

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

## FAKULTA INFORMAČNÍCH TECHNOLOGIÍ



IP1 – Technická zpráva

## Analýza výkonnosti síťového procesoru NXP

## Abstrakt

S příchodem nové platformy jsou známy pouze parametry udávané výrobcem. Tyto parametry jsou však udávány zpravidla za optimálních podmínek, nebo jsou pouze částečné. Proto je vhodné provést alespoň základní výkonnostní měření takovýchto technologií.

U platformy NXP QorlQ LS2088A výrobce uvádí, že zvládne zpracovat všechny pakety na plně vytížené síťové lince. Cílem této práce je zjistit reálné parametry procesoru LS2088A na vývojové desce LS2088A-RDB a objevit funkční řešení, která budou poskytovat požadovaný výkon. Procesor LS2088A obsahuje akcelerační jednotku DPAA2, která je velmi komplexní a celkově tak dosahuje řádově tisíce možných konfigurací. Měřený procesor je první ze série předprodukčních vzorků a řada funkcí uvedených v dokumentaci je nepředvídatelných nebo dokonce nefunkčních.

## Abstract

Parameters of new released platforms are usually provided only from the manufacturer. However those parameters are usually measured in the optimal environment and also some important parameters can be omitted or partial. So it is appropriate to perform at least a basic measurement of the performance of new platforms.

For the NXP QorlQ LS2088A, the manufacturer states it can handle process all packets on a fully loaded network. The goal of this work is to determine the real parameters of the LS2088A processor on the LS2088A-RDB development board and find out a functional configuration, which will provide the required performance. The main feature of the LS2088A processor is DPAA2 subsystem for the acceleration of processing network packets. The DPAA2 is highly complex subsystem and has more than thousand configurations. The processor measured in this work is preproduction sample and many of advertised featured have unpredictable behaviour or they are even non-functional.

# Obsah

<b>1</b>	<b>Úvod</b>	<b>3</b>
<b>2</b>	<b>DPDK</b>	<b>4</b>
2.1	TestPMD . . . . .	4
2.2	L2FWD . . . . .	4
<b>3</b>	<b>Platforma</b>	<b>5</b>
3.1	Referenční návrhová deska LS2088A-RDB . . . . .	5
3.2	DPAA2 . . . . .	6
<b>4</b>	<b>Měření</b>	<b>7</b>
4.1	Fronty . . . . .	7
4.2	Spirent . . . . .	7
4.3	Měřené konfigurace . . . . .	7
4.4	Postup při měření . . . . .	9
4.5	Měření s TestPMD . . . . .	9
4.6	Měření pod Linuxem . . . . .	10
4.7	Problémy při měření . . . . .	10
<b>5</b>	<b>Výsledky měření</b>	<b>11</b>
5.1	Výsledky měření s DPDK . . . . .	11
5.2	Výsledky měření pod Linuxem . . . . .	14
<b>6</b>	<b>Závěr</b>	<b>18</b>

# 1 Úvod

Při vývoji aplikací spoléháme, že platforma, na které aplikace poběží je dostatečně výkonná tak, aby splnila určité limity. Jedna z možností, jak zjistit zda platforma dané limity opravdu splňuje, je provést výkonnostní měření.

Vzhledem k faktu, že Linux používá network stack, který kopíruje pakety z paměti pro operační systém do uživatelské paměti, což způsobuje zpomalení zpracování paketů, bude pro primární měření použit framework DPDK. Poté budou pro porovnání uvedena měření čistě pod Linuxem, dále s modifikací MMAP, jež obchází výše zmíněný network stack a poslední měření s knihovnou libpcap.

Nová platforma je postavena na ARM architektuře a disponuje velkou škálou akceleratorů se zaměřením především na zpracování síťového provozu. Vzhledem k faktu, že množství zařízení používaných jako NAT, Firewall, či jiné aplikační servery jsou založeny na procesorech Intel, bude pro zajímavost uvedeno také měření stejné konfigurace jako u NXP na takovémto zařízení s procesorem Intel.

## 2 DPDK

DPDK<sup>1</sup> je soubor knihoven a ovladačů pro rychlé zpracování paketů. Dnes podporuje nepoužívanější typy procesorů, ale pouze v linuxovém prostředí (podmnožina funkcí DPDK funguje také na FreeBSD)[1]. PMD<sup>2</sup> je ovladač, který ukládá pakety přímo do user-space (paměť vyhrazená pro uživatele) a nedochází tak ke kopírování mezi kernel-space (paměť vyhrazená pro OS) a user-space. Navíc PMD přistupuje k deskriptorům portů přímo bez jakýchkoli přerušení (s výjimkou přerušení změn stavu propojení), aby se pakety rychleji přijímaly, zpracovávaly a doručovaly.

Klíčové části, ze kterých se DPDK skládá, jsou EAL<sup>3</sup> která zajišťuje komunikaci s operačním systémem a přístup k nízkourovňovým zdrojům, jako je paměťový prostor, PMD ovladače síťových karet a mempool knihovna, která umožňuje naalokovat paměť při spuštění a tím se vyhnout dalším alokacím na hromadě (část paměti, kterou si program za běhu rezervuje), které by vedly k přepnutí kontextu procesoru.

### 2.1 TestPMD

TestPMD je jedna z referenčních aplikací distribuovaných s balíčkem DPDK. Jak název napovídá, je určena k testování PMD ovladače, ale lze ji použít i pro měření výkonnosti DPDK[2].

Aplikace TestPMD poskytuje následující možnosti použití:

1. Vstupně-výstupní mód: Tento režim je obecně označován jako režim přeposílání/forwarding. Jedná se o výchozí mód při spuštění aplikace TestPMD. V tomto režimu jádro přijímá pakety z jednoho síťového rozhraní a odešle je na jiné síťové rozhraní. Je možné použít i jeden port pro příjem paketů a zároveň tak pro jejich odeslání.
2. Pouze příjem: Tento režim je obecně označován jako režim Rx-only. V tomto režimu aplikace zpracuje pakety přijaté z portů a uvolní je bez jejich přenosu.
3. Pouze odesílání: Tento režim je obecně označován jako režim Tx-only. V tomto režimu vygeneruje aplikace pakety o nastavitelné velikosti a odesílá je přes zvolené porty. Aplikace se tedy chová jako generator paketů.

Další módy jež už se při měření nepoužily jsou: mac, macsw ap, flow gen, csum a icmpecho.

### 2.2 L2FWD

L2FWD, stejně jako TestPMD, je jedna z referenčních aplikací distribuovaných s balíčkem DPDK a provádí L2 forwarding (– přeposílání na jiné síťové rozhraní) pro každý paket přijatý na příchozí port. Cílový port je přilehlý port z povolené portmasky, to znamená, že pokud jsou povoleny první čtyři porty, tj. portmaska 0xF, porty 1 a 2 si vzájemně předávají pakety mezi sebou. Stejně tak porty 3 a 4.

Tato aplikace může být, stejně jako TestPMD, použita k porovnávání výkonu pomocí generátoru síťového provozu.

---

<sup>1</sup>Data plane development kit

<sup>2</sup>Poll Mode Driver

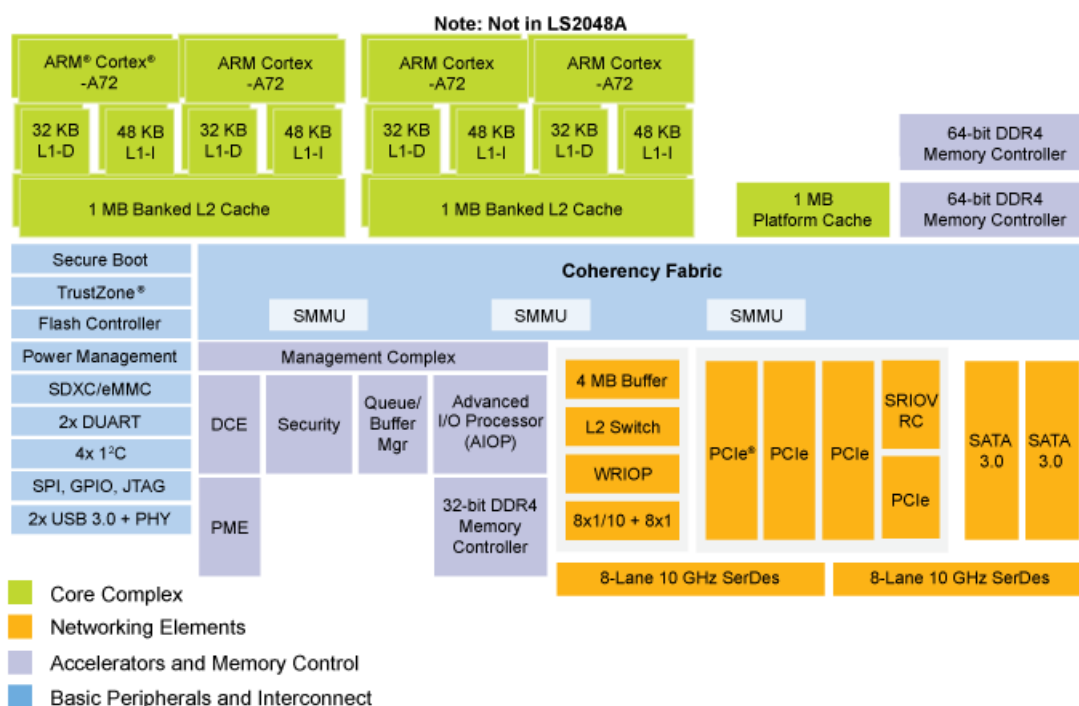
<sup>3</sup>Environment Abstraction Layer

