 0, \\ A_{0212}, & y' = y - 1, \\ 0, & \text{поинаку,} \end{cases} \\
 A_{0211}(z, z') &= \begin{cases} \beta\mu_s, & z' = z - 1, \\ 0, & \text{поинаку,} \end{cases} \\
 A_{0212}(z, z') &= \begin{cases} \mu_s, & z' = z, \\ 0, & \text{поинаку.} \end{cases}
 \end{aligned}$$

На сличен начин, под-матрицата A_2 ја дефинира фазната дистрибуција кога бројот на пакети во контролерот се намалува за 1:

$$\begin{aligned}
 A_2(w, w') &= \begin{cases} A_{21}, & w' = w, \\ A_{22}, & w' = w + 1, \\ 0, & \text{поинаку,} \end{cases} \\
 A_{21}(x, x') &= \begin{cases} A_{211}, & x' = x, \\ 0, & \text{поинаку,} \end{cases} \\
 A_{211}(y, y') &= \begin{cases} A_{2111}, & y' = y + 1, \\ 0, & \text{поинаку,} \end{cases} \\
 A_{2111}(z, z') &= \begin{cases} \mu_c, & z' = z, \\ 0, & \text{поинаку,} \end{cases} \\
 A_{22}(x, x') &= \begin{cases} A_{221}, & x' = x, \\ 0, & \text{поинаку,} \end{cases} \\
 A_{221}(y, y') &= \begin{cases} A_{2211}, & y' = y, \\ 0, & \text{поинаку.} \end{cases} \\
 A_{2211}(z, z') &= \begin{cases} \mu_c, & z' = z, \\ 0, & \text{поинаку.} \end{cases}
 \end{aligned}$$

Под-матрицата A_1 , ја претставува фазната дистрибуција кога бројот на пакети во контролерот не се менува и притоа има пакети во баферот на контролерот:

$$\begin{aligned}
 A_1(w, w') &= \begin{cases} A_{11}, & w' = 0, \\ A_{12}, & w' = w, \\ A_{13}, & w' = w - 1, \\ 0, & \text{поинаку,} \end{cases} \\
 A_{11}(x, x') &= \begin{cases} A_{111}, & x' = x - 1, \\ 0, & \text{поинаку,} \end{cases} \\
 A_{111}(y, y') &= \begin{cases} A_{1111}, & y' = y, \\ 0, & \text{поинаку,} \end{cases} \\
 A_{1111}(z, z') &= \begin{cases} (1 - \beta)\mu_s, & z' = z, \\ 0, & \text{поинаку,} \end{cases}
 \end{aligned}$$

$$\begin{aligned}
A_{12}(x, x') &= \begin{cases} A_{121}, & x' = x, \\ A_{122}, & x' = x + 1, \\ 0, & \text{поинаку,} \end{cases} \\
A_{121}(y, y') &= \begin{cases} A_{1211} & y' = 0, \\ A_{1212}, & y' = y, \\ A_{1213}, & y' = y - 1, \\ 0, & \text{поинаку,} \end{cases} \\
A_{1211}(z, z') &= \begin{cases} (1 - \beta)\mu_s, & z' = z - 1, \\ 0, & \text{поинаку,} \end{cases} \\
A_{1212}(z, z') &= \begin{cases} N\lambda, & z' = z + 1, \\ 0, & \text{поинаку,} \end{cases} \\
A_{1213}(z, z') &= \begin{cases} \mu_s, & z' = z, \\ 0, & \text{поинаку,} \end{cases} \\
A_{122}(y, y') &= \begin{cases} A_{1221}, & y' = y, \\ 0, & \text{поинаку,} \end{cases} \\
A_{1221}(z, z') &= \begin{cases} N\lambda, & z' = z, \\ 0, & \text{поинаку,} \end{cases} \\
A_{13}(x, x') &= \begin{cases} A_{131}, & x' = x, \\ 0, & \text{поинаку,} \end{cases} \\
A_{131}(y, y') &= \begin{cases} A_{1311}, & y' = y, \\ 0, & \text{поинаку,} \end{cases} \\
A_{1311}(z, z') &= \begin{cases} \mu_s, & z' = z, \\ 0, & \text{поинаку.} \end{cases}
\end{aligned}$$

Дијагоналните елементи на A_{12} , каде $z'=z$, имаат четири засебни случаи:

1. $w=0, x=0$

$$A_{1212}(z, z') = \begin{cases} -2N\lambda - \mu_c, & y = 0, \quad z = 0, \\ -2N\lambda - \mu_c - \mu_s, & y = 0, \quad 0 < z < K_2, \\ -N\lambda - \mu_c - \mu_s, & y = 0, \quad z = K_2, \\ -N\lambda - \mu_c - \mu_s, & 0 < y \leq K_1, \quad 0 \leq z \leq K_2, \\ 0, & \text{поинаку.} \end{cases}$$

2. $w=0, 0 < x < K_2$

$$A_{1212}(z, z') = \begin{cases} -2N\lambda - \mu_c - \mu_s, & y = 0, \quad z = 0, \\ -2N\lambda - \mu_c - 2\mu_s, & y = 0, \quad 0 < z < K_2, \\ -N\lambda - \mu_c - 2\mu_s, & y = 0, \quad z = K_2, \\ -N\lambda - \mu_c - 2\mu_s, & 0 < y \leq K_1, \quad 0 \leq z \leq K_2, \\ 0, & \text{поинаку.} \end{cases}$$

3. $w=0, x=K_2$ и $0 < w < K_1, 0 \leq x \leq K_2$

$$A_{1212}(z, z') = \begin{cases} -N\lambda - \mu_c - \mu_s, & y = 0, \quad z = 0, \\ -N\lambda - \mu_c - 2\mu_s, & y = 0, \quad 0 < z < K_2, \\ -\mu_c - 2\mu_s, & y = 0, \quad z = K_2, \\ -\mu_c - 2\mu_s, & 0 < y \leq K_1, \quad 0 \leq z \leq K_2, \\ 0, & \text{поинаку.} \end{cases}$$

4. $w=K_1, 0 \leq x \leq K_2$

$$A_{1212}(z, z') = \begin{cases} -N\lambda - \mu_c - \mu_s, & y = 0, \quad z = 0, \\ -N\lambda - \mu_c - 2\mu_s, & y = 0, \quad 0 < z < K_2, \\ -2\mu_s, & y = 0, \quad z = K_2, \\ -2\mu_s, & 0 < y \leq K_1, \quad 0 \leq z \leq K_2, \\ 0, & \text{поинаку.} \end{cases}$$

Конечно, под-матрицата B_1 ја претставува фазната дистрибуција кога бројот на пакети во контролерот не се менува и е еднаков на нула:

$$B_1 = A_1 (\mu_c = 0).$$

Целта е да се пресметаат веројатностите на стационарната дистрибуција $(\pi_{v,w,x,y,z})$, за v пакети во конторолерот w пакети во комутаторот 1 (класа SP), x пакети во комутаторот 1 (класа FP), y пакети во комутаторот 2 (класа SP), z пакети во комутаторот 2 (класа FP). Протоколот на класата FP во комутаторот 1 е сума од веројатностите дека оваа класа има најмалку еден пакет за препраќање со средна рата на опслужување μ_s :

$$T_{fp1} = \mu_s \sum_{v=0}^{\infty} \sum_{x=1}^{K2} \sum_{y=0}^{K1} \sum_{z=0}^{K2} \pi_{v,0,x,y,z}. \quad (6.52)$$

Слично, протоколот на класата SP во комутаторот 2 и заедничката класа CP, се дадени како:

$$\begin{aligned} T_{sp2} &= \mu_s \sum_{v=0}^{\infty} \sum_{w=0}^{K1} \sum_{x=0}^{K2} \sum_{y=1}^{K1} \sum_{z=0}^{K2} \pi_{v,w,x,y,z}, \\ T_{cp} &= \mu_c \sum_{v=1}^{\infty} \sum_{w=0}^{K1} \sum_{x=0}^{K2} \sum_{y=0}^{K1} \sum_{z=0}^{K2} \pi_{v,w,x,y,z}. \end{aligned} \quad (6.53)$$

Просечната должина на ред во класата FP во комутаторот 1 е одредена со следниот израз:

$$E(N)_{fp1} = \sum_{w=1}^{K2} \pi_x \times x, \quad (6.54)$$

каде, π_x е маргиналната веројатност за просечните пакети во класата FP во комутаторот 1 и е дефинирана како:

$$\pi_x = \sum_{v=0}^{\infty} \sum_{w=0}^{K1} \sum_{y=0}^{K1} \sum_{z=0}^{K2} \pi(v, w, x, y, z). \quad (6.55)$$

Слично, просечната должина на редот во класата SP во комутаторот 2 ($E(N)_{sp2}$) и класата CP ($E(N)_{cp}$) е следната:

$$\begin{aligned} E(N)_{sp2} &= \sum_{y=1}^{K1} \pi_y \times y, \\ \pi_y &= \sum_{v=0}^{\infty} \sum_{w=0}^{K1} \sum_{x=0}^{K2} \sum_{z=0}^{K2} \pi(v, w, x, y, z), \\ E(N)_{cp} &= \sum_{x=1}^{\infty} \pi_v \times v, \\ \pi_v &= \sum_{w=0}^{K1} \sum_{x=0}^{K2} \sum_{y=0}^{K1} \sum_{z=0}^{K2} \pi(v, w, x, y, z). \end{aligned} \quad (6.56)$$

Се користи теоремата на Литл за да се пресмета просечното време на процесирање на пакет во секоја од трите класи (t_{fp1} , t_{sp2} , t_{cp}):

$$\begin{aligned}
t_{fp1} &= \frac{E(N)_{fp1}}{T_{fp1}}, \\
t_{sp2} &= \frac{E(N)_{sp2}}{T_{sp2}}, \\
t_{cp} &= \frac{E(N)_{cp}}{T_{cp}}.
\end{aligned} \tag{6.57}$$

