

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE



# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA ELEKTROTECHNIKY**

**A KOMUNIKAČNÍCH TECHNOLOGIÍ**

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

**ÚSTAV TELEKOMUNIKACÍ**

DEPARTMENT OF TELECOMMUNICATIONS

## SOFTWAREVĚ DEFINOVANÉ OPTICKÉ SÍŤ

SOFTWARE DEFINED OPTICAL NETWORK

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

Tomáš Kloda

**VEDOUCÍ PRÁCE**

SUPERVISOR

Ing. Lukáš Kočí

BRNO 2016

# Bakalářská práce

bakalářský studijní obor **Teleinformatika**  
Ústav telekomunikací

**Student:** Tomáš Kloda

**ID:** 154767

**Ročník:** 3

**Akademický rok:** 2015/16

**NÁZEV TÉMATU:**

## Softwarově definované optické sítě

### POKYNY PRO VYPRACOVÁNÍ:

Cílem bakalářské práce je detailně prozkoumat princip DWDM softwarově definovaných optických sítí. Práce bude obsahovat navázání komunikace mezi OpenFlow agentem a kontrolerem, implementaci rozšířeného záhlaví pro přenos v optických sítích a návrh řízení pomocí centrálního kontroleru. Simulace budou prováděny v prostředí OMNeT++. Výsledkem práce bude návrh na řízení a zlepšení využívání šířky pásma v OXC pomocí softwarově definovaných optických sítí.

### DOPORUČENÁ LITERATURA:

- [1] CHANNEGOWDA, Mayur, Reza NEJABATI a Dimitra SIMEONIDOU. 2013. Software-Defined Optical Networks Technology and Infrastructure: Enabling Software-Defined Optical Network Operations [Invited]. Journal of Optical Communications and Networking. 5(10): A274-. DOI: 10.1364/JOCN.5.00A274. ISSN 1943-0620.
- [2] BAHNASY, M., K. IDOUDI a H. ELBIAZE. 2014. Software-defined DWDM optical networks: OpenFlow and GMPLS experimental study. 2014 IEEE Global Communications Conference. IEEE, : 2173-2179. DOI: 10.1109/GLOCOM.2014.7037130. ISBN 978-1-4799-3512-3.

**Termín zadání:** 1.2.2016

**Termín odevzdání:** 1.6.2016

**Vedoucí práce:** Ing. Lukáš Kočí

**Konzultant bakalářské práce:**

**doc. Ing. Jiří Mišurec, CSc., předseda oborové rady**

### UPOZORNĚNÍ:

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

## **ABSTRAKT**

Tato práce je zaměřena na vysvětlení principu softwarově definovaných sítí a komunikace přes protokol OpenFlow s popisem přenášených zpráv. Dále se práce zabývá softwarově definovanými optickými sítěmi, kde jsou popsány kombinace OpenFlow s dalšími technologiemi a způsoby řízení optické cesty. V další části se práce soustředí na popis fyzické vrstvy páteřních optických sítí, kde se nachází popis multiplexů, DWDM technologie a optických prvků. Tato práce dále zahrnuje návrhy softwarově definované optické sítě s implementací rozšířeného záhlaví pro přenos v optické doméně. Výstupem práce jsou simulace obsahující synchronizaci mezi jednotlivými prvky během inicializace softwarově definované optické sítě s výskytem zpoždění. V závěru je práce soustředěna na simulace samotného datového přenosu uvnitř navržené softwarově definované optické sítě.

## **KLÍČOVÁ SLOVA**

Multiplex, OpenFlow agent, OpenFlow protokol, OpenFlow směrovač, SDN kontrolér, softwarově definované síť

## **ABSTRACT**

This thesis is focused on explanation of software defined networking and communication via OpenFlow protocol with description of transmitted messages. This thesis deals with software defined optical networking, which describes combinations of OpenFlow with other technologies and methods to control the optical path. In another part, the thesis is focused on the description of the physical layer of optical backbone networks, where there is a description of multiplexers, DWDM technology and optical elements. This thesis also includes designs of software defined optical network with implementation of extended header for transmission in the optical domain. Outcome of this thesis are simulations, which include synchronization between the elements during initialization of software defined optical network with occurrence of delay. At the end, the thesis is focused on simulations of the actual data transfer within the proposed software defined optical network.

## **KEYWORDS**

Multiplex, OpenFlow agent, OpenFlow protocol, OpenFlow router, SDN controller, software defined networking

KLODA, Tomáš *Softwarově definované optické sítě*: bakalářská práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2016. 70 s. Vedoucí práce byl Ing. Lukáš Kočí

## PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Softwarově definované optické sítě“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

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

Brno .....

.....

(podpis autora)

## PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu bakalářské práce panu Ing. Lukášovi Kočímu za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Brno .....

.....

(podpis autora)



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

## PODĚKOVÁNÍ

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

Brno .....

.....

(podpis autora)



EVROPSKÁ UNIE  
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ  
INVESTICE DO VAŠÍ BUDOUCNOSTI



# OBSAH

Úvod	12
<b>1 Softwarově definované sítě</b>	<b>13</b>
1.1 Model softwarově definované sítě	14
1.2 OpenFlow	15
1.2.1 SDN (OpenFlow) směrovače	15
1.2.2 Druhy zpráv přenášené pomocí OpenFlow protokolu	16
1.2.3 Typy OpenFlow zpráv	17
1.2.4 Vývoj OpenFlow protokolu	21
<b>2 Softwarově definované optické sítě</b>	<b>23</b>
2.1 Rozšíření OpenFlow protokolu	24
2.2 Řízení optické cesty pomocí OpenFlow	24
2.3 Kombinace OpenFlow s výpočetní jednotkou PCE	26
2.4 Kombinace OpenFlow s GMPLS	26
<b>3 Sdílení kanálů v páteřních optických sítích</b>	<b>28</b>
3.0.1 Optický časový multiplex	29
3.0.2 Vlnový multiplex	29
3.1 DWDM technologie	30
3.1.1 DWDM dle ITU-T G.694.1	31
<b>4 Optické prvky v páteřních DWDM sítích</b>	<b>33</b>
4.1 Optický křížový spoj	33
4.1.1 Architektura OXC	33
4.2 Rekonfigurovatelný optický add/drop multiplexer	35
4.2.1 Architektura ROADM	36
4.3 Optické linkové zakončení	37
<b>5 Navázání spojení SDN kontroléru a OpenFlow agenta v OMNeT++</b>	<b>38</b>
5.1 Průběh výměny zpráv	38
5.2 Obsah a nastavení zpráv	40
5.2.1 OFPT_HELLO	40
5.2.2 OFPT_FEATURES_REQUEST/REPLY	41
5.2.3 OFPT_ECHO_REQUEST/REPLY	42
5.3 Rozšířené navázání spojení OpenFlow agentů s SDN kontrolérem	42



<b>6</b>	<b>Návrh DWDM optické sítě na bázi OXC</b>	<b>45</b>
6.1	Princip směrování optického signálu . . . . .	46
6.2	Zapouzdření IP paketu . . . . .	46
<b>7</b>	<b>Simulace datového přenosu DWDM na bázi OXC</b>	<b>48</b>
7.1	Průběh simulace datového přenosu . . . . .	49
7.2	Řízení šířky pásma OXC . . . . .	51
<b>8</b>	<b>Návrh DWDM optické sítě na bázi ROADM</b>	<b>54</b>
8.1	Princip směrování optického signálu . . . . .	54
<b>9</b>	<b>Simulace datového přenosu DWDM na bázi ROADM</b>	<b>56</b>
9.1	Průběh simulace datového přenosu . . . . .	57
<b>10</b>	<b>Závěr</b>	<b>59</b>
	<b>Literatura</b>	<b>61</b>
	<b>Seznam symbolů, veličin a zkratk</b>	<b>64</b>
	<b>Seznam příloh</b>	<b>66</b>
<b>A</b>	<b>Příloha A</b>	<b>67</b>
A.1	Ukázky zdrojových kódů . . . . .	67
A.1.1	Porovnávání přijatých záznamů v tabulce toků . . . . .	67
A.1.2	Navázání spojení SDN kontroléru . . . . .	68
A.1.3	Navázání spojení OpenFlow agenta . . . . .	69
<b>B</b>	<b>Příloha B</b>	<b>70</b>
B.1	Obsah přiloženého CD . . . . .	70

# SEZNAM OBRÁZKŮ

1.1	Model softwarově definované sítě [20]. . . . .	14
1.2	Struktura OpenFlow směrovače [21]. . . . .	16
1.3	OpenFlow záhlaví [21]. . . . .	17
1.4	Zpráva <i>OFPT_HELLO</i> [21]. . . . .	18
1.5	Zpráva <i>OFPT_ERROR</i> [21]. . . . .	18
1.6	Zpráva <i>OFPT_FEATURES_REQUEST/REPLY</i> [21]. . . . .	19
1.7	Zpráva <i>OFPT_SET_CONFIG</i> [21]. . . . .	19
1.8	Zpráva <i>OFPT_PACKET_IN</i> [21]. . . . .	20
1.9	Zpráva <i>OFPT_PACKET_OUT</i> [21]. . . . .	21
3.1	Princip multiplexování [3]. . . . .	28
3.2	Optický časový multiplex [26]. . . . .	29
3.3	Vlnový multiplex [3]. . . . .	29
3.4	Fixní rozestup kanálů pro 50 GHz [19]. . . . .	31
3.5	Příklad přenášených šířek pásma u flexibilního rozdělení DWDM kanálu [19]. . . . .	32
4.1	Základní architektura OXC [24]. . . . .	33
4.2	Vnitřní struktura přepínacího jádra OXC [7]. . . . .	34
4.3	Schéma ROADM [11]. . . . .	35
4.4	Architektura colored ROADM [11]. . . . .	36
5.1	Propojení modulů. . . . .	38
5.2	Navázání spojení. . . . .	39
5.3	Výsledek simulace navázání spojení. . . . .	40
5.4	Obsah zprávy <i>OFPT_HELLO</i> implementované v OMNeT++. . . . .	40
5.5	a) <i>OFPT_FEATURES_REQUEST</i> , b) <i>OFPT_FEATURES_REPLY</i> implementované v OMNeT++. . . . .	41
5.6	Obsah zprávy <i>OFPT_ECHO_REQUEST</i> implementované v OMNeT++. . . . .	42
5.7	Propojení OpenFlow agentů s SDN kontrolérem v OMNeT++ [18]. . . . .	42
5.8	Průběh navázání spojení OpenFlow agentů s SDN kontrolérem [18]. . . . .	43
6.1	Návrh DWDM optické sítě v rámci OXC [18]. . . . .	45
6.2	Princip přepínání signálu. . . . .	46
6.3	Zapouzdření paketu. . . . .	46
7.1	Topologie datového přenosu v rámci OXC v OMNeT++. . . . .	48
7.2	Vnitřní struktura OpenFlow agenta v rámci abstrakce OXC v OMNeT++. . . . .	48
7.3	Vnitřní struktura SDN kontroléru v OMNeT++. . . . .	49
7.4	Obsah zapouzdřeného paketu v OMNeT++. . . . .	50
7.5	Obsah zprávy <i>OFPT_FLOW_MOD</i> implementované v OMNeT++. . . . .	50
7.6	Kompletní výstup simulace datového přenosu v rámci OXC. . . . .	51

7.7	Obsazenost prvního portu. . . . .	52
7.8	Obsazenost druhého portu. . . . .	52
7.9	Obsazenost třetího portu. . . . .	53
7.10	Obsazenost čtvrtého portu. . . . .	53
8.1	Návrh DWDM optické sítě v rámci ROADM. . . . .	54
9.1	Topologie datového přenosu v rámci ROADM. . . . .	56
9.2	Vnitřní struktura OpenFlow agenta v rámci abstrakce ROADM. . . . .	56
9.3	Vnitřní struktura add/drop modulu. . . . .	57
9.4	Kompletní výstup simulace datového přenosu v rámci ROADM. . . . .	58

# ÚVOD

V současné době je rozvoj síťové architektury jeden z nejdůležitějších trendů počítačových sítí. V důsledku neustálého vývoje síťových technologií dochází ke zvyšování nároků na stávající síťovou architekturu jak v tradičních IP sítích, tak i v sítích optických. Současná síťová architektura již začíná projevovat své nedostatky, které je nutné řešit v rámci budoucnosti síťových služeb a internetu jako takovém. Softwarově definované sítě s sebou přinášejí možnosti centralizovaného řízení, lepší automatizace, rychlejší inovaci sítě či přímou programovatelnost síťových prvků. Jedná se tedy o soubor řešení, jakým směrem se má vývoj síťové architektury ubírat.

Tato práce se zabývá problematikou softwarově definovaných optických sítí a popisuje jednotlivá řešení pro její realizaci. V úvodu teoretické části je práce zaměřena na popis softwarově definovaných sítí obecně. Je zde vysvětlen důvod jejího zavedení a seznámení s třívrstevným modelem softwarově definované sítě s popisem funkce jednotlivých vrstev. V dalším bodě teoretické části se nachází seznámení s protokolem OpenFlow. Je zde vysvětlena struktura OpenFlow směrovače s popisem vnitřních komponentů. Dále je probrána funkce a obsah jednotlivých typů OpenFlow zpráv, které jsou přenášeny během komunikace přes protokol OpenFlow.

Dále se již práce zaměřuje na konkrétní požadavky pro zavedení principu softwarově definovaných sítí v rámci páteřních optických sítí. Jsou zde popsány jednotlivé podvrstvy pro hardwarovou abstrakci optických sítí a kombinace rozšířeného OpenFlow protokolu s dalšími technologiemi pro zaručení kompatibility v rámci optické domény. V závěru teoretické části je práce soustředěna na fyzickou vrstvu optických sítí, kde jsou uvedeny způsoby sdílení kanálů s popisem jednotlivých multiplexů. Součástí popisu fyzické vrstvy je i vysvětlení architektury a funkce jednotlivých optických prvků využívaných v páteřních optických sítích.

Praktická část práce obsahuje návrhy softwarově definované optické sítě s abstraktním pohledem na její fyzickou vrstvu. V návrzích jsou implementovány komponenty softwarově definovaných sítí v kombinaci s optickými prvky. V rámci kompatibility softwarově definované sítě s optickými prvky je implementován princip rozšířeného záhlaví zapouzdřeného IP paketu pro přenos v optické doméně. Výstupem práce jsou vytvořené simulace průběhu navázání spojení mezi hlavními prvky softwarově definované optické sítě včetně nastavení konfiguračních zpráv. Během navázání spojení je zahrnut a odsimulován vznik zpoždění mezi prvky na základě reálných přenosových parametrů. V závěru praktické části je odsimulován samotný datový přenos a řízení šířky pásma na základě pracovní logiky použitých optických prvků. Veškeré simulace jsou realizovány v prostředí simulačního softwaru OMNeT++.

# 1 SOFTWAREOVĚ DEFINOVANÉ SÍTĚ

Požadavky na počítačové sítě se neustále vyvíjejí. Podniky a koncoví uživatelé kladou čím dál více nároků, na které již současné tradiční síťové architektury a technologie přestávají být vhodné. Tradiční počítačové sítě jsou hierarchické, složené z řady síťových prvků, uspořádaných nejčastěji do stromové struktury. Takovéto sítě dříve dávaly smysl, jelikož architektura klient-server byla nejvíce rozšířená a používaná. Avšak současné nároky, jako jsou zejména virtualizace serverů, rozšíření cloudových služeb či mobilních zařízení, vedou k tomu, aby tyto sítě byly přehodnoceny a optimalizovány [22].

Stávající počítačové sítě jsou pro současné nároky poněkud omezené. Mezi tato omezení patří především statická povaha sítí, která je v dnešní době v ostrém kontrastu s dynamickým vývojem serverového prostředí. Navíc je správa síťových zařízení od různých dodavatelů vysoce manuální a je potřeba tato zařízení konfigurovat zvlášť. Z tohoto důvodu vyplývá, že stávající sítě nejsou schopny se dynamicky přizpůsobovat měnícím se požadavkům [22].

Myšlenka softwarově definovaných sítí SDN (Software Defined Networking) vychází z výše zmíněných omezení. Jedná se o novodobou síťovou architekturu, jejíž hlavním principem je úplné oddělení řídicí roviny od datové roviny. Oddělená řídicí rovina, která byla dříve pevně vázána v jednotlivých síťových zařízeních, je poté přímo programovatelná a umožňuje vytvořit abstrakci pro aplikace a síťové služby, které mohou spravovat síť jako logický či virtuální subjekt [22].

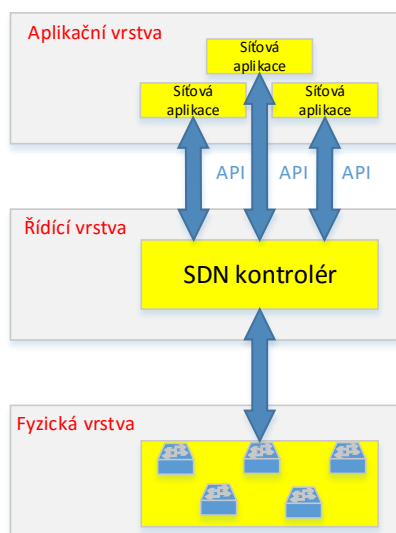
SDN přinášejí, oproti stávajícím sítím, mnoho výhod. Mezi primární výhody patří především [22]:

- centralizovaná správa síťových zařízení od různých dodavatelů,
- lepší automatizace a řízení pomocí běžných rozhraní API (Application Programming Interface),
- rychlá inovace sítě prostřednictvím schopnosti realizovat nové funkce a služby bez nutnosti individuální konfigurace jednotlivých síťových zařízení, což vede k nižším nákladům,
- programová konfigurace pomocí SDN aplikací, nikoliv přes příkazový řádek,
- přímá programovatelnost pomocí otevřených programů umožňuje správcům, podnikům, nezávislým výrobcům a koncovým uživatelům nové příležitosti k rozvoji sítě,
- centralizace a automatizované řízení vede ke zlepšení spolehlivosti a bezpečnosti sítě.

