

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ  
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION  
DEPARTMENT OF TELECOMMUNICATION

MĚŘENÍ PARAMETRŮ DATOVÝCH PŘENOSŮ V  
POČÍTAČOVÝCH SÍTÍCH

DIPLOMOVÁ PRÁCE  
MASTER'S THESIS

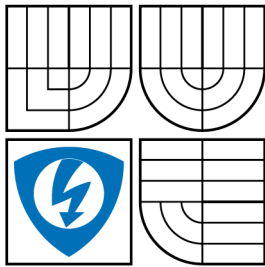
AUTOR PRÁCE  
AUTHOR

BC. PAVEL ŠEDA

BRNO 2008



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY  
A KOMUNIKAČNÍCH TECHNOLOGIÍ  
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND  
COMMUNICATION  
DEPARTMENT OF TELECOMMUNICATION

MĚŘENÍ PARAMETRŮ DATOVÝCH PŘENOSŮ V  
POČÍTAČOVÝCH SÍTÍCH  
TITLE OF STUDENT'S THESIS

DIPLOMOVÁ PRÁCE  
MASTER'S THESIS

AUTOR PRÁCE  
AUTHOR

BC. PAVEL ŠEDA

VEDOUCÍ PRÁCE  
SUPERVISOR

ING. MICHAL SKOŘEPA

BRNO 2008

ZDE VLOŽIT LIST ZADÁNÍ

Z důvodu správného číslování stránek

ZDE VLOŽIT PRVNÍ LIST LICENČNÍ  
SMOUVY

Z důvodu správného číslování stránek

ZDE VLOŽIT DRUHÝ LIST LICENČNÍ  
SMOUVY

Z důvodu správného číslování stránek

## **ABSTRAKT**

Diplomová práce je souhrně zaměřena na základní parametry datových přenosů počítačových sítí. V první části jsem přehledně popsal základní parametry sítí, způsob jakým se dané parametry měří a zohledňují. Zaměřil jsem se na pevné počítačové sítě LAN, a také na bezdrátové sítě Wifi. V druhé části jsem realizoval vlastní návrh měření těchto parametrů. Tento návrh jsem aplikoval při vývoji SW aplikace. Aplikace má za úkol měřit zpoždění přenášených paketů, jitter, ztrátovost a dostupnou šířku pásma. V závěru práce jsem uvedl provedená měření spolu s podrobným popisem SW aplikace.

## **KLÍČOVÁ SLOVA**

UDP, zpoždění, jitter, ztrátovost, šířka pásma, IPv4, IPv6.

## **ABSTRACT**

Master's thesis is focused on basic characteristics of data transmissions in computer networks. In forepart I digestedly described basic characteristics computer networks, methods of their measurement and valuation. Computer networks type LAN as well as wireless computer networks Wifi, are mentioned. In second part I realized my own proposition of basic characteristics measurement. This proposition is used for SW applications development. Main goal of these applications is measurement of packet delay, packet jitter, packet loss and available bandwidth. I described details of SW applications and realized measurement of real computer networks in the end.

## **KEYWORDS**

UDP, delay, jitter, loss, bandwidth, IPv4, IPv6.

ŠEDA P. *Měření parametrů datových přenosů v počítačových sítích*. Místo: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2008. 70s. Diplomová práce. Vedoucí práce byl Ing. Michal Skořepa.

## PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Měření parametrů datových přenosů v počítačových sítích“ jsem vypracoval samostatně pod vedením vedoucího diplomové 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 diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne .....

.....

(podpis autora)

Sem lze vložit poděkování

Není však nutné

# OBSAH

<b>1 Úvod</b>	<b>14</b>
Úvod	14
<b>2 QoS</b>	<b>15</b>
2.1 Definice QoS . . . . .	15
2.2 Koncept QoS . . . . .	15
2.3 Identifikace a označení . . . . .	15
2.4 QoS síťového segmentu . . . . .	16
2.4.1 Managment povolovacího okénka . . . . .	16
2.4.2 Managment front . . . . .	16
2.4.3 Výkonnost spoje . . . . .	17
2.5 Dohled a utváření provozu . . . . .	17
<b>3 Základní parametry počítačové sítě</b>	<b>18</b>
3.1 Šířka pásma . . . . .	18
3.1.1 Měření dostupné šířky pásma . . . . .	18
3.1.2 Iterační algoritmus měření A . . . . .	19
3.2 Zpoždění paketů a jitter . . . . .	21
3.2.1 Definice zpoždění paketů . . . . .	21
3.2.2 Měření RTT pomocí UDP paketů . . . . .	22
3.2.3 Definice jitter . . . . .	23
3.2.4 Měření jitter pomocí UDP paketů . . . . .	23
<b>4 Další nástroje pro měření parametrů sítí</b>	<b>26</b>
4.1 Netperf . . . . .	26
4.2 Iperf . . . . .	26
4.3 Pathrate . . . . .	26
4.4 Pathload . . . . .	26
4.5 Pipechar . . . . .	27
4.6 WEB 100 . . . . .	27
<b>5 Bezdrátová síť</b>	<b>30</b>
5.1 Měření obecně rozdělit do tří částí . . . . .	30
<b>6 Vlastní řešení měření</b>	<b>32</b>
6.1 Měřené parametry . . . . .	32
6.2 Protokol . . . . .	32

