VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

IP1 – Projektová praxe 1

Technická zpráva

18.10.2017 Filip Kočica

Abstrakt

S příchodem nové platformy nejsou s jistotou známy její reálné výkonnostní parametry. U platformy NXP QorlQ LS2088A výrobce uvádí, že zvládne wirespeed – zpracovat všechny pakety na plně vytížené lince. Cílem této práce je zjistit realné parametry dané platformy a objevit funkční řešení, která budou poskytovat požadovaný výkon. Platforma však má řádově tisíce možných konfigurací a parametrů, přičemž se ještě pořád jedná o produkční vzorek a řada věcí nefunguje dle očekávání.

Abstract

With the release of the new platform, its real performance parameters are not known. At the NXP QorlQ LS2088A platform, the manufacturer states that it can handle wirespeed – process all packets on a fully loaded line. The aim of this work is to determine the real parameters of that platform and to discover functional solutions that will provide the required performance. The platform, however, has thousands of possible configurations and parameters, while it is still a production sample and many things do not work as expected.

1 Úvod

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

Vzhledem k faktu, že Linux podporuje network stack, který kopíruje pakety z paměti pro operační systém do uživatelské paměti, což způsobuje zpomalení zpracování paketů, bude k měření použit framework DPDK. DPDK tuto vlastnost obchází – zero copy. Za účelem dosažení co nejlepších výsledků bude potřeba provést optimální konfiguraci softwaru CPU, jež je teprve v betaverzi. Pro porovnání bude také uvedeno měření na zařízení s procesorem Intel.

2 DPDK

DPDK[1] je soubor knihoven a ovladačů pro rychlé zpracování paketů. Dnes podporuje nejpoužívanější typy procesorů, ale pouze v linuxovém prostředí (podmnožina funkcí DPDK funguje také na FreeBSD) [1]. PMD[2] ovladač může pracovat v reálném a virtualizovaném prostředí a ukládá pakety přímo do user-space (pamět vyhrazená pro uživatele) a nedochází tak ke kopírování mezi kernel-space (paměť vyhrazená pro OS) a user-space.

Klíčové části, ze kterých se DPDK skládá, jsou EAL[3] která zajišťuje komunikaci s operačním systémem a přístup k nízkoúrovňovým zdrojům, poll-mode 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 a mimo jiné podporuje mód přesměrovávání paketů mezi ethernetovými porty a síťovým rozhraním [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 přijímá jádro pakety z jednoho portu a odešle je z jiného portu. 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 Rx 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[4] pro každý paket přijatý na Rx port. Cílový port je přilehlý port z povolené portmasky.

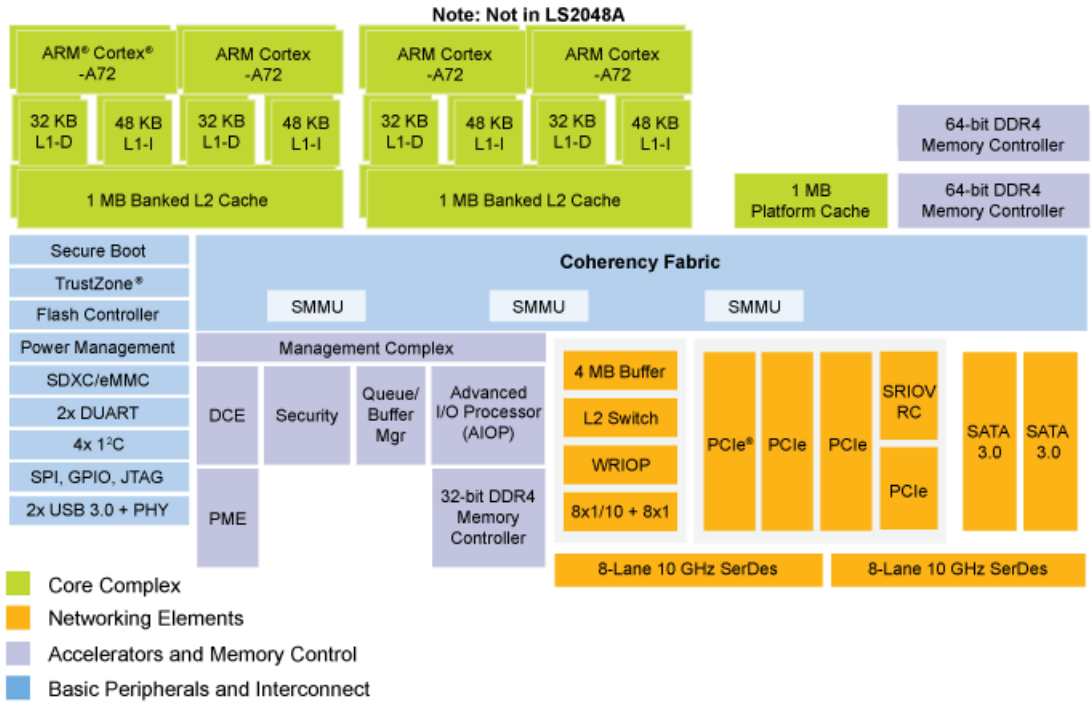
[1] Data plane development kit
[2] Poll Mode Driver
[3] Environment Abstraction Layer
[4] L2F je tunelovací protokol vyvinutý společností Cisco Systems, Inc. k vytvoření připojení virtuálních privátních sítí přes internet.