Вкупното доцнење при хендовер кое го доживува MN е сума од следните компоненти:

$$t_{tot}^{pfb} = t_{fp1} + 2t_{cp} + 2t_{sp2}. \tag{6.58}$$

Повторно, целта е да се истражи проблемот на димензионирање на баферот, овој пат за моделот PFB. Слично како во претходното поглавје, се користи M/M/1 ред за моделирање на комутаторот и се дефинира минимален капацитет на баферот (K_{1min} е за класата SP и K_{2min} е за класата FP) како:

$$\begin{aligned}
K_{1min} &\geq \frac{\log[PER]}{\log[\rho_1]} \times 100\%, \\
K_{2min} &\geq \frac{\log[PER]}{\log[\rho_2]} \times 100\%,
\end{aligned} \tag{6.59}$$

каде ρ_1 е искористувањето на класата FP, а ρ_2 е искористувањето на класата SP во комутаторот, кои се дефинираат како:

$$\begin{aligned}
\rho_1 &= \frac{N\lambda}{\mu_s}, \\
\rho_2 &= \frac{\beta N\lambda}{\mu_s}.
\end{aligned} \tag{6.60}$$

Минималниот капацитет на баферот е сума од K_1 and K_2 :

$$K_{min}^{pfb} = K_1 + K_2. \tag{6.61}$$

6.5.3 Анализа на перформанси

Во оваа глава, основната цел е да се споредат моделот SFB и моделот PFB во смисла на вкупно доцнење при хендовер кое MN го доживува при “тврд” хендовер. Дополнително, се истражуваат и одредени аспекти на димензионирање на баферот во комутаторот со намера да се разбере кој од двата понудени модели овозможува подобри резултати и под кои услови. Табелата 3 ги дефинира користените параметарски вредности. Веројатноста на наидување на *Packet-in* пораките варира во опсегот од 0.1 до 1, додека средната рата на опслужување е фиксирана на 10000 pkt/s. Средната рата на опслужување на контролер-комутатор, $r = \mu_c / \mu_s$, е во опсегот од 0.1 – 2, а надворешната рата на пристигнувањето на пакетите по MN е 400 pkt/s и 600 pkt/s. *Maximum transmission unit* (MTU) е 1500 Bytes, BER е 10^{-12} (PER е 1.2×10^{-8}), а бројот на MN е во рангот 1 - 40. Во MATLAB се врши нумеричка евалуација, но исто така се извршуваат и симулации користејќи дискретен симулатор на

настани, со цел валидација на предложениот аналитички модел. Се користи Monte Carlo и симулациите се повторуваат 50 пати.

Табела 6.3. Дефиниција на вредности на параметрите

Параметар	Вредност
Веројатност на наидување на <i>Packet-In</i> пораки, β	0.1 - 1
Средно рата на опслужување на ком., μ_s (pkt/s)	10000
Средна рата на опслужување контролер-кон-ком., r	0.1 - 2
Рата на пристигнување, λ (pkt/sec)	400, 600
<i>Bit Error Rate</i> , BER	10^{-12}
Големина на пакет (Bytes)	1500
Број на MN по комутатор, N	1 - 40

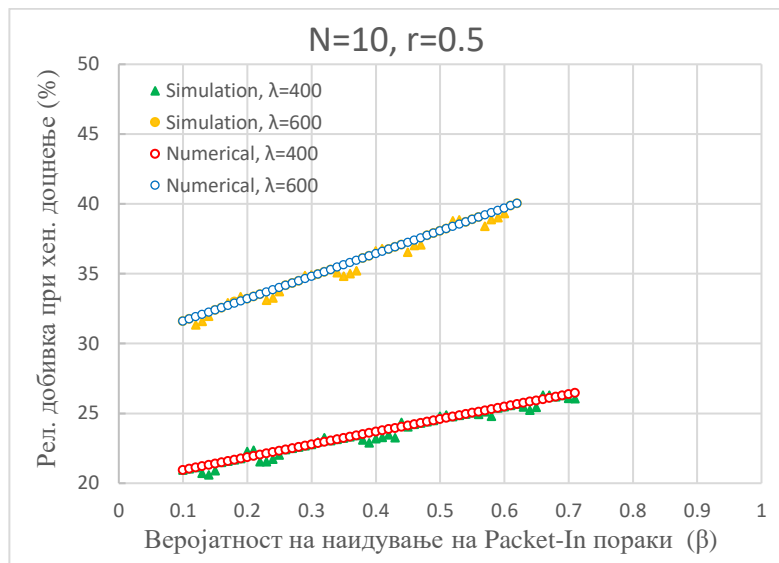
Намерата е да се спореди доцнењето при хендовер за двата предложени модели (претходно означени како t_{tot}^{sfb} и t_{tot}^{pfb} , а дефинирани во равенките (6.48) и (6.58)). Се дефинира нова перформансна метрика наречена релативна добивка при доцнење од хендовер (на англ. *relative handover delay gain*), t_{rel} , на следниот начин:

$$t_{rel} = \frac{(t_{tot}^{sfb} - t_{tot}^{pfb})}{t_{tot}^{sfb}} \times 100\%. \quad (6.62)$$

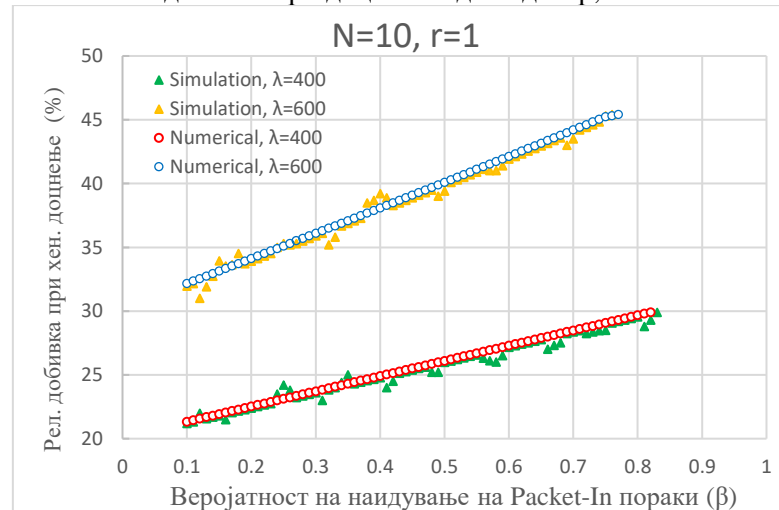
Позитивна вредност на t_{rel} значи дека моделот PFB има пониско вкупно доцнење при хендовер во однос на моделот SFB.

Сликата 6.18а ја покажува релативната добивка при доцнење од хендовер за различни вредности на β и промена на параметарот r ($r=0.5$, $r=1$, $r=2$ на сликите 6.18а, 6.18б и 6.18с, соодветно) и подразбира фиксен број на MN, $N = 10$. Се анализираат две различни вредности на ратата на пристигнување, 400 pkt/s и 600 pkt/s. Гледајќи ги резултатите, прво се заклучува дека симулациските резултати се во согласност со нумеричките резултати, разликата помеѓу нумеричките и симулациските резултати е помеѓу 1.6% - 3.3%. Како што се зголемува β , се зголемува и релативната добивка при доцнење од хендовер и како што се зголемува λ , релативната добивка при доцнење од хендовер исто така има повисоки вредности и се зголемува дури и побрзо со раснење на β . Кога се споредуваат сликите 6.18а, 6.18б и 6.18с, заклучокот е дека како што r се зголемува, t_{rel} исто така се зголемува и за $r=2$, $\beta \sim 0.84$, t_{rel} е околу 50%. Исто така, кога r расне, ова во основа значи дека средното време на опслужување на контролерот се зголемува и од резултатите се забележува подобрување во t_{rel} , што е уште повидливо при повисоки рати на пристигнување. Очигледно е дека моделот PFB покажува супериорни резултати во однос на моделот SFB, а ова е поради дедицираниот приоритетен ред за контролниот сообраќај кој значајно го намалува доцнењето во контролната рамнина поради природата на приоритизацијата. Понатаму, се заклучува дека користење на бафери со приоритет го подобрува корисничкото искуство

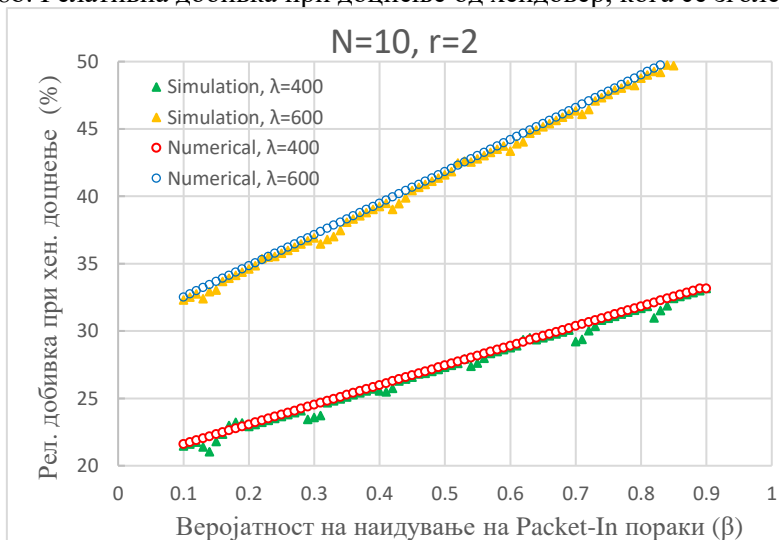
поради пократкото доцнење при хендовер во сите анализирани сценарија на варијабилни β и r .