6.3	Návrh . . . . .	33
<b>7</b>	<b>Realizace SW</b>	<b>34</b>
7.1	Úpravy návrhu měření . . . . .	34
7.2	Serverová aplikace . . . . .	36
7.2.1	Kategorie měření . . . . .	36
7.2.2	Adresy klientských stanic . . . . .	38
7.2.3	IPv4 vs. IPv6 . . . . .	39
7.2.4	Parametry měření . . . . .	39
7.2.5	Zobrazování dat . . . . .	40
7.2.6	Funkce, spuštění a uložení . . . . .	41
7.3	Klientská aplikace . . . . .	42
7.3.1	Funkce . . . . .	42
7.4	Podmínky použití . . . . .	43
7.5	Vývojové diagramy aplikací . . . . .	43
7.5.1	Serverová aplikace . . . . .	43
7.5.2	Zpoždění paketů . . . . .	44
7.5.3	Jitter paketů . . . . .	44
7.5.4	Ztrátovost paketů . . . . .	44
7.5.5	Časovače řídicí běh měření a zobrazování . . . . .	44
7.5.6	Dostupná šířka pásma . . . . .	49
7.5.7	Klientská aplikace . . . . .	49
<b>8</b>	<b>Měření parametrů sítě</b>	<b>53</b>
8.1	Měření parametrů sítě protokolem IPv4 . . . . .	53
8.2	Měření parametrů sítě protokolem IPv6 . . . . .	62
<b>9</b>	<b>Závěr</b>	<b>65</b>
	<b>Literatura</b>	<b>67</b>
	<b>Seznam symbolů, veličin a zkratk</b>	<b>68</b>

# SEZNAM OBRÁZKŮ

2.1	Základní implementace QoS . . . . .	16
3.1	Jednosměrné zpoždění přenosu, pro $R > A$ . . . . .	20
3.2	Jednosměrné zpoždění přenosu, pro $R < A$ . . . . .	20
3.3	Jednosměrné zpoždění přenosu, pro $R \approx A$ . . . . .	20
3.4	Princip NetDyn . . . . .	22
3.5	$RTT_n$ pro $0 \leq n \leq 800$ při $\delta = 50ms$ . . . . .	23
3.6	<i>Jitter</i> pro $0 \leq n \leq 800$ při $\delta = 50ms$ . . . . .	24
3.7	Model provozu měřených paketů a sítě . . . . .	24
3.8	<i>Jitter</i> pro $0 \leq n \leq 800$ při $\delta = 500ms$ . . . . .	25
4.1	Velikost povolovacího okénka v závislosti na čase, při $RTT = 10ms$ . . . . .	28
4.2	Velikost povolovacího okénka v závislosti na čase, při $RTT = 50ms$ . . . . .	28
4.3	Poměr měřené a dosažitelné šířky pásma v závislosti na $RTT$ pro přehled přenosů, 1s po pomalém startu . . . . .	29
4.4	Poměr měřené a dosažitelné šířky pásma v závislosti na $RTT$ pro přehled přenosů, 10s po pomalém startu . . . . .	29
6.1	Model UDP paketu . . . . .	32
7.1	Použitý UDP paket . . . . .	34
7.2	Schéma komunikace klient-server . . . . .	35
7.3	Serverová aplikace . . . . .	36
7.4	Kategorie měřených parametrů . . . . .	37
7.5	IP adresy klientských stanic . . . . .	38
7.6	Parametry měření . . . . .	40
7.7	Zpoždění paketu . . . . .	41
7.8	Grafické nástroje . . . . .	42
7.9	Formulář zadávání IPv6 adresy klientské stanice . . . . .	43
7.10	Vývojový diagram serverové aplikace . . . . .	45
7.11	Vývojový diagram funkce zpoždění paketu . . . . .	46
7.12	Vývojový diagram funkce jitter paketu . . . . .	47
7.13	Vývojový diagram funkce ztrátovost paketu . . . . .	48
7.14	Vývojový diagram funkce časovačů serverové aplikace . . . . .	50
7.15	Vývojový diagram funkce dostupné šířky pásma . . . . .	51
7.16	Vývojový diagram funkce klientské aplikace . . . . .	52
8.1	Ověření dostupnosti klientských stanic . . . . .	53
8.2	Zpoždění paketů stanice číslo 1. po dobu 5s . . . . .	54
8.3	Zpoždění paketů stanice číslo 1. po dobu 10s . . . . .	54
8.4	Zpoždění paketů stanice číslo 2. po dobu 5s . . . . .	55
8.5	Zpoždění paketů stanice číslo 2. po dobu 10s . . . . .	56

8.6	Jitter paketů stanice číslo 1. po dobu 5s . . . . .	56
8.7	Jitter paketů stanice číslo 1. po dobu 10s . . . . .	57
8.8	Jitter paketů stanice číslo 2. po dobu 5s . . . . .	57
8.9	Jitter paketů stanice číslo 2. po dobu 10s . . . . .	57
8.10	Ztrátovost paketů stanice číslo 1. po dobu 5s . . . . .	58
8.11	Ztrátovost paketů stanice číslo 1. po dobu 10s . . . . .	58
8.12	Ztrátovost paketů stanice číslo 2. po dobu 5s . . . . .	59
8.13	Ztrátovost paketů stanice číslo 2. po dobu 10s . . . . .	59
8.14	Dostupná šířka pásma stanice číslo 1. . . . .	60
8.15	Dostupná šířka pásma stanice číslo 2. . . . .	61
8.16	Dostupná šířka pásma stanice přes WIFI spoj . . . . .	61
8.17	Zpoždění paketů stanice číslo 1. po dobu 5s . . . . .	62
8.18	Zpoždění paketů stanice číslo 2. po dobu 5s . . . . .	62
8.19	Jitter paketů stanice číslo 1. po dobu 5s . . . . .	63
8.20	Jitter paketů stanice číslo 2. po dobu 5s . . . . .	63
8.21	Ztrátovost paketů stanice číslo 1. po dobu 5s . . . . .	63
8.22	Ztrátovost paketů stanice číslo 2. po dobu 5s . . . . .	64
8.23	Dostupná šířka pásma stanice číslo 1. . . . .	64
8.24	Dostupná šířka pásma stanice číslo 2. . . . .	64

# SEZNAM TABULEK

7.1 Schéma IP portů . . . . .	40
-------------------------------	----

# 1 ÚVOD

Vlastnosti počítačových sítí se v průběhu času mění a správa takových to sítí si vyžádá spoustu času pokud nemáme představu co se v naší síti právě odehrává. Nejjednodušší cestou jak se dozvědět maximum o stavu sítě je provést sérii měření a testů, při kterých se zaměříme na parametry popisující vlastnosti této sítě.