### 3 Platforma

Hlavním výpočetním prvkem platformy je procesor NXP QorIQ LS2088A. Procesor disponuje osmi 64bitovými výpočetními jádry ARM Cortex-A72 s maximálním taktům 2 GHz. Každé výpočetní jádro má k dispozici 32KB datové L1 cache paměti a 48KB instrukční L1 cache paměti. Jádra jsou spárována do dvojic a každá dvojice má k dispozici 1MB sdílené L2 cache paměti. Dohromady platforma disponuje 4MB L2 cache paměti. Disponuje také akcelerační jednotkou DPAA2 a L2 Switchem, který na základě nastavených pravidel funguje jako demultiplexor a podle MAC adresy paketu jej přesměruje do vybrané fronty, kde každé jádro má svou frontu [3].

Další, již neměřené functionality tohoto procesoru v oblasti zpracování síťového toku

1. 10 Gb/s Pattern Matching regulérních výrazů
2. 20 Gb/s Datové komprese
3. 20 Gb/s SEC krypto akcelerace



Obrázek 1: Blokové schéma procesoru QorIQ LS2088A

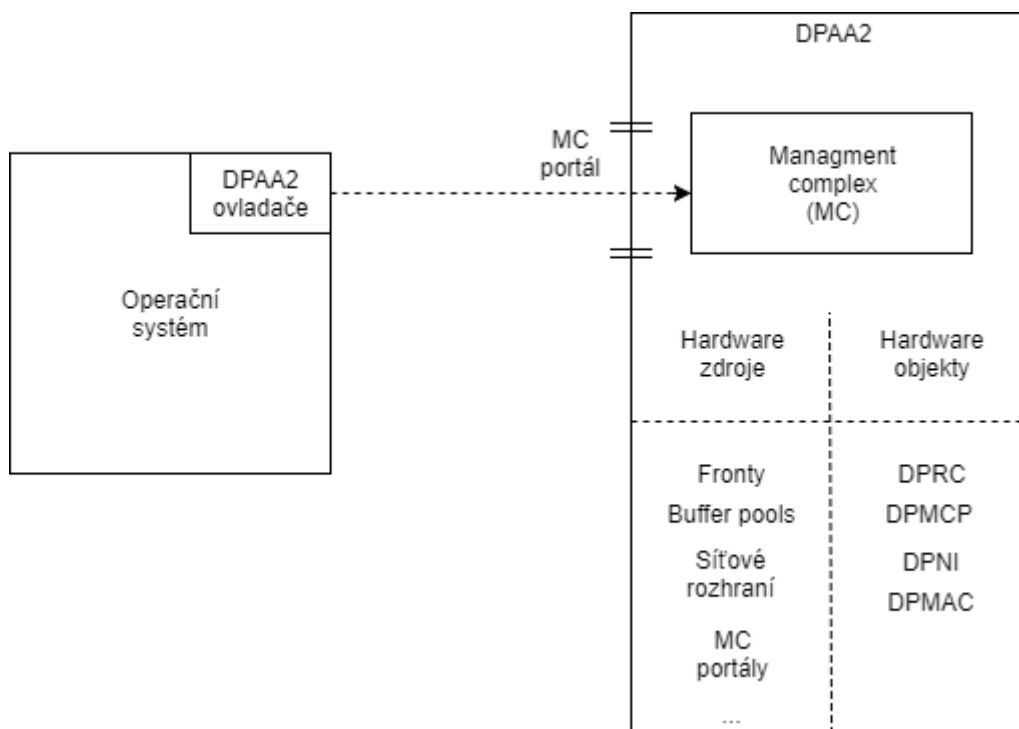
#### 3.1 Referenční návrhová deska LS2088A-RDB

Tato deska poskytuje komplexní platformu, která umožňuje využití procesoru LS2088A ve standardním linuxovém prostředí a síťovou inteligenci s novou generací Datapath (DPAA2). Pro zpracování síťového provozu deska poskytuje 4 SFP+ a 4 RJ45 porty s rychlostí až 10 Gb/s. Na desce jsou také vyvedeny čtyři porty 1Gb/s. Disponuje 128 MB NOR a 2GB 8-bit NAND flash paměti. Obsahuje 64 MB EEPROM a dvě 72bitové DDR4, 4 GB na slot. Pro periferie jsou připraveny dvě SATA rozhraní, dva USB 3.0 porty a sběrnici PCI Express třetí generace.

### 3.2 DPAA2

DPAA2<sup>5</sup> je hardwarová architektura určená pro vysokorychlostní síťové zpracování paketů. DPAA2 se skládá ze sofistikovaných mechanismů pro zpracování ethernetových paketů, správu fronty, správy vyrovnávacích pamětí, autonomního přepínání L2, virtuálního přemostění sítě ethernet a akceleratorů – například kryptografických či akceleratoru zpracování paketů o rychlosti 20Mp/s.

Hardwarová součást DPAA2 s názvem Management Complex (dále jen MC) spravuje hardwarové prostředky DPAA2. MC poskytuje objektovou abstrakci pro softwarové ovladače určené pro použití hardwaru DPAA2. MC používá hardwarové prostředky DPAA2, jako jsou fronty, vyrovnávací paměti a síťové porty pro vytvoření funkčních objektů / zařízení, jako jsou network interface (síťová rozhraní), L2 Switch nebo akcelerátory. MC také poskytuje paměťově mapované vstupně-výstupní příkazové rozhraní (MC portály), které softwarové ovladače DPAA2 používají pro práci s objekty DPAA2[4].



Obrázek 2: DPAA2 diagram

<sup>5</sup>Data Path Acceleration Architecture Gen2

## 4 Měření

Abychom byli schopni určit výkonnostní parametry platformy, je třeba provést měření. K získání co nejpřesnějších výsledků se bude měřit 5 konfigurací počínaje nejjednoduššími, pouze příjem či odesílání z jednoho portu, až po nejsložitější, kdy všechny porty NXP odesílají i přijímají maximum a L2 Switch. Měření výkonnostních parametrů probíhá od nejmenšího možného paketu velikosti 64 Bytů a postupně se velikost zvyšuje až po 1400 Bytů pro každou konfiguraci. Pokud nebude řečeno jinak, je zátěž linky implicitně nastavena na maximum, tedy 10 Gb/s.

### 4.1 Fronty

Fronty paketů jsou základní součástí libovolného síťového zařízení. Umožňují komunikaci asynchronních modulů a zvyšují výkon.

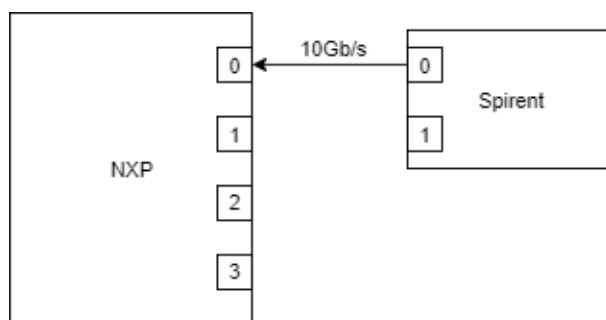
Fronta paketů je zpravidla implementována jako vyrovnávací paměť vstupu (first-in, first-out – FIFO) s pevnou velikostí. Fronta neobsahuje paketová data. Místo toho se skládá z deskriptorů, které odkazují na pakety v operační paměti. Pakety se do síťové karty následně dostanou pomocí DMA přenosů.

### 4.2 Spirent

Spirent je nástroj pro testování síťových zařízení. Základní vlastností Spirentu je generování paketových toků. Toky jsou definovány v tzv. streamblocku, v nichž je možné definovat hodnoty jednotlivých položek v hlavičkách podporovaných protokolů. V některých případech bylo ke generování paketů nebo zjišťování rychlosti linky použito zařízení Spirent SPT-2000A. V daném spirentu byl použit rozšiřující modul se dvěma SFP+ porty, které jsou schopny přenést až 10 Gb/s.