## 1.1 Model softwarově definované sítě

Architektura SDN vychází z již dobře známého modelu běžného operačního systému. Model SDN architektury, který se nachází na obr. 1.1, lze rozdělit do tří vrstev [20]:

- **Aplikační vrstva (Application Layer)** – skládá se z koncových SDN aplikací. Na hranici mezi aplikační vrstvou a řídicí vrstvou se nachází rozhraní API.
- **Kontrolní vrstva (Control Layer)** – nachází se zde síťový operační systém pracující na zařízení označovaném jako SDN kontrolér, který je logicky centralizován a obsahuje služby, které umožňují propojení s jednotlivými prvky v síti a poskytují otevřená programovatelná rozhraní pro SDN aplikace. Dále můžeme pomocí této vrstvy vytvořit abstrakci fyzické vrstvy pro snadnější programovatelnost zařízení.
- **Fyzická vrstva (Infrastructure Layer)** – skládá se ze síťových prvků, které umožňují přepínání a přeposílání paketů. Označují se jako SDN směrovače (směrovače, podporující SDN funkce) a ve spolupráci se dvěma nadřazenými vrstvami (kontrolní a aplikační) řeší kompletní správu fyzické vrstvy. Úkolem SDN směrovačů je pouze přijímat pakety, upravovat záhlaví paketů a přeposílat je na porty ostatních síťových zařízení. Veškeré řídicí instrukce, které určují, jak s pakety zacházet, se nacházejí v SDN kontroléru.



Obr. 1.1: Model softwarově definované sítě [20].

Po obdržení paketu SDN směrovačem je provedena analýza jeho záhlaví. Pokud SDN směrovač nenalezne ve svých směrovacích tabulkách potřebné informace o tom, co se má s daným paketem stát, či kam má být směrován, vyšle dotazovací zprávu na SDN kontrolér. Na základě programovatelné vnitřní logiky je rozhodnuto, jak má

být s paketem zacházeno. Tato informace je poté vyslána zpět na SDN směrovač přes SDN kontrolér, který má funkci kompilátoru. SDN směrovač poté, na základě obdržené informace, kterou zavede do své směrovací tabulky, provede požadovanou akci (zahození paketu, modifikaci záhlaví atd.). Tento proces se opakuje u každého dalšího SDN směrovače v síti až do doby, kdy paket SDN opustí. Avšak prioritně SDN směrovač nahlíží do své vnitřní směrovací tabulky a provádí směrování bez pomoci vyšších vrstev [12].

## 1.2 OpenFlow

Jednou z nejvíce rozšířených implementací pro SDN je OpenFlow. Jedná se o standard, který umožňuje komunikaci mezi datovou rovinou a oddělenou řídicí rovinou. OpenFlow specifikuje základní primitiva, která mohou být použita externí SDN aplikací k programování síťových zařízení, nacházejících se v datové rovině. Tento způsob řízení se dá přirovnat k instrukční sadě procesoru, který by takto programoval počítačový systém [22].

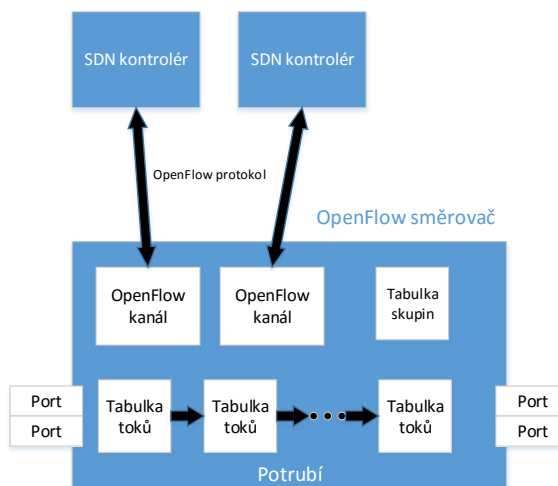
OpenFlow je implementován mezi řídicí a fyzickou vrstvou a využívá myšlenku datového toku pro identifikaci síťového provozu na základě předem definovaných pravidel. Tato pravidla mohou být staticky či dynamicky programovatelná pomocí SDN aplikací. Na základě požadavků lze poté definovat, jak bude síťový provoz protékat přes síťová zařízení. Aktuálně se OpenFlow nachází ve verzi 1.5.1, ze které je v práci vycházeno [22].

### 1.2.1 SDN (OpenFlow) směrovače

Směrovače, podporující SDN, se označují jako SDN směrovače. V některých publikacích lze najít označení OpenFlow směrovač. Struktura OpenFlow směrovače obsahuje tabulku skupin Group Table a jednu nebo více tabulek toků Flow Table, které jsou součástí tzv. potrubí. Každá tabulka toků obsahuje sadu záznamů, které poskytují informace, kam mají být pakety směrovány. Každý záznam obsahuje tyto části [21]:

- match header – slouží k porovnávání toku se záznamem,
- priority – odpovídající priorita toku,
- counter – pokud paket odpovídá záznamu, dojde k jeho navýšení,
- instruction – akce, které mají být provedeny během zpracování paketů,
- timeout – maximální doba toku v OpenFlow směrovači,
- cookie – data, která mohou být použita SDN kontrolérem pro filtrování záznamu,
- flags – určují, jakým způsobem jsou záznamy řízeny.

Využitím OpenFlow protokolu je poté SDN kontrolér schopen přidávat, aktualizovat či mazat záznamy v tabulce toků. Dále se zde nacházejí vstupní a výstupní porty a zabezpečené OpenFlow kanály, které slouží jako výstup OpenFlow směrovače pro komunikaci s SDN kontrolérem přes OpenFlow protokol [21]. Vnitřní struktura OpenFlow směrovače je zobrazena na obr. 1.2.



Obr. 1.2: Struktura OpenFlow směrovače [21].

OpenFlow porty jsou síťová rozhraní, která logicky propojují jednotlivé OpenFlow směrovače v SDN a zároveň umožňují přeposílat pakety mezi nimi. Pakety jsou přijaty na vstupním portu OpenFlow směrovače a dále zpracovávány přes tabulky toků, nacházejících se v potrubí. Zároveň je paket potrubím směrován na výstupní port OpenFlow směrovače, který umožňuje paketu přístup dál do sítě. OpenFlow směrovač podporuje fyzické, logické a rezervované typy portů [21].

Po obdržení paketu OpenFlow směrovačem dojde k porovnání paketu se záznamy v první, případně i v následujících tabulkách toků a to na základě předem určené priority. Pokud je záznam o paketu nalezen, je poté vykonána instrukce, která určuje, co se s daným paketem stane. V případě nenalezení záznamu může dojít k zahození paketu či k přeposlání paketu na SDN kontrolér. Po obdržení záznamu mohou být pakety zařazeny do skupiny, která poté specifikuje jejich další zpracování pomocí sady instrukcí [21].

### 1.2.2 Druhy zpráv přenášené pomocí OpenFlow protokolu

Komunikace přes OpenFlow protokol probíhá na základě třech druhů zpráv [21]:

- **Controller-to-Switch zprávy** – jsou posílány SDN kontrolérem na OpenFlow směrovač. Pomocí těchto zpráv je SDN kontrolér schopen získat veškeré informace o OpenFlow směrovači, provádět jeho konfiguraci a přidávat, mazat



či modifikovat záznamy v tabulkách toků. Dále může SDN kontrolér vkládat či mazat sady instrukcí v tabulce skupin nebo nastavovat vlastnosti portů OpenFlow směrovače. Pomocí packet-out zprávy jsou pakety zasílány na specifické porty OpenFlow směrovače.

- **Asynchronní zprávy** – jsou posílány OpenFlow směrovačem na SDN kontrolér v případě změny stavu OpenFlow směrovače. Pokud OpenFlow směrovač nenalezne pro příchozí paket žádný záznam v tabulkách toků, je pomocí packet-in zprávy veškeré řízení paketu přenecháno SDN kontroléru. Dále může OpenFlow směrovač informovat SDN kontrolér o vymazání záznamu z tabulky toků, zaslat informace o konfiguračních změnách portů či oznámit změny stavu OpenFlow kanálu.
- **Synchronní zprávy** – jsou zasílány jak OpenFlow směrovačem tak i SDN kontrolérem. Nacházejí se zde zprávy, které slouží zejména k sestavení a průběžné kontrole spojení či k zasílání informací o chybách.

### 1.2.3 Typy OpenFlow zpráv

Každá OpenFlow zpráva obsahuje povinné pole záhlaví. Struktura záhlaví je zobrazena na obr. 1.3. Pole version o velikosti 8 bit specifikuje nejvyšší verzi OpenFlow protokolu, na které jsou obě zařízení schopna pracovat. Length (16 bit) indikuje celkovou délku zprávy, type (8 bit) obsahuje typ přenášené zprávy. Identifikátor xid (32 bit) je jedinečná hodnota, která se používá pro usnadnění párování OpenFlow směrovače a SDN kontroléru. Každá žádost má své jedinečné xid, na kterou OpenFlow směrovač odpovídá právě tímto xid [21].

version	type	length
xid		

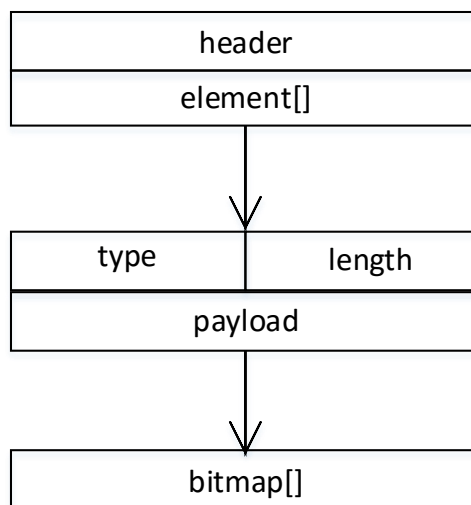
Obr. 1.3: OpenFlow záhlaví [21].

V následujícím výčtu se nacházejí základní zprávy pro sestavení spojení mezi OpenFlow směrovačem a SDN kontrolérem a konfiguraci OpenFlow směrovače.

#### OFPT\_HELLO

Během sestavení spojení dojde k výměně synchronní zprávy *OFPT\_HELLO*. Skládá se ze záhlaví a od verze 1.3.1 i z nepovinné sady *HELLO* elementů. *HELLO* element je struktura, která obsahuje délku a typ *HELLO* elementu a také datovou zátěž, označovanou jako bitmapové pole. Bitmapové pole obsahuje sadu podporovaných verzí OpenFlow protokolu a může být použito během vyjednávání verze OpenFlow

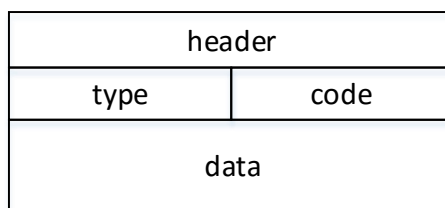
na počátku komunikace mezi OpenFlow směrovačem a SDN kontrolérem [21]. Zpráva *OFPT\_HELLO* je zobrazena na obr. 1.4.



Obr. 1.4: Zpráva *OFPT\_HELLO* [21].

### **OFPT\_ERROR**

Zprávy *OFPT\_ERROR* jsou během spojení OpenFlow směrovače a SDN kontroléru zasílány v případě výskytu chyby či selhání operace. Mezi tato chybová hlášení patří především poškození zpráv, neúspěšné jednání o verzi OpenFlow protokolu či selhání OpenFlow směrovače. Zpráva *OFPT\_ERROR* je znázorněna na obr. 1.5. Skládá se ze záhlaví, z polí *type* a *code* o velikosti 16 bit a z datového pole. *Type* udává úroveň chybové zprávy, zároveň *code* udává její přesný údaj. Datové pole obsahuje alespoň 512 bit a přenáší celkovou chybovou zprávu [21].

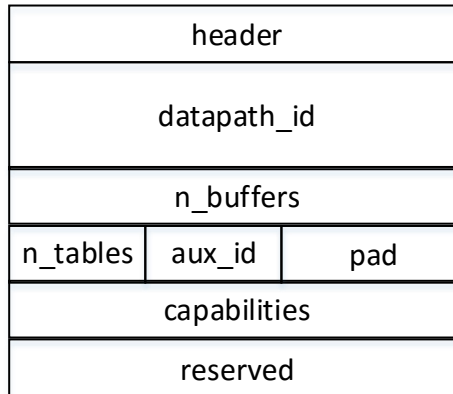


Obr. 1.5: Zpráva *OFPT\_ERROR* [21].

### **OFPT\_FEATURES\_REQUEST/REPLY**

Zpráva *OFPT\_FEATURES\_REQUEST* je používána SDN kontrolérem pro identifikaci a zjištění základních funkcí a schopností OpenFlow směrovače, na kterou

OpenFlow směrovač reaguje zprávou *OFPT\_FEATURES\_REPLY*. Na obr. 1.6 je zobrazena struktura zprávy *OFPT\_FEATURES\_REQUEST*, která obsahuje záhlaví a pole *datapath\_id* (64 bit), *n\_buffers* (32 bit), *n\_tables* (8 bit), *auxiliary\_id* (8 bit), *pad* (16 bit), *capabilities* (32 bit) a *reserved* (32 bit).

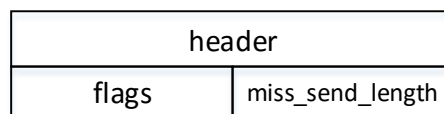


Obr. 1.6: Zpráva *OFPT\_FEATURES\_REQUEST/REPLY* [21].

*Datapath\_id* identifikuje IP adresu OpenFlow směrovače, kdy jeden fyzický OpenFlow směrovač může mít více identifikátorů (např. při virtualizaci). *N\_buffers* identifikuje, kolik paketů je OpenFlow směrovač schopen zařadit do fronty v rámci odeslání paketů na SDN kontrolér. *N\_tables* udává, kolik tabulek OpenFlow směrovač podporuje. *Auxiliary\_id* vyjadřuje přídavná spojení mezi OpenFlow směrovačem a SDN kontrolérem. Pole *pad* je v případě potřeby využito pro zarovnání bitů. *Capabilities* slouží pro uložení schopností a statistik OpenFlow směrovače. Pole *reserved* slouží pro rezervaci bitů [21].

### **OFPT\_GET\_CONFIG\_REQUEST/OFPT\_SET\_CONFIG**

Zpráva *OFPT\_GET\_CONFIG\_REQUEST* je zasílána SDN kontrolérem na OpenFlow směrovač a slouží ke zjištění konfiguračních parametrů pro manipulaci IP fragmentů v paketovém potrubí. Dále také určuje, kolik paketů bude sdíleno s SDN kontrolérem. Na tento konfigurační požadavek poté OpenFlow směrovač reaguje zprávou *OFPT\_GET\_CONFIG\_REPLY*. Pro nastavení konfiguračních parametrů slouží zpráva *OFPT\_SET\_CONFIG*, kterou zasílá SDN kontrolér. Struktura zprávy *OFPT\_SET\_CONFIG* je znázorněna na obr. 1.7.



Obr. 1.7: Zpráva *OFPT\_SET\_CONFIG* [21].

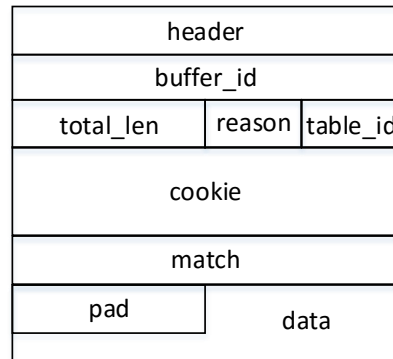
Pole flags (16 bit) určuje, jakým způsobem bude OpenFlow směrovač manipulovat s IP fragmenty. Může jít o standardní řízení (porovnávání se záznamy), zahození či znovuospořádání IP fragmentů. Pole miss\_send\_length (16 bit) definuje počet bitů každého paketu, který je poslán na SDN controller [21].

### OFPT\_ECHO\_REQUEST/REPLY

Zprávu *OFPT\_ECHO\_REQUEST* zasílá SDN kontrolér ke zjištění informací o velikosti zpoždění během přenosu či k měření šířky pásma. Reakcí na obdrženou zprávu *OFPT\_ECHO\_REQUEST* je zpráva *OFPT\_ECHO\_REPLY*, která slouží jako odpověď na požadované informace. Obě zprávy se skládají ze záhlaví a z datového pole [21].

### OFPT\_PACKET\_IN

Zpráva *OFPT\_PACKET\_IN* představuje způsob odeslání zachyceného paketu OpenFlow směrovačem na SDN kontrolér. K odeslání packet-in zprávy dochází tehdy, kdy tabulka toků OpenFlow směrovače neobsahuje žádný záznam o příchozím paketu, a tudíž nedojde k porovnání příchozího paketu se záznamem. Struktura zprávy *OFPT\_PACKET\_IN* je zobrazena na obr. 1.8.

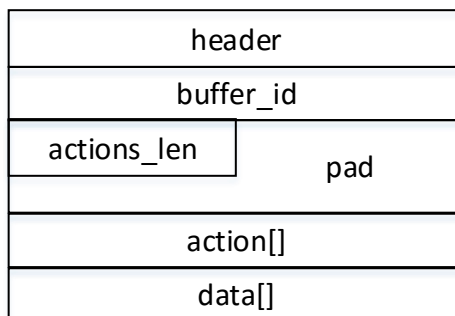


Obr. 1.8: Zpráva *OFPT\_PACKET\_IN* [21].

Pole buffer\_id (32 bit) obsahuje unikátní hodnotu využívanou pro identifikaci paketu, nacházejícího se ve vyrovnávací paměti OpenFlow směrovače. Total\_len (16 bit) vyjadřuje úplnou délku zachyceného paketu. Reason (8 bit) obsahuje informaci, z jakého důvodu byl paket vyslán na SDN kontrolér. Table\_id (8 bit) obsahuje jedinečný identifikátor tabulky toků, ve které došlo k hledání záznamu zachyceného paketu. Cookie (64 bit) slouží k identifikaci toku, kterému odpovídá zachycený paket. Match (32 bit) je struktura, která obsahuje metadata paketu. Pad (8 bit) slouží k zarovnání bitů. Pole data (8 bit) již obsahuje část či celý zachycený paket [21].

## OFPT\_PACKET\_OUT

Zpráva *OFPT\_PACKET\_OUT* je zasílána SDN kontrolérem na OpenFlow směrovač jako reakce na asynchronní zprávu *OFPT\_PACKET\_IN*. Struktura zprávy *OFPT\_PACKET\_OUT* je zobrazena na obr. 1.9.



Obr. 1.9: Zpráva *OFPT\_PACKET\_OUT* [21].

Pole `buffer_id` (32 bit) obsahuje stejnou hodnotu, jako u *OFPT\_PACKET\_IN*. `Actions_len` (16 bit) obsahuje délku pole akcí. `Pad` (8 bit) opět slouží k zarovnání bitů. Pole `action[]` obsahuje soubor akcí definujících, jak má být paket zpracován OpenFlow směrovačem. Mezi tyto akce může být zahrnuta modifikace paketu, skupinové zpracování paketů nebo výstupní port, na který bude paket poslán. Pole `data[]` může být buď prázdné nebo, v případě vyrovnávací paměti, může obsahovat zpracovávaný paket [21].