Mezi základní parametry počítačové sítě patří zpoždění přenášených paketů, jitter paketů, ztrátovost a také dostupná šířka pásma. Tento souhrn nám osvětlí představu, například jak je síť vytížená, zda komunikující uzly nejsou zahlceny nepotřebným provozem, nebo zda některý ze směrovacích prvků sítě není poškozen a nezahazuje pakety bez zjevného důvodu. Zmiňované parametry jsou hlavním určujícím faktorem pro kvalitu poskytovaných služeb v síti a jejich kontrola je rozhodující jednak při řešení zmíněných problému, ale také při rozdělování dostupných prostředků sítě. Služba QoS vypovídá o schopnosti sítě poskytovat lepší služby vybraným síťovým spojům pomocí různých technologií, které se nabízejí.

Přehlednost a dostupnost síťových prostředků je rozhodující pro správný chod sítě a dostupnost všech jejich uživatelských stanic. Z tohoto důvodu jsem se zaměřil na definici a určení stavu těchto základních parametrů. Hlavní částí tohoto dokumentu je popis realizované SW aplikace, měřící základní parametry sítě. Ovšem ještě před samotným popisem je první část dokumentu zaměřena na charakteristické vlastnosti sítí, specifikaci QoS a další měřící nástroje sítí. Specifikoval jsem zpoždění přenášených paketů, jitter paketů, ztrátovost a také dostupnou šířku přenosového pásma. Mezi nástroje umožňující měření síťových parametrů jsem se zaměřil na Iperf, Netperf, WEB100 a další.

Vzhledem k velkému nárůstu bezdrátových sítí, jsem se zaměřil také na síť Wifi a způsoby měření těchto sítí, které mají svá specifika. V hlavní části dokumentu jsem provedl návrh vlastního způsobu měření základních parametrů datových přenosů počítačovou sítí. Tento návrh jsem použil při realizaci vlastní SW aplikace, která popsané parametry dokáže měřit a patřičným způsobem vyhodnocovat. Aplikační SW byl vyvinut v programovacím prostředí Visual Studio 2005 a komunikace probíhá formou klient-server. Pro nasazení na reálných sítí je program schopen měřit stanice komunikující protokolem IPv4 tak IPv6. Aplikace byli testovány jak na pevné síti LAN, tak i na bezdrátové síti Wifi, pro širší nasazení v praxi.

## 2 QoS

### 2.1 Definice QoS

QoS (kvalita služeb – Quality of Services) vypovídá o schopnosti sítě poskytovat lepší služby vybraným síťovým spojům pomocí různých technologií jako např. ATM, Frame Relay, Ethernet, Sonet atd. Hlavním cílem QoS je poskytnout prioritu přidělené šířce pásma, kontrolu jitteru a zpoždění (pro přenosy v reálném čase) a upravené charakteristiky ztrátovosti paketů. Důležitou součástí je také zabezpečení, aby přiřazená priorita jednomu datovému toku nezapříčinila kolaps čili ztrátu jiného datového toku.

### 2.2 Koncept QoS

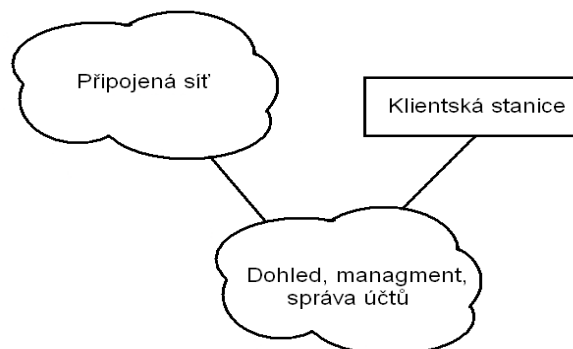
Podstata QoS spočívá ve změně priorit datových toků, zvýšení priorit jednoho a omezení priorit ostatních toků. Jestliže se používá povolovacích nástrojů je možné pomocí obsluhy nebo zpoždování front měnit priority konkrétních toků. Řízení procesů front pro vyrovnání se s rostoucími prioritami používá postup zahazování dat s nižší prioritou na úkor dat s prioritou vyšší. Dohled a formování poskytuje prioritu datovému toku omezením propustnosti toků ostatních. Řízení QoS může pomoci řešit zahlcující problémy v síti. Ovšem častokrát je příliš mnoho požadavků ke spojení vzhledem k dostupné šířce pásma.

Základní prvky jsou tři.

1. Technika označení a identifikace QoS pro řízení přenosu bod-bod mezi dvěma uzly.
2. QoS uvnitř jednoho uzlu sítě (harmonogram přenosů, řazení do front, nástroje pro utváření provozu).
3. Management dohledu, funkce účtů ke správě provozu v síti mezi uživateli.

### 2.3 Identifikace a označení

K poskytování preferenčních služeb různým typům provozu je nutné v první řadě určitým způsobem identifikovat. Pakety mohou a nebo nemohou být označovány. Jestliže pakety nemohou být označeny, jsou klasifikovány při každém hopu znovu. Takováto klasifikace náleží pouze aktuálnímu uzlu. Jestliže jsou pakety označeny, tak pro celou síť jednotně v poli IP (internet protokol – Internet Protocol) priority.



Obr. 2.1: Základní implementace QoS

Stejné metody identifikace obsahuje ACL (seznam oprávněného přístupu – Access Control List), dohledové přepínání, CAR (angažovaná míra přístupu – Committed Access Rate) a NBAR (síťové aplikace rozpoznávání – Network-Based Application Recognition).