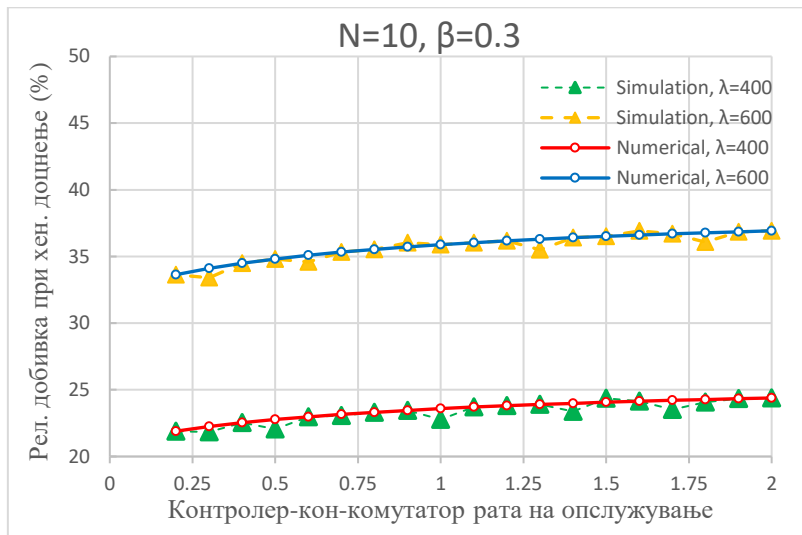
Слика 6.18а. Релативна добивка при доцнење од хендовер, кога се зголемува β ($r=0.5$)



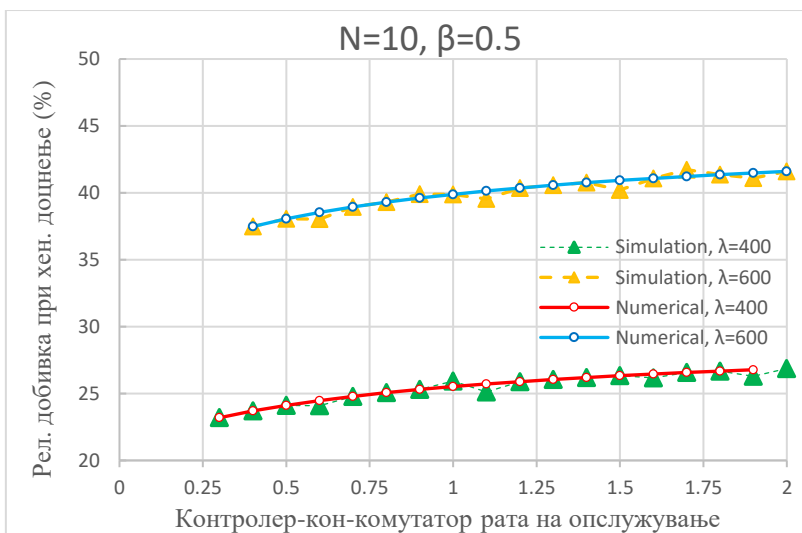
Слика 6.18б. Релативна добивка при доцнење од хендовер, кога се зголемува β ($r=1$)



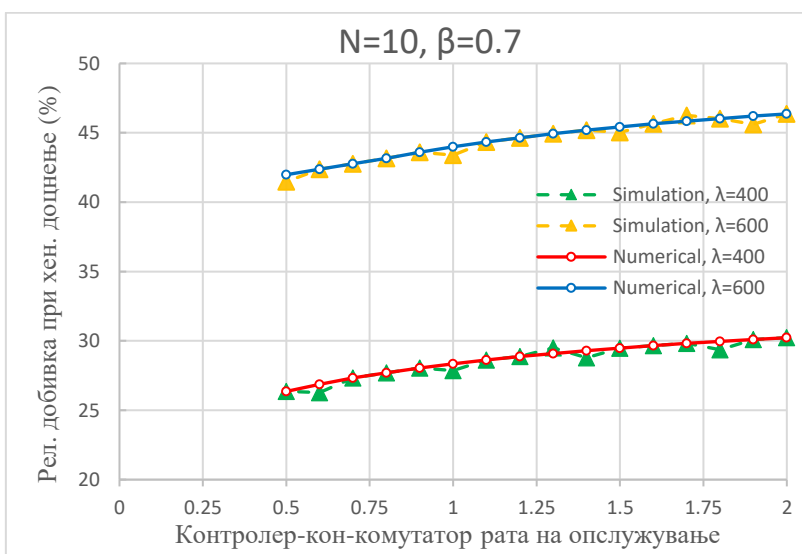
Слика 6.18с. Релативна добивка при доцнење од хендовер, кога се зголемува β ($r=2$)



Слика 6.19а. Релативна добивка при доцнење од хендовер, кога се зголемува r ($\beta=0.3$)



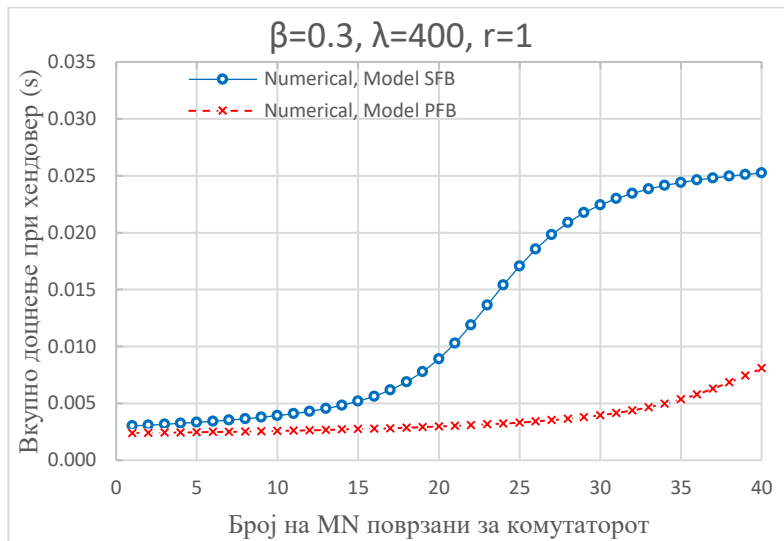
Слика 6.19б. Релативна добивка при доцнење од хендовер, кога се зголемува r ($\beta=0.5$)



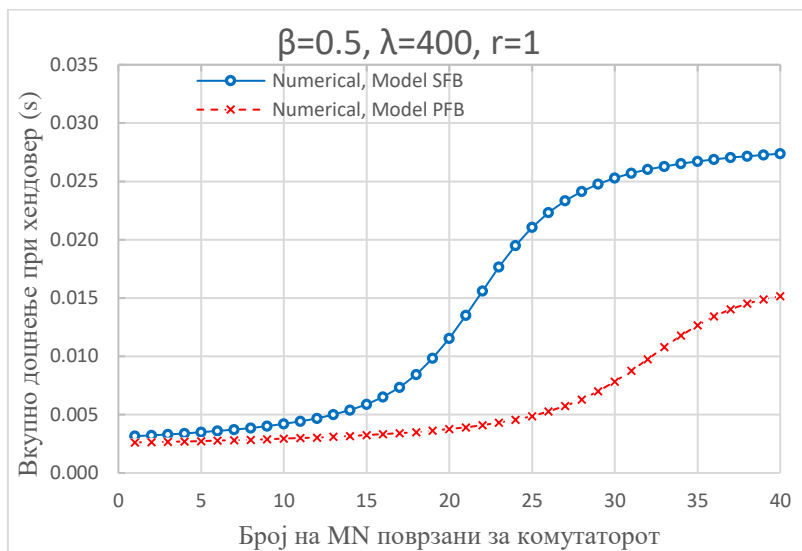
Слика 6.19с. Релативна добивка при доцнење од хендовер, кога се зголемува r ($\beta=0.7$)

Слично како на сликата 6.18, на сликата 6.19 се анализира t_{rel} , но сега за променлив r на X-оската. Повторно, симулациските резултати доста точно ги следат нумеричките вредности, разликата помеѓу нумеричките и симулациските резултати е помеѓу 1.4% - 2.9%. На сликите 6.19a, 6.19b и 6.19c, се истражува ефектот на променлив β за три различни вредности: 0.3, 0.5 and 0.7. Ако се анализира случајот за $\lambda=600$ на сликата 6.18a, кога се споредуваат $r=0.5$ и $r=2$, t_{rel} се зголемува од 33.5% на 37%, или пораст од 3.5%. Разликата во промена на вредноста е слична и на сликите 6.18b и 6.18c, па може да се заклучи дека со зголемување на средната рата на опслужување на контролерот, се врши мало влијание на релативната добивка при доцнење од хендовер кога β има константна вредност. Од друга страна, зголемувањето на β од 0.3 до 0.7 (слика 6.18a до слика 6.18c) има многу повидливо влијание на t_{rel} . Може да се забележи пораст од 10% за $\lambda=600$ кога се споредуваат сликите 6.18a и 6.18c. Се заклучува дека користењето на приоритетни бафери влијае позитивно на корисничкото искуство поради пократкото доцнење при хендовер во сите анализирани сценарија на варијабилни β и r .

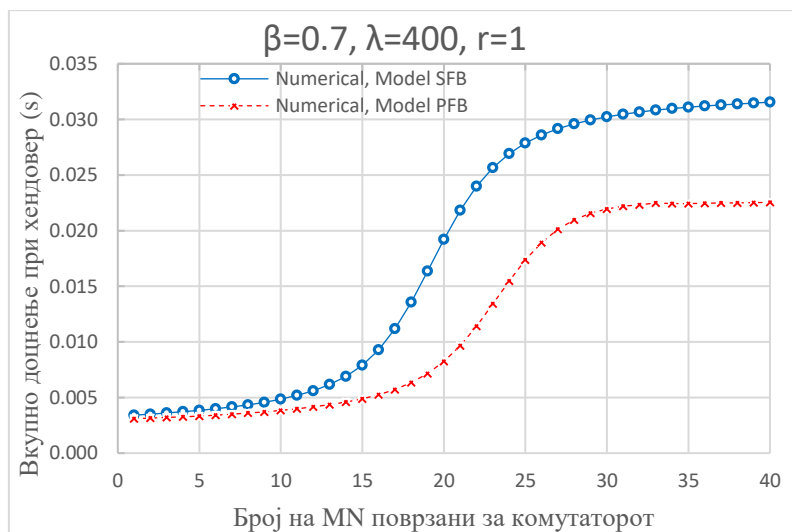
Понатаму, целта е да се анализира ефектот на промена на бројот на MN поврзани за комутаторот. Претходно, се претпоставуваше фиксен број на MN, $N=10$, но овој пат N се менува помеѓу 1 и 40 и се анализира ефектот врз вкупното време на доцнење при хендовер. Дополнително, овој пат се цели кон одредување на апсолутната вредност на доцнењето при хендовер (во секунди) за да се разбере магнитудата на доцнењето кое го доживува MN кога врши “тврд” хендовер. Во сите три случаи (слика 6.20a до 6.20c), $r=1$, додека β се менува за три различни вредности 0.3, 0.5 и 0.7. Кога се гледа на сликата 6.20, се забележува дека со зголемување на бројот на MN, t_{rel} е значајно понизок за моделот PFB споредено со моделот SFB. Уште повеќе, како што бројот на MN се зголемува, разликата во вкупното доцнење при хендовер за моделот PFB е повидлива. За помал број на MN, апсолутните вредности на вкупното доцнење при хендовер во случај на моделот PFB имаат тенденција да бидат во рангот на милисекунди, до 8 ms, 15 ms и 23 ms кога β се менува со вредности од 0.3, 0.5 и 0.7, соодветно. Кога се користи моделот SFB, истите вредности се во рангот на милисекунди, но сега до максимум 25 ms, 28 ms and 32 ms кога β се менува со вредности од 0.3, 0.5 and 0.7, соодветно. Бидејќи во мобилните мрежи се очекуваат голем број на корисници, се заклучува дека користењето на бафери со приоритизација значително го намалува вкупното доцнење при хендовер и треба да биде префериран избор за користење во мобилните мрежи.