### 1.2.4 Vývoj OpenFlow protokolu

Základní OpenFlow protokol verze 0.2.0 byl vydán v roce 2008 a neobsahoval výraznější specifikace. V tomtéž roce byla vydána verze 0.8.0 a přinesla řadu vylepšení. Byla provedena reorganizace OpenFlow zpráv (byly přidány *OFPT\_ERROR* zprávy a zprávy popisující stav tabulek a stav portů OpenFlow směrovače). Dále byl přidán virtuální port pro odeslání paketu zprávou `packet-out` do tabulky toku a byla také přidána priorita toku [21].

Další výrazné změny přinesla verze 0.8.9, která byla vydána na konci roku 2008. V této verzi je již možné přiřadit záznamu masku podsítě. Mezi hlavní vylepšení dále patří rozšíření zprávy, která obsahuje informace o stavu portu OpenFlow směrovače, přidání zpráv *OFPT\_ECHO\_REQUEST/REPLY* a přidání podpory STP protokolu do záznamu v tabulce toků [21].

V roce 2009 byla vydána verze 0.9, ve které byl přidán mechanismus Failover. OpenFlow směrovač může být spravován více SDN kontroléry, které se nacházejí za sebou v seznamu. Pokud dojde k selhání prvního řadiče, mechanismus Failover

automaticky přepne na následující SDN kontrolér v seznamu. Dále byly přidány havarijní záznamy do tabulky toků, které jsou aktivovány v případě výpadku spojení mezi OpenFlow směrovačem a SDN kontrolérem, či bariérový příkaz, který oznámí, kdy OpenFlow zpráva v OpenFlow směrovači dokončila činnost [21].

Konec roku 2009 přinesl verzi 1.0, která již podporuje garanci minimální šířky pásma na výstupním portu OpenFlow směrovače pomocí front. Dále byla ve zprávě *OFPT\_DESC* přidána položka pro popis OpenFlow směrovače, byla implementována možnost porovnávání IP polí v ARP paketu a upraveno vyjádření doby toku a zpráv v OpenFlow směrovači (vyjádřeno v nanosekundách) [21].

Verze 1.1, vydána v roce 2011, přináší vylepšení OpenFlow potrubí, ve kterém se již nachází více tabulek toků. Dále byla přidána tabulka skupin, podpora značkování VLAN a MPLS a rozšíření fyzických portů o virtuální porty [21].

V tomtéž roce, v rámci verze 1.2, došlo k výrazné změně metody porovnávání paketů se záznamy v tabulce toků. V předchozích verzích měla tato metoda statickou strukturu, což často způsobovalo přeplnění porovnávacích polí. V této verzi se již metoda nachází v TLV formátu a je pojmenována jako OpenFlow Extensible Match (OXM). Každé logické porovnávací pole je tedy unikátní, čímž je zajištěna větší flexibilita. Dále byla v této verzi přidána podpora IPv6 protokolu a podpora mechanismu Failover pro SDN kontroléry [21].

Verze 1.3, publikována v roce 2012, již obsahuje flexibilnější framework pro popis schopností OpenFlow směrovače a záznam table-miss, popisující situaci, kde není nalezen záznam pro zpracováváný paket. Dále byla přidána podpora per-flow metru, který může být spojený se záznamem v tabulce toků a jehož úkolem je měřit a řídit míru paketů, které jsou posílány na SDN kontrolér. Dalším rozšířením jsou pomocná spojení mezi OpenFlow směrovačem a SDN kontrolérem, které obvykle přenáší zprávy packet-in a packet-out [21].

Ve verzi 1.4, vydanou v roce 2013, je implementována podpora optických portů, zavedena možnost monitoringu toku z pohledu SDN kontroléru v reálném čase, je vytvořen mechanismus, který při přeplnění tabulky toků umožňuje OpenFlow směrovači uvolňovat místo v tabulce mazáním méně potřebných záznamů či varovat SDN kontrolér v případě rizika přeplnění [21].

## 2 SOFTWAREVĚ DEFINOVANÉ OPTICKÉ SÍTĚ

Koncept SDN nezahrnuje pouze řešení v rámci paketových sítí, ale obsahuje proprietární řešení pro optické síťové spoje. Konfigurace jednotlivých prvků v současných páteřních sítích je prováděna manuálně pomocí systému Element Management System (EMS) či Network Management System (NMS). Takovéto řešení je na jednu stranu spolehlivé, avšak je nákladné a nezajišťuje dynamickou správu páteřních optických sítí. Navíc kvůli rozdílné architektuře IP a optických sítí musí obě sítě pracovat odděleně bez vzájemné interakce. Další možností je správa optických sítí pomocí technologie Generalized Multi-Protocol Label Switching (GMPLS). Tato technologie je však příliš komplexní na to, aby mohla být nasazována do reálného provozu. Pro efektivnější a méně nákladnou správu optických sítí byla využita myšlenka SDN, která je založena na rozšířeném OpenFlow protokolu [12].

Aby však bylo možné vytvořit jednotné řízení optické sítě na bázi SDN, musí být splněny konkrétní požadavky [6]:

- návrh a implementace hardwarové abstrakce, která umožní skrýt detaily optické přenosové vrstvy a realizovat pouze její zobecněné funkce,
- definice optického toku, který bude všeobecný pro různé optické přenosové technologie a bude kompatibilní s technologií přepínání paketů,
- vzít v potaz specifikace a omezení fyzické vrstvy různých optických přenosových technologií.

Jak již bylo řečeno, cílem hardwarové abstrakce je skrýt technologické detaily optické přenosové vrstvy a umožnit programovatelné rozhraní pro konfiguraci hardwarových prostředků této vrstvy. Hardwarová abstrakce se skládá ze 3 podvrstev [6]:

- **Hardwarová prezentační vrstva (Hardware Presentation Layer)** - nejnižší vrstva, která umožňuje skrýt hardwarové složitosti a poskytnout pouze funkce a schopnosti hardwarových zařízení v optické přenosové vrstvě na základě jednotného informačního modelu, který posílá následující vyšší vrstvě.
- **Hardwarová vrstva rozhraní (Hardware Interface Layer)** - využívá surových informací, poskytnuté hardwarovou prezentační vrstvou, k sestavení ucelené abstrakce. V abstrakci jsou poté zpřístupněny pouze funkce a informace, které mohou být využity technologií OpenFlow.
- **Rozhraní pro programování aplikací (Application Programming Interface)** - jedná se o aplikace, které mapují abstraktní informace, poskytnuté hardwarovou vrstvou rozhraní, do OpenFlow protokolu.

Řešení pomocí podvrstev je implementováno do již známého OpenFlow směrovače. Takto rozšířený OpenFlow směrovač se nazývá OpenFlow agent, který umožňuje abstraktní pohled na hardwarové zařízení optické přenosové vrstvy.

Optický tok lze definovat pomocí identifikátoru toku, který obsahuje port, vlnovou délku, šířku pásma a typ signálu, spojeného se specifickou optickou přenosovou technologií. V definici je zahrnuta i řada problémů, týkající se fyzické vrstvy (např. citlivost či výkonový rozsah). Celková definice je dostačující k tomu, aby mohl být optický tok zařazen do optických přenosových technologií [6].

## 2.1 Rozšíření OpenFlow protokolu

Aby byla zaručena komunikace mezi OpenFlow agentem a SDN kontrolérem, je nutné přidat nové zprávy do OpenFlow protokolu. Mezi tyto nové zprávy patří *OFPT\_SWITCH\_FEATURE* a *OFPT\_FLOW\_MOD*. Informace o dostupných funkcích prvků v optické síti jsou obsaženy ve zprávě *OFPT\_SWITCH\_FEATURE*, kterou vyšle OpenFlow agent na SDN kontrolér [6]. Zprávy *OFPT\_FLOW\_MOD* slouží k přímé konfiguraci prvků, nacházejících se v optické síti [5].

Kromě základních OpenFlow zpráv se v OpenFlow protokolu navíc vyskytují další dvě rozšiřující zprávy. Jedná se o zprávy switching constraints a power equalization. Zprávy switching constraints popisují, jak jsou fyzické porty navzájem propojeny a jaký optický signál může protékat mezi porty. Zprávy power equalization slouží ke spuštění funkce pro rozložení výkonu přes vnitřní signálovou cestu mezi porty [6].

## 2.2 Řízení optické cesty pomocí OpenFlow

Implementace OpenFlow protokolu pro aplikaci v optických sítích spočívá v rozšíření stávajících standardních fyzických zařízení o vyšší řídicí vrstvy podporující OpenFlow. Aby bylo možné řídit optické uzly, nacházející se v optické síti, jsou agentovi implementovány virtuální ethernetová rozhraní veths. Jedná se o virtualizované porty optického uzlu, podle kterých si agent dokáže vytvořit přehled jeho fyzické struktury. Informace o stavu portů optického uzlu poté agent může posílat na SDN kontrolér pomocí zprávy *OFPT\_FEATURES\_REPLY*. SDN kontrolér je poté schopen pomocí OpenFlow agenta řídit činnost optického uzlu prostřednictvím zprávy *OFPT\_FLOW\_MOD* [12].

Jakmile SDN kontrolér obdrží informace o příchozím datovém toku a získá jeho zdrojovou a cílovou IP adresu, může na základě své znalosti celé sítě provést směrování a přiřazení vlnové délky optickému signálu. Po obdržení výsledků poté dojde



k přiřazení nového záznamu SDN kontrolérem do tabulky toků OpenFlow agenta. OpenFlow agent na základě záznamu v tabulce toků automaticky vyšle příkazy přes TCP protokol na odpovídající porty daného uzlu pro sestavení optické cesty. Příkazy, vysílané OpenFlow agentem na optický uzel se nacházejí ve formátu TL1 (Transaction Language 1) a slouží pro konfiguraci optického uzlu [12].

Pro dynamické řízení optické cesty jsou definována celkem čtyři řešení. V případě sestavení cesty se jedná o sekvenční a zpožděné řešení, pro odpojení cesty je využito aktivní a pasivní řešení [12].

Sestavení cesty [12]:

- **Sekvenční řešení** – jedná se o způsob, kdy SDN kontrolér řídí OpenFlow směrovač a optické uzly v postupném pořadí. Necht nastane situace, kdy IP tok dorazí na OpenFlow směrovač v tradiční IP síti. Pokud IP tok (první paket toku) neodpovídá žádnému záznamu v tabulce toků, vyšle OpenFlow směrovač tento paket na SDN kontrolér. SDN kontrolér vypočítá cestu, přiřadí vlnovou délku optického signálu a nakonec přidá nový záznam do tabulky toků v OpenFlow směrovači a OpenFlow agentům v optické síti. OpenFlow agenti poté vyšlou TL1 příkazy na své optické prvky, které vytvoří propojení mezi uzly v optické síti, čímž dojde k vytvoření optické cesty a IP tok dorazí do cíle. Toto řešení je z hlediska realizace přímočaré a rychlé, avšak jeho nevýhodou je výskyt zpoždění během konfigurace jednotlivých uzlů a také během komunikace mezi OpenFlow agentem a uzlem. Zpoždění může způsobit nekompletní sestavení optické cesty, což může vést ke ztrátám paketů.
- **Zpožděné řešení** – na rozdíl od sekvenčního řešení nejprve dojde k sestavení optické cesty SDN kontrolérem. SDN kontrolér poté záměrně čeká s určitým časovým zpožděním až bude cesta kompletně sestavena. Po uplynutí čekací doby dojde k přidání záznamu do tabulek toků SDN kontrolérem. Výhodou tohoto řešení je garance doručení paketu do cíle, avšak za cenu delší doručovací doby, způsobené čekáním SDN kontroléru na sestavení optické cesty.

Odpojení cesty [12]:

- **Aktivní řešení** – jedná se o řešení, ve kterém SDN kontrolér vloží do záznamu v tabulce toků OpenFlow agenta pevný časový limit timeout, který indikuje, jak dlouho má být optická cesta sestavena. Po vypršení limitu OpenFlow agent vymaže záznam z tabulky a následně vyšle uzlu příkaz k odpojení cesty. Aktivní řešení může být použito v případě, kdy SDN kontrolér předem zná množství dat, které bude přenášeno.
- **Pasivní řešení** – na rozdíl od aktivního řešení je pasivní řešení založeno na indikaci, která oznamuje SDN kontroléru, že přenos byl úspěšně ukončen. Cílový klient tedy vyšle paket, který oznamuje ukončení přenosu. Po obdržení paketu SDN kontrolérem dojde k vymazání záznamu v tabulce toků OpenFlow

agenta, který následně vyšle uzlu příkazy k odpojení cesty.

## 2.3 Kombinace OpenFlow s výpočetní jednotkou PCE

Velkým problémem, se kterým se potýká OpenFlow je škálovatelnost. V důsledku centralizace jsou na SDN kontrolér kladeny vysoké požadavky, které jsou náročné na procesor SDN kontroléru (výpočet optické cesty, přiřazování vlnových délek, signalizace přes OpenFlow protokol). Z tohoto důvodu je nutné zavést řešení, které umožní optimální rozložení zátěže SDN kontroléru a tím pádem lepší škálovatelnost celého systému [12].

Pro vyřešení tohoto problému je využita samostatná výpočetní jednotka Path Computation Element (PCE), nacházející se v řídicí rovině. Úkolem jednotky PCE je pomocí implementovaného Dijkstrova algoritmu vypočítat nejkratší optickou cestu, přes kterou bude vedena komunikace v optické síti a zároveň přiřadit odpovídající vlnovou délku. Aby však bylo možné výpočet uskutečnit, musí PCE jednotka znát topologii dané sítě, což vede k zavedení takzvaného topologického serveru [12].

Topologický server se může vyskytovat samostatně nebo jako součást SDN kontroléru. Jeho úkolem je shromažďovat a ukládat veškeré Traffic Engineering (TE) informace o síti a také informace, týkající se fyzické vrstvy. V případě sestavení či odpojení optické cesty, dojde k aktualizaci TE informací SDN kontrolérem. Aktualizované informace poté SDN kontrolér vyšle na topologický server ve formě XML souboru. Topologický server si poté tyto informace uloží do své databáze zvané Traffic Engineering Database (TED). Následně SDN kontrolér vyšle PCE jednotce zprávu PCReq pro provedení výpočtu cesty. Pro komunikaci mezi SDN kontrolérem a jednotkou PCE je využit klient Path Computation Client (PCC), který je součástí SDN kontroléru. Po obdržení zprávy PCReq dojde k vyslání zprávy TED Req na topologický server. Server následně odpoví zprávou TED Rep, která obsahuje XML soubor s aktuálními informacemi o síti, což znamená, že jednotka PCE je tedy schopna začít provádět výpočet optické cesty. Po dokončení výpočtu dojde k vyslání zprávy PCRep jednotkou PCE, která obsahuje informace a atributy o optické cestě. SDN kontrolér poté na základě obdržené PCRep zprávy provede konfiguraci jednotlivých uzlů prostřednictvím OpenFlow [12].

## 2.4 Kombinace OpenFlow s GMPLS

Pro zajištění společné správy IP a optických sítí bylo navrženo řešení, které je založeno na začlenění OpenFlow do již existujících GMPLS technologií. Tato integrace

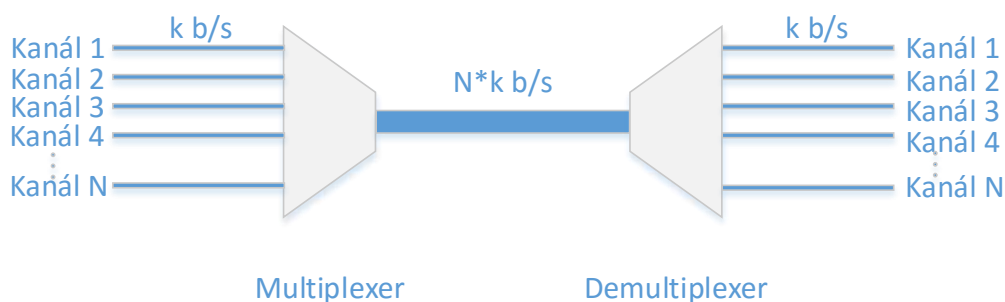
může být provedena třemi způsoby [12]:

- **Paralelní způsob** – jedná se o způsob, kdy řídicí roviny OpenFlow a GMPLS jsou zcela oddělené. SDN kontrolér komunikuje s řídicí rovinou GMPLS přes proprietární rozhraní. Nechtě nastane situace, kdy paket dorazí na SDN kontrolér. SDN kontrolér na základě zdrojové a cílové IP adresy zažádá řídicí rovinu GMPLS pro sestavení cesty LSP (Label Switched Path) v optické síti, což se provede přes protokol RSVP (Resource Reservation Protocol). SDN kontrolér poté čeká na úplné sestavení LSP cesty. Po vypršení čekací doby je následně SDN kontrolérem vložen nový záznam do tabulky toků OpenFlow směrovače. Nevýhodou tohoto řešení jsou možné problémy v komunikaci mezi SDN kontrolérem a řídicí rovinou GMPLS, jelikož komunikační rozhraní není založeno na standardním protokolu. Pro vyřešení zmíněného problému je možné využít překrytého způsobu začlenění.
- **Překrytý způsob** – v tomto případě je ke každému GMPLS kontroléru přiřazen OpenFlow agent, což umožní SDN kontroléru současnou správu IP a optické sítě pomocí OpenFlow protokolu. Po ukončení výpočtu cesty a přiřazení vlnové délky dojde k přidání nového záznamu OpenFlow agentům. Zároveň OpenFlow agenti obdrží od SDN kontroléru zprávu *OFPT\_FLOW\_MOD*, která obsahuje informace o vypočítané cestě. OpenFlow agenti poté informují GMPLS kontroléry k sestavení optické cesty.
- **Integrovaný způsob** – jedná se o řešení, ve kterém jsou GMPLS kontroléry a OpenFlow agenti integrováni do jednoho výsledného kontroléru. Tento kontrolér se poté nazývá OpenFlow-enabled GMPLS (OF-GC) a je schopen komunikovat s SDN kontrolérem přes OpenFlow protokol. Výhodou tohoto řešení je fakt, že všechny operace, které probíhají mezi OpenFlow agenty a GMPLS kontroléry, se vyskytují uvnitř jednoho společného OF-GC kontroléru, což má za následek celkové snížení latence. Nevýhodou je však vyšší zátěž OF-GC kontroléru.

### 3 SDÍLENÍ KANÁLŮ V PÁTEŘNÍCH OPTICKÝCH SÍTÍCH

Nezbytnou součástí v současných páteřních optických sítích je technologie multiplexování. Tato technologie umožňuje mnoha uživatelům vysílat data současně přes jednu linku, ať už se jedná o koaxiální kabel, optické vlákno či rádiové nebo satelitní spojení. Multiplexování je široce využíváno v komunikačních systémech a to zejména kvůli schopnosti zvětšit přenosovou kapacitu či efektivněji využít přenosový kanál a zároveň snížit celkové náklady na systém [3].