## 2.4 QoS síťového segmentu

### 2.4.1 Management povolovacího okénka

Bloková povaha hlasového/obrazového/datového přenosu, někdy přesahuje množství provozu a rychlost spoje. Daný segment musí řešit podobné problémy. Obsahuje nástroje řízení, jako PQ (prioritní fronta – Priority Queuing), CQ (účastnická fronta – Custom Queuing), WFQ (přímá váhovací fronta – Weighted Fair Queuing) a CBWFQ (klasifikovaná přímá váhovací fronta – Class-Based Weighted Fair Queuing).

### 2.4.2 Management front

Protože fronty nemají nekonečnou délku, může se stát, že se naplní nebo dokonce přetečou. Jestliže je fronta plná a přijde další paket, nebude přijat a řízení jej zahodí pryč. Problém s tímto koncovým zahazováním paketů spočívá v tom, že směrovač není schopen zajistit, aby nebyly pakety tímto způsobem zahazovány (dokonce i pakety s vyšší prioritou.)

Management musí zabezpečit dva základní mechanismy.

1. Zajistit, aby se fronta nenaplnila natolik, aby bylo nutné zahazovat pakety. Vytvořit prostor pro pakety s vyšší prioritou.

2. Poskytovat nějaké druhy kritérií pro zahazování paketů s nižší prioritou před zahazováním paketů s vyšší prioritou.

Mechanismus WRED (prvotní náhodná detekce – Weighted Early Random Detect) poskytuje zmíněné mechanismy.

### 2.4.3 Výkonnost spoje

Častokrát pomalé linky jsou představovány jako záležitost pouze pro menší pakety. Například sériové (frontovací) zpoždění paketu 1500B na 56Kb/s lince dosahuje až 214ms. Jestliže se hlasový paket dostane za tento velký paket, celkový součet zpoždění přesáhne tuto hodnotu ještě dříve než paket opustí směrovač. Fragmentace spoje a prokládání umožňuje tento velký paket rozdělit a proložit hlasovými pakety. Prokládání je stejně důležité jako fragmentace. Není důvod fragmentovat paket a posílat hlasový paket za všemi fragmentovanými pakety.

## 2.5 Dohled a utváření provozu

Utváření je využíváno k vytvoření datového toku takového, který dokáže využít plného potenciálu šířky pásma. Mnoho sítí využívajících Frame relay v sítích topologie Hub&Spoke využívá utváření pro řešení problémů s přetečením bufferů. Centrální síť má širokopásmové spoje v porovnání se vzdálenými sítěmi, které mají úzkopásmové spoje (např. 384Kb/s). V tomto případě je možné, že při provozu z centrální sítě směrem k úzkopásmové dojde k přetečení bufferů v této vzdálené síti. Management utváření provozu je tím pravým nástrojem jak přiblížit provoz k 384Kb/s a tím se vyhnout přetečení. Přenos takto konfigurován, je ukládán pro pozdější přenos a podporu konfigurace.

Management dohledu je podobným nástrojem správy přenosu, ale provoz, který přesahuje konfigurovanou mez není ukládán, ale je přímo odpojen. Problematika QoS je podrobněji rozebrána v prameni[1].

## 3 ZÁKLADNÍ PARAMETRY POČÍTAČOVÉ SÍTĚ

### 3.1 Šířka pásma

Dostupnost šířky přenosového pásma byla vždy kritickým parametrem kapacity přenosu, směrování dat, modelování, QoS, streamování videa (audia) a mnoha dalších oblastí. Definujme si co to vlastně šířka pásma je. Cesta P v dané síti je sled jednotlivých spojení H přenášejících pakety od vysílací k přijímací straně. Za předpokladu, neměnnosti dané cesty, směrovacích tabulek. Každý spoj  $i$  může přenášet data rychlostí  $C_i$  bit/s, definované jako kapacita spoje. Dostupná šířka pásma na dané trase P je kapacita definována jako suma.  $C = \min C_i$  pro  $i=1..H$  což je maximální hodnota dostupná pro daný tok na cestě P. Šířka pásma na dané cestě je A. Za předpokladu, že linka  $i$  umožňuje přenést  $C_i u_i(t_0, t_0 + \tau)$  bitů v časovém intervalu  $(t_0, t_0 + \tau)$ .

Funkce  $u_i(t_0, t_0 + \tau)$ , nebo také  $u_i^\tau(t_0)$  je průměrný užitek spoje  $i$  v intervalu  $(t_0, t_0 + \tau)$ , kde tato fce nabývá hodnoty  $0 \leq u_i^\tau(t_0) \leq 1$ . Dostupná šířka pásma  $A_i^\tau(t_0)$  spoje  $i$  v intervalu  $(t_0, t_0 + \tau)$  může být definována zlomek kapacity spoje, která nebyla využita v daném intervalu.

$$A_i^\tau(t_0) = C_i[1 - u_i^\tau(t_0)] \quad (3.1)$$

pro  $i=1..H$

V podobném smyslu ovšem pro celou cestu, která je dostupná přenosu dat. Dostupná šířka pásma  $A^\tau(t_0)$  v intervalu  $(t_0, t_0 + \tau)$  je minimální dostupná šířka pásma ve všech spojích dané cesty P

$$A^\tau(t_0) = \min C_i[1 - u_i^\tau(t_0)] \quad (3.2)$$

pro  $i=1..H$ .

Tudíž je dostupná šířka pásma definována jako maximální rychlost dané cesty, která k dispozici pro přenos bez omezení ostatních přenosů na dané cestě. Pro přesnější definice je nutné uvést dva nové výrazy.

Omezené spojení - je to spojení s minimální kapacitou a je určujícím pro kapacitu dané cesty. Pevné spojení - je spojení s minimální dostupnou šířkou pásma a je určujícím pro dostupnou šířku pásma dané cesty.