Слика 6.20а. Вкупно доцнење при хендовер кога се менува бројот на MN ($\beta=0.3$)



Слика 6.20б. Вкупно доцнење при хендовер кога се менува бројот на MN ($\beta=0.5$)

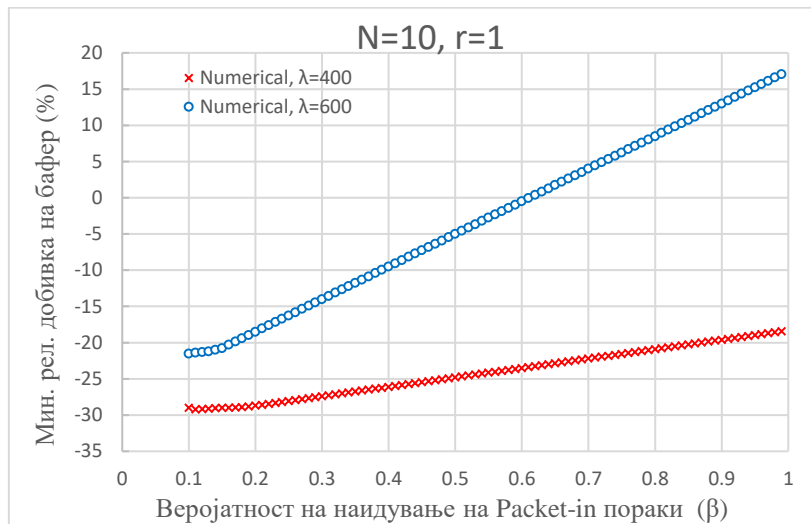


Слика 6.20с. Вкупно доцнење при хендовер кога се менува бројот на MN ($\beta=0.7$)

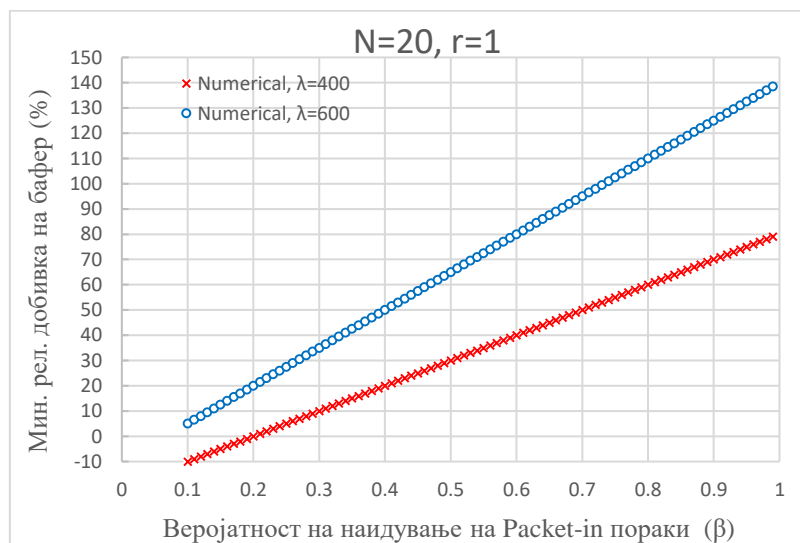
Се дефинира нова перформансна метрика со цел да се истражи димензионирањето на баферот кога се споредуваат двата модели. Релативната добивка на минимален капацитет во баферот, K_{min}^{rel} се дефинира како:

$$K_{min}^{rel} = \frac{(K_{min}^{sfb} - K_{min}^{pfb})}{K_{min}^{sfb}} \times 100, \quad (6.63)$$

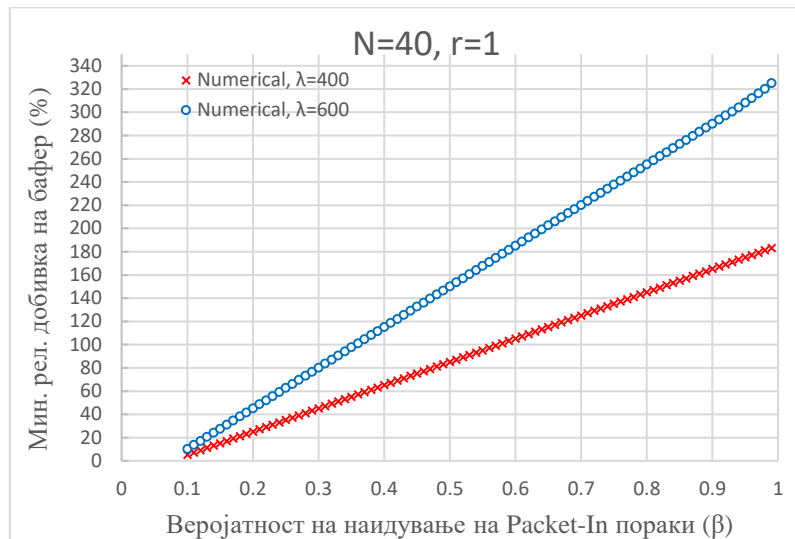
каде K_{min}^{sfb} and K_{min}^{pfb} се дефинирани во равенката (6.49) и равенката (6.61), соодветно. Релативната добивка на минимален капацитет во баферот ја дава процентуалната разлика во минималната должина на баферот која е потребна да се постигне одредена веројатност на загуби на линкот (во овој случај $BER=10^{-12}$). Сликата 6.21 ги прикажува кривите за K_{min}^{rel} при варијабилно β . Негативна вредност на K_{min}^{rel} значи дека моделот PFB бара повеќе капацитет од моделот SFB, додека во спротивниот случај, позитивна вредност имплицира резултати за минимален бафер во корист на моделот PFB.



Слика. 6.21a. Релативна добивка на минимален капацитет во баферот ($N=10$)



Слика. 6.21b. Релативна добивка на минимален капацитет во баферот ($N=20$)



Слика. 6.21с. Релативна добивка на минимален капацитет во баферот ($N=40$)

Се анализира K_{min}^{rel} за $N=10$, $N=20$, and $N=40$ (слика 6.21a, слика 6.21b, слика 6.22c, соодветно). За $N=10$ (слика 6.21a), кога $\lambda=400$, резултатите се во корист на моделот SFB и се забележуваат негативни вредности на K_{min}^{rel} за сите вредности на β . Кога се анализира истата слика, но за $\lambda=600$, се гледа дека до $\beta = 0.6$, заедничкиот бафер има подобри перформанси, додека за $\beta > 0.6$, моделот PFB побарува помала големина на бафер. За $N=20$ (слика 6.21b), за $\lambda=400$, резултатите се во корист на моделот SFB, но само за $\beta < 0.2$. За повисоки вредности на β , моделот PFB е супериорниот модел. Во случајот на $\lambda=600$, истата слика јасно покажува дека моделот PFB има предност за сите вредности на β , со максимална добивка за $\beta=1$, која достигнува 140%. Последната анализа за $N=40$ (слика 6.21c) покажува дека за двете вредности на ратата на пристигнување, моделот PFB има потреба од помал капацитет на баферот и максималната добивка за $\lambda=600$ и $\beta=1$, е околу 330%. Бидејќи во мобилните мрежи има голем број на корисници, може да се заклучи дека користењето на приоритетни бафери ги намалува потребите во однос на големина на бафер и претставува соодветен избор за сите типови мрежи кои опслужуваат голем број на корисници.

6.6 Заклучок

Во оваа глава се предложени три аналитички модели кои може да се искористат за моделирање на OF-базирани SDN мобилни јадрени мрежи. Со користење на теоријата на редови, се анализира просечната рата на опслужување во комутаторите и контролерите, со конечна цел квантификација на вкупното доцнење при хендовер и анализа на најважните параметри кои влијаат на ова доцнење.

Во првиот аналитички модел, се моделира сценарио со повеќе комутатори и единствен контролер. Нумеричките резултати покажуваат дека ратата на пристигнување на

пакети, бројот на комутатори и оптоварувањето на контролерот се битни фактори при моделирањето на SDN-базирани мрежи. Дополнително, висока веројатност на наидување на *Packet-in* пораки влијае негативно на времето на опслужување и ги деградира перформансите во мрежата. Се утврдува и зависноста на доцнењето при хендовер од вкупниот проток на мрежата, а резултатите може да се искористат од SDN дизајнерите при конструирање на гаранции за максимално доцнење при хендовер и подобрување на корисничкото искуство.