Základní princip multiplexování je zobrazen na obr. 3.1. Do multiplexu vstupuje několik  $n$  kanálů. Multiplex tyto kanály zkombinuje do jediné linky tak, aby byly oddělitelné. Pokud každý kanál na vstupu přenáší  $k$  b/s dat, celková přenosová rychlost výsledné linky je poté  $N \times k$  b/s. Na konci linky se nachází demultiplexer, který má opačnou funkci jako multiplexer. Jeho úkolem je oddělit jednotlivé kanály z jedné linky a doručit je do příslušných výstupních linek [3].



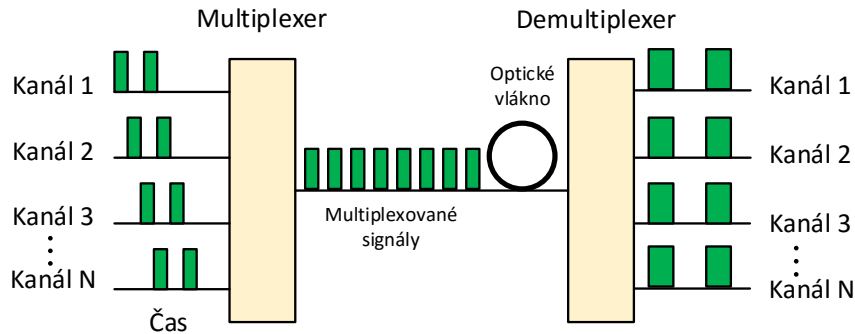
Obr. 3.1: Princip multiplexování [3].

Mezi nejznámější techniky multiplexování v optických sítích patří [24]:

- **Prostorový multiplex SDM (Space Division Multiplexing)** – jedná se o techniku, ve které jsou nasazována další optická vlákna. Je udržována stejná přenosová rychlost, ale pro přenos je využito více optických vláken. V současné době se již téměř nepoužívá.
- **Optický časový multiplex OTDM (Optical Time Division Multiplexing)** – způsob multiplexování, který umožňuje zvýšit přenosovou rychlost optickým vláknem. Vzhledem k vyšším nákladům na provoz je od této techniky ustupováno.
- **Vlnový multiplex WDM (Wavelength Division Multiplexing)** – umožňuje přenášet více vlnových délek jediným optickým vláknem. V současné době se jedná o nejpoužívanější techniku multiplexování.

### 3.0.1 Optický časový multiplex

Jedná se o techniku, která umožňuje sloučit několik signálů do jediného toku, jehož výsledná přenosová rychlost je dána součtem přenosových rychlostí jednotlivých signálů. Na obr. 3.2 je znázorněn princip optického časového multiplexu pro N kanálů přenášející signály se stejnou vlnovou délkou a stejnou přenosovou rychlostí X.



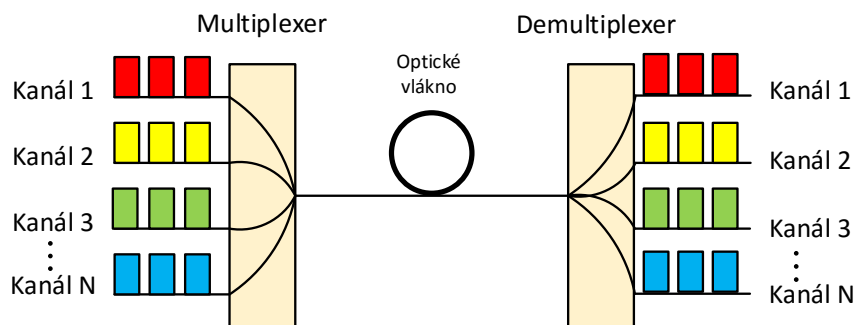
Obr. 3.2: Optický časový multiplex [26].

Na vstupu se nachází multiplexer, který pracuje na principu přepínače. Multiplexer opakovaně přepíná mezi jednotlivými kanály a připojuje je do výsledného datového toku. Průběh přepínání probíhá opakovaně s frekvencí F, kdy každému kanálu je vyhrazen časový interval  $T_0$ , zvaný timeslot. V tomto časovém intervalu je pak přenášén jeden symbol signálu [1].

Jak již bylo zmíněno, výhodou této metody je získání vyšších přenosových rychlostí. Na druhou stranu, nevýhodou je vyšší citlivost systému na nelinearity optického vlákna a vysoká cena.

### 3.0.2 Vlnový multiplex

Vlnový multiplex, znázorněný na obr. 3.3, umožňuje nezávislým uživatelům přenášet svá data přes jediné optické vlákno využitím různých vlnových délek.



Obr. 3.3: Vlnový multiplex [3].

Na vysílací straně se nachází  $N$  uživatelů, z nichž každý může využívat pro přenos specifickou modulaci, kde navíc každý kanál využívá specifické vlnové délky. Tyto vlnové délky jsou určeny dle standardu ITU-T. Vlnový multiplexer poté tyto optické signály zkombinuje do jediného vlákna. Na přijímací straně se nachází demultiplexer, který, s pomocí  $N$  optických filtrů, rozdělí optické signály do správných kanálů. Celková kapacita WDM linky závisí na tom, jak blízko mohou být jednotlivé kanály rozmístěny [3]. Hlavní nevýhodou WDM je nízké využití kanálů a spektrální účinnosti, jelikož na jednoho uživatele připadá jedna vlnová délka. Proto pro multiplexování  $N$  uživatelů je vyžadováno  $N$  vlnových délek neboli  $N$  světelných zdrojů s  $N$  optickými filtry, což navyšuje celkové náklady na systém.

Pro WDM řešení se využívají 3 vlnové multiplexy [10, 13]:

- **Wide Wavelength Division Multiplexing (WWDM)** – nejstarší technologie, označována jako široký multiplex. Využívá vlnové délky v oblasti kolem 1300 a 1310 nm s odstupem 25 nm.
- **Coarse Wavelength Division Multiplexing (CWDM)** – hrubý vlnový multiplex, který dle ITU-T G.694.2 využívá vlnové délky v rozsahu od 1271 nm do 1611 nm s odstupem 20 nm. Z toho vyplývá, že tento multiplex umožňuje přenášet až 18 optických signálů.
- **Dense Wavelength Division Multiplexing (DWDM)** – hustý vlnový multiplex, který je charakteristický užším odstupem kanálů než multiplex CWDM. V důsledku zaměření této práce na využití technologie DWDM pro SDN je princip DWDM detailněji rozebrán v následujících sekcích.

### 3.1 DWDM technologie

Vznik DWDM je jedním z nejvýznamnějších jevů ve vývoji technologie přenosu optickým vláknem. Hlavní výhodou DWDM technologie je schopnost poskytovat potenciálně neomezenou kapacitu přenosu, což je obrovskou výhodou z technického i ekonomického hlediska. Zejména v současné době, kdy požadavky na šířku přenosového pásma se neustále zvětšují, je nasazením DWDM technologie kapacitu možné zvýšit buď jednoduchou modernizací zařízení nebo zvýšením počtu vlnových délek přenášených v optickém vlákně [8].

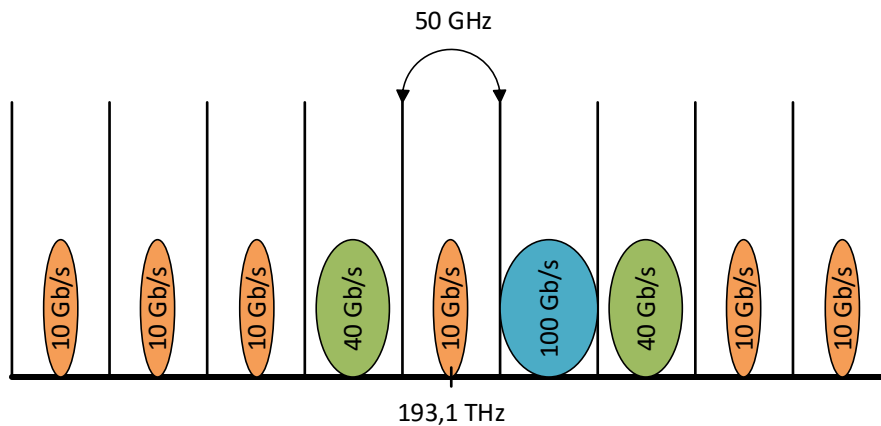
Jak již bylo řečeno, DWDM se od klasického WDM liší v užších rozestupech jednotlivých vlnových délek (kanálů), což zajišťuje větší celkovou kapacitu. Další vlastností DWDM je možnost zesílit všechny vlnové délky najednou bez nutnosti konverze na elektrické signály a také možnost současného přenosu různých typů signálu o různých přenosových rychlostech. Maximální možná kapacita DWDM není přesně známá a pravděpodobně se jí ještě nepodařilo dosáhnout [8].

Celý systém DWDM je založen na pěti primárních funkcích [8]:

- **Generování signálu** – zdrojem světla je laser, který zajišťuje stabilní světlo v rámci konkrétního pásma, které přenáší digitální data modulované jako analogový signál.
- **Kombinace signálů** – moderní DWDM systémy využívají ke kombinaci příchozích signálů do jediného vlákna multiplexery.
- **Přenos signálu** – během přenosu signálu je nutné vzít v úvahu možné přeslechy či zhoršení kvalit signálu.
- **Oddělení přijatých signálů** – na přijímací straně musí být multiplexované signály odděleny.
- **Přijímání signálů** – demultiplexované signály jsou přijímány fotodetektorem.

### 3.1.1 DWDM dle ITU-T G.694.1

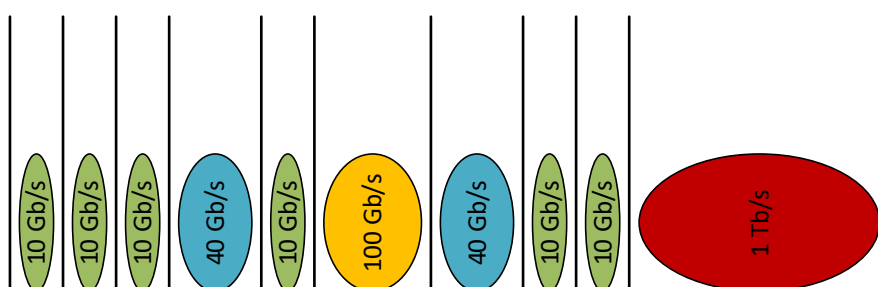
Obecně platí, že lasery, používané v systému DWDM, využívají kontrolní mechanismus, který jim umožňuje splnit požadavky, týkající se frekvenční stability. Dle ITU-T G.694.1 jsou rozestupy kanálů rozděleny na fixní a flexibilní. Fixní rozestupy jsou dle standardu definovány na 12,5 GHz, 25 GHz, 50 GHz, 100 GHz a výše s centrální vlnovou délkou o frekvenci 193,1 THz neboli 1552,52 nm [14]. Fixní rozestup pro 50 GHz je znázorněn na obr. 3.4.



Obr. 3.4: Fixní rozestup kanálů pro 50 GHz [19].

Pro jednotlivé přenosové kanály je definován rozsah vlnových délek od 1490 nm (200,95 THz) do 1620 nm (186,00 THz) [9]. Například při rozsahu 100 GHz je definován odstup jednotlivých kanálů cca 0,8 nm, v případě rozsahu 50 GHz (dvojnásobný počet přenosových kanálů) je definován odstup cca 0,4 nm [9]. Pro určení vlnové délky fixního rozestupu platí obecný vzorec  $X + n \times Y$ , kde  $X$  je centrální vlnová délka,  $n$  je kladné nebo záporné celé číslo definující číslo kanálu a  $Y$  je hodnota fixního rozestupu [14].

Pro podporu současných a zároveň budoucích přenosových rychlostí je zaveden flexibilní rozestup, který je zobrazen na obr. 3.5. Hlavní motivací pro zavedení flexibilního rozestupu je možnost dynamického nastavení šířky pásma optických kanálů. Dle ITU-T G.694.1 je nominální centrální frekvence flexibilního DWDM definována jako  $193,1 + n \times 0,00625$ , kde  $n$  je záporné nebo kladné celé číslo a  $0,00625$  THz je výchozí nominální centrální frekvence. Dále je dle standardu definován výchozí rozestup kanálů na  $12,5$  GHz s možností definovat spektrální šířku tzv. superkanálu. Tyto superkanály lze flexibilně definovat tak, aby bylo možné pojmout jakoukoliv kombinaci optických nosičů, modulace či přenosových rychlostí. Šířka superkanálu je definována jako  $12,5 \times m$ , kde  $m$  je kladné celé číslo [14].



Obr. 3.5: Příklad přenášených šířek pásma u flexibilního rozdělení DWDM kanálu [19].



## 4 OPTICKÉ PRVKY V PÁTEŘNÍCH DWDM SÍTÍCH

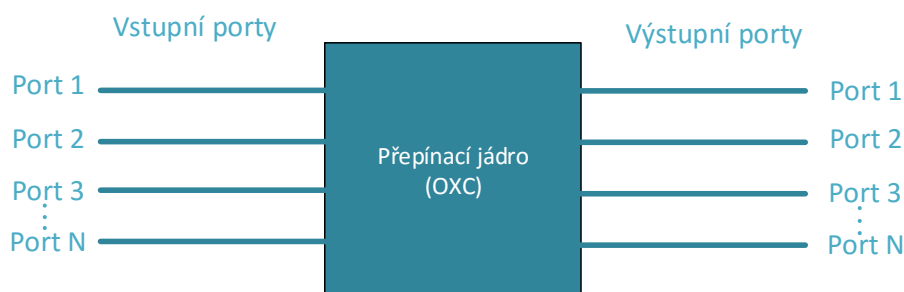
### 4.1 Optický křížový spoj

Optický křížový spoj OXC (Optical Crossconnect) je optický prvek, který je nasazován zejména na rozsáhlé a komplexní optické sítě s velkým množstvím vlnových délek. OXC také hraje klíčovou roli v zajištění rekonfigurovatelných optických sítí, ve kterých je možné dálkově sestavovat a odpojovat optické trasy dle potřeby [24]. Obecně lze klíčové funkce OXC shrnout do následujících bodů:

- **Poskytování služeb** – OXC může být v rozsáhlých optických sítích řízen automatizovaným způsobem. Tato schopnost je důležitá zejména v případě, kdy je nutné se vypořádat s velkým množstvím vlnových délek v uzlu nebo velkým množstvím těchto uzlů v síti. Dále je tato schopnost důležitá v případě, kdy je nutné reagovat na náhlé změny síťového provozu. V případě manuální správy, kdy je nutné vyslat pověřeného pracovníka k obsluze každého uzlu zvlášť, je správa jednotlivých uzlů nákladná a více náchylná k výskytu chyb.
- **Zabezpečení** – OXC je inteligentní optický prvek, který dokáže detekovat výpadky v optické síti (např. přerušení optických vláken či selhání zařízení). V případě vyhnutí se výpadku umožňuje také rychle přeměřovat optické trasy.
- **Monitoring** – OXC umožňuje monitorovat parametry signálu v mezilehlých uzlech.
- **Konverze vlnových délek** – v případě přepínání signálu může OXC obsahovat funkce pro konverzi vlnových délek [24].

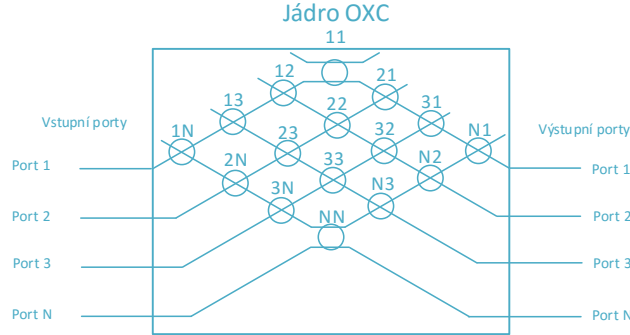
#### 4.1.1 Architektura OXC

Základní architektura OXC je zobrazena na obr. 4.1.



Obr. 4.1: Základní architektura OXC [24].

Z funkčního hlediska lze OXC rozdělit na dvě části. Hlavní část je tzv. přepínací jádro, ve kterém se nachází přepínač, který provádí funkci křížového přepínání příchozích vlnových délek. Vnitřní struktura přepínacího jádra je zobrazena na obr. 4.2. Druhá část je tzv. komplex portů. Jedná se o vstupní a výstupní porty, které umožňují komunikaci OXC s dalšími zařízeními v optické síti [24].



Obr. 4.2: Vnitřní struktura přepínacího jádra OXC [7].

Architekturu OXC je možné matematicky vyjádřit pomocí maticového vztahu:

$$\begin{pmatrix} I_1 \\ I_2 \\ I_3 \\ I_N \end{pmatrix} \times \begin{pmatrix} T_{11} & T_{1N} \\ T_{21} & T_{2N} \\ T_{31} & T_{3N} \\ T_{N1} & T_{NN} \end{pmatrix} = \begin{pmatrix} O_1 \\ O_2 \\ O_3 \\ O_N \end{pmatrix}$$

V tomto případě  $I_N$  vyjadřuje amplitudovou úroveň dopadajícího světla na vstupním portu  $N$  a  $O_N$  amplitudovou úroveň dopadajícího světla na výstupním portu  $N$ . Za  $T_{NN}$  je poté v ideálním případě, kdy zanedbáváme absorpci a disperzi světla, možné dosadit 1 či 0, vyjadřující připojení (1) či odpojení (0) portů [16].

V současné době existují dvě základní konfigurace OXC lišících se v metodě přepínání, které může být prováděno z elektrického nebo optického hlediska. První je tzv. neprůhledná (opaque) konfigurace, ve které je optický signál na vstupních portech OXC převáděn pomocí konvertorů na elektrický. Jádro OXC tedy přepíná čistě elektrický signál, který je na výstupních portech převáděn zpět na optický [15]. Výhodou této konfigurace je možnost jednoduššího monitoringu signálu s měřením bitové chybovosti. Nevýhodou je však vyšší pořizovací cena, větší spotřeba a zejména nižší rychlost přepínání signálu z důvodu konverze na vstupních a výstupních portech [24].

Druhá konfigurace je tzv. plně optická (transparent), ve které nedochází ke konverzi signálu, a tedy jádro OXC přepíná čistě optický signál. Tento typ se v některých referencích označuje jako fotonický křížový spoj PXC (Photonic Crossconnect). Hlavní výhodou plně optické konfigurace je zejména mnohem vyšší rychlost přepínání, nižší pořizovací a provozní náklady a především možnost zakomponování do

čistě optických tras s vysokou přenosovou rychlostí [15].