3 Platforma

Hlavním výpočetním prvkem platformy je procesor NXP QorlQ LS2088A. Procesor disponuje osmi 64–bitovými výpočetními jádry ARM Cortex-A72 s maximálním taktem 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 akcelerátorem zpracování paketů o rychlosti 20Mp/s a L2 Switchem, který na základě nastavených pravidel funguje jako multiplexor a podle MAC adresy paketu jej přesměruje do vybrané fronty, kde každé jádro má svou frontu [3].

Další, již netestované funcionality tohoto procesoru v oblasti zpracování síťového toku

1. 10 Gb/s Pattern Matching regulerních 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 také 64 MB EEPROM a dvě 72 bitové DDR4, 4 GB na slot. Pro periferie jsou připraveny dvě SATA, dvě USB 3.0 a 1×8 PCI Express třetí generace nebo 2×4 PCI Express sběrnice.

3.2 DPAA2

DPAA2[5] je hardwarová architektura určená pro zpracování vysokorychlostních síťových paketů. DPAA2 se skládá ze sofistikovaných mechanismů pro zpracování ethernetových paketů, správu fronty, správy vyrovnávacích pamětí, autonomního přepínání L2, hardwarovou podporou virtualizace Ethernetu a sdílení akcelerátorů (například kryptografických).

Hardwarová součást DPAA2 s názvem Management Complex (dále jen MC) spravuje hardwarové prostředky DPAA2. MC poskytuje objektovou abstrakci pro softwarové ovladače 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 instance akcelerátoru. MC také poskytuje paměťově mapované vstupně–výstupní příkazové rozhraní (MC portály), které softwarové ovladače DPAA2 používají pro práci s objekty DPAA2.

4 Měření

Abychom byli schopni určit výkonnostní parametry platformy, je třeba provést měření. K získaní 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 přijímaji maximum. 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.

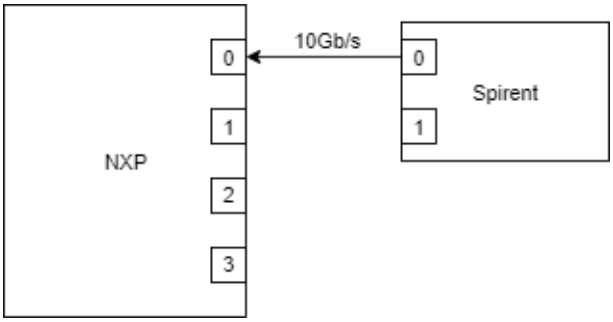
Fronta paketů je zpravidla implementována jako vyrovnávací paměť vstupu (first-in, first-out – FIFO) s pevnou velikostí. Fronta neobsahuje paketová data. Místo toho se skládá z deskriptorů, které odkazují na jiné datové struktury nazvané socket kernel buffers (SKB), které drží paketová data a používají se v jádře.

