

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

IPK – Projekt 2

Varianta 1 – Bandwidth Measurement

## **Obsah**

<b>1</b>	<b>Úvod</b>	<b>2</b>
<b>2</b>	<b>Implementace</b>	<b>2</b>
<b>3</b>	<b>Výsledky měření</b>	<b>2</b>
<b>4</b>	<b>Omezení</b>	<b>3</b>
<b>5</b>	<b>Literatura</b>	<b>3</b>

# 1 Úvod

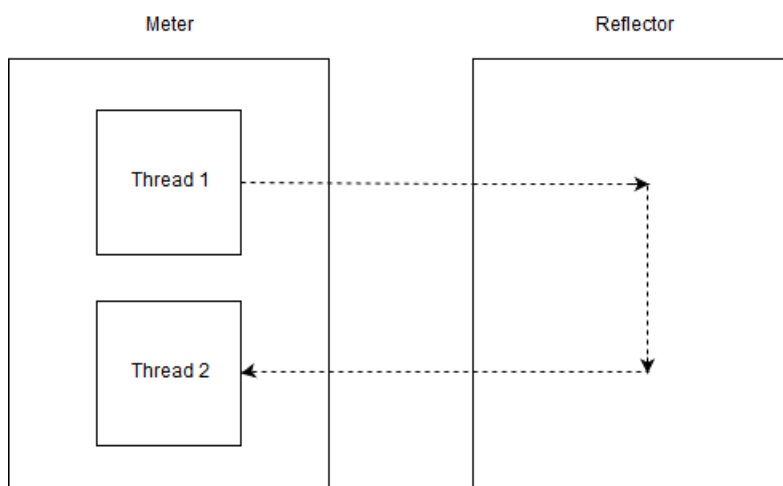
Úkolem bylo vytvořit aplikaci, jež provede měření přenosové rychlosti mezi dvěma stanicemi v internetu. K měření bylo zadáno použít protokol UDP. *"Na rozdíl od protokolu TCP totiž nezaručuje, zda se přenášený datagram neztratí, zda se nezmění pořadí doručených datagramů, nebo zda některý datagram nebude doručen vícekrát."*[1] Proto bude implementace zjištění přenosové rychlosti trochu složitější.

## 2 Implementace

K měření byly použity funkce `recvfrom()` a `sendto()`. Jsou to systémová volání jež jsou celkem náročná a při měření výkonnosti by bylo vhodnější použít spíše framework DPDK či MMAP (princip zero-copy).

Každý paket má časovou známku, z níž se pak v měřáku počítá průměrné RTT.

Měření probíhá ve dvou vláknech. První ve smyčce po zadaný čas s určitým uspaním odesílá pakety a druhé stále jen počítá příchozí pakety pomocí blokujícího `recvfrom()`.