### 4.3 Měřené konfigurace

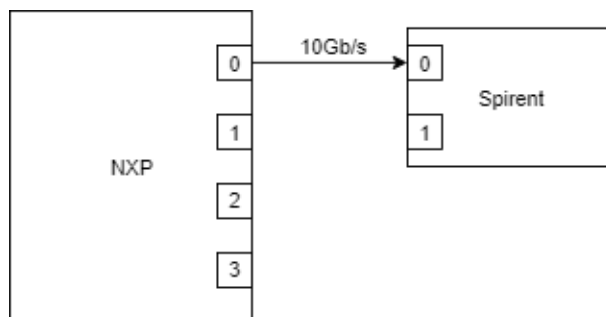
1. *Rx only* – Spirent generuje a posílá pakety na maximální rychlosti přes jednu 10 Gbps linku do NXP procesoru. Na NXP procesoru běží aplikace TestPMD, která pakety přijímá. Přijmuté pakety jsou po započítání zahozeny.



Obrázek 3: RX only schéma

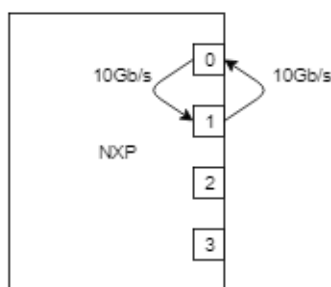


2. *Tx only* – Generování paketů v tomto případě zajišťuje aplikace TestPMD a pakety se odesílají přes jednu linku přivedenou na port Spirentu, z kterého se odečítá výsledná rychlost.



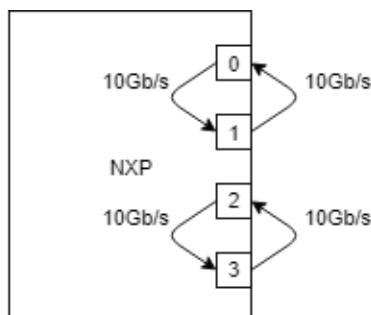
Obrázek 4: TX only schéma

3. *Malý loopback* – Generování paketů pomocí aplikace TestPMD a odesílání přes dva porty, kde pakety z portu 0 jsou přesměrovány na port 1 a opačně. Výsledná rychlost se vypočítá jako průměr rychlostí RX obou portů.



Obrázek 5: Malý loopback schéma

4. *Malý loopback – Intel* – Stejný princip jako u malého loopbacku s NXP, pouze se zařízením s procesorem Intel. Měření se provádělo pro porovnání.
5. *Malý loopback × 4* – Dva malé loopbacky mezi čtyřmi porty, první malý loopback je mezi porty 0 a 1. Druhý malý loopback je mezi porty 2 a 3. Výsledná rychlost je vypočtena jako průměr rychlostí RX portů.



Obrázek 6: Malý loopback × 4 schéma

6. *L2 Switch* – Na základě nastavených pravidel funguje jako demultiplexer a podle MAC adresy paketu jej přesměruje do vybrané fronty, kde každé jádro má svou frontu.

## 4.4 Postup při měření

1. Konfigurace Spirentu
  - (a) Nastavení parametrů (velikost paketů, burst, ...)
  - (b) Nastavení zátěže linky/linek
2. Konfigurace TestPMD
  - (a) Konfigurace DPAA2 sběrnice zahrnující alokaci a konfiguraci MAC, síťových rozhraní a jejich front
  - (b) Přiřazení jader k portům/frontám
3. Spuštění generování paketů pomocí Spirentu/TestPMD
4. Odečtení výsledků měření ze Spirentu/TestPMD
5. Zastavení generování, změna parametrů měření a vrácení se na bod č. 3)
6. Změna zapojení/přesměrování linek a vrácení se na bod č. 1)

## 4.5 Měření s TestPMD

Aplikaci TestPMD použijeme v některých případech jako generátor paketů a k zobrazení statistik datových toků na jednotlivých portech NXP.

Zde je postup příkazů měření malého loopbacku s TestPMD.

```
# Alokovat rozhrani/mac/dpaa2 kontejner
# dynamic_dpl.sh je originalni skript dodavany
# k procesoru od NXP
source /usr/odp/scripts/dynamic_dpl.sh \
dpmac.1 dpmac.2 dpmac.3 dpmac.4
# Vstup do interaktivniho rezimu
testpmd -- -i
# Nastavit velikost paketu v bytech a burst
set txpkts 64
set burst 256
# Nastavit presmerovani paketu mezi porty 0->1, 1->0
set portlist 0,1
# Vymaskovat jadra a porty, 2 porty 2 jadra
set portmask 3
set coremask 6
# Nastavit rezim forwardingu
set fwd io
# Spusteni generovani
start tx_first
# Zobrazit aktualni hodnoty portu
show port info (PORT_ID | all)
# Zastaveni generovani
stop
```

TestPMD zobrazuje datový tok na portech v FPS (Frames per second), proto pro výslednou rychlost v Gbps musíme přepočítat hodnoty pomocí vzorce[5]

$$Rychlost[b/s] = FPS * 8 * (VelikostPaketu[Byte] + 8 + 12)$$

kde 8 je počet bajtů preamble paketu a 12 je počet bajtů pro mezi paketovou mezeru.

Pro kontrolu hodnot naměřených pomocí TestPMD, byla použita aplikace L2FWD. Ve všech měřených případech jsou výsledky shodné jako u aplikace TestPMD.

## 4.6 Měření pod Linuxem

Pro měření bylo vytvořeno několik testů v C++, které pomocí socketů přijímají a odesílají ze zadaných síťových rozhraní pakety. Jejich úkolem je získat množství takto přijatých/odeslaných paketů za jednotku času.

První z testů má za úkol pouze jednoduše ve smyčce přijímat a odesílat jeden paket. Další z testů pracuje obdobně, pouze místo jednoho paketu jich přijímá/odesílá 10 zároveň. Oba tyto testy mají dvouvláknovou modifikaci, kdy jedno vlákno čte a ukládá pakety do sdíleného kontejneru. Druhé vlákno z něj pakety odebírá a odesílá je. Pro synchronizaci vláken byly použity zámky. Další z testů, pojmenovaný MMAP, odstraňuje náročná systémová volání a pakety ukládá do naalokovaného kruhového bufferu. Poslední z testů je založený na knihovně libpcap.

Abychom byli schopni na dané platformě měření spustit, bylo potřeba vytvořit pomocí yocty vlastní systém s Linuxovou architekturou a přidat naše testy jako recepty. Poté stačilo naboťovat tento systém na NXP a po menších konfiguracích síťových rozhraní a připojení generátoru síťového provozu Spirent měřit výsledky.

## 4.7 Problémy při měření

K měření pod DPDK bylo potřeba provést alokaci DPAA2 objektů jako je například alokace front, 'buffer pools', síťových rozhraní apod. Velkým problémem byly ovladače. Ovladač pod DPDK se jmenuje vfio-fsl-mc a linuxový ovladač fsl-mc. Problém je, že při akcích bind – svázat, či unbind – rozsvázat se náhodně vyskytly problémy.

Problémy se vyskytly také při měření L2 switche. Při použití demultiplexoru na rozdělení příchozích paketů (podle MAC adresy) do jednotlivých front pro jednotlivá jádra se rychlost výrazně snížila oproti přímému připojení (přemostění demultiplexeru) na jádra, které dávalo plnou rychlost linky (10Gb/s). Ani přes různé konfigurace L2 switche či zvyšování IP adresy generovaných paketů se rychlost výrazně nezvýšila. Problém je hlavně ten, že neexistuje ovladač pro switch, který by byl DPDK podporován.

Při konfiguraci DPAA2 se NXP procesor často zasekl a následně bylo potřeba celý systém restartovat. Ze začátku při instalaci nových ovladačů se neviděly. Dokumentace ne vždy úplně odpovídala implementaci DPDK, což je způsobeno tím, že software k CPU je ještě v betaverzi a někdy bylo nutné najít informace až v implementaci.

Při měření pod Linuxem bylo nejsložitější sestavit systém pomocí yocty, kde bylo potřeba vyřešit spoustu chyb které při sestavování z různých důvodů vznikaly. Další problém byla nízká rychlost která vedla k názorům, že se NXP zaseklo, ovšem po přeměření hodnot s ujištěním, že NXP zaseklé není a zkontrolování zdrojového kódu testů byly hodnoty stejné.