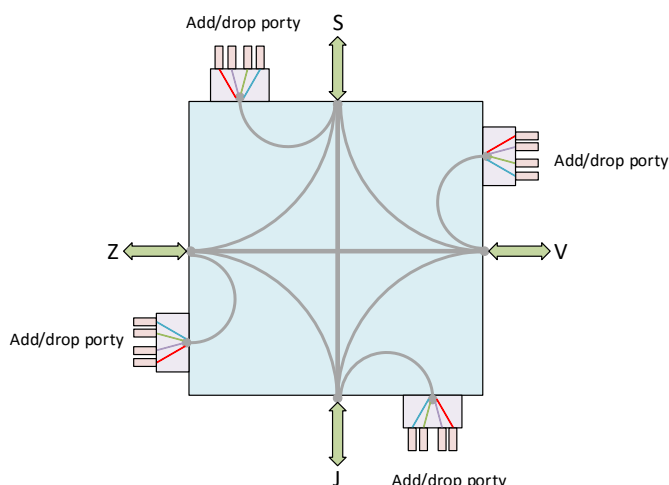
Plně optické OXC mohou pracovat na bázi tří metod:

- Prostorově optické přepínání (směrování vlnových délek) – využívají se Mach-Zehnderovy interferometry společně se směrovačem WGR (Waveguide Grating Router).
- Optické polovodičové zařízení – využívají se akusticko-optické a elektro-optické spoje.
- Elektromechanické zařízení – pracují na bázi zrcadel [16].

## 4.2 Rekonfigurovatelný optický add/drop multiplexer

Rekonfigurovatelný optický add/drop multiplexer ROADM (Reconfigurable Optical Add-Drop Multiplexer) je subsystém, který umožňuje dynamicky přidávat, vydělovat nebo propouštět procházející vlnové délky. Přidávání či vydělování vlnových délek v mezilehlých uzlech ovlivňuje kvalitu ostatních procházejících kanálů. V optických DWDM sítích je nasazován zejména za účelem zajištění flexibility při řízení vlnových délek a také za účelem snížení celkových nákladů na provoz sítě [11].

ROADM, nacházející se v DWDM sítích, je často rozlišován na základě stupňů. Každý stupeň označuje směr, ve kterém se uzel v DWDM síti propojí s jiným uzlem. Standardní ROADM je čtyřstupňový a je orientován v severním, jižním, východním a západním směru. Na obr. 4.3 se nachází schéma základního čtyřstupňového ROADM.



Obr. 4.3: Schéma ROADM [11].

V některých případech jsou využívány dvoustupňové, třístupňové či vícestupňové

ROADM. Zařazení určitého stupně vychází z topologie a řešení sítě [11].

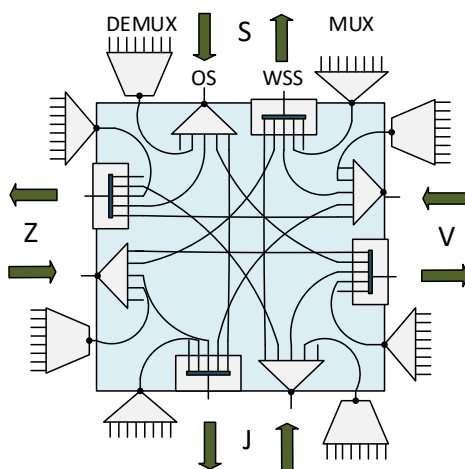
Současné ROADM je také nutné rozlišovat na základě svých vlastností a nastavení. Kromě již zmíněných stupňů se ROADM dále rozděluje na tzv. colored, colorless a directionless. Colored ROADM umožňuje přidat či vydělit pouze specifickou vlnovou délku, tj. každým portem může procházet pouze jedna barva světla. Colorless ROADM umožňuje na každém portu přidat či vydělit libovolné vlnové délky, které mohou být zároveň softwarově konfigurovatelné. Directionless ROADM umožňuje přidat či vydělit vlnové délky v libovolném směru, který je dán softwarově [2].

V rámci implementace ROADM je kromě vlnových multiplexerů a demultiplexerů používáno několik dalších základních stavebních prvků:

- Přepínač vlnových délek WSS (Wavelength Selective Switch) – hlavní komponenta, která umožňuje nezávisle přepínat vybrané vlnové délky. V současnosti se v ROADM používají WSS, které mají obvykle 5 nebo 9 vstupů a 1 výstup.
- Optický splitter – pasivní optický prvek, který umožňuje rozbočovat optický signál v závislosti na frekvenčním pásmu. Pokud je optický splitter použit v opačném směru, jedná se o optický coupler, který naopak optický signál slučuje.
- Optické zesilovače – umožňují obnovit a zesílit procházející optický signál [11].

#### 4.2.1 Architektura ROADM

Na obr. 4.4 se nachází způsob implementace čtyřstupňového ROADM, ve kterém jsou jednotlivé kanály směřovány do daných stupňů nebo do tzv. add/drop struktury, která je složena z vlnového multiplexeru a demultiplexeru. Tento způsob směřování je zajištěn využitím optického splitteru a WSS [11].



Obr. 4.4: Architektura colored ROADM [11].

Na vstupu každého stupně je optický signál přiveden do optického splitteru, který příchozí signál rozdělí na jednotlivé kanály a nasměruje je do WSS ostatních stupňů nebo do své add/drop struktury. V add/drop struktuře dojde k rozdělení kanálů demultiplexerem a následnému přiřazení jednotlivých vlnových délek do příslušných výstupních portů, čímž dojde k jejich vydělení. Naopak pomocí multiplexeru jsou kanály přidávány zkombinováním příchozích vlnových délek, které multiplexer vyše na porty WSS [11].

Nevýhodou této architektury je značné omezení způsobené add/drop strukturou, která je fixně nastavená jako colored, a tedy každým portem může procházet pouze jedna barva světla [11].

Dalším způsobem je možnost implementace ROADM v colorless provedení. Na rozdíl od colored architektury je v add/drop struktuře vlnový demultiplexer nahrazen WSS, který umožňuje přiřadit libovolnou vlnovou délku kterémukoliv portu. Nevýhodou této architektury je možný výskyt přeslechů mezi kanály [11].

V případě directionless realizace je jediná add/drop struktura připojena ke všem stupňům ROADM. Toto řešení je realizováno přidáním dalšího optického splitteru a WSS do add/drop struktury. Nevýhodou tohoto řešení je nutnost připojení optických zesilovačů na vstupu ROADM, jelikož vlivem využití přídavných komponentů v add/drop struktuře dochází k většímu oslabení optického signálu [11].

### 4.3 Optické linkové zakončení

Optické linkové zakončení OLT (Optical Line Terminal) je optický prvek, skládající se obvykle z transpondéru, vlnového multiplexeru a demultiplexeru a případně i z optického zesilovače. Hlavní funkce OLT je přizpůsobovat přicházející signál koncového uživatele na signál, který je kompatibilní pro přenos páteřní optickou sítí. Obvykle se jedná o správné přiřazení vlnové délky signálu. OLT dále umožňuje provádět dopřednou korekci chyb, která je využita zejména u signálů s přenosovou rychlostí 10 Gb/s a vyšší. OLT také typicky monitoruje signál na síťovém rozhraní k zachycení bitové chybovosti signálu. OLT se využívá především v souvislosti se zakončovacím zařízením ONU (Optical Network Unit) v pasivní optické síti PON (Passive Optical Network) a umísťuje se zejména na rozhraní mezi přístupovou a páteřní sítí. ONU se v PON obvykle využívá pro převod elektrického signálu na signál optický a naopak [24, 25].

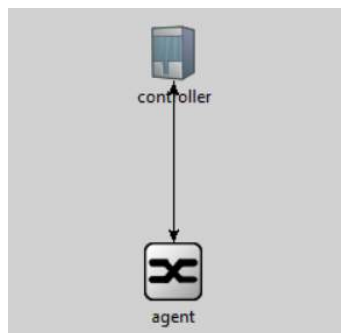
## 5 NAVÁZÁNÍ SPOJENÍ SDN KONTROLÉRU A OPENFLOW AGENTA V OMNET++

Pro simulaci softwarově definované optické sítě byl využit software OMNeT++ verze 4.6. Jedná se o objektově orientovaný simulační framework, který umožňuje modelovat síť a následně provést její simulaci. Každá síť je v OMNeT++ složena z tzv. modulů, které se dělí na jednoduché a složené. Jednoduché moduly je možno programovat v jazyce C++ s využitím simulačních knihoven. Složené moduly se poté mohou skládat z těchto jednoduchých modulů.

Softwarově definované optické sítě jsou realizovány na bázi OpenSource, stejně, jako SDN, a není proto jasně definována výsledná podoba. V současné době již existují simulace vyšších vrstev SDN, avšak pouze v elektrické doméně [17].

V simulaci je zobrazeno navázání spojení mezi SDN kontrolérem a jedním OpenFlow agentem, které probíhá na základě výměny inicializačních a konfiguračních zpráv. Sestavení spojení tvoří základ pro budoucí konfiguraci OpenFlow agenta přes SDN kontrolér. Jednotlivé simulace jsou vytvořeny na základě sekvenčního řešení pro řízení optické cesty.

Sestavená simulace je složena ze dvou jednoduchých modulů. Jedná se o modul OpenFlow agenta a SDN kontroléru. Oba moduly mají jeden vstup a jeden výstup a jsou vzájemně propojeny přes ideální kanál. Propojení obou modulů v prostředí softwaru OMNeT++ je zobrazeno na obr. 5.1.



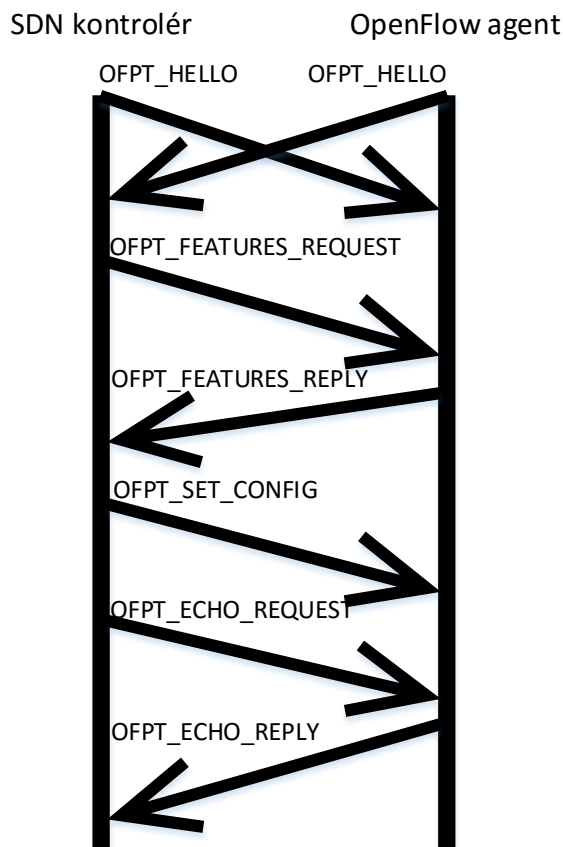
Obr. 5.1: Propojení modulů.

### 5.1 Průběh výměny zpráv

Kompletní sestavení spojení je zobrazeno na obr. 5.2. Jako první inicializuje spojení SDN kontrolér, který vyšle zprávu *OFPT\_HELLO* obsahující záhlaví. SDN kontrolér v něm nastaví svou nejvyšší podporovanou verzi OpenFlow protokolu, typ přenášené zprávy, celkovou délku zprávy a identifikátor xid. OpenFlow agent

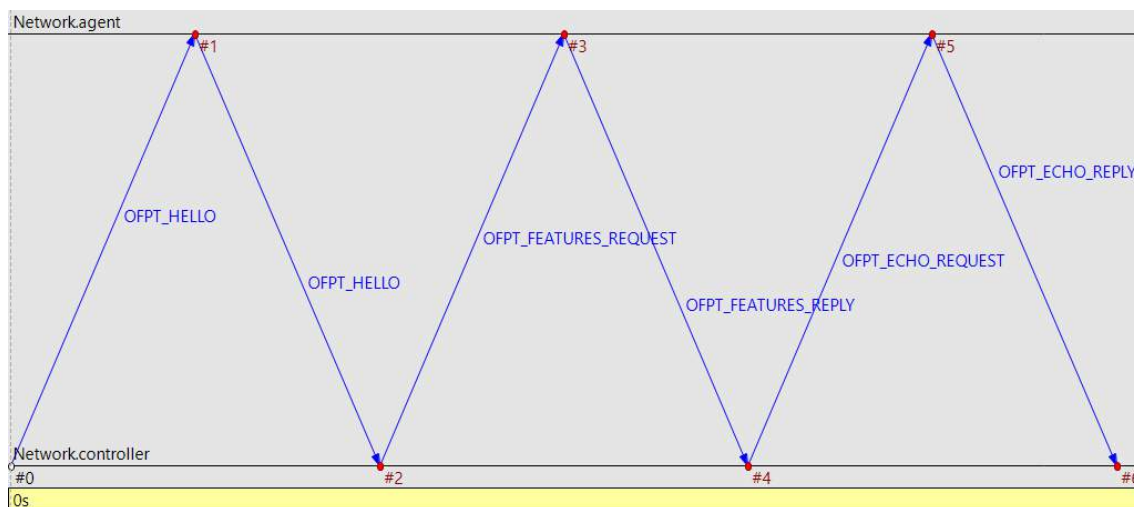
tuto zprávu přijme a odpoví na ni taktéž zprávou *OFPT\_HELLO*, čímž dojde k její výměně a k nastavení správné verze OpenFlow protokolu, na které bude spojení probíhat. Po přijetí *OFPT\_HELLO* zprávy, vyslanou OpenFlow agentem, si SDN kontrolér uloží její obsah do svých proměnných a následně vyšle zprávu *OFPT\_FEATURES\_REQUEST* pro získání funkcí a parametrů OpenFlow agenta.

Po jejím obdržení vyšle OpenFlow agent zprávu *OFPT\_FEATURES\_REPLY*, čímž odpovídá na požadavky zprávy *OFPT\_FEATURES\_REQUEST* vyslanou SDN kontrolérem. SDN kontrolér si obsah zprávy *OFPT\_FEATURES\_REPLY* opět uloží do svých proměnných. Zpráva *OFPT\_SET\_CONFIG* je v rámci simulace vynechána, jelikož OpenFlow agent je nastaven na standardní manipulaci s IP fragmenty a není nutné tento konfigurační parametr měnit. V dalším kroku je spojení mezi SDN kontrolérem a OpenFlow agentem udržováno na základě výměny zpráv *OFPT\_ECHO\_REQUEST* a *OFPT\_ECHO\_REPLY*.



Obr. 5.2: Navázání spojení.

Výsledek simulace navázání spojení ze softwaru OMNeT++ lze vidět na obr. 5.3.

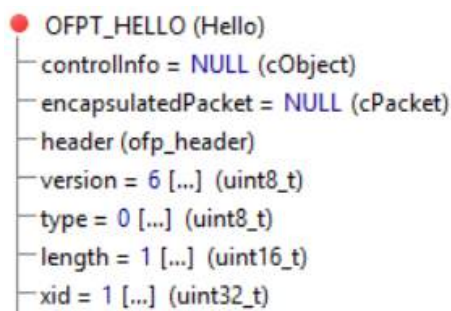


Obr. 5.3: Výsledek simulace navázání spojení.

## 5.2 Obsah a nastavení zpráv

### 5.2.1 OFPT\_HELLO

Obsah *OFPT\_HELLO* zprávy je zobrazen na obr. 5.4. Pole *version* obsahuje nejvyšší podporovanou verzi OpenFlow protokolu standardně nastavenou na hodnotu 0x06 pro SDN kontrolér i pro OpenFlow agenta. Tato hodnota vyjadřuje nejvyšší verzi OpenFlow protokolu, tedy 1.5.1. Pole *type* obsahuje typ aktuálně přenášené zprávy, které je definováno pomocí výčtu. Pole *length* vyjadřuje délku zprávy, jejíž hodnota je 32 bitů. Pole *xid* vyjadřuje identifikátor zprávy, který je definován pomocí počítadla. Počítadlo se při každé nové zprávě vyslané SDN kontrolérem zvýší o 1. OpenFlow agent odpovídá vždy stejným *xid*, které se nachází v přijaté zprávě.

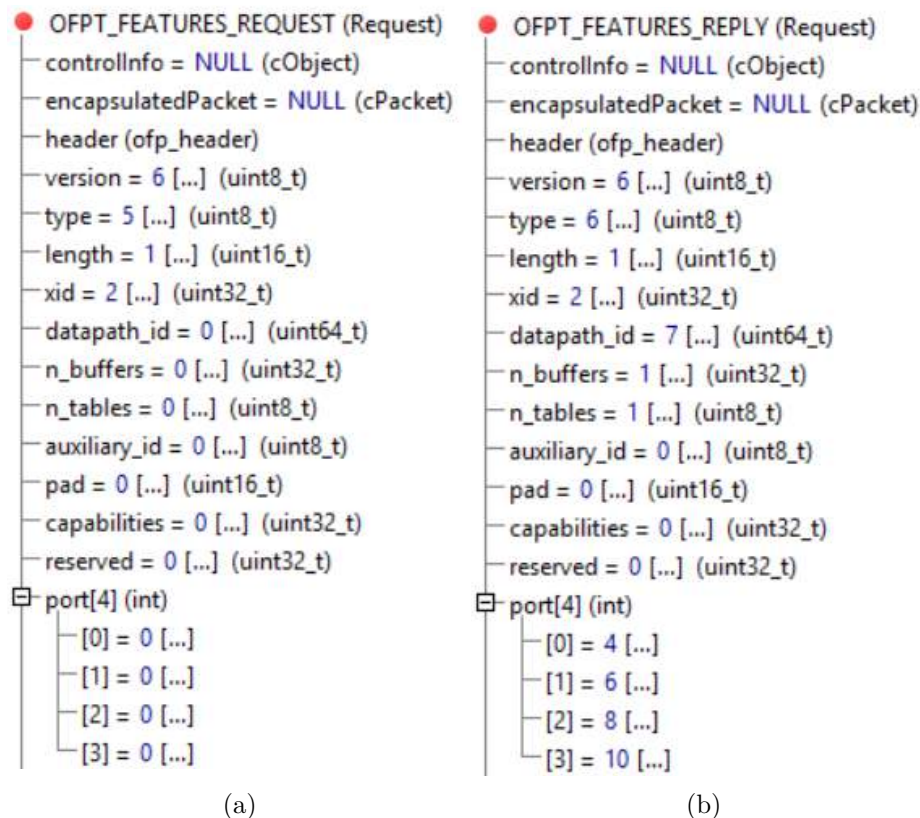


Obr. 5.4: Obsah zprávy *OFPT\_HELLO* implementované v OMNeT++.