Вториот математички пристап, моделира систем со повеќе контролери во контролната рамнина. Повторно се потврдува дека ратата на пристигнување, оптоварувањето на контролерот, бројот на мрежни елементи (комутатори и контролери) и веројатноста на наидување на *Packet-in* пораките се најкритичните фактори кои влијаат на вкупното доцнење при “тврд” хендовер. Дополнително, се докажува дека употребата на голем број на контролери влијае негативно на времето на опслужување поради големиот број на синхронизациски пораки кои се разменуваат помеѓу соодветните домени со цел освежување на мрежната информација. Се прави и анализа за естимација на оптималниот број на контролери при фиксна рата на синхронизација и фиксна веројатност на наидување на *Packet-in* пораките. Аналитичкиот модел е верифициран со опширни симулации извршени во MATLAB.

Последниот предложен математички пристап, моделира два типа системи со различен тип SDN комутатори: комутатор со единствен делен бафер без приоритизација (модел SFB) и комутатор со два изолирани бафери кои користи приоритизација (модел PFB). Двата системи се споредуваат во однос на релативната добивка при доцнење од хендовер и релативната добивка на минимален капацитет во баферот. Анализата на перформанси покажува дека симулациските резултати со голема точност ги следат соодветните нумерички резултати и може јасно да се увидат предностите и супериорноста на моделот PFB во најголем дел од сценаријата од интерес. Како што расне средната рата на опслужување на контролерот, вкупното доцнење при хендовер се намалува побрзо за моделот PFB, додека при висока веројатност на наидување на *Packet-in* пораките, моделот PFB покажува уште подобри резултати. Исто така, како што расне бројот на MN, доцнењето при хендовер е значајно помало за моделот PFB и може да се увиди > 10 ms разлика во доцнењето при хендовер кога се споредува со моделот SFB. Во однос на анализата за димензионирање на бафер, за пониски рати на пристигнување, мал број на корисници и ниска веројатност на наидување на *Packet-in* пораките, моделот SFB е преферираниот

модел. Како што горе наведените параметри раснат, моделот PFB покажува подобри резултати.

За понатамошна работа, се планира да се инкорпорираат хардверски влијанија во моделите и да се споредат резултатите од предложените модели со реалните хардверски имплементации. Од посебен интерес може да биде поставување на мулти-контролер тест-систем со цел експериментална евалуација на точноста на вториот аналитички модел, при што би се користеле синтетички генератори на сообраќај кои подобро би ја пресликале реалната себеслична природа на интернет сообраќајот. Понатаму, последниот модел може да се прошири преку воведување на дистрибуирана контролна рамнина за да се види ефектот на доцнењето при хендовер и ефектот врз минималната големина на баферот во комутаторите.

7. Заклучок

Примената на SDN во мобилните јадрени мрежи овозможува широко поле за истражување и унапредување на постојните сознанија и концепти во областа. На крајот од оваа дисертација, се јавува потребата за сублимација на сите добиени резултати, како и каде може да се применат овие резултати и кои се можните идни насоки за проширување и подобрување на квалитетот на истражувањата кои беа претставени. Софтверски дефинираното омрежување е генерален концепт кој овозможува многу придобивки за мобилните оператори: програмабилност преку отворени интерфејси, намалување на оперативните трошоци, многу полесна иновација и воведување на нови услуги. Во LTE, SDN концептот е воведен подоцна, преку надградба на EPC во 5G EPC, додека SDN е длабоко втемелен во основата на архитектурата на новата 5та генерација мобилни мрежи. Во LTE не постои гаранција за минимална латентност, додека 5Г мрежата мора да гарантира одредено минимално ниво на латентност со цел да се задоволат барањата на новите кориснички случаи. Во рамките на оваа дисертација, прво детално беа испитани сите фактори кои влијаат на латентноста, а потоа се предложи алгоритам за брза детекција и опоравување од испад за кој се докажа дека може ефикасно да се примени во 5Г. Понатаму, преку користење на теоријата на редови, беа конструирани неколку аналитички модели кои може да се искористат од SDN дизајнерите при моделирањето на SDN-базирани јадрени мрежи. Овие математички пристапи беа детално нумерички евалуирани, па беа споредени со симулациските резултати и беше јасно покажано дека аналитичките модели претставуваат брз и ефективен начин за предлагање на нови механизми и разбирање на зависностите помеѓу најважните параметри кои влијаат на доцнењето во SDN мрежите.

Предмет на истражувањето во рамките на оваа дисертација беше анализата на перформансите во SDN-базирани јадрени мобилни мрежи. Основна цел на истражувањето беше да се докаже дека SDN може ефикасно да се употреби во LTE и 5Г мобилните мрежи поради тоа што генерално позитивно влијае на намалување на латентноста, времето на опоравување од испад може да се минимизира со цел да се задоволат 5Г побарувањата и дека корисничкото искуство при изведување на процедурата на хендовер може да се подобри преку намалување на вкупното доцнење поради размена на OF пораките. Значењето на истражувањата во рамки на оваа дисертација беше јасно да се покаже дека теоријата на редови и аналитичкото моделирање може ефикасно да се искористат за брзо и евтино моделирање на SDN-базирани мобилни мрежи, како прв чекор пред имплементација во тестен систем. Понатаму, целта беше да се истражи и преку симулации да се докаже, дека се уште постојат предлози за усовршување на OF стандардот. Беа предложени два

аналитички модели кои го моделираат доцнењето во SDN-базирани мобилни мрежи и го анализираат влијанието на основните претпоставки употребени од истражувачката заедница: неограничен бафер и рата на опслужување која е независна од оптоварувањето. Конечно, преку моделирањето на OF-базиран хендовер во LTE/5G, се покажа дека процедурата на хендовер може да се изврши само во рамки на стандардниот OF протокол, без никаква потреба за додавање на нови протоколи или дополнителен сообраќај заради сигнализација. Анализата покажа дека при дистрибуирана контролна рамнина, треба да се води сметка за магнитудата на разменетите синхронизациски и инвокациски пораки. Дополнително, детално се истражи и употребата на бафери кои користат приоритизација и нивниот бенефит при употребата во мобилните јадрени мрежи.

Во главата 2 се анализираше SDN концептот и најпопуларниот јужен протокол, *OpenFlow*. Потребата за вклучување на контролерот за било кое процесирање на состојба и за секое освежување на *match/action* полињата, води кон непотребна деградација на перформансите на мрежата, зголемување на процесирачкото доцнење и воведува дополнително сигнализациско оптоварување. Поради ова, родена е идејата за воведување на корисничка рамнина на состојби, каде се прави проверка за совпаѓање на секој пакетски тек кон една нова табела на состојба. На овој начин се овозможува брзо опоравување од испад директно во корисничката рамнина без потреба за контактирање на контролерот. Детално беа дискутирани пристапите за анализа на перформанси во SDN мрежи, со посебен осврт на употребата на софтверски-базирани алатки за симулации и математичкото моделирање, двата пристапи кои беа употребени во рамки на оваа дисертација. Конечно, беше направен краток осврт на теоријата на редови и QBD процесот, како вовед кон аналитичките модели предложени во наредните глави.

Главата 3 ја анализираше примената на SDN во рамки на мобилните комерцијални мрежи, со посебен осврт на конституентите на латентност кои ја сочинуваат вкупната латентност и кои технологии ефективно се користат за намалување на доцнењето. Дискусијата беше со цел да се разберат строгите барања за латентност во 5G и генерално ниската толеранција на било каков тип на доцнење. Иако не постојат експлицитни 5G барања за времето на опоравување од грешка во мобилните јадрени мрежи, сепак се заклучи дека се очекуваат многу строги барања ($< 20\text{ms}$). Понатаму, се анализираше специфично сценарио за софтверска надградба на постојните LTE пакетски мрежи, односно концептот на парцијална виртуелизација на EPC, кој овозможува максимално реискористување на постојната опрема и најбрз начин за воведување на 5G EPC. Се даде и детална анализа на постојните типови и механизми за опоравување од грешка и јасно се одредија категориите

според кои се класифицираше предлогот за нов механизам за опоравување од испад. На крајот се разгледа процедурата на хендовер која се изведува исклучиво со размена на OF сигнални пораки, која подоцна е искористена во главата 6 како основа за моделирање на доцнењето при хендовер.

Следуваат три глави кои директно беа поврзани со објавените трудови во рамки на истражувањата при оваа дисертација. Главата 4 го деталizираше предлогот за нов механизам за брза детекција и корекција на грешка, имплементиран директно во корисничката рамнина, кој се нарекува *Stateful Fast Failover (SFF)*. Предложениот механизам е базиран на OF FF и на механизмот опишан во [36] SPIDER, кој всушност е екстензија на популарниот концепт на “симнување” на дел од интелигенција во контролерот, наречен *OpenState*. Предложеното решение интелегентно ги комбинира најдобрите страни на SPIDER и OF FF и е дизајнирано да го оптимизира времето на опоравување во случај на единствена далечна или локална грешка. Резултатите од симулациите ги покажаа перформансите на протоколот и потврдија дека времето на опоравување од грешка може да се задржи под 20 ms, што е во согласност со *carrier-grade* мрежните барања. Дополнително, преку симулации се потврди дека решенијата кои не вклучуваат контакт со SDNC мора да бидат имплементирани директно во корисничката рамнина бидејќи волуменот на OF контролниот сообраќај и времето на опоравување од испад не се во согласност со 5G очекувањата.

Во главата 5, акцентот беше ставен на мрежната латентност која е воведена помеѓу комутаторите и контролерот, која има и најголемо и најкритично влијание на целокупните SDN перформанси. Дополнително, се анализираа предностите и недостатоците при користење на единствен контролер наспроти употребата на дистрибуирана контролна рамнина. Се заклучи дека иако постојат перформансни проблеми при користење на еден контролер, сепак мора внимателно да се дизајнира контролната рамнина преку додавање на повеќе контролери. Се даде предлог за нов математички пристап со чија помош може да се моделира латентноста крај-до-крај и латентноста поради процесирање во контролерот. Се заклучи дека доминантен фактор кој влијае на латентноста во SDN делот на јадрената мрежа претставува времето на процесирање на пакетите во контролерот. Резултатите покажаа дека преку додавање на повеќе контролери во системот, не се афектира времето на опслужување при повисоки вредности на оптоварувањето, во споредба со случајот кога се користи помал број на контролери. Дополнително, се анализираа и најчестите хипотези кои редовно се користат при моделирање на SDN мрежи: неограничен бафер и рата на опслужување која е независна од оптоварувањето. Се предложи нов аналитички модел за рата на опслужување

која зависи од состојбата и ограничен бафер и се изведоа равенки со чија помош се мери влијанието на земените хипотези при споредба со стандарден математички модел кој не ги зема во предвид. За претпоставката неограничен бафер, средното време на опслужување е пониско, споредено со случајот кога баферот има конечна вредност, во кој случај средното време на опслужување се влошува побрзо. За претпоставката на варијабилна средна рата на опслужување, деградацијата е многу повидлива, особено ако l е пониско. Конечно, се заклучи дека и двете хипотези имаат значително влијание на точноста на моделирањето и ова влијание мора да се земе во предвид при моделирање на SDN мрежи.