Obrázek 1: Schéma komunikace klient–server

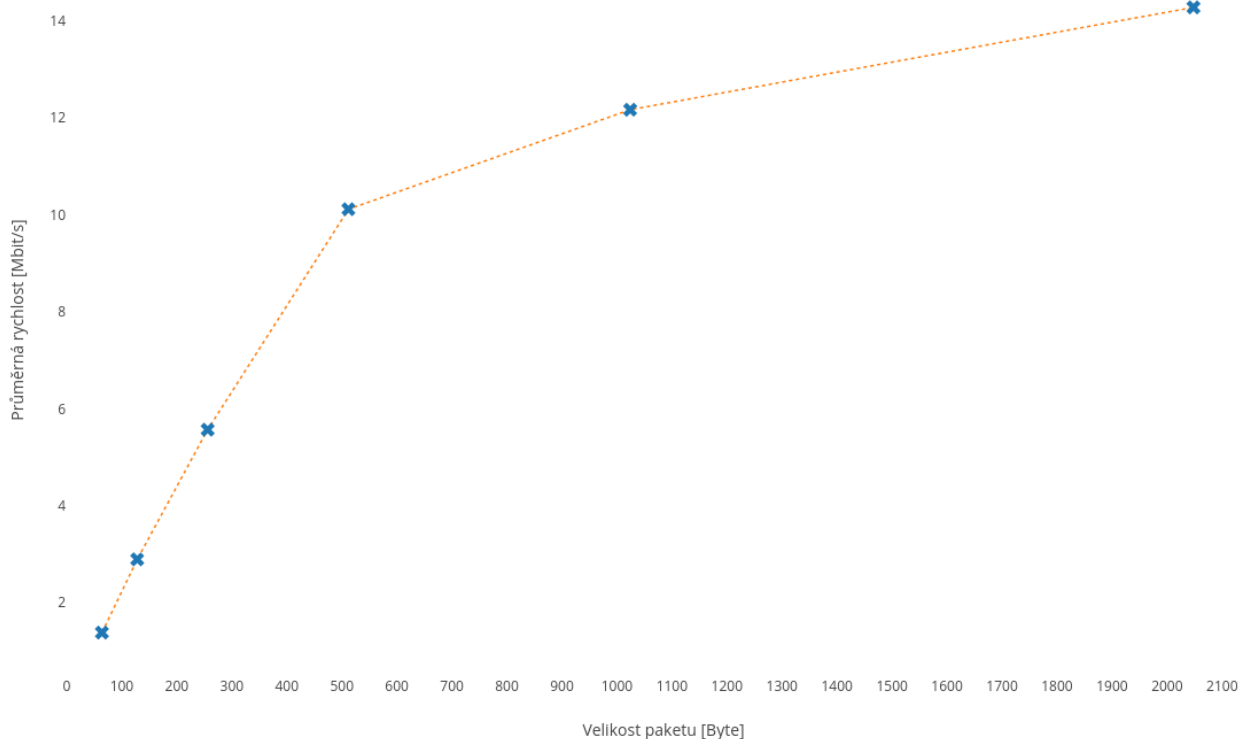
Pro rychlost přenosu dat platí vztah[2]

$$RychlostPrenosu = PocetPrenesenychDat / DobaPrenosu$$

Do tohoto vztahu může negativně zasáhnout odezva. K zjištění přenosové rychlosti jsem využil princip postupného zvyšování rychlosti odesílání až do doby, kdy se pakety začnou zahazovat (resp. ztrácet). V tu chvíli dokážu zjistit maximální přenosovou rychlost díky vztahu zmíněnému výše.

## 3 Výsledky měření

Tato tabulka zobrazuje průmětnou rychlost v závislosti na velikosti paketu. Vzhledem k tomu, že při větších paketech probíhá méně (náročných) systémových volání, méně parsování paketů a méně kopírování mezi kernel–user space pamětí než u menších paketů, tak by měla přenosová rychlost zákonitě stoupat. Všechny měření v grafu byly prováděny s dobou měření (parametr `-t`) 10 sekund, velikostmi paketů (bez ethernetové a IPv4 hlavičky) [64, 128, 256, 512, 1024, 2048] a měřák u mě na PC + reflektor na školním serveru eva.



Obrázek 2: Graf rychlostí v závislosti na velikosti "probe"paketu

Průměrná RTT se pohybovala v rozmezí 30 až 60 milisekund. Pro porovnání přikládám výsledky speedtestu[3] k serveru Brno University of Technology.

**Odezva:** 32ms

**Download:** 28.35Mbps

**Upload:** 4.09Mbps

## 4 Omezení

Sonda musí být alespoň tak velká, aby se do ní vešla struktura struct timeval (mělo by být na všech systémech 16 Bytů), jež přenáší časovou známku pro zjištění průměrné RTT. Počítal jsem s tím, že velikost ethernetové a IPv4 hlavičky se do této zadané velikosti nepočítá. Velikost sondy je na straně serveru omezena max na 2KiB (2048B) – při měření se velikost měnila dle potřeby.

## 5 Literatura

[1]*Used Datagram Protocol: Wikipedia* [online]. [cit. 2018-03-30]. Dostupné z: [https://cs.wikipedia.org/wiki/User\\_Datagram\\_Protocol](https://cs.wikipedia.org/wiki/User_Datagram_Protocol)

[2]*What is bandwidth: Veber* [online]. [cit. 2018-03-30]. Dostupné z: <https://www.veber.co.uk/what-is-bandwidth>

[3]*Speedtest by Ookla: Speedtest* [online]. [cit. 2018-03-30]. Dostupné z: <http://www.speedtest.net/>