## 5.2.2 OFPT\_FEATURES\_REQUEST/REPLY

Zpráva *OFPT\_FEATURES\_REQUEST* a *OFPT\_FEATURES\_REPLY* je zobrazena na obr. 5.5. Záhlaví zprávy *OFPT\_FEATURES\_REQUEST* se od zprávy *OFPT\_HELLO* liší v typu aktuálně přenášené zprávy a jejím xid. Zpráva dále obsahuje připravená pole pro získání parametrů OpenFlow agenta. K vyplnění těchto polí dojde ve zprávě *OFPT\_FEATURES\_REPLY*, kterou nastaví OpenFlow agent.

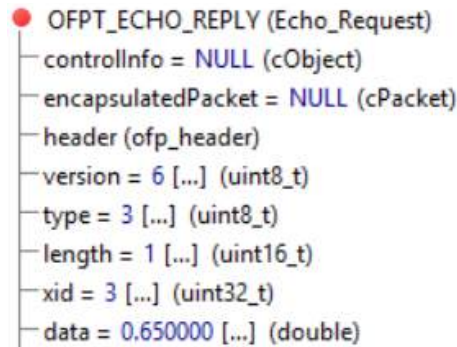


Obr. 5.5: a) *OFPT\_FEATURES\_REQUEST*, b) *OFPT\_FEATURES\_REPLY* implementované v OMNeT++.

V poli *datapath\_id* se nachází unikátní číslo, které slouží k rozpoznání a identifikaci OpenFlow agenta SDN kontrolérem. V softwaru OMNeT++ má každý modul své vlastní unikátní číslo, které se automaticky přiřadí ihned po vytvoření modulu či brány. OpenFlow agent dále obsahuje jednu tabulku toku a je schopen zařadit do fronty pouze jeden paket. Hodnota *auxiliary\_id* označuje 0 přidavných spojení a tedy je komunikace mezi OpenFlow agentem a SDN kontrolérem realizována přes jediné hlavní spojení. Pole *portů* obsahuje unikátní čísla vstupních bran modulu OpenFlow agenta, které jsou později využity SDN kontrolérem ke kontrole obsazenosti jednotlivých portů. Pole *pad*, *capabilities* a *reserved* nejsou v rámci simulace využívány.

### 5.2.3 OFPT\_ECHO\_REQUEST/REPLY

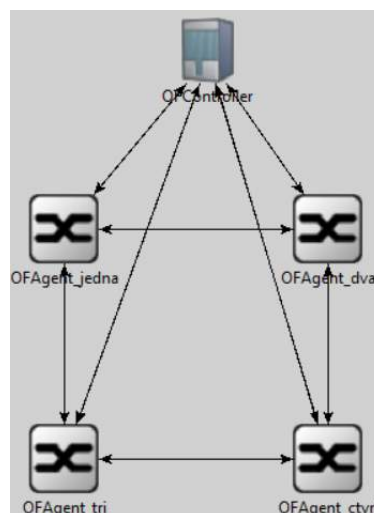
Na obr. 5.6 se nachází obsah zprávy *OFPT\_ECHO\_REQUEST/REPLY*. Zpráva obsahuje pouze záhlaví a pole dat, ve kterém se nachází informace o zpoždění, které vzniká při navázání spojení mezi SDN kontrolérem a OpenFlow agentem.



Obr. 5.6: Obsah zprávy *OFPT\_ECHO\_REQUEST* implementované v OMNeT++.

## 5.3 Rozšířené navázání spojení OpenFlow agentů s SDN kontrolérem

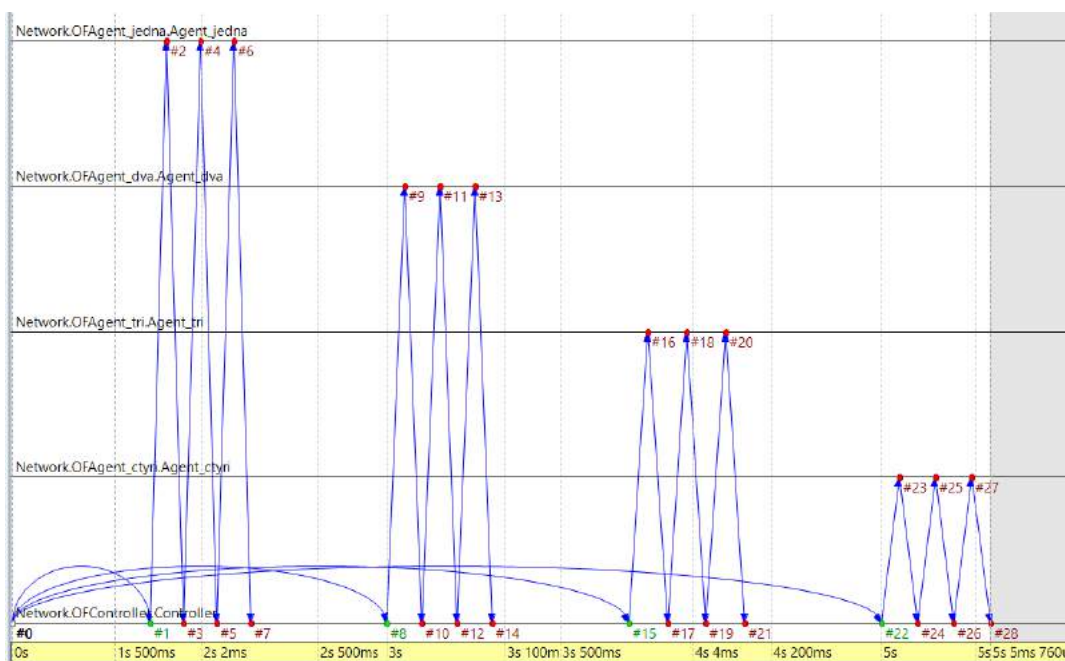
Pro sestavení softwarově definované optické sítě byla vytvořena topologie pro zajištění komunikace čtyř OpenFlow agentů s jedním SDN kontrolérem. Propojení jednotlivých OpenFlow agentů a SDN kontroléru v softwaru OMNeT++ je zobrazeno na obr. 5.7.



Obr. 5.7: Propojení OpenFlow agentů s SDN kontrolérem v OMNeT++ [18].

Během vytváření topologie je počítáno s reálným zpožděním, které se vyskytuje již při navázání spojení OpenFlow agentů a SDN kontroléru. V tomto případě však OpenFlow v současné době neobsahuje žádný standardní mechanismus, který by byl schopen měřit zpoždění již při úvodním navázání spojení. V dostupných referencích se však vyskytuje několik lišících se experimentálních metod a mechanismů, které umožňují zpoždění měřit. Z důvodu přiblížení se reálným hodnotám jsou v simulaci využity naměřené průměrné hodnoty zpoždění na základě referencí [4, 23, 27].

Každý OpenFlow agent je s SDN kontrolérem spojen přes zpožďovací kanál, který má definované konstantní zpoždění. Zpoždění je u jednotlivých kanálů různé a liší se na základě vzdálenosti OpenFlow agenta od SDN kontroléru. Pro jednotlivé kanály bylo definováno zpoždění 0,65 ms, 0,82 ms, 1,2 ms a 0,96 ms. Hodnoty zpoždění pro zpožďovací kanály jednotlivých OpenFlow agentů byly zvoleny na základě metody opakovaného měření pro maximálně 20 skoků (OpenFlow agentů) v rámci jednoho kanálu, kde prvnímu skoku odpovídá průměrný rozsah zpoždění od 0 ms do 2 ms [18]. Výsledek simulace ze softwaru OMNeT++ lze vidět na obr. 5.8.



Obr. 5.8: Průběh navázání spojení OpenFlow agentů s SDN kontrolérem [18].

V grafu lze vidět navázání spojení všech čtyř OpenFlow agentů s SDN kontrolérem, které proběhlo v maximálně stanoveném simulačním čase  $T = 5$  s. Jelikož dojde v průběhu navázání spojení k výměně celkem šesti OpenFlow zpráv, je celková doba zpoždění navázání spojení pro jediného OpenFlow agenta  $6 \times X$  ms, kde  $X$  je definované zpoždění kanálu. V případě prvního OpenFlow agenta (OFAgent\_jedna), jehož kanál má stanovené zpoždění 0,65 ms, je celková doba zpoždění v rámci navá-

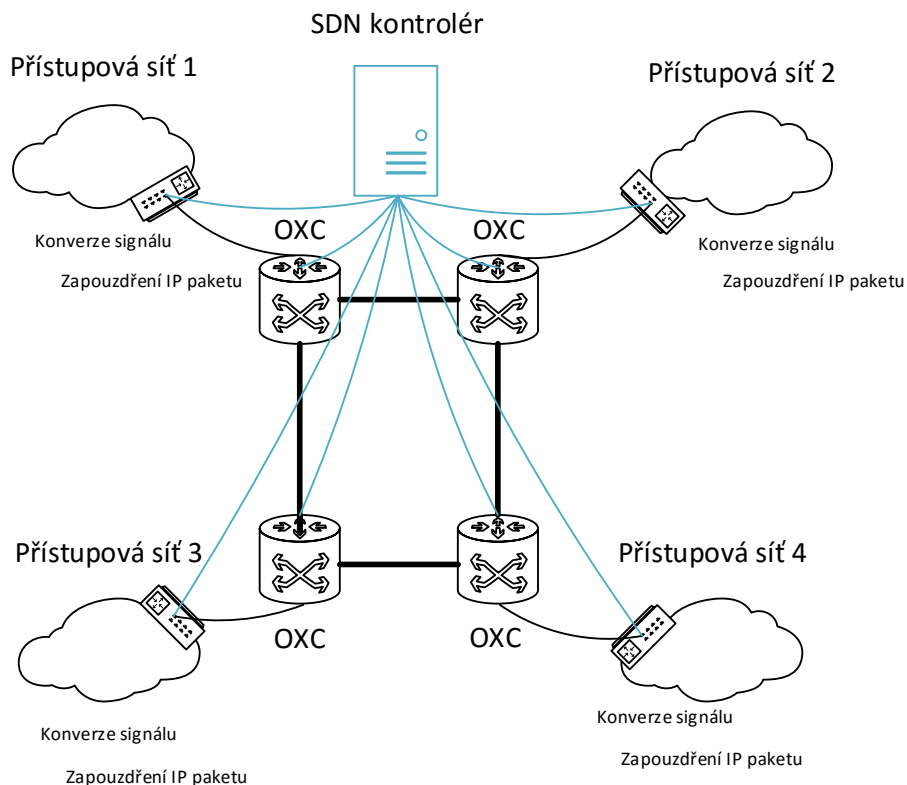
zání spojení  $6 \times 0,65 = 3,9$  ms.

Pro správné rozlišení OpenFlow agentů SDN kontrolérem má každý OpenFlow agent definovaný svůj identifikátor v poli `datapath_id`, nacházející se v asynchronní zprávě `OFPT_FEATURES_REPLY`. Ve zprávě `OFPT_ECHO_REPLY` se pak nacházejí rozdílné hodnoty zpoždění určené na základě zpoždovacích kanálů.

## 6 NÁVRH DWDM OPTICKÉ SÍTĚ NA BÁZI OXC

Na obr. 6.1 se nachází navržená topologie DWDM optické sítě. Topologie je zaměřená na páteřní síť, která je složena ze 4 OXC řízených jediným SDN kontrolérem. Na každý OXC je napojena přístupová síť. Přístupové sítě jsou v navržené topologii považovány za zdroje poskytující pakety pro přenos páteřní sítě do příslušné přístupové sítě. Z důvodu získání potřebných informací o příchozích paketech je každá přístupová síť spojena s SDN kontrolérem. Propojení OXC a SDN kontroléru je provedeno na různé vzdálenosti, které s sebou přináší rozdílná zpoždění během navázání spojení a následné komunikace. Výsledné zpoždění má vliv na dobu doručení dat páteřní sítě do příslušných přístupových sítí.

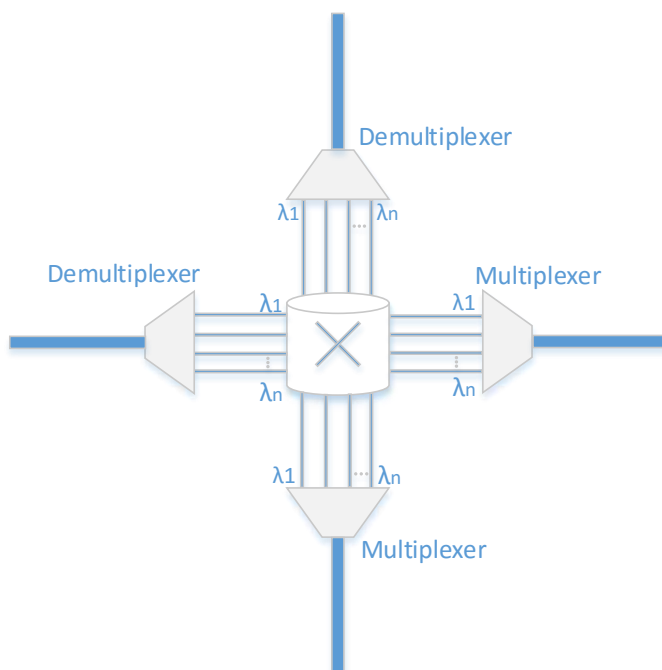
V rámci simulace navržené topologie bylo stanovené řešení pro abstrakci fyzické vrstvy. Optická vlákna páteřní sítě jsou nahrazena přenosovými kanály, kde jednotlivé vlnové délky jsou reprezentovány pomocí zapouzdřených paketů. OXC jsou reprezentovány pomocí OpenFlow agentů, které mají úlohu přepínačů pro jednotlivé pakety resp. vlnové délky.



Obr. 6.1: Návrh DWDM optické sítě v rámci OXC [18].

## 6.1 Princip směrování optického signálu

Na obr. 6.2 se nachází princip směrování optického signálu uvnitř OXC, který se nachází v navržené páteřní síti. Na vstupu OXC se nachází demultiplexer, který optický signál rozdělí na  $N$  vlnových délek. Vlnové délky jsou vedeny do přepínače, který, na základě logiky SDN kontroléru, může přepnout směr konkrétní vlnové délky a nasměrovat ji do příslušné přístupové sítě nebo do dalšího OXC v rámci páteřní sítě. Na výstupu OXC směrem k páteřní a přístupové síti se nachází multiplexer, který  $N$  vlnových délek sloučí do jediného optického vlákna.



Obr. 6.2: Princip přepínání signálu.

## 6.2 Zapouzdření IP paketu

Přicházející paket z přístupové sítě je před vstupem do páteřní sítě zapouzdřen. Obsah zapouzdřeného paketu je zobrazen na obr. 6.3 a skládá se ze 4 polí.

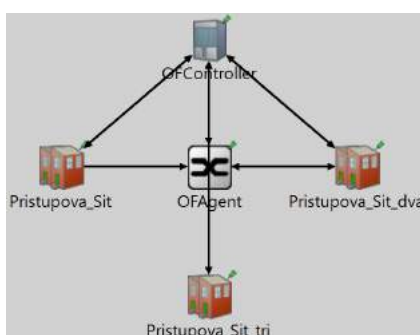
ID portu	$193,1 + n * 0,00625$	$12,5 * m$	IP paket
Veth X	Vlnová délka	Šířka superkanálu	Data

Obr. 6.3: Zapouzdření paketu.

První pole obsahuje virtuální ethernetové rozhraní. Jedná se o ID neobsazeného vstupního portu OXC resp. OpenFlow agenta, které určuje SDN kontrolér. Druhé pole obsahuje přiřazenou vlnovou délku SDN kontrolérem pro flexibilní rozestup kanálů. Třetí pole obsahuje informaci o šířce superkanálu. Ve čtvrtém poli se již nachází zapouzdřená data paketu pocházejícího z přístupové sítě.

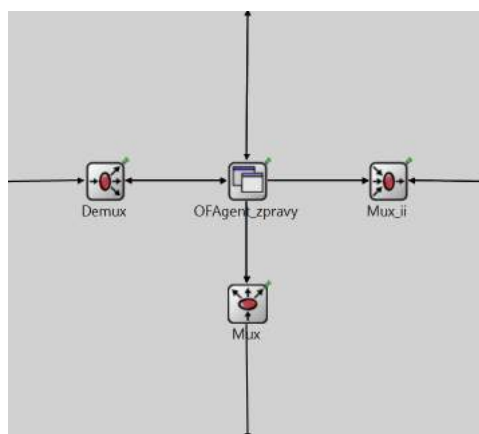
## 7 SIMULACE DATOVÉHO PŘENOSU DWDM NA BÁZI OXC

Na obr. 7.1 se nachází sestavená topologie pro simulaci datového přenosu v rámci OXC. Topologie se skládá ze třech jednoduchých modulů a ze dvou složených modulů. Jednoduché moduly představují tři přístupové sítě, naopak složené moduly představují OpenFlow agenta a SDN kontrolér. Jednotlivé přístupové sítě jsou přes kanály propojeny s OpenFlow agentem, který tvoří abstrakci fyzického OXC. SDN kontrolérem je poté spojen s jednotlivými přístupovými sítěmi pro získání potřebných informací o příchozím datovém toku.



Obr. 7.1: Topologie datového přenosu v rámci OXC v OMNeT++.

Na obr. 7.2 je zobrazena vnitřní struktura složeného modulu OpenFlow agenta s abstrakcí OXC.



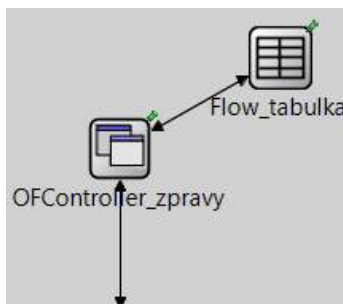
Obr. 7.2: Vnitřní struktura OpenFlow agenta v rámci abstrakce OXC v OMNeT++.

Z důvodu abstrakce OXC byly OpenFlow agentovi implementovány celkem čtyři jednoduché moduly. Na západní straně se nachází modul demultiplexeru, který obsahuje jednu vstupní bránu představující jeden vstup a čtyři výstupní brány, představující čtyři výstupy. Na východní a jižní straně se nacházejí moduly multiplexerů,



kteře obsahují čtyři vstupní a jednu výstupní bránu. Hlavní modul, který zároveň tvoří jádro OpenFlow agenta, leží uprostřed složeného modulu a nazývá se OFAgent\_zpravy. OFAgent\_zpravy je naprogramován tak, aby plnil funkci přepínání OXC a zároveň komunikoval s SDN kontrolérem a se sousedními multiplexery a demultiplexery.