4.2 Spirent

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 dvěmi SFP+ porty, které jsou schopny přenést až 10 Gb/s.

4.3 Měřené konfigurace

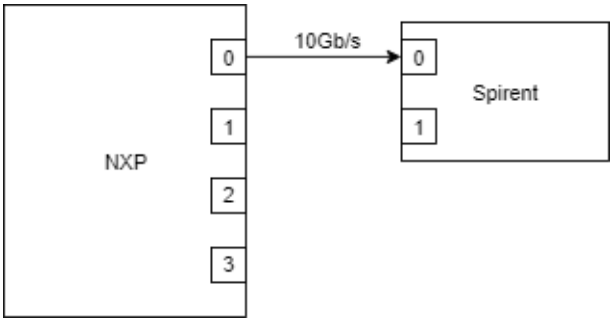
1. *Rx only* – Generování paketů pomocí Spirentu a odesílání přes jednu linku přivedenou na port NXP, ze kterého se pomocí aplikace TestPMD odečítala výsledná rychlost.



[5]Data Path Acceleration Architecture Gen2

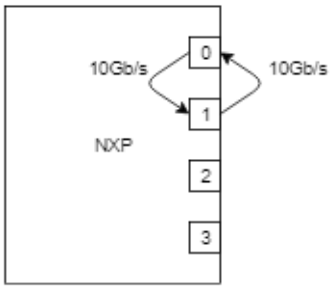
Obrázek 2: RX only schéma

2. *Tx only* – Generování paketů pomocí aplikace TestPMD a odesílání přes jednu linku přivedenou na port Spirentu, ze kterého se odečítala výsledná rychlost.



Obrázek 3: TX only schéma

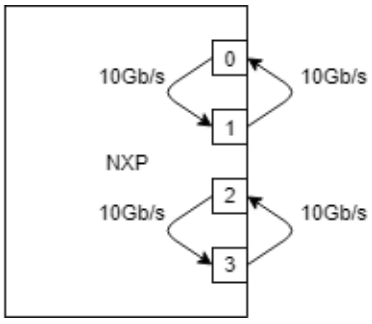
3. *Malý loopback* – Generování paketů pomocí aplikace TestPMD a odesílání přes dva porty, kde pakety z portu 0 jsou přesměrováný na port 1 a opačně. Výsledná rychlost se vypočítala jako průměr rychlostí RX obou portů.



Obrázek 4: Malý loopback schéma

4. *Malý loopback – Intel* – Stejný princip jako u malého loopbacku s NXP, pouze se zařízením s procesorem od Intelu. Měření se provádělo pro porovnání.

5. *Malý loopback* $\times$ *4* – Dva malé loopbacky mezi čtyřmi porty, první malý loopback je mezi porty 0 a 1. Druhý loopback je mezi porty 2 a 3. Výsledná rychlost byla vypočtena jako průměr rychlostí RX portů.



Obrázek 5: Malý loopback $\times$ 4 schéma

6. *L2 Switch* – Na základě nastavených pravidel funguje jako multiplexer 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 testu 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 jednoduchá ukázka 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[6]

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

kde 8 je preambule paketu a 12 je mezera mezi pakety.

Pro kontrolu hodnot naměřených pomocí TestPMD, byla použita aplikace L2FWD. Ukázalo se, že naměřené hodnoty jsou totožné jako u aplikace TestPMD.

4.6 Problémy při měření

Největší problémy se vyskytly při měření L2 switche. Při použití multiplexeru na rozdělení příchozích paketů (podle MAC adresy) do jednotlivých front pro jednotlivá jádra se rychlost výrazně snížila oproti přímému připojení (přemostění multiplexeru) na jádra, které dávalo plnou rychlost linky (10Gb/s) i při nejmenších paketech. Ani přes různé konfigurace L2 switche či zvyšování IP adresy generovaných paketů se rychlost výrazně nezvýšila.

Při snaze o dosažení co nejvyššího výkonu bylo potřeba provést optimální konfiguraci NXP. Ovšem platforma má řádově tisíce možných konfigurací a parametrů. Také 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.