#### 3.1.1 Měření dostupné šířky pásma

Jedna z možných metod měření je SLoPS (opakované přenášení dat – Self-loading Periodic Streams). Princip spočívá ve vzrůstajícím zpoždění periodického přenosu paketů, jestliže je přenos vyšší než dostupná šířka pásma. Periodický přenos paketů

je složen z  $K$  paketů o délce  $L$  vyslaných po dané cestě rychlostí  $R$ . Tento přenos  $R$  je vyšší než dostupná šířka pásma  $A$ , potom jednosměrné zpoždění přijatých paketů je vzrůstající. Jestliže předpokládáme neměnnost v čase a plynulý provoz na spoji  $i$ , dostupná šířka pásma je  $A_i$  a vytíženost spoje

$$u_i = (C_i - A_i)/C_i \quad (3.3)$$

a provoz je  $u_i C_i \tau$  bitů mezi vysílačem a přijímačem na spoji  $i$  v intervalu  $\tau$ . Pakety jsou odesílány s periodou  $T = L/R_0$ . Pro měření je využito možnosti porovnání dynamiky jednosměrného zpoždění při opakujícím se přenosu. Jednosměrné zpoždění  $D^k$  mezi vysílací a přijímací částí pro paket  $k$  je dáno jako

$$D^k = \sum_{i=1}^H \left( \frac{L}{C_i} + \frac{q_i^k}{C_i} \right) = \sum_{i=1}^H \left( \frac{L}{C_i} + d_i^k \right), \quad (3.4)$$

kde  $q_i^k$  je velikost fronty spoje  $i$  při příchodu paketu  $k$  (bez paketu  $k$ ), a  $d_i^k = q_i^k/C_i$  je zpoždění paketu  $k$  vznikající při čekání ve frontě spojení  $i$ . Rozdíl mezi za sebou příchozími pakety  $k$  a  $k+1$  je

$$\Delta D_k \equiv D_{k+1} - D_k = \sum_{i=1}^H \frac{\Delta q_i^k}{C_i} = \sum_{i=1}^H \Delta d_i^k, \quad (3.5)$$

kde  $\Delta q_i^k = q_i^{k+1} - q_i^k$  a  $\Delta d_i^k = \Delta q_i^k/C_i$ .

Návrhy měření:

1. Jestliže  $R_0 > A$  potom  $\Delta D^k > 0$  pro  $k=1, \dots, K-1$ . V opačném případě, kdy  $R_0 \leq A$  potom  $\Delta D^k = 0$  pro  $k=1, \dots, K-1$ .

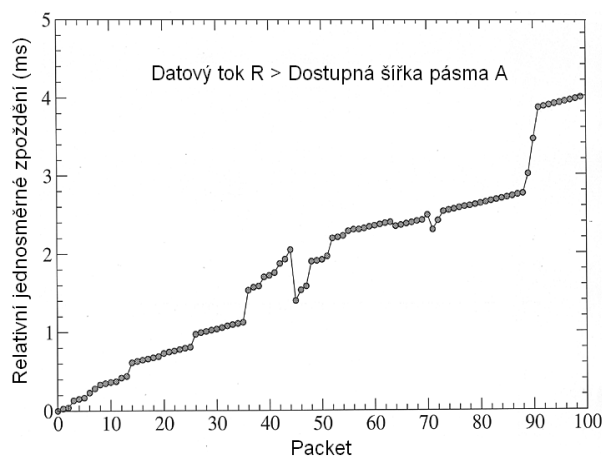
Dostupná šířka pásma může být spočítána z přenosu paketů přijatých v přijímači. Tento postup je také uplatňován v technice řízeného paketového rozptylu.

2. Rychlost přenosu  $R_H$  na přijímací straně je všeobecně funkcí kapacity spoje  $C_i$  a šířkou pásma  $A_i$  pro všechna  $i=1, \dots, H$ .

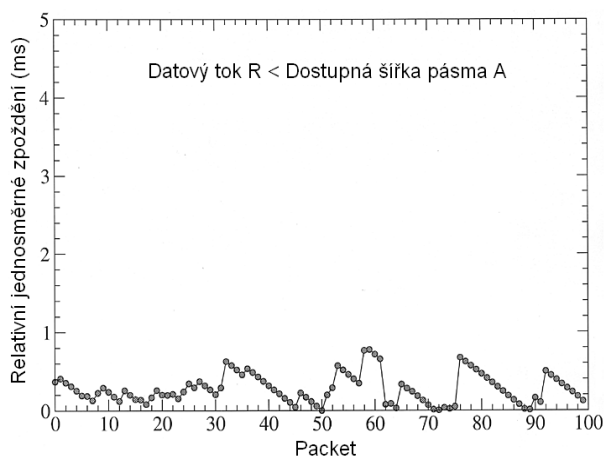
### 3.1.2 Iterační algoritmus měření A

Algoritmus vychází z návrhu číslo 1. a předpokládá, že vysílací strana vyšle  $n$  opakujících se přenosů paketů s rychlostí  $R(n)$ . Princip poměrně názorně vystihují obrázky 3.1, 3.1 a v neposlední řadě také obrázek 3.3, který představuje jakýsi kompromis oproti předchozím dvěma variantám.

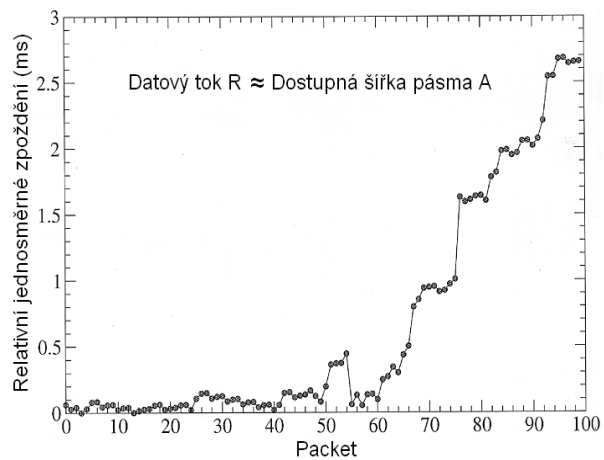
Následně přijímací strana analyzuje změnu jednosměrného zpoždění a dle návrhu určí, zda platí, že  $R(n) > A$  a nebo naopak. Následně přijímací strana oznámí výsledek relace mezi  $R(n)$  a  $A$ .