Во последната Глава 6, фокусот беше ставен на хендоверот на мобилен корисник помеѓу различни комутатори, а целта беше да се моделира доцнењето при хендовер поради OF контролните пораки преку користење на нов математички пристап. Од интерес беше единствен контролер, но и сценарио со мулти-контролери во контролната рамнина. Беа предложени три аналитички модели конструирани со користење на теоријата на редови, при што се анализираше просечната рата на опслужување во комутаторите и контролерите, со конечна цел квантификација на вкупното доцнење при хендовер и анализа на најважните параметри кои влијаат на ова доцнење. Во првиот аналитички модел, се моделираше контролна рамнина со единствен контролер. Нумеричките резултати покажаа дека ратата на пристигнување на пакети, веројатноста на наидување на *Packet-in* пораките, бројот на комутаторите и оптоварувањето на контролерот се најважните фактори при моделирањето на SDN-базираните мрежи. Во наредниот математички пристап, се моделираше систем со повеќе контролери во контролната рамнина. Покрај потврдата на критичните параметри кои влијаат на доцнењето при хендовер, дополнително се докажа дека употребата на преголем број на контролери негативно влијае на времето на опслужување поради големиот број на синхронизациски пораки кои се разменуваат помеѓу соодветните домени со цел освежување на мрежната информација. Се направи и обид за одредување на оптималниот број на контролери при фиксна рата на синхронизација и фиксна веројатност на наидување на *Packet-in* пораките. Третиот предложен аналитички модел, всушност конструира два типа системи кои инкорпорираат два вида SDN комутатори: комутатор со единствен делен бафер без приоритизација (модел SFB) и комутатор со два изолирани бафери кои користи приоритизација (модел PFB). Двата системи се споредуваа во однос на релативната добивка при доцнење од хендовер и релативната добивка на минимален капацитет во баферот. Заклучокот е дека моделот PFB покажува подобри резултати скоро во сите сценарија од интерес. Со пораст на средната рата на опслужување на контролерот, вкупното доцнење при хендовер се намалува побрзо за моделот PFB, додека при висока веројатност на

наидување на *Packet-in* пораките, моделот PFB покажува уште подобри резултати. Супериорноста на PFB е уште повидлива при анализата на доцнењето при хендовер во сценарија со голем број на MN. Што се однесува на делот за димензионирање на бафер, за пониски рати на пристигнување, мал број на корисници и ниска веројатност на наидување на *Packet-in* пораки, моделот PFB покажува подобри резултати. Во спротивен случај, каде горе наведените параметри раснат, моделот PFB е повторно супериорен.

Така, придонесот на оваа дисертација е предлогот за нов механизам за опоравување од испад имплементиран директно во корисничката рамнина, како и нови аналитички модели базирани на теоријата на редови кои ефикасно може да искористат при моделирање на вкупната латентност како и за доцнењето кое се воведува при процедура на “тврдо” хендовер. Во оваа насока, врз база на изложеното, може да се издвојат следните најзначајни придобивки од оваа докторска дисертација:

- Анализа на 5Г барањата за латентност и идентификација на клучните фактори кои влијаат на различните типови доцнење во мобилни јадрени мрежи базирани на SDN,
- Нов предлог за механизам за опоравување од испад дизајниран преку интелигентна интеграција на OF FF и SPIDER решенијата со цел брзо опоравување од испад во случај на локални и далечни испади. Предложениот алгоритам може да се употреби во 5Г јадрените мобилни мрежи бидејќи ги задоволува 5Г барањата за латентност,
- Се предложи математички модел базиран на теоријата на редови за моделирање на средното време на опслужување на комутаторот и контролерот преку користење на M/M/1 и M/M/c моделите на редови, соодветно. Се заклучи дека оптоварувањето и бројот на контролери, ратата на пристигнување на пакети, средната рата на опслужување и веројатноста на наидување на *Packet-in* пораките се главни фактори кои влијаат на латентноста и перформансите во SDN мрежи.
- Се предложи аналитички модел базиран на M/M/m/K и M/M/1 (со рата на опслужување зависна од состојбата) редови за да се квантифицива ефектот на ограничен бафер и ратата на опслужување која зависи од оптоварувањето. Се заклучи дека и двете хипотези имаат значајно влијание врз точноста на моделирањето и мора да се земат во предвид при дизајнирање на SDN мрежи,
- Се даде предлог аналитички пристап за моделирање на доцнењето при “тврдо” хендовер. Комутаторите и контролерот се моделираа како M/M/1 ред и се моделираа само *Packet-in* пораките. Добиените резултати укажуваат дека бројот на комутатори, ратата на пристигнување, веројатноста на наидување на *Packet-in* пораките и

вкупниот системски проток, се критични фактори при моделирањето на SDN-базирани мрежи.

- Се конструираше нов математички пристап за моделирање на перформансите во SDN мрежи со повеќе комутатори, но и повеќе контролери во контролната рамнина. Комутаторите се моделираа како M/M/1 ред, додека контролерите се моделираа како M/PN/1 ред. Се направи експанзија на претходните трудови со тоа што не се моделираа само *Packet-in*, туку и *Ports-status* пораките. *Port-status* пораките се од клучен интерес кога MN изведува “тврд” хендовер и претставуваат првите пораки кои се разменуваат во контролната рамнина при иницијализација на процедурата на хендовер. Заклучокот може да се искористат при дизајнирање на гаранции за доцнењето при хендовер во мобилна мрежа за одреден сообраќај кој се насочува кон комутаторот.
- Се предложи аналитички модел кој ги моделира комутаторите користејќи GI/M/1/K ред и M/M/1 за контролерот. Се конструираа и се споредуваа математички модели за два типа на комутатори: (i) кога се користи единствен делен бафер без приоритизација и (ii) кога се користат два посебни бафери со не-преемтивна приоритизација. Двата системи се споредуваа во однос на доцнењето при хендовер, како и за потребната големина на бафер за да се постигне соодветна рата на пакетска загуба на линк. Заклучокот е дека моделот PFB покажа супериорни резултати и треба да биде преферираниот избор за употреба во мобилни мрежи кои опслужуваат голем број на корисници.

Резултатите од истражувањето во оваа дисертација, може да се искористат од SDN дизајнерите при моделирање на SDN-базирани јадрени мрежи. Во однос на предложениот алгоритам за опоравување од испад, идните екстензии на ова истражување може да бидат во насока на проширување на предложениот протокол со цел користење на истиот за опоравување од повеќекратни грешки. Во однос на аналитичките модели изложени во главата 5, предлог за идна работа претставува насоката која се однесува на проблемот на поставување на контролер со цел минимизирање на вкупната латентност. Друга насока може да биде експанзија на анализата со користење на теоријата на мрежен калкулус или можеби вметнување на реални хардверски влијанија во аналитичкото моделирање. Конечно, за аналитичките модели изложени во главата 6, може да се инкорпорираат хардверски влијанија во моделите и да се споредат резултатите од предложените модели со реалните хардверски имплементации. Понатаму, последниот модел може да се прошири

преку воведување на дистрибуирана контролна рамнина за да се види ефектот на доцнењето при хендовер и ефектот врз минималната големина на баферот во комутаторите.

На крајот може да се заклучи дека примената на SDN во мобилните јадрени мрежи претставува широко поле за истражувања, кое и во иднина ќе биде предмет од интерес со цел добивање на уште подобри модели, алгоритми, оптимизации и поефикасни резултати. Резултатите од оваа дисертација претставуваат значаен придонес кон разгледуваната научно-истражувачка област и може ефикасно да се искористат како готов производ при дизајнирање на SDN мобилни мрежи, но и да се прошират и надградат во голем број на насоки, дел од кои веќе беа презентирани.

Работна библиографија

- [1] Open Networking F., “Openflow switch specification version 1.3.1,” Tech. Republic, 2012. Available: <https://www.opennetworking.org/>.
- [2] Y. Oktian, S. Lee, H. Lee, J. Lam, “Distributed SDN controller system: A survey on design choice”, *Computer Networks*, Vol. 121, pp. 100-111, 2017.
- [3] N. L. Van Adrichem, B. J. Van Asten, and F. A. Kuipers, “Fast Recovery in Software Defined Networks” (EWSDN), 2014 Third European Workshop on. IEEE, pp. 61–66, 2014.
- [4] G. Bianchi, M. Bonola, A. Capone, C. Cascone, "OpenState: Programming Platform-independent Stateful OpenFlow Applications Inside the Switch," *ACM SIGCOMM Computer Communication Review*, pp 44-51, Vol 44, Issue 2, 2014.
- [5] P. Ameigeiras, J. J. Ramos-Munoz, L. Schumacher, J. Prados-Garzon, J. Navarro-Ortiz, and J. M. Lopez-Soler, “Link-Level Access Cloud Architecture Design Based on SDN for 5G Networks,” *IEEE Network*, pp 24-31, vol.29, no.2, 2015.
- [6] C. Marquezan, X. An, Z. Despotovic, R. Khalili, and A. Hecker, “Identifying latency factors in sdn-based mobile core networks”, *Symposium on Computers and Communication*, pp. 484–491, 2016.
- [7] K. Tantayakul, R. Dhaou, and B. Paillassa, “Impact of SDN on mobility management”, 30th International Advanced Information Networking and Applications Conference, pp. 260–265, 2016.
- [8] M. F. Neuts, “Explicit Steady-State Solutions to Some Elementary Queueing Models”, *Operations Research* 30(3), pp 480-489, 1982.
- [9] H. Takagi, *Stochastic analysis of computer and communication systems*, Elsevier Science Inc., 1990.
- [10] M. F. Neuts, *Matrix-geometric solutions in stochastic models - an algorithmic approach.*, Dover Publications, 1994.
- [11] Open Networking Foundation. (2012, April) *Software Defined Networking: The new norm for networks*.
- [12] X. Jin, L. E. Li, L. Vanbever, J. Rexford, “SoftCell: scalable and flexible cellular core network architecture”, *CoNEXT '13, Proceedings of the ninth ACM conference on Emerging networking experiments and technologies*, pp. 163-174, 2013.