Struktura složeného modulu SDN kontroléru je zobrazena na obr. 7.3 a obsahuje dva jednoduché moduly. Jádrem SDN kontroléru je modul OFController\_zpravy, který je zodpovědný za řízení OpenFlow agenta. Druhým modulem je tabulka toků (Flow\_tabulka), která je implementována nikoliv jako součást OpenFlow agenta, ale jako součást SDN kontroléru. Důvodem tohoto řešení je princip fungování OXC v datové rovině. Je nutné brát v úvahu, že OXC se zde chová pouze jako přepínací prvek, který naslouchá SDN kontroléru, jak má s příchozím datovým tokem naložit. Veškeré instrukce, jak má OXC sestavit optickou cestu a na který neobsazený port datový tok zaslat, jsou řízeny SDN kontrolérem. Tabulka toků v SDN kontroléru je vytvořena na základě pole struktur o maximální kapacitě 50 polí a slouží pro uložení a porovnání zdrojové a cílové IP adresy zapouzdřených paketů z přístupové sítě.



Obr. 7.3: Vnitřní struktura SDN kontroléru v OMNeT++.

## 7.1 Průběh simulace datového přenosu

Simulace datového přenosu je nastavena na maximálně tři příchozí zapouzdřené pakety, které postupně vysílá přístupová síť nacházející se na západní straně navržené topologie. Vliv zpoždění během zpracování datového toku OpenFlow agentem není brán v úvahu, neboť majoritní doba zpoždění vzniká během navázání spojení OpenFlow agenta s SDN kontrolérem.

Po úspěšném navázání spojení SDN kontroléru s OpenFlow agentem dojde k vyslání prvního zapouzdřeného paketu přístupovou sítí na SDN kontrolér. Struktura paketu obsahuje cílovou a zdrojovou IP adresu, která slouží pro identifikaci paketu

SDN kontrolérem a zároveň informaci, do které přístupové sítě má být paket nasměrován. Obsah zapouzdřeného paketu s připraveným rozšířeným záhlavím pro přenos optickou sítí lze vidět na obr. 7.4.

```

● Packet (Packet)
├── controllInfo = NULL (cObject)
├── encapsulatedPacket = NULL (cPacket)
├── portID = 0 [...] (int)
├── vlnova_delka = 0.000000 [...] (double)
├── superkanal = 0.000000 [...] (double)
└── paket (MAC)

```

Obr. 7.4: Obsah zapouzdřeného paketu v OMNeT++.

Po obdržení paketu SDN kontrolérem dojde k porovnání IP adres paketu v tabulce toků. Spuštění funkce porovnání je vyřešeno pomocí signalizační zprávy, kterou vysílá modul OFController\_zpravy na tabulku toků. Porovnání probíhá na základě cyklu, který porovnává jednotlivá pole s IP adresami obdrženého paketu. Na základě výsledku porovnání dojde k uložení cílové a zdrojové IP adresy do tabulky toků. Pokud se již IP adresy v tabulce nacházejí, SDN kontrolér pouze převezme již uložené hodnoty a k uložení záznamu nedojde.

V dalším kroku vyšle SDN kontrolér zprávu *OFPT\_FLOW\_MOD* zobrazenou na obr. 7.5.

```

● OFPT_FLOW_MOD (Flow_Mod)
├── controllInfo = NULL (cObject)
├── encapsulatedPacket = NULL (cPacket)
├── header (ofp_header)
├── version = 6 [...] (uint8_t)
├── type = 14 [...] (uint8_t)
├── length = 1 [...] (uint16_t)
├── xid = 4 [...] (uint32_t)
├── buffer_id = 0 [...] (uint32_t)
├── portID = 4 [...] (int)
├── vlnova_delka = 193.112500 [...] (double)
├── superkanal = 50.000000 [...] (double)
├── SA = '73175213822' [...] (string)
└── DA = '226905419450024' [...] (string)

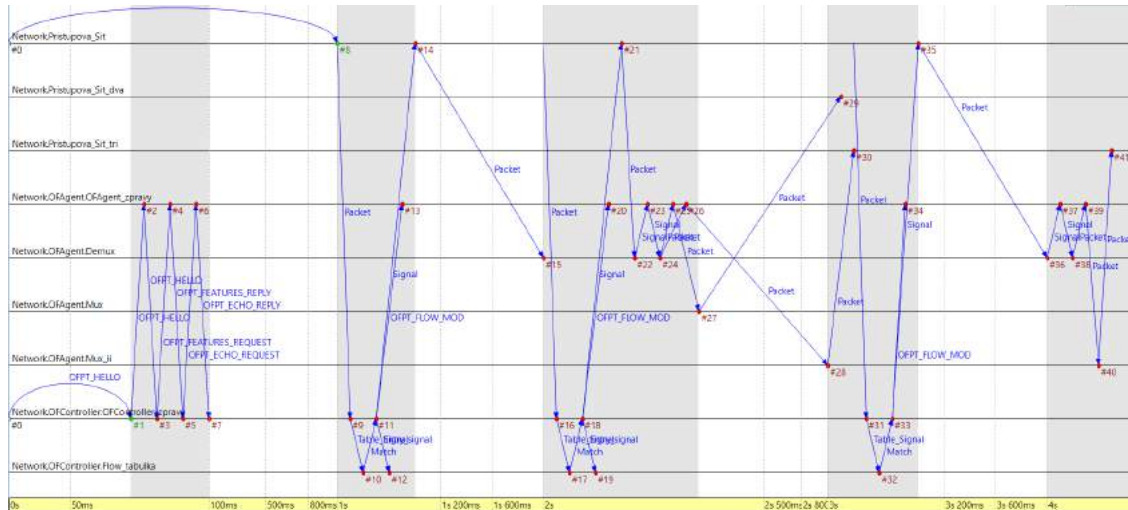
```

Obr. 7.5: Obsah zprávy *OFPT\_FLOW\_MOD* implementované v OMNeT++.

Zpráva *OFPT\_FLOW\_MOD* slouží k sestavení optické cesty nastavením vstupních portů OpenFlow agenta. Pole PortID určuje, přes který vstupní port modulu

OFAgent\_zpravy bude spojení sestaveno. Obsahem zprávy jsou dále pole, která obsahují parametry fyzické vrstvy a zdrojová a cílová IP adresa v desítkové soustavě. Po přijetí zprávy *OFPT\_FLOW\_MOD* dále dojde k vyslání zbylého datového toku přístupovou sítí směrem na demultiplexer OpenFlow agenta.

Výstup kompletní simulace je zobrazen na obr. 7.6.



Obr. 7.6: Kompletní výstup simulace datového přenosu v rámci OXC.

## 7.2 Řízení šířky pásma OXC

Řízení šířky pásma OXC probíhá na principu obsazenosti vstupních portů OpenFlow agenta. Postupná obsazenost všech vstupních portů je stanovena pro maximální simulační čas  $T = 4\text{s}$ . Na počátku simulace jsou všechny porty neobsazené a SDN kontrolér sestaví optickou cestu přes první port modulu OpenFlow agenta. Zároveň si do paměti uloží stav obsazenosti tohoto portu (číslo 0 znamená obsazený, číslo 1 volný). Ve vnitřní struktuře modulu OpenFlow agenta dorazí datový tok na demultiplexer. Po přijetí datového toku demultiplexerem dojde k vyslání druhého paketu z přístupové sítě. SDN kontrolér sestaví optickou cestu přes druhý vstupní port OpenFlow agenta, neboť první port je stále obsazený. V dalším kroku dojde modulem OFAgent\_zpravy k vyslání signalizační zprávy na demultiplexer. Ve zprávě je obsažena informace, na který vstupní port (vstupní bránu) má demultiplexer datový tok poslat.

Na základě obsazenosti výstupních portů dojde k přesměrování obou datových toků na neobsazené výstupní porty, které následně dorazí na multiplexer. V okamžiku, kdy oba datové toky kompletně opustí složený modul OpenFlow agenta, dojde k vyslání signálu na SDN kontrolér, který ve své paměti změní stav portů

zpět na neobsazený. Následně datový tok dorazí do své přístupové sítě a přenos je označen za úspěšný.

Tento proces se dále opakuje pro další datový tok, který SDN kontrolér nasměruje na třetí vstupní port OpenFlow agenta. Čtvrtý port je po celý simulační čas neobsazený.

V grafech na obr. 7.7, 7.8, 7.9 a 7.10 se nachází průběh obsazenosti všech čtyř vstupních portů vyexportovaných z prostředí OMNeT++.



Obr. 7.7: Obsazenost prvního portu.



Obr. 7.8: Obsazenost druhého portu.



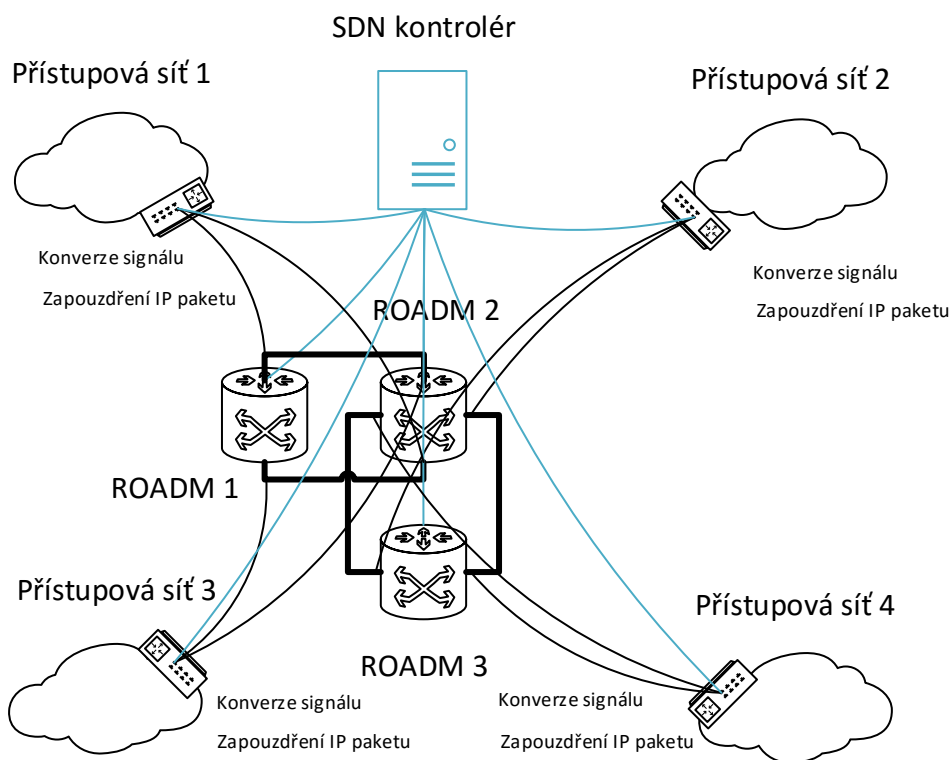
Obr. 7.9: Obsazenost třetího portu.



Obr. 7.10: Obsazenost čtvrtého portu.

## 8 NÁVRH DWDM OPTICKÉ SÍTĚ NA BÁZI ROADM

Na obr. 8.1 se nachází návrh topologie DWDM optické sítě s jedním čtyřstupňovým a dvěma dvoustupňovými ROADM. Ke každé add/drop struktuře příslušného ROADM je napojena přístupová síť. Přístupové sítě jsou v rámci topologie považovány za zdroje a příjemce zapouzdřených paketů. Uvnitř páteří sítě jsou ROADM napojené na SDN kontrolér, který umožňuje v rámci každého ROADM vzdáleně řídit činnost přidání a vydělení zapouzdřených paketů získaných z přístupových sítí. ROADM jsou obdobně, jako u DWDM optické sítě s OXC, reprezentovány pomocí OpenFlow agentů.



Obr. 8.1: Návrh DWDM optické sítě v rámci ROADM.

### 8.1 Princip směrování optického signálu

Předpokládanou funkci navržené DWDM optické sítě na bázi ROADM lze postavit na konkrétním příkladu. Nechť nastane situace, kdy přístupová síť 1 potřebuje navázat komunikaci s přístupovou sítí 4. Přístupová síť 1 vyšle svůj paket na rozhraní

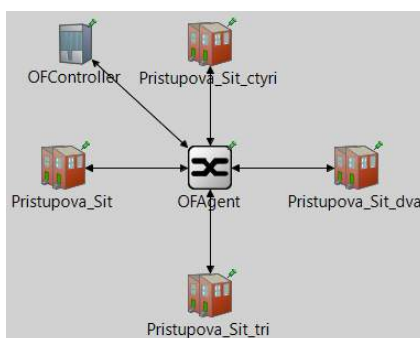
s páteřní sítí, kde dojde k zapouzdření paketu s parametry fyzické vrstvy a k převodu elektrického signálu na optický. SDN kontrolér, na základě znalosti kompletní sítě topologie, vypočítá a sestaví neoptimálnější optickou trasu a zároveň přiřadí optickému signálu odpovídající vlnovou délku.

Uvnitř páteřní sítě je optický signál přiveden na WSS ROADM 1, který provede směrování optického signálu na základě vytvořené optické trasy SDN kontrolérem. Tímto způsobem je optický signál přiveden na optický splitter uzlu ROADM 2. Optický signál je následně pomocí optického splitteru rozbočen do příslušné add/drop struktury, kde dojde k jeho vydělení na přístupovou síť 4. V případě výpadku spojení s přístupovou sítí 4 dojde ke zvolení alternativní optické trasy. Optický splitter ROADM 2 provede rozbočení optického signálu do směru ROADM 3, ve kterém dojde k alternativnímu vydělení optického signálu do přístupové sítě 4. Na rozhraní přístupové sítě 4 dojde ke konverzi optického signálu zpět na elektrický a k odpouzdření paketu.

Stejný princip směrování je využíván pro všechny ostatní přístupové sítě v návrhu.

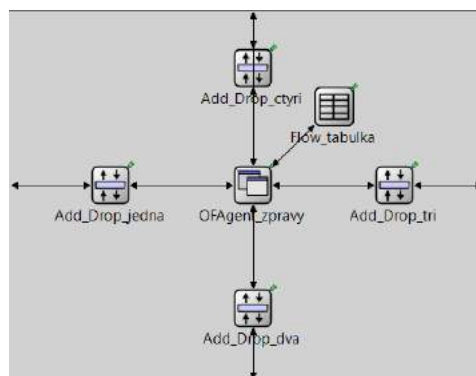
## 9 SIMULACE DATOVÉHO PŘENOSU DWDM NA BÁZI ROADM

Na obr. 9.1 se nachází sestavená topologie pro datový přenos v rámci čtyřstupňového ROADM. Složený modul OpenFlow agenta zde tvoří abstrakci fyzického ROADM, kde na každý stupeň je napojena přístupová síť. Přístupové sítě jsou, jako u simulace s OXC, vytvořeny pomocí jednoduchých modulů. Modul SDN kontroléru je v tomto řešení spojen pouze s OpenFlow agentem, neboť informace o datových tocích z přístupových sítích se ukládají přímo do modulu OpenFlow agenta.



Obr. 9.1: Topologie datového přenosu v rámci ROADM.

Na obr. 9.2 je zobrazena vnitřní struktura složeného modulu OpenFlow agenta s abstrakcí ROADM.



Obr. 9.2: Vnitřní struktura OpenFlow agenta v rámci abstrakce ROADM.

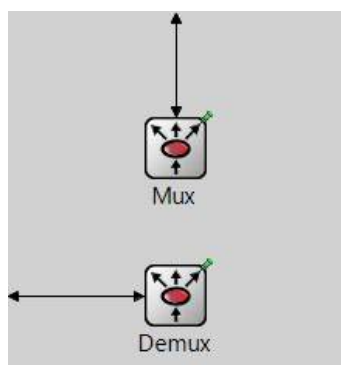
Z důvodu abstrakce čtyřstupňového ROADM byly OpenFlow agentovi implementovány celkem čtyři složené moduly add/drop struktur. Každá add/drop struktura je připojena k příslušnému stupni ROADM a umožňuje přijímat či vydělovat datové toky z okolních přístupových sítí. OpenFlow agentovi je dále implementován jednoduchý modul tabulky toků, který vykonává stejnou porovnávací funkci příchozích paketů, jako u SDN kontroléru v rámci OXC. Jádrem OpenFlow agenta je



jednoduchý modul `OFAgent_zpravy`, který zajišťuje komunikaci s SDN kontrolérem a s jednotlivými add/drop strukturami.

Na základě pracovní logiky ROADM se tabulka toků nachází pouze v režii OpenFlow agenta. Toto řešení umožňuje OpenFlow agentovi, v případě již existujícího záznamu o datovém toku, vydělovat procházející datové toky přes add/drop strukturu rovnou do okolních přístupových sítí.

Každý složený modul add/drop struktury je vytvořen z jednoduchého modulu multiplexeru a demultiplexeru, jak je znázorněno na obr. 9.3. Multiplexer umožňuje přidávat příchozí datový tok do jádra OpenFlow agenta. Demultiplexer naopak vyděluje datový tok do příslušné přístupové sítě.



Obr. 9.3: Vnitřní struktura add/drop modulu.

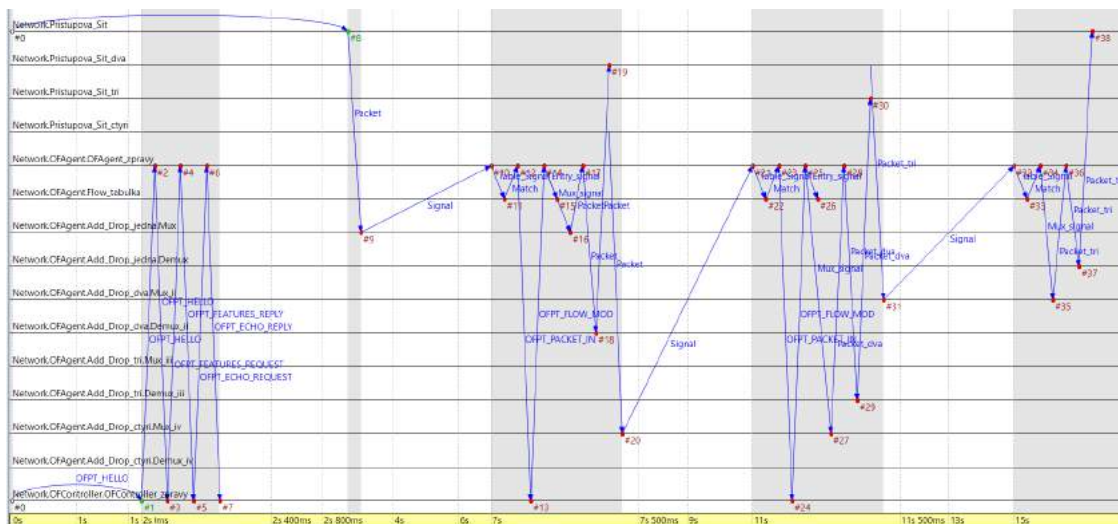
## 9.1 Průběh simulace datového přenosu

Simulace datového přenosu je založena na třech zapouzdřených paketech, které jsou vysílány postupně z okolních přístupových sítí v maximálním simulačním čase  $T = 15$  s. Po úspěšném navázání spojení je západní přístupovou sítí vyslán zapouzdřený paket na add/drop strukturu OpenFlow agenta. Signalizační zpráva add/drop struktury umožňuje spustit funkci porovnání IP adres v tabulce toků OpenFlow agenta. Pokud v tabulce toků není nalezen žádný záznam o příchozím paketu, dojde k vyslání zprávy `OFPT_PACKET_IN` na SDN kontrolér. Pomocí konfigurační zprávy `OFPT_FLOW_MOD` dojde k nastavení optických parametrů SDN kontrolérem. Po přijetí zprávy `OFPT_FLOW_MOD` dojde k přidání záznamu do tabulky toků a následně k vyslání signalizační zprávy OpenFlow agentem na add/drop strukturu příchozího paketu. V dalším kroku je paket nasměrován na demultiplexer add/drop struktury příslušného stupně, kde dojde k vyslání paketu do cílové přístupové sítě.