Obr. 3.1: Jednosměrné zpoždění přenosu, pro  $R > A$



Obr. 3.2: Jednosměrné zpoždění přenosu, pro  $R < A$



Obr. 3.3: Jednosměrné zpoždění přenosu, pro  $R \approx A$

Jestliže  $R(n) > AR_{max} = R(n)$ , a také  $R(n) \leq AR_{min} = R(n)$ ,

$$R(n + 1) = (R_{max} + R_{min})/2. \quad (3.6)$$

Jestliže je  $R(n) > A$ , vysílací strana vyšle další sled paketů  $n+1$  s rychlostí  $R(n+1) < R(n)$ . V opačném případě bude rychlost přenosu paketů  $n+1$   $R(n+1) > R(n)$ .  $R_{min}$  a  $R_{max}$  jsou spodní a horní hranice pro dostupnou šířku pásma. Rozsáhlé pojetí zaujímá literatura [2], která byla výchozím zdrojem, pro tuto část přehledu měření.

## 3.2 Zpoždění paketů a jitter

### 3.2.1 Definice zpoždění paketů

Zpoždění paketů můžeme definovat jako čas, který uplyne mezi dobou odeslání paketu  $n$  ze zdroje vysílání a dobou přijetí na přijímači paketů. Rozdělujeme dva hlavní druhy zpoždění, zpoždění jednosměrné a zpoždění obousměrné. Zpoždění jednosměrné odpovídá ve své podstatě předešlé definici tzn. čas odeslání paketu zdrojem vysílání a časem jeho přijetí na straně přijímače paketu. Obousměrné zpoždění je také nazýváno jako RTT (oboustranné zpoždění – Round Trip Time), které zahrnuje dobu cesty k přijímači i dobu cesty od přijímače tak i dobu zpracování na přijímači. Obousměrné zpoždění se používá častěji, protože jej zle měřit z jednoho místa.

Zpoždění sítě je klíčovým parametrem definovaným poskytovatelem internetového připojení ISP (poskytovatel služby Internet – Internet Service Provider)[3]. Při zvyšování provozu daného ISP internetových obchodů a streamování videí se definované zpoždění stává dosti hlídaným parametrem. Zjišťování zpoždění pomocí příkazu *ping* je dosti nepřesné a to s toho důvodu, že *ping* využívá protokolu ICMP (kontrolní protokol zpráv internetu – Internet Control Message Protokol) a konkrétně dva typy rámců typ 0 (Echo request) a typ 8 (Echo reply). Při odeslání uloží do paketu časový údaj o době odeslání a cílový uzel udělá to samé při příjmu tohoto paketu a odešle nově vytvořený paket typu 8 nazpět. Výchozí uzel data zpracuje a vyhodnotí. Nastávají tu dva problémy, a to synchronizace systémových hodin daných uzlů a také čas pro zpracování na cílovém uzlu, který v některých případech může být značný vzhledem k prioritám paketů protokolu ICMP, které nepřevyšují prioritu například datových paketů TCP (Transmission Control Protocol) a UDP (User Datagram Protocol). Proto se spíše *ping* používá k zjišťování dostupnosti uzlu než ke zpoždění.[3]

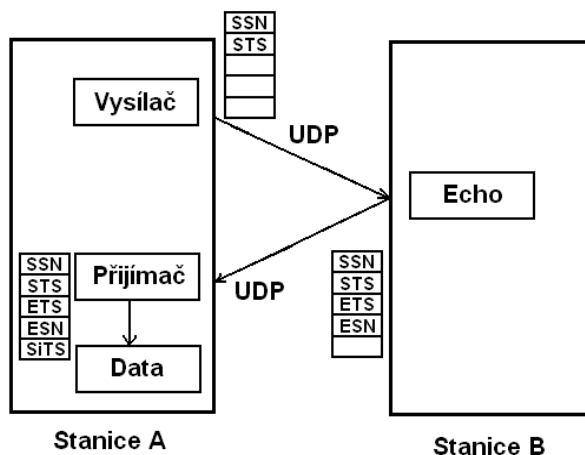
### 3.2.2 Měření RTT pomocí UDP paketů

Měření je realizováno pomocí krátkých UDP paketů a zpětné echo vazby na zdroj vysílání, které se často používají pro rychlejší přenos dat, vzhledem k jejich vlastnostem, a právě jejich RTT. Pakety jsou zasílány v pravidelných intervalech. Při vyhodnocování výsledků se klade hlavní důraz na zpoždění paketů a jejich ztrátovost v prostředí reálné sítě. Díky měnícím se intervalům mezi vysílanými pakety je možné studovat strukturu provozu sítě v různých časových intervalech. Vlastní zpoždění každého paketu vzniká při přenosu od zdroje k jeho cíli v mezilehlých uzlech na této trase. Celkové zpoždění je dáno sumou těchto zpoždění. Každá dílčí část zpoždění se skládá ze dvou hlavních složek. První z nich je fixní zpoždění vzniklé při přenosu mezi mezilehlými uzly a druhou proměnnou část tvoří režie a procesy zpracování, řazení do front atd., které je nutné uskutečnit v každém uzlu.

K analýze paketového zpoždění v počítačových sítích se využívá rozborů zpoždovacích modelů sítí. Na základě těchto rozborů je možné definovat množství parametrů, včetně velikosti vyrovnávacích paměti na směrovačích (aby nedocházelo k jejich přetečení při směrování a tudíž zahazování paketů) a zpoždění v dané síti.

Díky proměnné době odesílání bloku paketů je možné sledovat zpoždění a ztrátovost v různých časových úsecích. V malých intervalech odesílání bloků paketů dochází k shlukování bloků paketů a zvýšenému kolísání zpoždění vznikajícím při řazení do front.