- [13] J. H. Cox, J. Chung, S. Donovan, J. Ivey, R. J. Clark, G. Riley, "Advancing Software-Defined Networks: A Survey," *IEEE Access*, vol 5, pp 25487 - 25526, 2017.
- [14] N. L. Van Adrichem, B. J. Van Asten, and F. A. Kuipers, "Fast Recovery in Software Defined Networks" (EWSDN), 2014 Third European Workshop on. IEEE, pp. 61–66, 2014.
- [15] M. N. Sadiku and S. M. Musa, *Performance analysis of computer networks*. Springer, 2013.
- [16] F. Benamrane, M. B. Mamoun, and R. Benaini, "Performances of OpenFlow-Based Software-Defined Networks: An overview.," *JNW*, vol. 10, no. 6, pp. 329–337, 2015.
- [17] T. Huang, F. R. Yu, C. Zhang, J. Liu, J. Zhang, and Y. Liu, "A survey on large-scale software defined networking (SDN) testbeds: Approaches and challenges," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 2, pp. 891–917, 2016.
- [18] F. Hu, Q. Hao, and K. Bao, "A survey on software-defined network and openflow: From concept to implementation," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 4, pp. 2181–2206, 2014.
- [19] H. Farhady, H. Lee, and A. Nakao, "Software-defined networking: A survey," *Computer Networks*, vol. 81, pp. 79–95, 2015.
- [20] A. Ost, *Performance of communication systems: A model-based approach with matrix-geometric methods*. Springer Science & Business Media, 2013.
- [21] D. Singh, "Analysis of OpenFlow Switches in Software-Defined Networking", Doctoral thesis, Victoria University of Wellington, 2019.
- [22] D. G. Kendall, "Stochastic processes occurring in the theory of queues and their analysis by the method of the imbedded Markov chain," *The Annals of Mathematical Statistics*, pp. 338–354, 1953.
- [23] T. Dayar, W. Sandmann, D. Spieler, and V. Wolf, "Infinite leveldependent QBD processes and matrix-analytic solutions for stochastic chemical kinetics," *Advances in Applied Probability*, vol. 43, no. 4, pp. 1005–1026, 2011.
- [24] G. Latouche and V. Ramaswami, *Introduction to matrix analytic methods in stochastic modeling*, vol. 5. Siam, 1999.
- [25] G. Latouche and V. Ramaswami, "A logarithmic reduction algorithm for quasi-birth-death processes," *Journal of Applied Probability*, pp. 650–674, 1993.
- [26] W. J. Stewart, *Probability, Markov chains, queues, and simulation: the mathematical basis of performance modeling*. Princeton University Press, 2009.

- [27] J. Garzon, “Architecture, Modeling, Planning, and Dynamic Provisioning of Softwarized 5G mobile Core Networks”, Doctoral thesis, University of Granada, 2018.
- [28] Ericsson Mobility Report: 5G uptake even faster than expected, Available: <https://www.ericsson.com/en/press-releases/2019/6/ericsson-mobility-report-5g-uptake-even-faster-than-expected>, 2019.
- [29] A. Colazzo, R. Ferrari, R. Lambiase, “Achieving low-latency communication in future wireless networks: the 5G NORMA approach”, EuCNC, 2016.
- [30] I. Parvez, A. Rahmati, I. Guvenc, A. Sarwat, H. Dai, “A Survey on Low Latency Towards 5G: RAN, Core Network and Caching Solutions”, IEEE Communications Surveys & Tutorials, Volume: 20 , Issue: 4, pp 3098 – 3130, 2018.
- [31] 3GPP TS 22.261 V16.5.0 (2018-09), “Technical Specification Group Services and System Aspects, Service requirements for the 5G system, Stage 1, Release 16”, 3GPP, 2018.
- [32] Ericsson, “Evolution from EPC to 5G Core made easy”, Available: <https://www.ericsson.com/en/blog/2018/5/core-network-evolution-from-epc-to-5g-core-made-easy>, 2018.
- [33] P. Ameigeiras, J. J. Ramos-Munoz, L. Schumacher, J. Prados-Garzon, J. Navarro-Ortiz, and J. M. Lopez-Soler, “Link-Level Access Cloud Architecture Design Based on SDN for 5G Networks,” IEEE Network, pp 24-31, vol.29, no.2, 2015.
- [34] B. Awoyemi, A. Alfa, B. Maharaj, “Network Restoration for Next-Generation Communication and Computing Networks”, Journal of Computer Networks and Communications, Hindawi, Volume 2018, 2018.
- [35] P. H. Franklin, I. Tavorovsky, and R. Ames, “A strategy for optimal management of spares,” in Proceedings of the 2016 Annual Reliability and Maintainability Symposium (RAMS), pp. 1–6, 2016.
- [36] C. Cascone, D. Sanvito, L. Pollini, A. Capone, B. Sansò, “Fast Failure Detection and Recovery in SDN with Stateful Data Plane,” International Journal of Network Management, vol. 27, no. 1, 2016.
- [37] P. Thorat, S. Jeon, S. M. Raza, R. Challa, H. Choo, "Scalable and Efficient Forwarding Table Design for Multi-Link Failover in OpenFlow-Enabled Networks," IETE Technical Review, pp 27-38, 34:sup1, 2017.
- [39] M. Alotaibi, A. Helmy, A. Nayak, "Modeling Handover Signaling Messages in OpenFlow-Based Mobile Software-Defined Network", Journal of Computer Networks and Communications, vol. 2018, 2018.

- [40] H. D. Vu, J. But, "How RTT Between the Control and Data Plane on a SDN Network Impacts on the Perceived Performance," International Telecommunication Networks and Application Conference (ITNAC)," pp 179-184, 2015.
- [41] V. G. Nguyen, A. Brunstrom, K. J. Grinnemo, and J. Taheri, "SDN/NFV-based Mobile Packet Core Network Architectures: A Survey," IEEE Commun. Surv. Tutor., vol. 19, issue. 33, pp. 1567–1602, 2017.
- [42] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, and D. Sabella, "On Multi-Access Edge Computing: A Survey of the Emerging 5G Network Edge Architecture Orchestration," IEEE Commun. Surv. Tutor., vol. 19, issue 3, pp. 1657–1681, 2017.
- [43] B. Briscoe, A. Brunstrom, A. Petlund, D. Hayes, D. Ros, I. J. Tsang, S. Gjessing, G. Fairhurst, C. Griwodz, and M. Welzl, "Reducing Internet Latency: A Survey of Techniques and Their Merits," IEEE Commun. Surv. Tutor., vol. 18, no. 3, pp. 2149–2196, third quarter 2016.
- [44] S. Srivastava and S. P. Singh, "A Survey on Latency Reduction Approaches for Performance Optimization in Cloud Computing," in Proc. Int. Conf. Comput. Intell. Commun. Technol. (CICT), pp. 111–115, 2016.
- [45] J. Page, J Dricot, "Software-Defined Networking for Low-Latency 5G Core Network", 2016 International Conference on Military Communications and Information Systems (ICMCIS), 2016.
- [46] N. L. Van Adrichem, B. J. Van Asten, and F. A. Kuipers, "Fast Recovery in Software Defined Networks" (EWSN), 2014 Third European Workshop on. IEEE, pp. 61–66, 2014.
- [47] S. Sharma, D. Staessens, D. Colle, M. Pickavet, and P. Demeester, "OpenFlow: Meeting carrier-grade recovery requirements," Computer Communications, vol. 36, no. 6, pp. 656 – 665, 2013.
- [48] S. Lee, K.-Y. Li, K.-Y. Chan, G.-H. Lai, and Y.-C. Chung, "Path layout planning and software based fast failure detection in survivable OpenFlow networks," in Design of Reliable Communication Networks (DRCN), 2014 10th International Conference on the, pp. 1–8, 2014.
- [49] M. Borokhovich, L. Schiff, and S. Schmid, "Provable data plane connectivity with local fast failover: Introducing Openflow graph algorithms," in Proceedings of the Third Workshop on Hot Topics in Software Defined Networking, ser. HotSDN '14. ACM, pp. 121–126, 2014.
- [50] M. S. M. Zahid, B. Isyaku, and F. A. Fadzil, "Recovery of software defined network from multiple failures: Openstate vs openflow," in Computer Systems and Applications (AICCSA), 2017 IEEE/ACS 14th International Conference on. IEEE, pp. 1178–1183, 2017.