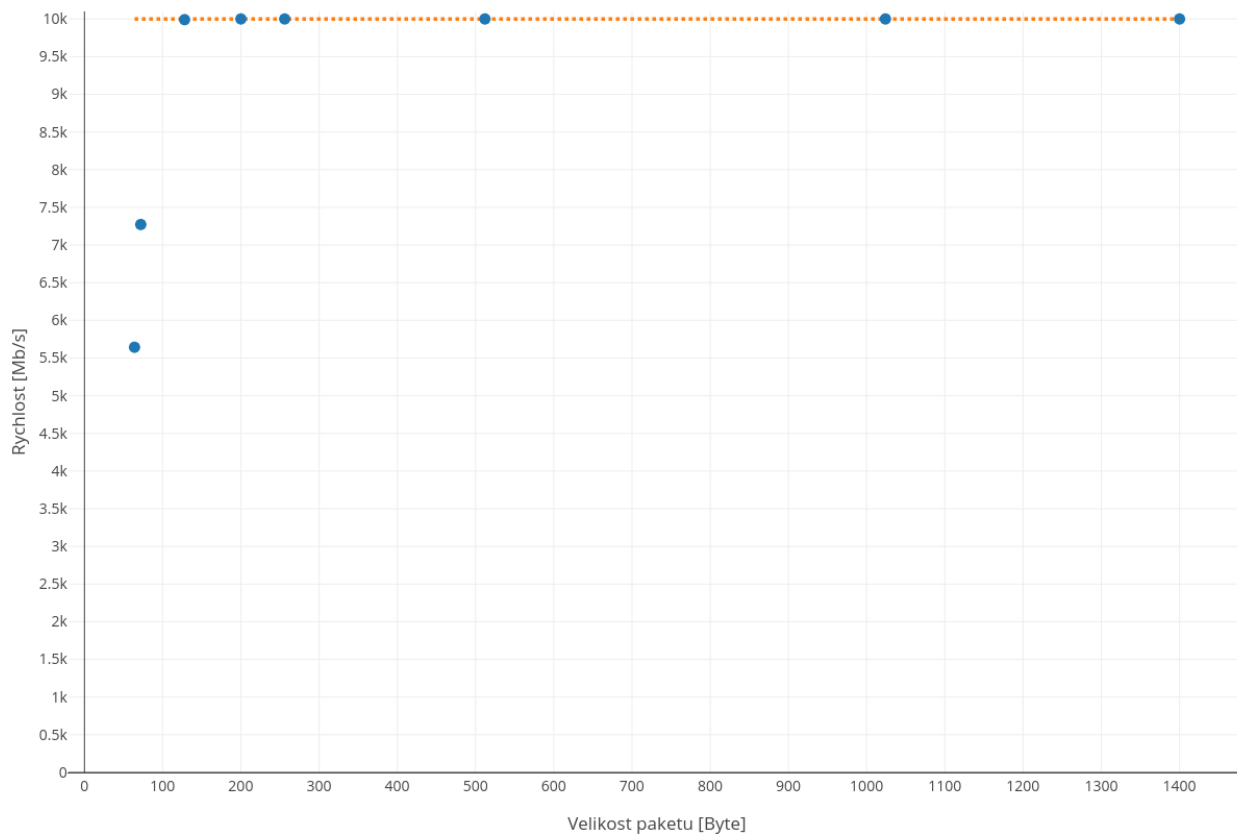
## 5 Výsledky měření

Všechny výsledky s DPDK byly nameřeny při jednom jádru na každý použitý port a při 8 frontách na jedno síťové rozhraní, jenž poskytovalo nejlepší výsledky. Při měření pod Linuxem bylo použito jedno jádro a pouze jedna fronta, protože ovladač jiný počet front nepodporuje.

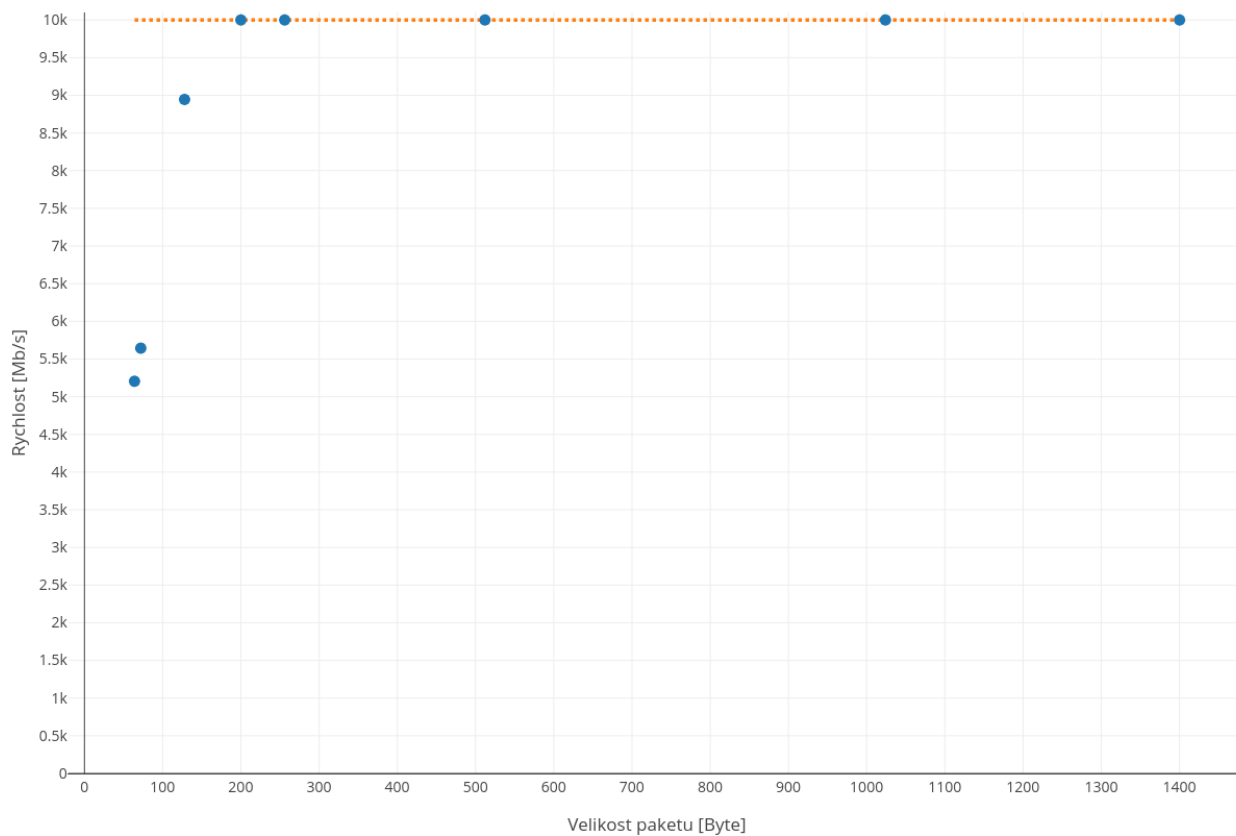
Konfigurace	Jader	Front na SR	Rych. 64B paket [FPS]	Rych. 1400B paket [FPS]
Pouze RX	1	8	8 397 505 (56%)	876 070 (100%)
Pouze TX	1	8	7 746 353 (53%)	878 280 (100%)
Malý loopback	2	8	7 038 974 (47%)	877 787 (100%)
Malý loopback - Intel	2	8	14 880 930 (100%)	877 016 (100%)
Malý loopback × 4	4	8	4 088 664 (27%)	877 801 (100%)
L2 Switch	–	8	5 910 773 (40%)	876 254 (100%)
Linux – Jeden paket	1	1	160 000 (1%)	92 865 (10%)
Linux – Jeden paket	2	1	11 768 (0.1%)	15 637 (1.78%)
Linux – Vektor 10 paketů	1	1	162 420 (1%)	161 820 (18.47%)
Linux – Vektor 10 paketů	2	1	11 660 (0.1%)	13 641 (1.55%)
Linux – MMAP	1	1	377 905 (2.5%)	412 197 (47%)
Linux – libpcap	1	1	188 952 (1.26%)	206 098 (23%)

Tabulka 1: Tabulka rychlostí na nejmenších a největších paketech při dané konfiguraci