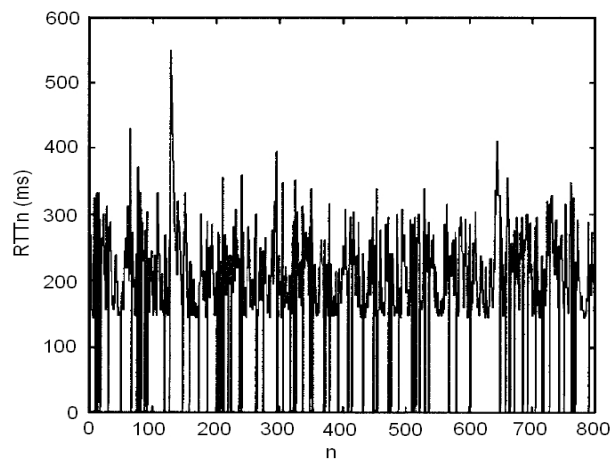
Při konkrétním měření, detailně popsáném ve zdroji [4], je naznačen postup pro měření RTT a dalších parametrů. V tomto měření byla data naměřena na specifické trase udávající vzhledem k dostupné šířce pásma přenosovou rychlost 128Kb/s. Tento pokus byl realizován programem NetDyn. Princip měření je naznačen na následujícím obrázku 3.4. Výhodou tohoto programu, je možnost nastavení množství



Obr. 3.4: Princip NetDyn

paketů, které budou odeslány, velikost paketů a také intervaly mezi odesláním. V případě, který zde uvádím se jedná o pakety velikosti 32B odesílaných v následujících intervalech: 8, 20, 50, 100, 200 a 500ms. Každý paket obsahuje tři 6bajtová pole s hodnotou času na daném uzlu. Při odeslání je do paketu uložen čas odesláním spolu s číslem paketu pro případ ztráty paketu během přenosu. Při přijetí paketu na cílovém uzlu se vloží do paketu čas přijetí a opět číslo paketu. Stejný proces se opakuje při příjmu paketu na vysílacím zdroji a dojde k vyhodnocení.

Nechť  $s_n$  je čas odeslání paketu  $n$  ze zdroje a  $r_n$  je čas přijetí paketu zdrojem nazpět. Měřené  $RTT_n$  pro ztracené pakety je nedefinováno resp. se rovná  $RTT_n = 0$ , v ostatních případech  $RTT_n = r_n - s_n$ . Obrázek 3.5 znázorňuje měřené  $RTT_n$  jako funkci  $n$  v rozsahu  $0 \leq n \leq 800$  při  $\delta = 50ms$ .



Obr. 3.5:  $RTT_n$  pro  $0 \leq n \leq 800$  při  $\delta = 50ms$

Dále je definována doba mezi vysíláním paketů jako  $\delta = s_{n+1} - s_n$  pro všechna  $n$ .

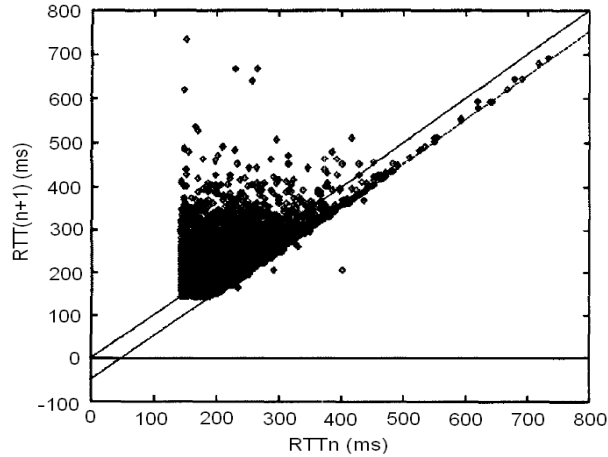
### 3.2.3 Definice jitter

Je rozdíl ve zpoždění  $RTT_n$  pro vzorek  $x = n$  a zpoždění  $RTT_n$  pro vzorek  $x = n+1$ . Jedná se o kolísání zpoždění přenášených paketů.

### 3.2.4 Měření jitter pomocí UDP paketů

Obrázek 3.6 zobrazuje jitter ve stejném rozsahu  $n$  odeslaných paketů se stejným  $\delta = 50ms$ .

Plynulý proud paketů nastane v případě, kdy bude provoz sítě slabý, pakety nebudou moc velké, aby vytěžovaly maximálně buffer. Potom zpoždění vznikající



Obr. 3.6: *Jitter* pro  $0 \leq n \leq 800$  při  $\delta = 50ms$

měřicími pakety je téměř konstantní a můžeme tedy psát

$$w_{n+1} = w_n + \epsilon_n, \quad (3.7)$$

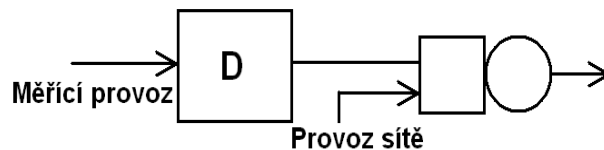
kde  $\epsilon_n$  je náhodný proces s střední hodnotou 0.

Potom

$$RTT_{n+1} = D + w_{n+1} + P/\mu a RTT_n = D + w_n + P/\mu \quad (3.8)$$

$$RTT_{n+1} - RTT_n = \epsilon_n \quad (3.9)$$

Pro měřené pakety dostáváme hodnotu  $D=140ms$ . Nechť  $\mu$  je využití serveru [bit/s] a  $K$  je velikost vyrovnávací paměti. Celkový provoz je sumou měřeného toku dat se zpožděním  $D$  a provozu sítě jak znázorňuje 3.7.



Obr. 3.7: Model provozu měřených paketů a sítě

Definujme si  $P$  jako délku měřeného paketového toku v bitech. Příspěvek síťového provozu si definujme jako  $b_n$  bitů v době mezi přijetím paketů právě od měřeného toku  $P$  v čase mezi  $n$  a  $n+1$ . Definujme si  $w_n$  jako čas čekání na paket  $n$ .