5 Výsledky měření

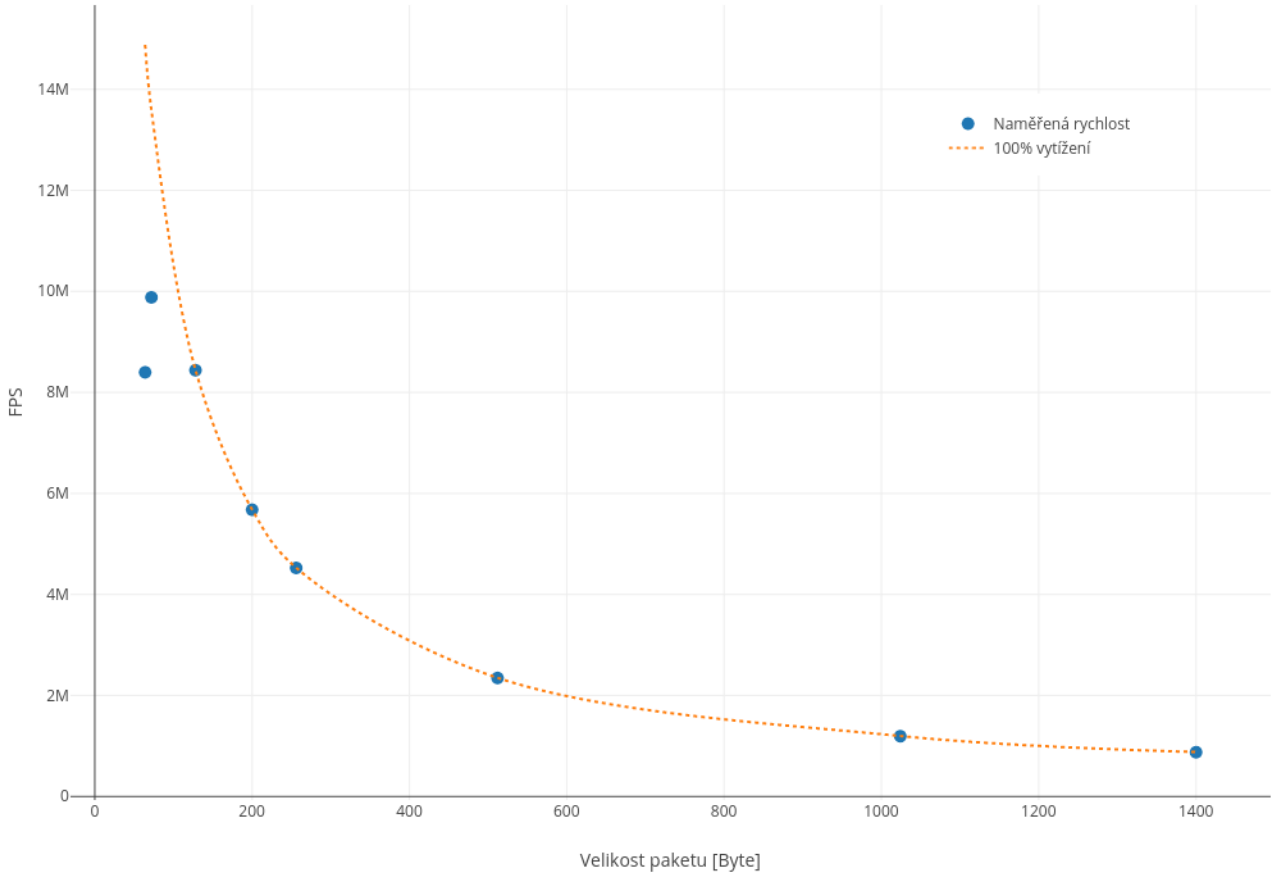
Všechny výsledky byly nameřeny při jednom jádru na každý použitý port a při 8 frontách na jeden NI[7] za účelem dosažení co nejvyšší rychlosti.