Tento způsob směřování se opakuje pro zbylé dva pakety, které jsou postupně vysílány ze severní a jižní přístupové sítě. V případě třetího paketu, jehož IP adresa

je totožná s prvním paketem, k vyslání zprávy *OFPT\_PACKET\_IN* již nedochází. Paket je, dle záznamu v tabulce toků, nasměrován do cílové přístupové sítě.

Výstup kompletní simulace je zobrazen na obr. 9.4.



Obr. 9.4: Kompletní výstup simulace datového přenosu v rámci ROADM.

## 10 ZÁVĚR

Princip SDN spočívá v úplném oddělení datové a řídicí roviny. Oddělená řídicí rovina poté nabízí přímou programovatelnost a vytváří abstrakci pro aplikace a síťové služby. V práci byl popsán třívrstvý model SDN a standard OpenFlow umožňující komunikaci mezi datovou a řídicí rovinou na základě OpenFlow zpráv.

Koncept SDN zahrnuje řešení i pro optické síťové spoje. Byl popsán rozšířený OpenFlow protokol obsahující nové zprávy SWITCH\_FEATURE a CFLOW\_MOD. Dále byl vysvětlen princip sestavení a odpojení optické cesty SDN kontrolérem pomocí sekvenčního a zpožděného řešení. Pro optimální rozložení zátěže SDN kontroléru byla představena výpočetní jednotka PCE, která umožňuje vypočítat nejkratší optickou přenosovou cestu s přiřazením vlnové délky. Dále byla popsána kombinace OpenFlow s technologií GMPLS pomocí paralelního, překrytého a integrovaného způsobu.

V závěru teoretické části práce byl vysvětlen princip sdílení kanálu v páteřních optických sítích s popisem multiplexů OTDM a WDM. OTDM umožňuje sloučit několik signálů do jediného toku na základě přepínání mezi kanály. V současné době nejvíce využívaný multiplex WDM umožňuje nezávislým uživatelům přenášet svá data přes jediné optické vlákno využitím různých vlnových délek. Dále se v práci nachází popis DWDM technologie pro fixní a flexibilní rozestupy kanálů. Na závěr byl popsán princip optických prvků OXC, ROADM a OLT.

Pomocí simulačního softwaru OMNeT++ bylo sestaveno spojení mezi SDN kontrolérem a OpenFlow agenty a odsimulován vznik zpoždění při navázání komunikace na základě reálných přenosových parametrů. Byla prokázána výměna inicializačních zpráv mezi SDN kontrolérem a jednotlivými OpenFlow agenty. Následně bylo prokázáno úspěšné navázání spojení všech OpenFlow agentů v rámci navržených softwarově definovaných optických sítí. Při realizaci bylo zahrnuto zpoždění, které vzniká majoritně vlivem rozdílné vzdálenosti mezi SDN kontrolérem a OpenFlow agenty.

V závěru praktické části byl odsimulován samotný datový přenos v rámci OXC a ROADM. V prostředí OMNeT++ byly sestaveny topologie s OXC a s ROADM, které se liší na základě pracovní logiky obou optických prvků. U topologie s OXC je SDN kontrolér propojen s okolními přístupovými sítěmi, neboť pro sestavení optické cesty přes OXC je nutné poskytnout SDN kontroléru informace o příchozích zapouzdřených paketech. Tabulka toků se nachází uvnitř SDN kontroléru a obsahuje záznamy o zapouzdřených paketech společně s parametry fyzické vrstvy. Řízení šířky pásma OXC bylo realizováno na základě obsazenosti vstupních portů OpenFlow agenta řízených SDN kontrolérem.

Topologie s čtyřstupňovým ROADM je tvořena čtyřmi přístupovými sítěmi.

Každá síť je napojena na jeden stupeň ROADM, kde každý stupeň obsahuje add/drop strukturu. Add/drop struktura slouží k přidávání a vydělování zapouzdřených paketů vyslaných okolními přístupovými sítěmi. Tabulka toků se nachází uvnitř OpenFlow agenta, kde dochází k porovnávání příchozích zapouzdřených paketů.

## LITERATURA

- [1] ABBOU, Fouad Mohammed a Hiew Chee CHOONG. *Optical transmission and networks for next generation internet traffic highways*. Hershey: IGI Global, 2015. ISBN 978-1-4666-6575-0.
- [2] ABEDIFAR, Vahid, Sadegh Abbasi SHAHKOOR, Ali EMAMI, Ali POURES-LAMI a S. A. AYOUGHI. Design and simulation of a ROADM-based DWDM network. *2013 21st Iranian Conference on Electrical Engineering (ICEE)*. Mashhad: IEEE, 2013, s. 1-4. DOI: 10.1109/IranianCEE.2013.6599722.
- [3] AMOUZAD, Ghafour a Ahmad FAUZI. Advanced Modulation Formats and Multiplexing Techniques for Optical Telecommunication Systems. *Trends in Telecommunications Technologies*. InTech, 2010-03-01. DOI: 10.5772/8494.
- [4] AZIZI, Mounir, Redouane BENAINI a Mouad Ben MAMOUN. Delay Measurement in Openflow-Enabled MPLS-TP Network. In: *Modern Applied Science*. Rabat, 2014, s. 90-101. ISSN 1913-1852.
- [5] BAHNASY, M., K. IDOUDI a H. ELBIAZE. Software-defined DWDM optical networks: OpenFlow and GMPLS experimental study. In: *2014 IEEE Global Communications Conference*. Austin: IEEE, 2014, s. 2173-2179. ISBN 978-1-4799-3512-3.
- [6] CHANNEGOWDA, Mayur, Reza NEJABATI a Dimitra SIMEONIDOU. Software-Defined Optical Networks Technology and Infrastructure: Enabling Software-Defined Optical Network Operations [Invited]. In: *Journal of Optical Communications and Networking*. Bristol: Ori Gerstel and Patrick Iannone, 2013, s. 274-282. ISSN 1943-0620.
- [7] CHIKAMA, Terumi, Hiroshi ONAKA a Satoshi KUROYANAGI. Photonic Networking Using Optical Add Drop Multiplexers and Optical Cross-Connects. *FUJITSU Sci. Tech. J.* 1999, s. 46-55. ISSN 0016-2523.
- [8] CISCO SYSTEMS. *Cisco ONS 15454 DWDM Engineering and Planning Guide: Software and Product Release 7.x*. San Jose, 2007.
- [9] FILKA, Miloslav. *Optické sítě - přednášky*. Brno, 2007. Skripta. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav Telekomunikací.
- [10] FILKA, Miloslav. *Přenosová média*. Brno, 2012. ISBN 978-80-214-4444-7. Skripta. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací.

- [11] GRINGERI, Steven, Bert BASCH, Vishnu SHUKLA, Roman EGOROV a Tiejun J XIA. Flexible architectures for optical transport nodes and networks. *IEEE Communications Magazine*. 2010, vol. 48, issue 7, s. 40-50. DOI: 10.1109/MCOM.2010.5496877.
- [12] HU, Fei. *Network innovation through OpenFlow and SDN: principles and design*. London: CRC Press, 2014. ISBN 978-1-4665-7210-2.
- [13] ITU-T G.694.2. *Spectral grids for WDM applications: CWDM wavelength grid*. Geneva, 2003.
- [14] ITU-T G.694.1. *Spectral grids for WDM applications: DWDM frequency grid*. Geneva, 2012.
- [15] IQBAL, M. Kaleem, M. Usman IQBAL, M. Bilal IQBAL a M. Hassan IQBAL. Optical fiber switches. In: *2012 International Conference on Open Source Systems and Technologies*. Lahore: IEEE, 2012, s. 40-45. ISBN 978-1-4673-3097-8.
- [16] KARTALOPOULOS, Stamatios V. *Introduction to DWDM technology: Data in a rainbow*. New York: Sons, 2000. ISBN 0-7803-5399-4.
- [17] KLEIN, Dominik a Michael JARSCHER. An OpenFlow extension for the OMNeT++ INET framework. In: *Proceedings of the 6th International ICST Conference on Simulation Tools and Techniques*. Belgium: Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, 2013, s. 322-329. ISBN 978-1-4503-2464-9.
- [18] KLODA, T. Connection Establishment between SDON Controller and OF Agents within OXC. In: *Proceedings of the 22nd Conference STUDENT EE-ICT 2016*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2016, s. 112-114. ISBN 978-80-214-5350-0.
- [19] MRV COMMUNICATIONS. *Optical Software Defined Networking*. 2014.
- [20] OPEN NETWORKING FOUNDATION. *OpenFlow-enabled Transport SDN*. 2014.
- [21] OPEN NETWORKING FOUNDATION. *OpenFlow Switch Specification: Version 1.5.0 (Protocol version 0x06)*. 2015.
- [22] OPEN NETWORKING FOUNDATION. *Software-Defined Networking: The New Norm for Networks*. 2012.

- [23] PHEMIUS, Kevin a Mathieu BOUET. Monitoring latency with OpenFlow. In: *Proceedings of the 9th International Conference on Network and Service Management (CNSM 2013)*. Zurich: IEEE, 2013, s. 122-125. ISBN 978-3-901882-53-1.
- [24] RAMASWAMI, Rajiv, Kumar N SIVARAJAN a Galen H SASAKI. *Optical networks: a practical perspective*. 3rd ed. Boston: Elsevier/Morgan Kaufmann, c2010. ISBN 978-0-12374092-2.
- [25] SCHLITTER, P. *Optické přístupové sítě*. Praha, 2004. ISSN 1214-9675. České vysoké učení technické v Praze.
- [26] TUCKER, R.S., G. EISENSTEIN a S.K. KOROTKY. Optical time-division multiplexing for very high bit-rate transmission. In: *Journal of Lightwave Technology*. IEEE, 1988, s. 1737-1749. ISSN 0733-8724.
- [27] VAN ADRICHEM, Niels L. M., Christian DOERR a Fernando A. KUIPERS. OpenNetMon: Network monitoring in OpenFlow Software-Defined Networks. In: *2014 IEEE Network Operations and Management Symposium (NOMS)*. Krakow: IEEE, 2014, s. 1-8. ISBN 978-1-4799-0913-1.

## SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

API	Application Programming Interface
ARP	Address Resolution Protocol
CWDM	Coarse Wavelength Division Multiplexing
DWDM	Dense Wavelength Division Multiplexing
EMS	Element Management System
GMPLS	Generalized Multi-Protocol Label Switching
IP	Internet Protocol
LSP	Label Switched Path
NMS	Network Management System
OLT	Optical Line Terminal
OTDM	Optical Time Division Multiplexing
OXC	Optical Crossconnect
OXM	OpenFlow Extensible Match
PCC	Path Computation Client
PCE	Path Computation Element
PON	Passive Optical Network
PXC	Photonic Crossconnect
ROADM	Reconfigurable Optical Add-Drop Multiplexer
RSVP	Resource Reservation Protocol
SDM	Space Division Multiplexing
SDN	Software Defined Network
STP	Spanning Tree Protocol
TCP	Transmission Control Protocol
TE	Traffic Engineering



TED	Traffic Engineering Database
TLV	Type Length Value
WDM	Wavelength Division Multiplexing
WGR	Waveguide Grating Router
WSS	Wavelength Selective Switch
WWDM	Wide Wavelength Division Multiplexing
XML	Extensible Markup Language

# SEZNAM PŘÍLOH

<b>A Příloha A</b>	<b>67</b>
A.1 Ukázky zdrojových kódů . . . . .	67
A.1.1 Porovnávání přijatých záznamů v tabulce toků . . . . .	67
A.1.2 Navázání spojení SDN kontroléru . . . . .	68
A.1.3 Navázání spojení OpenFlow agenta . . . . .	69
<b>B Příloha B</b>	<b>70</b>
B.1 Obsah přiloženého CD . . . . .	70

# A PŘÍLOHA A

## A.1 Ukázky zdrojových kódů

### A.1.1 Porovnávání přijatých záznamů v tabulce toků

---

```
1 void Flow_tabulka::matchEntry()
2 {
3
4     for(int i = 0; i < 50; i++) // Porovnavaci cyklus
        tabulky toku v ramci jeji maximalni kapacity.
5     {
6         if(zaznam[i].sAddress == iSA && zaznam[i].dAddress
            == iDA)
7         {
8             tabulka_zaznam = 1; // Pokud se jiz zaznam v
                tabulce nachazi, dojde k nastaveni porovnavaci
                promenne na 1 a cyklus se ukonci.
9             break;
10        }
11        else
12        {
13            tabulka_zaznam = 0; // Pokud není zaznam
                nalezen, dojde k nastaveni porovnavaci
                promenne na 0.
14        }
15    }
16
17
18    Match *msg = new Match("Match"); // Vytvoreni nove
        signalizacni zpravy.
19    msg->setZaznam(tabulka_zaznam); // Vlozeni porovnavaci
        promenne do zpravy.
20    msg->setPortID(zaznam[n].veth);
21    msg->setWavelength(zaznam[n].wavelength);
22    msg->setSuperkanal(zaznam[n].superchannel); // Vlozeni
        informaci o fyzicke vrstve.
23
24    send(msg, "table_out"); // Odeslani signalizacni zpravy.
25
26    if(tabulka_zaznam == 1)
27    {
28        n++; // V pripade jiz existujiciho zaznamu dojde k
            inkrementaci pomocne promenne pro ziskani
            spravnych informaci budouciho zaznamu.
29    }
30
31 }
```

---

## A.1.2 Navázání spojení SDN kontroléru

---

```
1 void OFController_zpravy::handleMessage(cMessage *msg)
2 {
3
4     if(msg->isSelfMessage())
5     {
6         send(msg, "out"); // Vyslani zpravy OFPT_HELLO na
7                             // OpenFlow agenta pomoci startovací zpravy.
8     }
9
10    if(counter > 0 && counter < 3) // Pocitadlo prijatych
11        // zprav.
12    {
13        Hello *of_msg = (Hello *)msg; // Vyvolani prijate zpravy
14        // OFPT_HELLO, ktera byla vyslana OpenFlow agentem.
15        header[0] = of_msg->getVersion(); // Ulozeni verze
16        // OpenFlow protokolu do promenne header.
17        header[1] = of_msg->getType(); // Ulozeni typu OpenFlow
18        // zpravy (OFPT_HELLO) do promenne header.
19        header[2] = of_msg->getXid(); // Ulozeni identifikatoru
20        // parovani do promenne header.
21        ofp_type type = (ofp_type)of_msg->getHeader().type;
22        //Vytvoreni promenne type na zaklade typu prijate
23        // zpravy.
24
25        switch(type)
26        {
27            case OFPT_HELLO:
28            {
29                handleRequestMessages(); // V pripade prijati
30                // zpravy OFPT_HELLO, dojde k zavolani funkce pro
31                // odeslani zpravy OFPT_FEATURES_REQUEST.
32                break;
33            }
34
35            case OFPT_FEATURES_REPLY:
36            {
37                Request *parametry = (Request *)msg; // V
38                // pripade prijati zpravy OFPT_FEATURES_REPLY,
39                // dojde k její vyvolani a k postupnému uložení
40                // jednotlivých parametru.
41                for(int i = 0; i < 4; i++)
42                {
43                    dostupne_porty[i] = parametry->getPort(i);
44                    uvolneni_portu[i] = dostupne_porty[i];
45                    obsazenost[i] = 1;
46                }
47
48                handleReplyMessages(); // Funkce pro odeslani
49                // zpravy OFPT_ECHO_REPLY na OpenFlow agenta.
50                break;
51            }
52        }
53    }
54 }
```

---

### A.1.3 Navázání spojení OpenFlow agenta

---

```
1 void OFAgent_zpravy::handleMessage(cMessage *msg)
2 {
3
4     if(msg->getKind() == 1)
5     {
6         handleHelloMessage(); // Odeslání zpravy OFPT_HELLO na
           SDN kontroler.
7     }
8
9     if(counter > 0 && counter < 3)
10    {
11        Hello *of_msg = (Hello *)msg; // Vyvolání přijaté
           zpravy OFPT_HELLO, která byla vyslána SDN
           kontrolerem.
12
13        ofp_type type = (ofp_type)of_msg->getHeader().type;
14        switch (type)
15        {
16            case OFPT_FEATURES_REQUEST:
17                handleReplyMessage(); // Odeslání zpravy
           OFPT_FEATURES_REPLY na SDN kontroler.
18                break;
19
20            case OFPT_ECHO_REQUEST:
21                handleEchoMessage(); // Odeslání zpravy
           OFPT_ECHO_REPLY na SDN kontroler.
22                break;
23
24        }
25    }
```

---

## B PŘÍLOHA B

### B.1 Obsah příloženého CD

Na příloženém CD jsou, kromě elektronické verze ve formátu pdf, umístěny složky vytvořených simulací pro optické prvky OXC a ROADM. Obě simulace jsou spustitelné v simulačním softwaru OMNeT++ verze 4.6. Pro spuštění simulací je nutné obě složky zkopírovat do hlavního adresáře softwaru OMNeT++. Jednotlivé simulace je poté možné spustit zavoláním příslušného exe souboru dané simulace a to přes příkazový řádek mingwenv.cmd, který je rovněž umístěn v hlavním adresáři. Tyto simulace, umístěné v archivu priloha.zip, jsou opatřeny heslem, které lze získat na emailové adrese xkloda00@stud.feec.vutbr.cz.