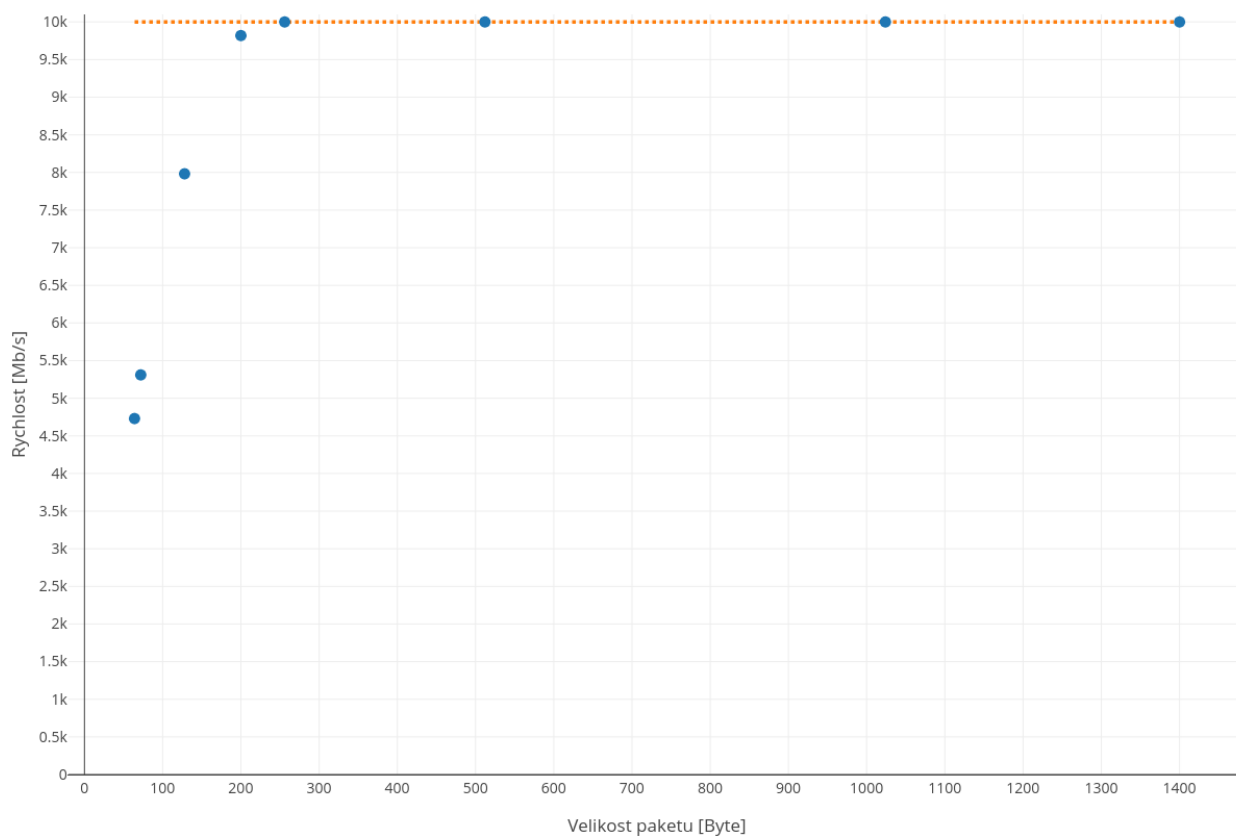
### 5.1 Výsledky měření s DPDK



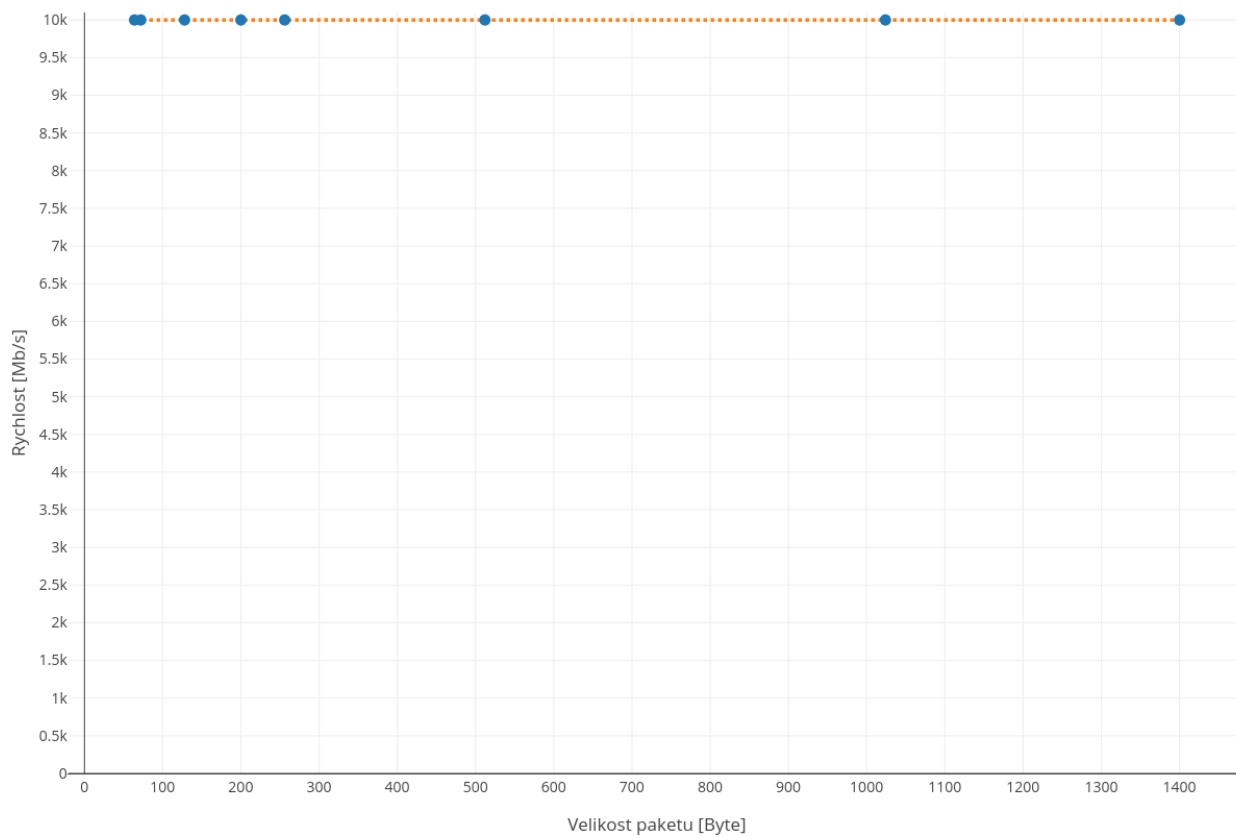
Obrázek 7: Pouze RX



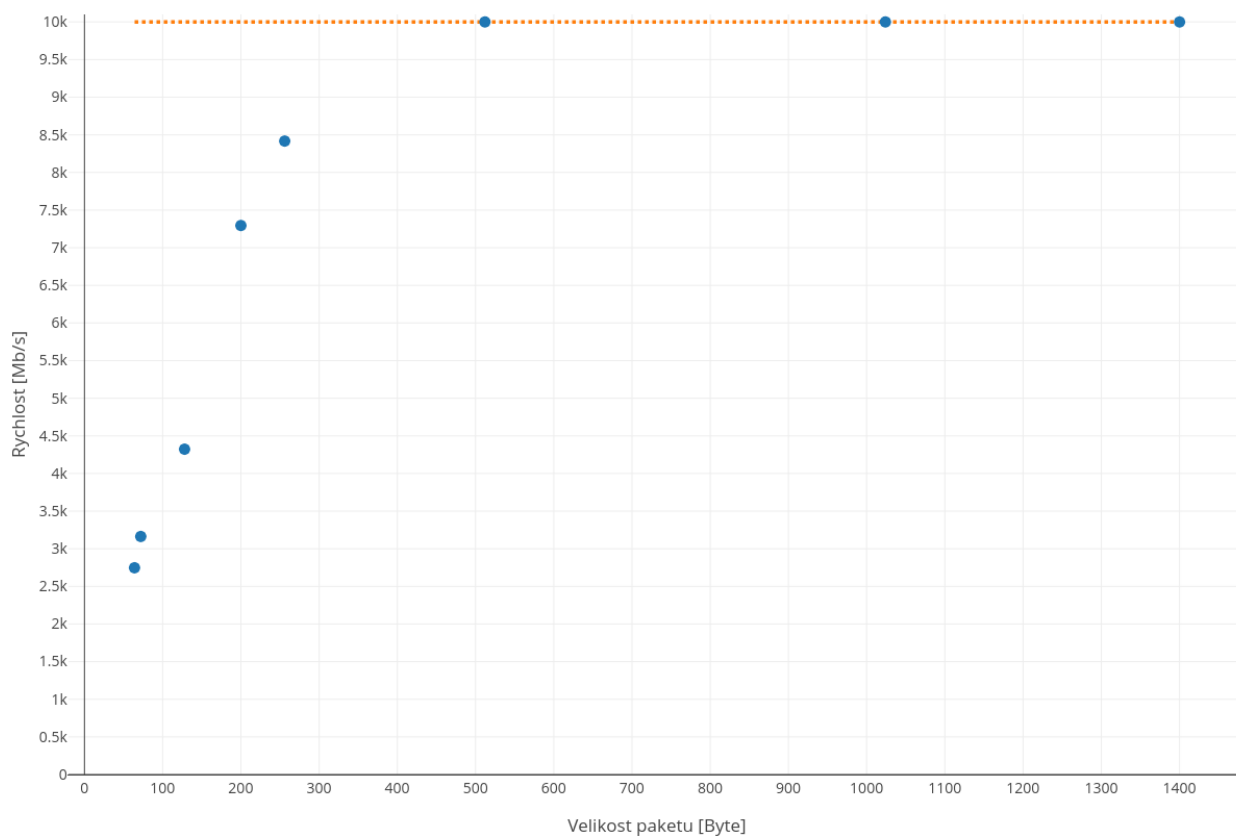
Obrázek 8: Pouze TX



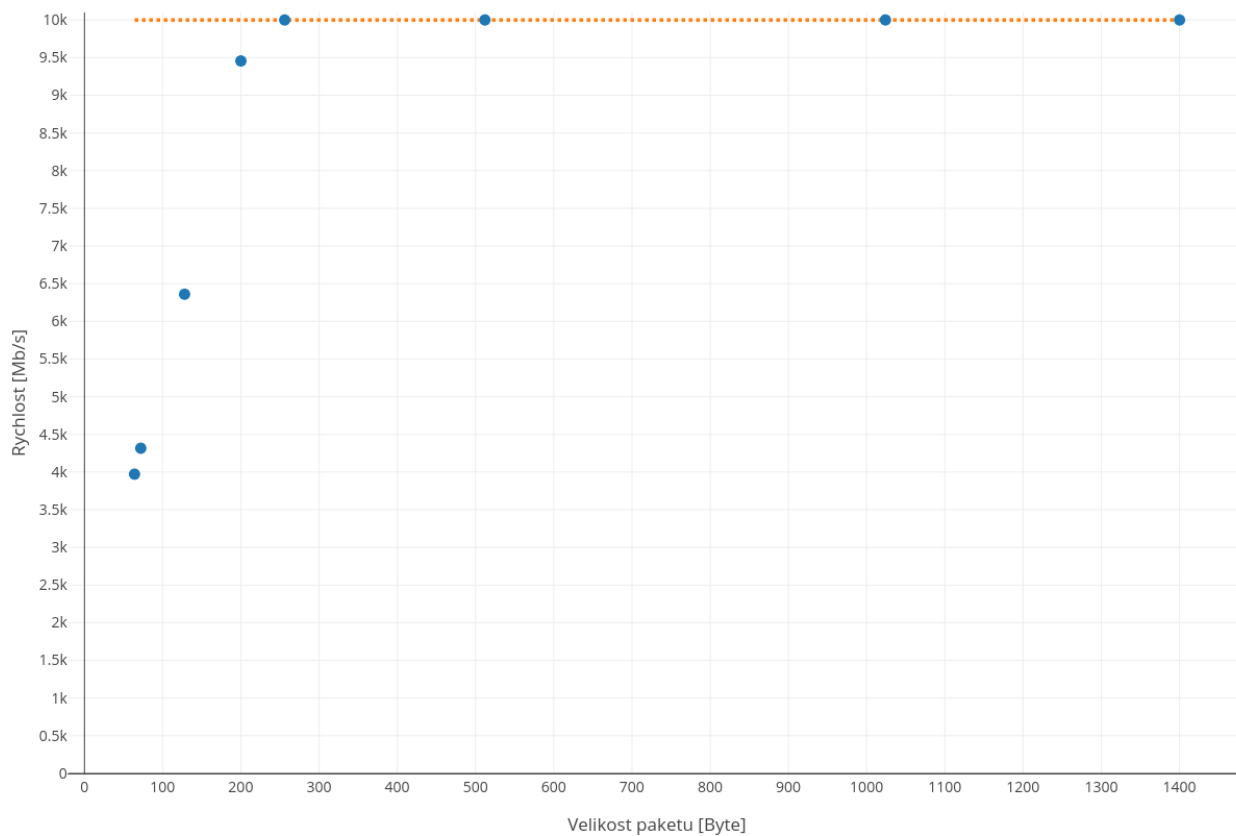
Obrázek 9: Malý loopback



Obrázek 10: Malý loopback – Intel

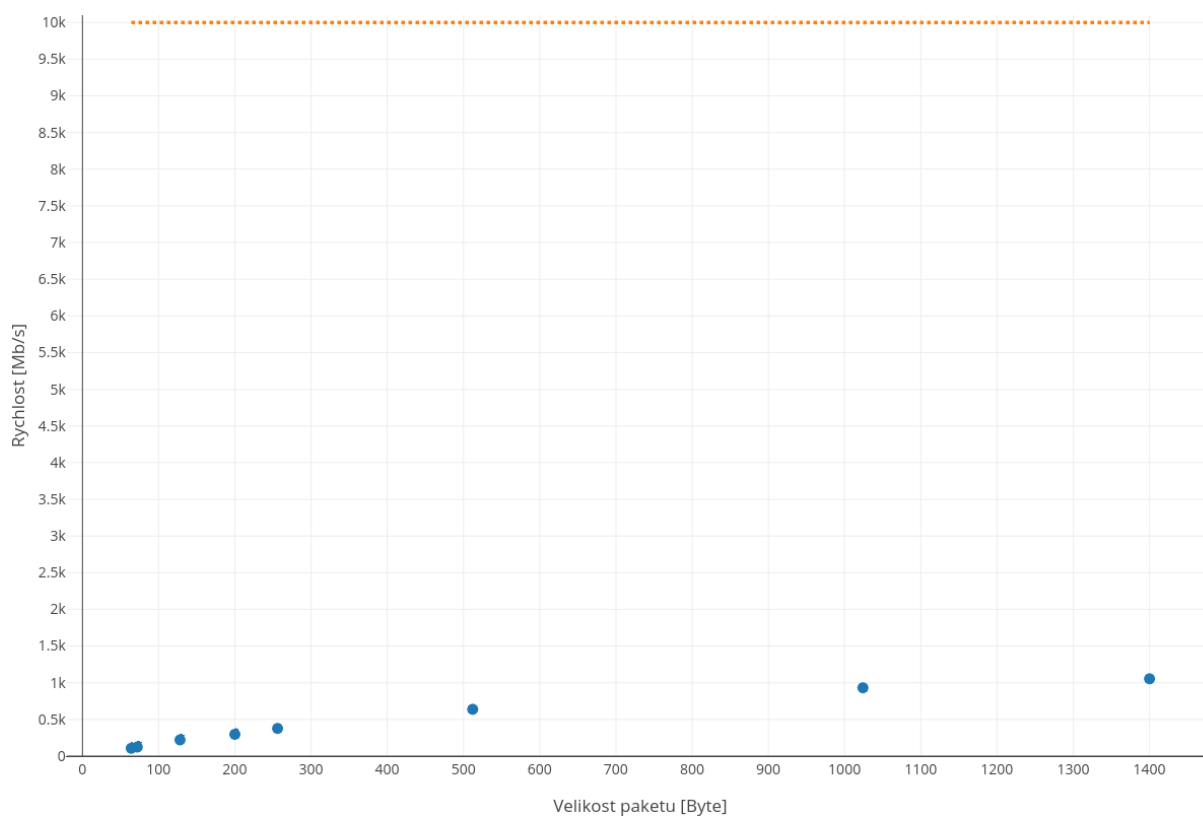


Obrázek 11: Malý loopback × 4

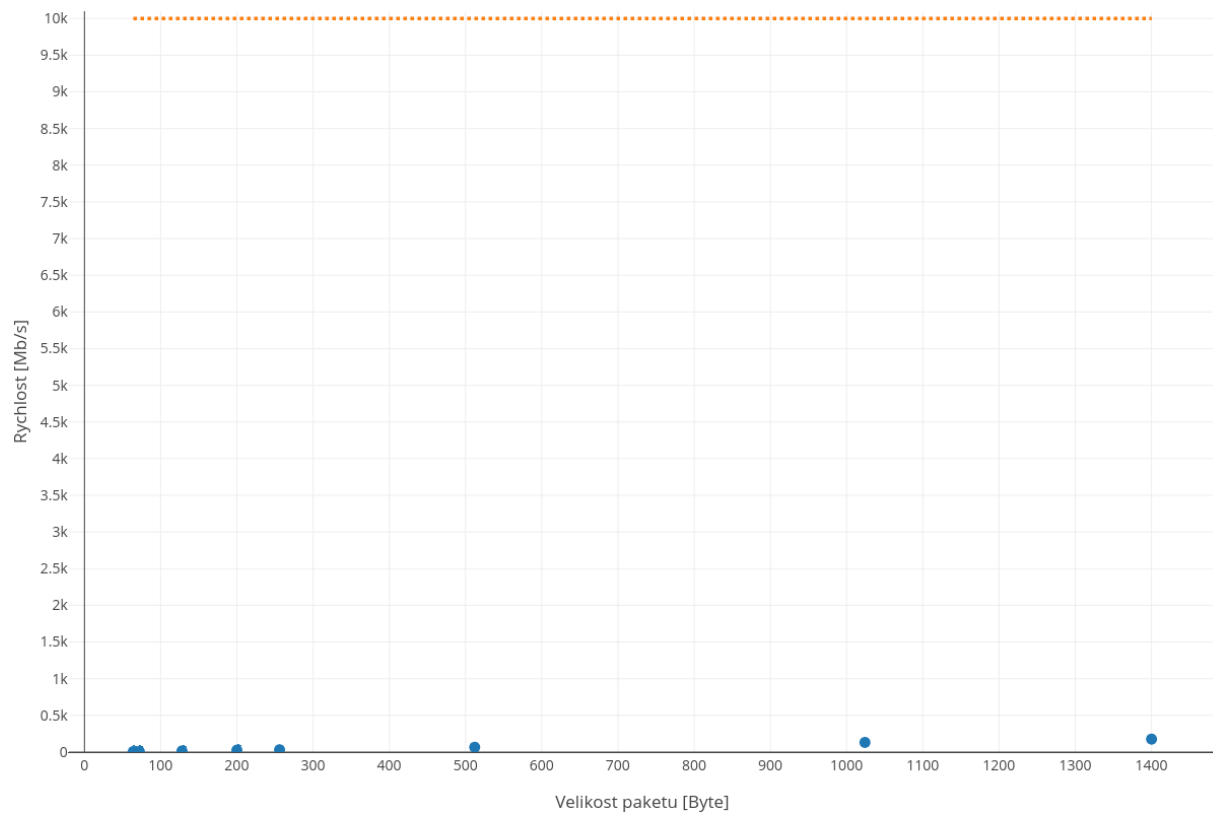


Obrázek 12: L2 Switch

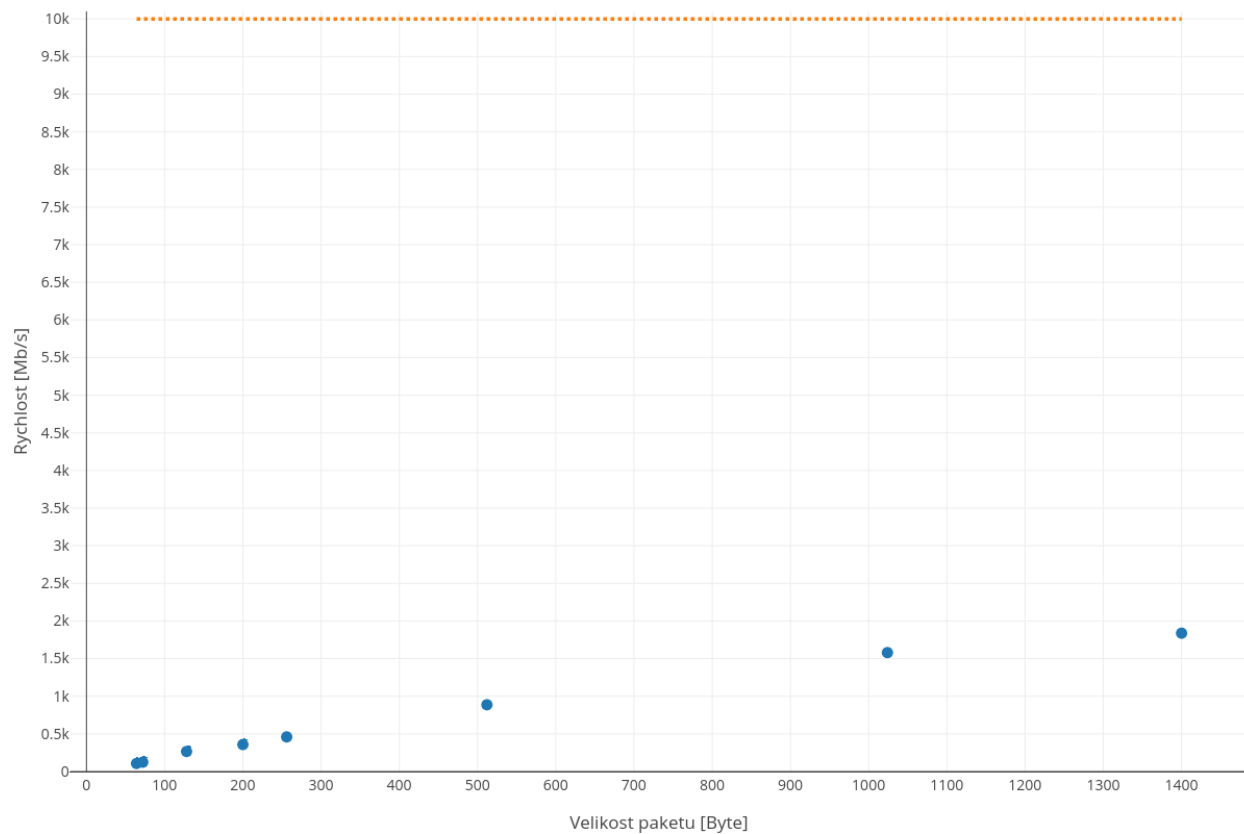
## 5.2 Výsledky měření pod Linuxem



Obrázek 13: Linux – Jeden paket

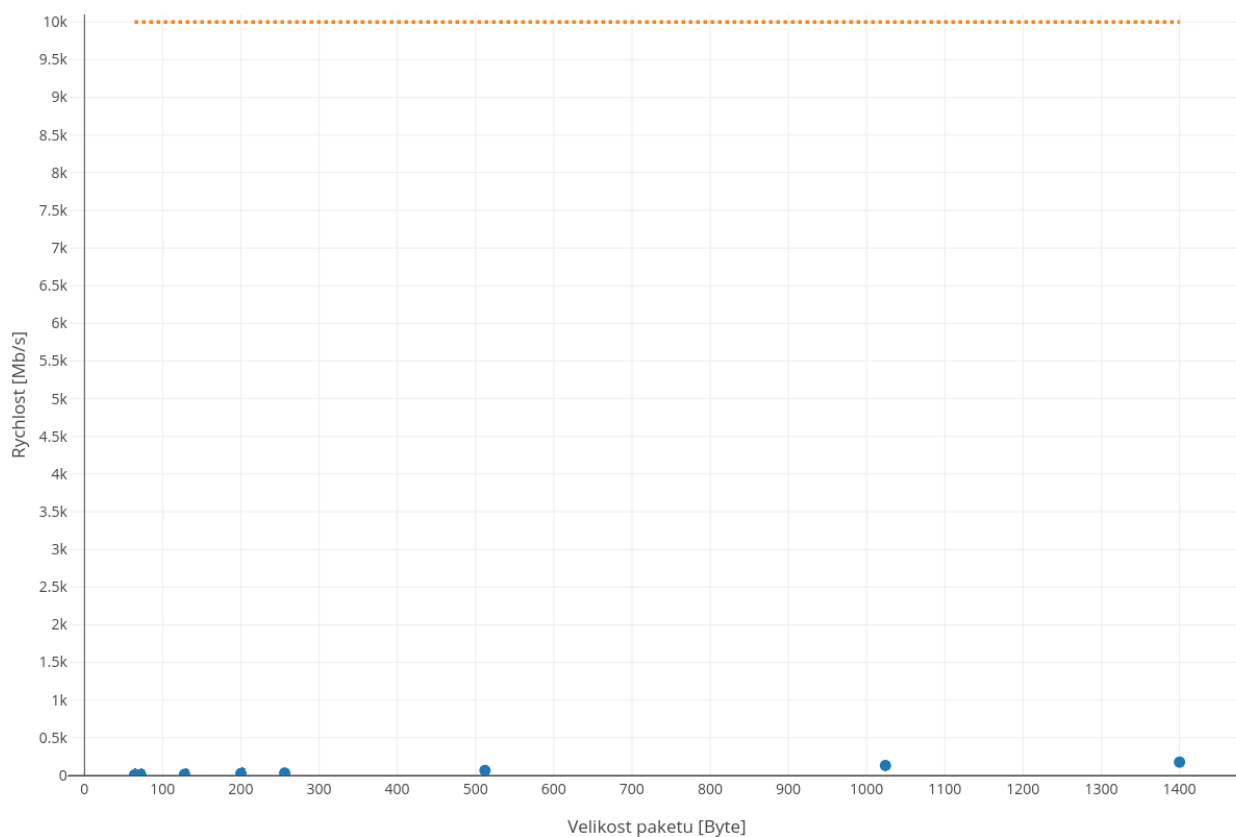


Obrázek 14: Linux – Jeden paket s dvěma vlákny

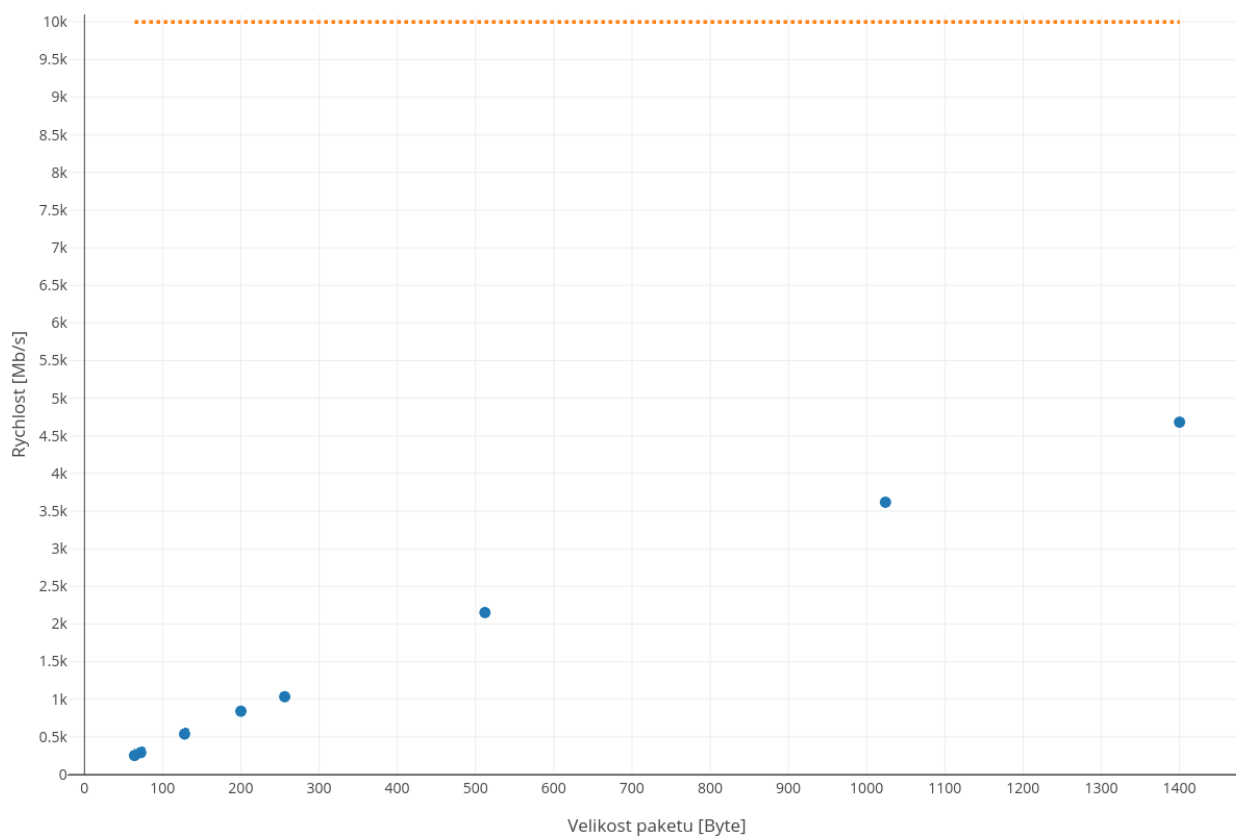




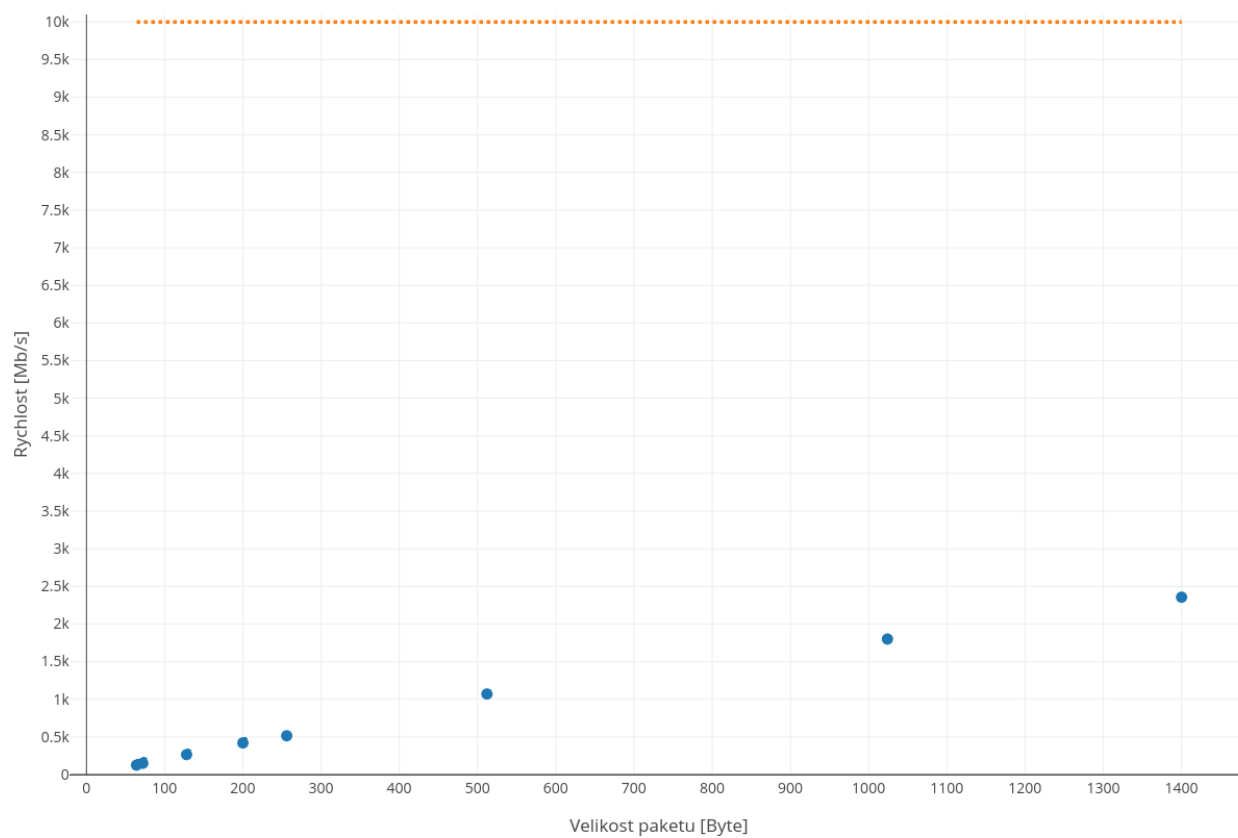
Obrázek 15: Linux – Vektor deseti paketů



Obrázek 16: Linux – Vektor deseti paketů s dvěma vlákny



Obrázek 17: Linux – MMAP



Obrázek 18: Linux – libpcap

## 6 Závěr

Z porovnání výsledků naměřených pod Linuxem a naměřených s frameworkem DPDK, vychází jednoznačně lépe framework DPDK.