Konfigurace	Jader	Front na NI	Rychlost 64B paket [FPS]	Rychlost 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	–	–	–	–

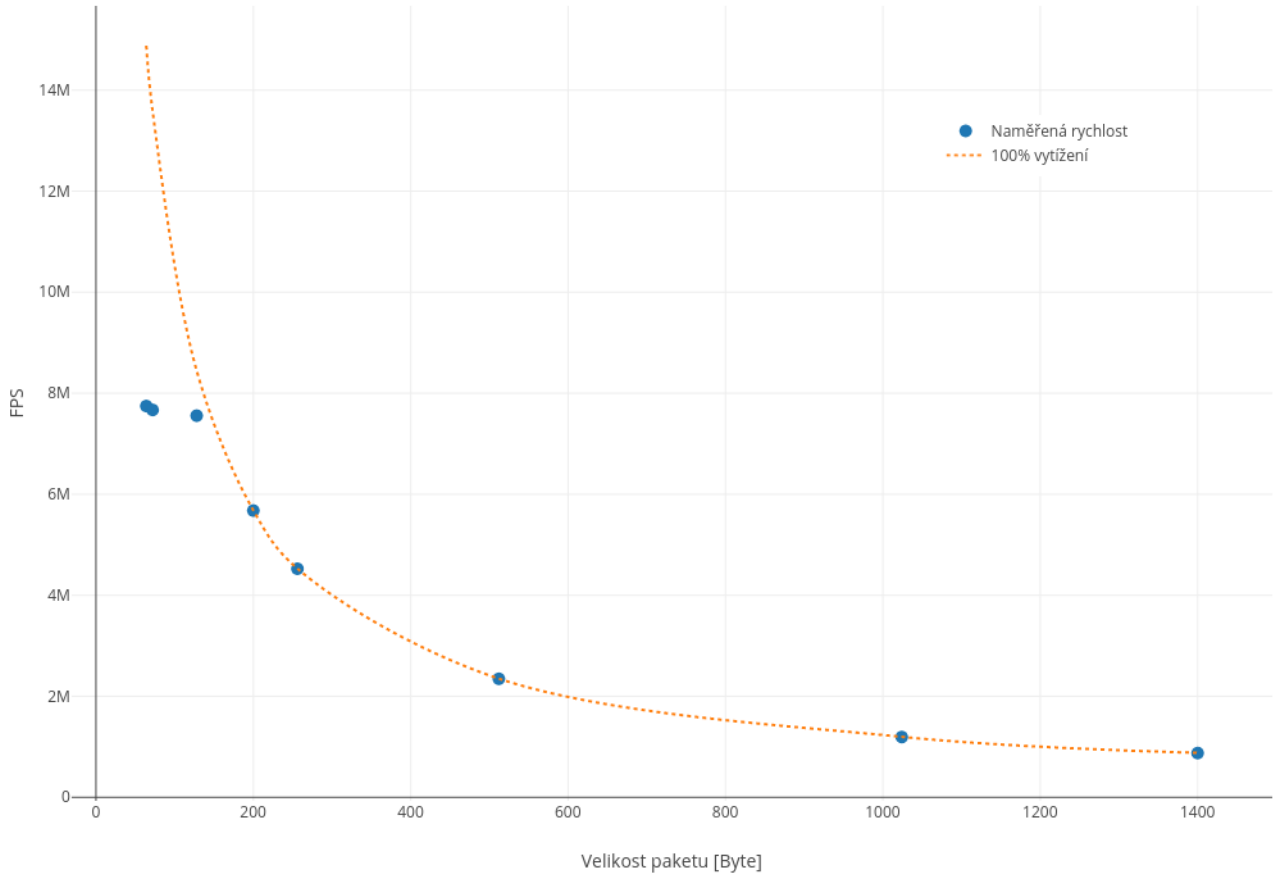
Tabulka 1: Tabulka rychlostí na nejmenších a největších paketech při dané konfiguraci

[6] Zdroj: `https://www.cisco.com/c/en/us/about/security-center/network-performance-metrics.html`