- [51] R. Ahmed, E. Alfaki, and M. Nawari, "Fast failure detection and recovery mechanism for dynamic networks using software-defined networking," in Basic Sciences and Engineering Studies (SGCAC), 2016 Conference of. IEEE, pp. 167–170, 2016.
- [52] S. Zhang, Y. Wang, Q. He, J. Yu, and S. Guo, "Backupresource based failure recovery approach in sdn data plane," in Network Operations and Management Symposium (APNOMS), 2016 18th Asia-Pacific. IEEE, pp. 1–6, 2016.
- [53] A. Alshra'a, P. Sewalkar, J Seitz, "Enhanced Failure Recovery Mechanism Using OpenState Pipeline in SDN", 2018 Tenth International Conference on Ubiquitous and Future Networks (ICUFN), 2018.
- [54] D. Wu, Y. Xia, X. Sun, X. Hiang, S. Dzinamarira, T. Eugene, "Masking Failures from Application Performance in Data Center Networks with Shareable Backup", SIGCOMM '18, 2018.
- [55] C. Cascone, L. Pollini, D. Sanvito, A. Capone, "Traffic Management Applications for Stateful SDN Data Plane", Fourth European Workshop on Software Defined Networks, 2015.
- [56] "MiniNet." [Online]. Available: <http://www.mininet.org>.
- [57] "Ryu OpenFlow controller," Available: <http://osrg.github.io/ryu/>.
- [58] "SPIDER source code repository." Available: <http://github.com/OpenState-SDN/spider>.
- [59] "CPqD OpenFlow 1.3 Software Switch," Available: <http://cpqd.github.io/ofsoftswitch13/>.
- [60] M. Borokhovich, L. Schiff, and S. Schmid, "Provable data plane connectivity with local fast failover: Introducing Openflow graph algorithms," in Proceedings of the Third Workshop on Hot Topics in Software Defined Networking, ser. HotSDN '14. ACM, pp. 121–126, 2014
- [61] G. Wang ; Y. Zhao, "An Effective Approach to Controller Placement in Software Defined Wide Area Networks", IEEE Transactions on Network and Service Management (Volume:15, Issue:1), pp 344-355, 2018.
- [62] T. Issa, Z. Raoul, A. Konaté, J. C. Adepo, B. Cousin, A. Olivier, "Analytical Load Balancing Model in Distributed Open Flow Controller System", Scientific research, Engineering, Vol.10 No.12, 2018.
- [63] S. Zhihao, H. Wu, K. Wolter, "Performance Evaluation of the Control Plane in Software Defined Networks", the 12th EAI Int. Conf, 2019.
- [64] M. Jarschel et al., "Modeling and performance evaluation of an openflow architecture", Proceedings of the 23rd International Teletraffic Congress, pp. 1–7, 2011.

- [65] C. Sarkar et al, "Analytical Model for OpenFlow-Based Software-Defined Network", *Progress in Computer, Analytics and Networking*, pp 583-592, 2018.
- [66] S. Azodolmolky, R. Nejabati, M. Pazouki et al., "An analytical model for software defined networking: a network calculus-based approach", *Global Communications Conference*, pp. 1397–1402, 2013.
- [67] C. Thieme, "Challenges for modelling of software-based packet processing in commodity hardware using queueing theory", *Network Architecture and Services*, vol. 49, 2017.
- [68] K. Mahmood, A. Chilwan, O. Osterbo, M. Jarschel, "Modelling of OpenFlow-based software-defined networks: the multiple node case", *IET Networks*, Volume: 4, Issue: 5, pp 278-284, 2015.
- [69] K. Mahmood, A. Chilwan, O. Østerbø, and M. Jarschel, "On the modeling of openflow-based sdns: the single node case", *Computer Science and Information Technology (CS & IT)*, Vol.4, pp 207-214, 2014
- [70] Z. Shang and K. Wolter, "Delay evaluation of openflow network based on queueing model", *12th European Dependable Computing Conference (EDCC 2016)*, 2016.
- [71] O. Bliat, M. Mamoun, "An Overview on SDN Architectures with Multiple Controllers", *Journal of Computer Networks and Communications*, Volume 2016, 2016.
- [72] Y. Oktian, S. Lee, H. Lee, J. Lam, "Distributed SDN controller system: A survey on design choice", *Computer Networks*, Vol. 121, pp. 100-111, 2017.
- [73] K. Sood, S. Yu, "A General QoS Aware Flow-Balancing and Resource Management Scheme in Distributed Software-Defined Networks", *IEEE Access*, Vol. 4, pp. 7176-7185, 2016.
- [74] G. Wang , Y. Zhao, "An Effective Approach to Controller Placement in Software Defined Wide Area Networks", *IEEE Transactions on Network and Service Management (Volume:15, Issue:1)*, pp 344-355, 2018.
- [75] G. Wang, J. Li, X. Chang, "Modeling and performance analysis of the multiple controllers' approach in software defined networking", *2015 IEEE 23rd International Symposium on Quality of Service (IWQoS)*, 2015.
- [76] P. Fonseca, R. Bennesby, E. Mota, A. Passito, "A replication component for resilient OpenFlow-based net-working," in *Proc. of NOMS 2012*, pp. 933–939, 2012
- [77] X. Chen et al., "Leveraging Master-Slave OpenFlow Controller Arrangement to Improve Control Plane Resiliency in SD-EONs", *Optics Express*, vol. 23, no. 6, pp. 7550-7558, 2015.
- [78] B. Heller, R. Sherwood, N. McKeown, "The controller placement problem", *Proceedings of the first workshop on Hot topics in software defined networks. ACM*, pp 7–12, 2012.

- [79] Y. Hu, W. Wendong, X. Gong, X. Que, C. Shiduan, “Reliability-aware controller placement for software-defined networks”, *Integrated Network Management (IM 2013)*, 2013 IFIP/IEEE International Symposium on. IEEE, pp 672–675, 2013.
- [80] G. Yao, J. Bi, Y. Li, L. Guo, “On the capacitated controller placement problem in software defined networks”. *IEEE Communications Letters* 18, 8 (2014), pp 1339–1342, 2014.
- [81] L. Yao et. al., “Evaluating the Controller Capacity in SDN”, *ICCCN*, 23rd International Conference, 2014.
- [82] X. Duan, A. Akhtar, and X. Wang, “Software-defined networking-based resource management: data offloading with load balancing in 5g hetnet”, *EURASIP Journal on Wireless Communications and Networking*, vol. 2015, no. 1, p. 181, 2015.
- [83] F. Meneses, D. Corujo, C. Guimaraes, A. Aguiar, “Extending SDN to End Nodes Towards Heterogeneous Wireless Mobility”, *2015 IEEE Globecom Workshops*, 2015.
- [84] T. Nguyen, C. Bonnet, J. Harri, “SDN-based distributed mobility management for 5G networks”, *2016 IEEE Wireless Communications and Networking Conference*, 2016.
- [85] M. Sanchez, A. Oliva, V. Mancuso, “Experimental evaluation of an SDN-based distributed mobility management solution”, *Conference: the Workshop*, 2016.
- [86] M. Alotaibi, A. Helmy, A. Nayak, "Modeling Handover Signaling Messages in OpenFlow-Based Mobile Software-Defined Network", *Journal of Computer Networks and Communications*, vol. 2018, 2018.
- [87] S. Kukliński, Y. Li, and K. Dinh, “Handover management in sdn-based mobile networks”, *Global Communications Conference Workshops*, pp. 194–200, 2014.
- [88] A. Mondal, S. Misra, I. Maity, *Buffer Size Evaluation of Open-Flow Systems in Software-Defined Networks*, *IEEE Systems Journal* (2018) 1–8.
- [89] P. J. Burke, *The Output of a Queuing System*, *Operations Research* 4 (6), pp 699–704, 1956.
- [90] A. Busic, B. Gaujal, and F. Perronnin, “Perfect Sampling of Networks with Finite and Infinite Capacity Queues.” in *ASMTA*, ser. *Lecture Notes in Computer Science*, K. Al-Begain, D. Fiems, and J.-M. Vincent, Eds., vol. 7314. Springer, pp. 136–149, 2012.
- [91] T.-C. Yen and C.-S. Su, “An SDN-based cloud computing architecture and its mathematical model,” in *Information Science, Electronics and Electrical Engineering (ISEEE)*, 2014 International Conference on, vol. 3, pp. 1728–1731, IEEE, 2014.
- [92] U. Javed, A. Iqbal, S. Saleh, S. A. Haider, and M. U. Ilyas, “A stochastic model for transit latency in OpenFlow SDNs,” *Computer Networks*, vol. 113, pp. 218–229, 2017.

- [93] Y.-C. Lai, A. Ali, M. M. Hassan, M. S. Hossain, and Y.-D. Lin, "Performance Modeling and Analysis of TCP Connections over Software Defined Networks," in GLOBECOM 2017-2017 IEEE Global Communications Conference, pp. 1–6, IEEE, 2017.
- [94] K. Sood, S. Yu, and Y. Xiang, "Performance analysis of softwaredefined network switch using M=Geo=1 model," IEEE Communications Letters, vol. 20, no. 12, pp. 2522–2525, 2016.
- [95] W. Miao, G. Min, Y. Wu, and H. Wang, "Performance Modelling of Preemption-Based Packet Scheduling for Data Plane in Software Defined Networks", IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity), 2015.
- [96] W. Miao, G. Min, Y. Wu, H. Wang, and J. Hu, "Performance Modelling and Analysis of Software-Defined Networking under Bursty Multimedia Traffic", ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), vol. 12, no. 5s, p. 77, 2016.
- [97] Y. Goto, H. Masuyama, B. Ng, W. K. Seah, and Y. Takahashi, "Queueing Analysis of Software Defined Network with Realistic OpenFlow-based Switch Model," in Proceedings of the IEEE International Symposium on Modelling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS), London, UK, 19-21 September 2016.
- [98] S. Ogasawara and Y. Takahashi, "Performance analysis of traffic Classification in an openflow switch," in Cloudification of the Internet of Things (CIoT). IEEE, 2016.
- [99] R. Schassberger, "Review of Reversibility and stochastic networks," Perform. Eval., vol. 1, no. 1, p. 90, 1981.
- [100] G. Bolch, S. Greiner, H. de Meer, and K. S. Trivedi, Queueing Networks and Markov Chains, 2nd ed. Hoboken, New Jersey: John Wiley & Sons, Inc., 2006.
- [101] L. Kleinrock, "Queueing systems: theory", Wiley Interscience, vol 1, 1975.
- [102] F. Gillent, G. Latouche, "Semi-explicit solutions for M/PH/1-like queuing systems", EU Journal of Oper. Research, Vol. 13, Is. 2, pp 151-160, 1983.
- [103] W.E. Leland, M.S. Taqqu, W. Willinger, and D.V. Wilson, "On the self-similar nature of Ethernet traffic (extended version)", Networking, IEEE/ACM Transactions on, pp 1-15, 1994.
- [104] I. Tomic, N. Maletic, "Comparison of Models for Self-Similar Network Traffic Generation", X int. symposium on industrial electronics (INDEL), 2014.