Při pohledu na výsledky měření pod Linuxem zjistíme, že žádný z testů nedosáhne ani na nejvyšších paketech maximální propustnosti linky. Nejlépe dopadl test MMAP, který na největších paketech dosáhl alespoň poloviny maximální propustnosti. Druhý nejrychlejší byl test s knihovnou libpcap, který měl zhruba poloviční rychlost testu MMAP. Další testy měly zanedbatelnou rychlost, zejména kvůli systémovým voláním. U vícevláknových testů bylo dosaženo nízkých propustností jež zaviniily systémové zámky. Bohužel už se nepodařilo přeměřit testy s atomickými operacemi.

Naopak při měření s DPDK a jedním jádrem na port bylo zjištěno, že kromě malého loopbacku  $\times 4$  použitý procesor dosáhne na nejmenších paketech zhruba poloviční propustnost při maximální zátěži linky. Při zvyšující se velikosti paketů se propustnost skokovitě zvyšuje až na 128 či 200 bytových paketech dosáhne 100% propustnosti. U malého loopbacku  $\times 4$ , tedy nejnáročnější konfigurace, kde jsou všechny porty NXP procesoru plně vytíženy, se při nejmenších paketech povedlo získat nejvyšší propustnost 27%. Při zvyšování velikosti paketů propustnost narůstá pomaleji a na 512 bytových paketech dosáhne maximální propustnosti. Z porovnání malých loopbacků s procesorem Intel a procesorem od NXP vyšel jednoznačně lépe Intel, který již na nejmenších paketech dosahoval maximální propustnosti, kdežto ARM pouze 47% propustnosti. Nejlepších výsledků bylo dosaženo při jednom jádru na port a osmi frontách na jedno síťové rozhraní. Při zvýšení počtu jader a front k nim přiřazeným propustnost klesla, což může být způsobeno kopírováním paketů místo rozdělení paketů mezi více jader.

Výsledkem práce tedy jsou reálné parametry dané platformy a konfigurace platformy, při kterých bylo dosaženo nejvyšší propustnosti u jednotlivých konfigurací. Na druhou stranu software k CPU je nový a způsobuje nestabilitu celého systému. Navíc některé výrobcem deklarované vlastnosti jsou stále nefunkční a tudíž nebylo možné využít všech možností konfigurace softwaru.

## Literatura

### Reference

- [1] *Data plane development kit: DPDK* [online]. [cit. 2017-12-12]. Dostupné z: <http://dpdk.org>
- [2] *TestPMD: DPDK* [online]. [cit. 2017-12-12]. Dostupné z: [http://dpdk.org/doc/guides/testpmd\\_app\\_ug/index.html](http://dpdk.org/doc/guides/testpmd_app_ug/index.html)
- [3] *QorIQ 2088A: NXP* [online]. [cit. 2017-12-12]. Dostupné z: <https://www.nxp.com/products/processors-and-microcontrollers/arm-based-processors-and-mcus/qoriq-layerscape-arm-processors/qoriq-layerscape-2088a-and-2048a-multicore-communications-processors:LS2088A?&fsrch=1&sr=1&pageNum=1>
- [4] *Data Path Acceleration Architecture Gen2: DPDK* [online]. [cit. 2017-12-13]. Dostupné z: <http://dpdk.org/doc/guides/nics/dpaa2.html>
- [5] *Network matrices* [online]. [cit. 2017-12-12]. Dostupné z: <https://www.cisco.com/c/en/us/about/security-center/network-performance-metrics.html>