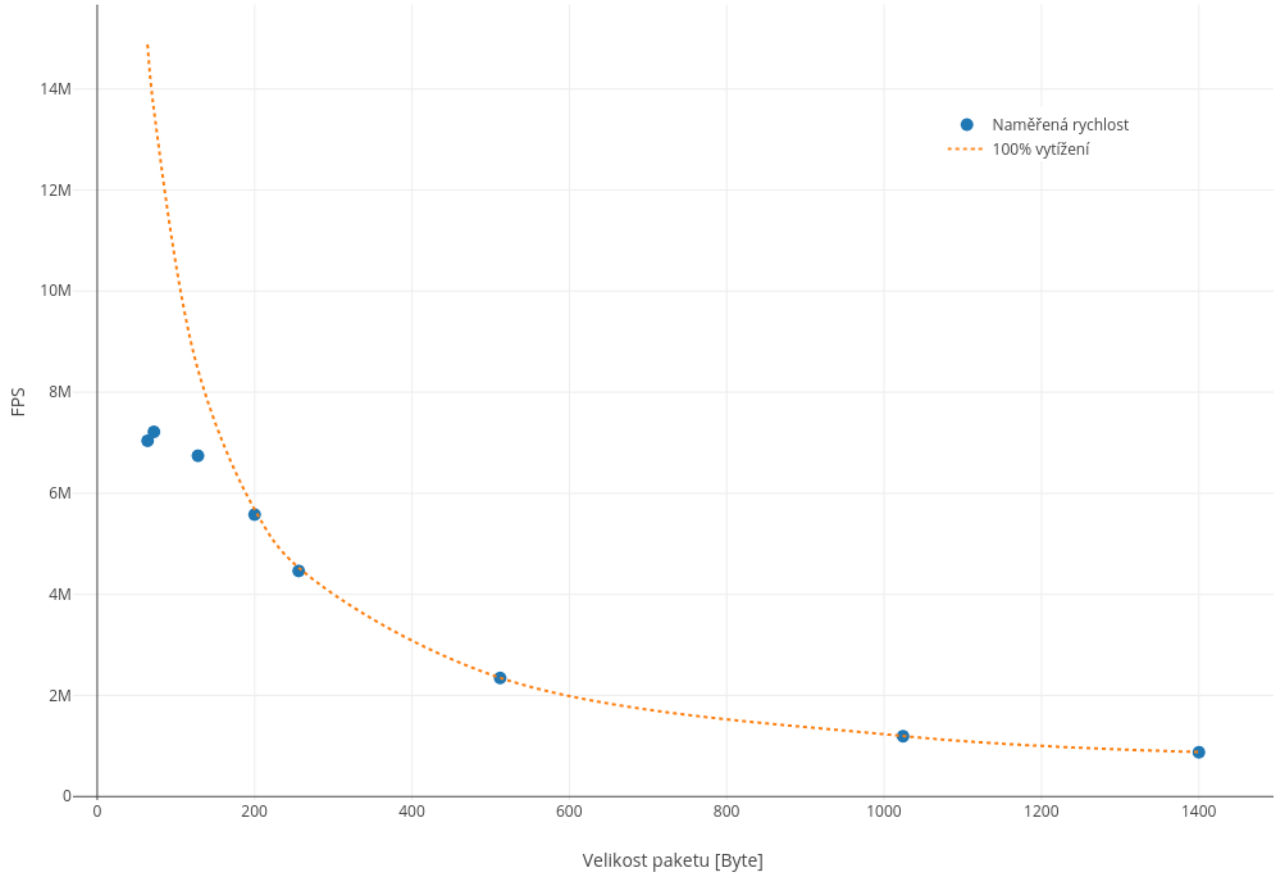
[7] Network interface



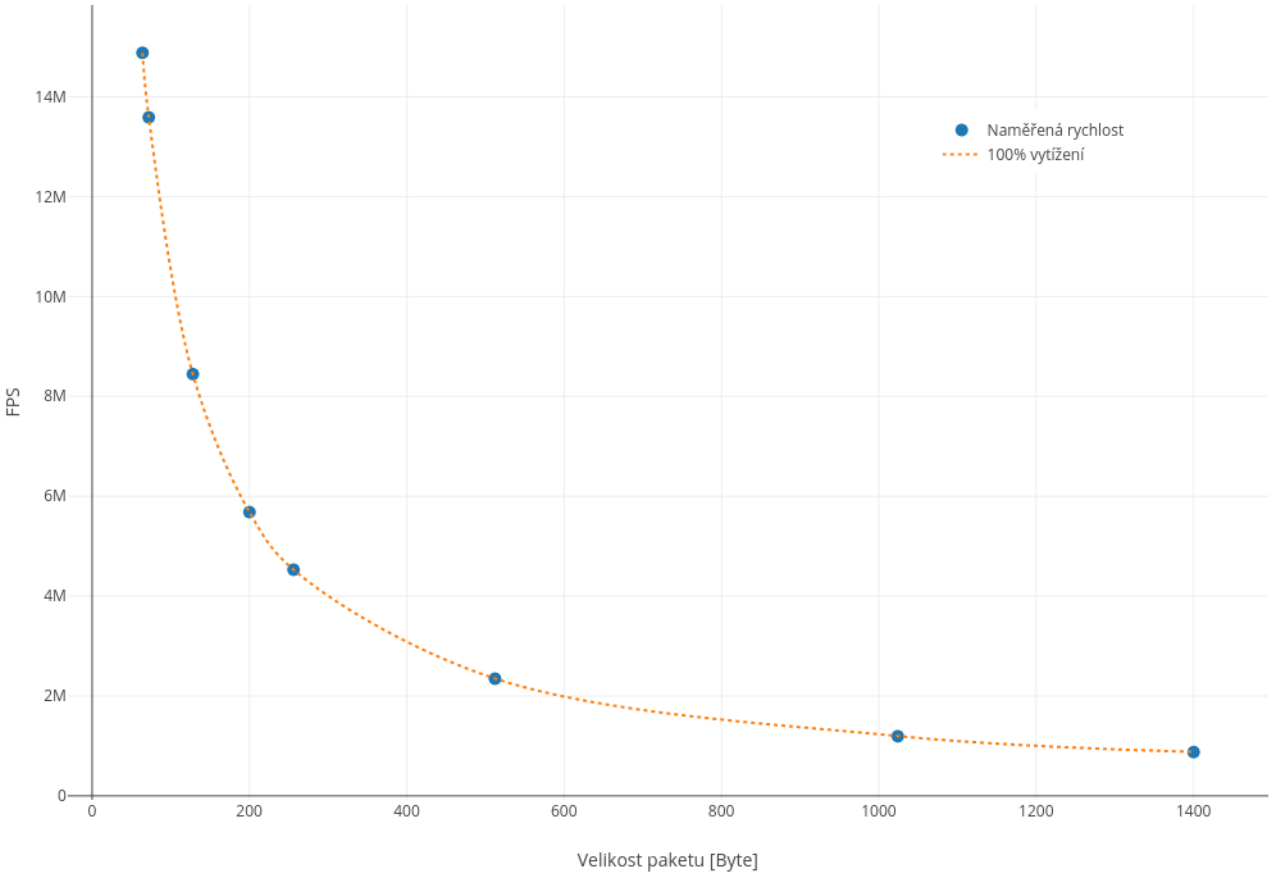
Obrázek 6: Pouze RX



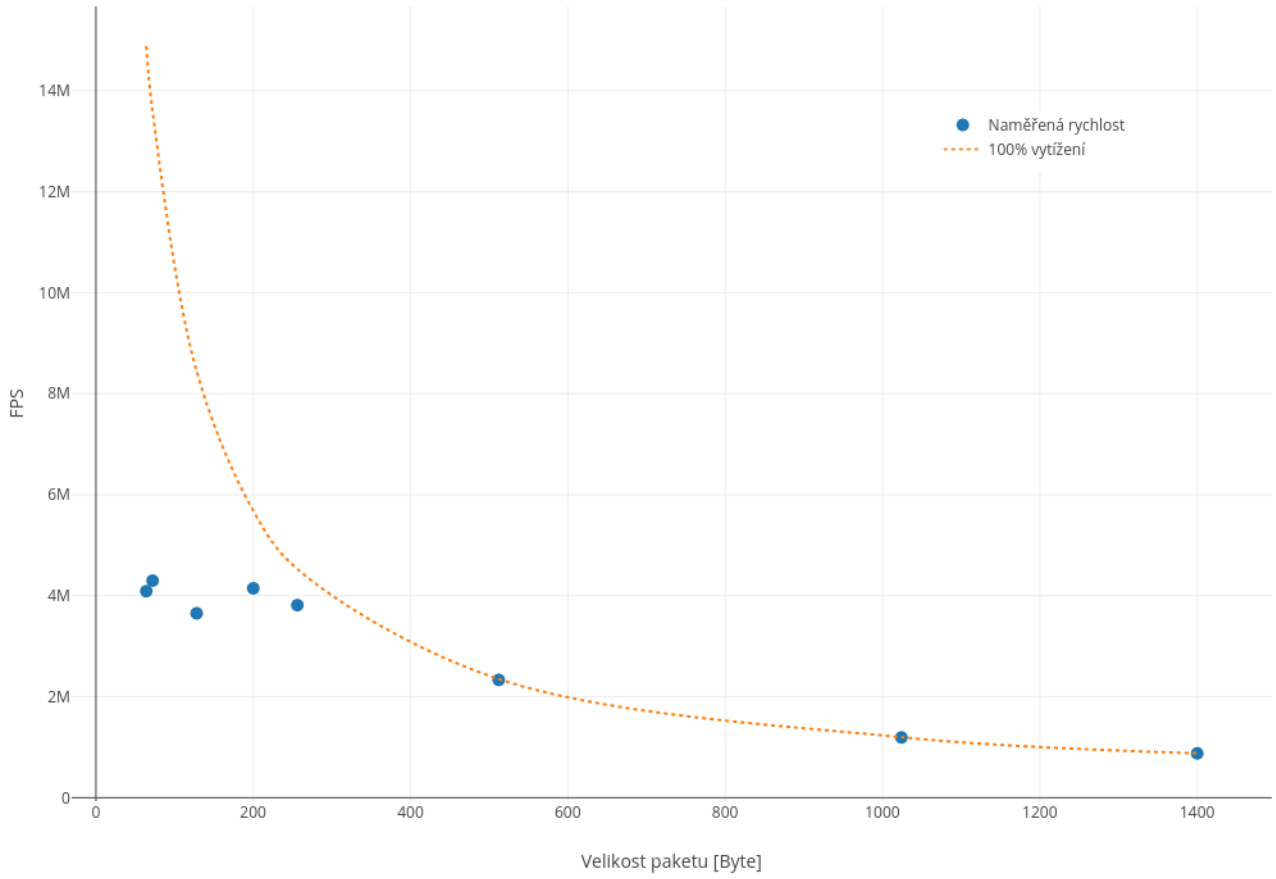
Obrázek 7: Pouze TX



Obrázek 8: Malý loopback



Obrázek 9: Malý loopback – Intel



Obrázek 10: Malý loopback × 4

TODO: L2 Switch

6 Závěr

Při měření 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 všechny porty NXP přijímají i odesílají maximum, se při nejmenších paketech povedlo získat nejvyšší propustnost 27%. Při zvyšování velikosti paketů propustost narůstá pomaleji a na 512 Bytových paketech dosáhne maximální propustnosti.

Z porovnání malých loopbacků s procesory Intel a ARM 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 jeden network interface. Při zvýšení počtu jader a front k nim přiřazeným rychlost 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 teprve v beta-verzi a tudíž nebylo možné využít všech možností konfigurace softwaru a ne vždy vše fungovalo v souladu s dokumentací.

Literatura

[1] Data plane development kit. DPDK [online]. [cit. 2017-12-02]. Dostupné z: `http://dpdk.org`
[2] TestPMD. DPDK [online]. [cit. 2017-12-02]. Dostupné z: `http://dpdk.org/doc/guides/testpmd_app_ug/index.html`
[3] QorIQ 2088A. NXP [online]. [cit. 2017-12-02]. Dostupné z: `https://www.nxp.com/products/processors-and-microcontrollers/arm-based-processors-and-mcus/qoriq-layerscape-arm/qoriq-layerscape-2088a-and-2048a-multicore-communications-processors:LS2088A?&fsrch=1&sr=1&pageNum=1`