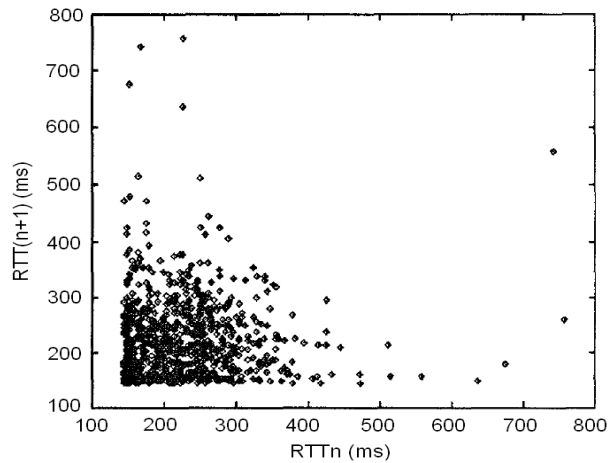
Nechť  $B$  je zatížení provozu sítě vyjádřené v bitech zařazených před měřený paket  $n+1$ . Potom

$$w_{n+1} = w_n + B/\mu \quad (3.10)$$

a také

$$RTT_{n+1} - RTT_n = B/\mu \quad (3.11)$$

V případě, že čekací doba  $w_{n+1} > \delta$  (vysílací interval zdroje) jeden nebo dokonce i více měřených paketů se bude shromažďovat před měřeným paketem  $n+1$  čekající na zpracování síťové zátěže serverem. Nechť  $k$  je počet takových to "čekajících" paketů. Obrázek 3.8 naznačuje, že již vůbec nedochází k hromadění po sobě jdoucích měřených paketů v důsledku vytíženosti serveru.  $x \neq 0$ .



Obr. 3.8: *Jitter* pro  $0 \leq n \leq 800$  při  $\delta = 500ms$

Jestliže nebude přijat žádný další síťový paket budou ze serveru odeslány v pravidelných intervalech definovaných hodnotou  $P/\mu$  s.

$$RTT_{n+3} - RTT_{n+2} = RTT_{n+k} - RTT_{n+k-1} = P/\mu - \delta \quad (3.12)$$

Takto "čekající" pakety se budou nacházet v blízkosti čárkované(jižnější) křivky  $RTT_{n+1} = RTT_n + P/\mu - \delta$  zobrazené také v grafu 3.6 . Výskyt bodů právě na této hranici  $RTT_{n+1} = RTT_n + P/\mu - \delta$  je dokladem toho, že se měřené pakety shromažďují za velkými pakety síťového provozu. V naznačeném případě se jedná o posun o zhruba 48ms oproti křivce  $RTT_{n+1} = RTT_n$ . Potom pro  $\delta = 50ms$  a  $P = 32$  bajtům dostáváme  $\mu = 130kb/s$ .

Nyní se podívejme na případ, kdy bude odesílací interval  $\delta$  mnohem větší. Delta je rovno  $\delta = 500ms$  potom  $\delta - P/\mu = 490ms$ . Pro tento případ jsou hodnoty jitteru naměřené s počtem přenesených paketů  $n = 0-800$  naprosto odlišné od předcházejícího obrázku. [4]

## 4 DALŠÍ NÁSTROJE PRO MĚŘENÍ PARAMETRŮ SÍTÍ

### 4.1 Netperf

Netperf byl vyvinut společností HP (Hewlett-Packard) v devadesátých letech minulého století k měření základních vlastností počítačových sítí. V principu se snaží dostat měřené spojení do saturace, nasytit spoj měřicími pakety. Výsledkem je míra propustnosti dostupná na dané cestě. Netperf je nutné nainstalovat na obou koncích měřené trasy. Netperf pracuje jako aplikace klient-server. Serverová aplikace defaultně poslouchá na portu číslo 12685 ovšem je nutné, aby byla spuštěna jako první z obou dvou. Jestliže klientská aplikace požádá o měření je vytvořeno kontrolní TCP spojení pro přenos konfigurační parametrů a naměřených výsledků. V zápětí je sestaveno měřicí spojení pro měření parametrů spoje. Přenos probíhá z klientské aplikace na aplikaci serverovou. Po ukončení měření jsou naměřená data zaslána klientské aplikaci a zobrazena na obou zúčastněných aplikacích.

### 4.2 Iperf

Iperf byl vyvinut společností NLANR (The National Laboratory for Applied Network Research). Je v podstatě dosti podobný jako Netperf, ovšem nabízí i další funkce jako nastavení povolovacího okénka a měření více spojení. Na rozdíl od FTP spoje, Iperf přenáší data z klientské operační paměti do paměti serveru a nezasahuje na disk stanice.

### 4.3 Pathrate

Pathrate je software pracující na principu rozptylu paketů z kterého následně stanoví odhad kapacity dané cesty. Výsledky měřené touto metodou jsou nezávislé na aktuálním provozu na dané cestě. Pracuje také na principu klient-server, kdy klientská aplikace zasílá měřicí pakety na server. Po ukončení měření server zasílá naměřené hodnoty zpět klientovi.

### 4.4 Pathload

je software, který nenarušuje provoz na síti přehlcování paketů. Je to nástroj k měření zpoždění paketů a také k určení dostupné šířky pásma pro daný spoj. Zjištěná

hodnota je reprezentována minimální a maximální hodnotou.

## 4.5 Pipechar

Pipechar stanovuje odhad kapacity a míru provozu na každém skoku (hopu) na své cestě. Je založen na Hop-by-Hop analýze paketového zpoždění u paketů s proměnou velikostí produkující nemalý provoz na měřené síti. Na rozdíl od předcházejících měřicích nástrojů nepotřebuje k měření serverovou aplikaci. Pipechar hlásí pouze nejpomalejší hopy na své cestě. Porovnání jednotlivých nástrojů můžete získat ve zdroji[5], kde jsou jednotlivé přehledy měření k dispozici.

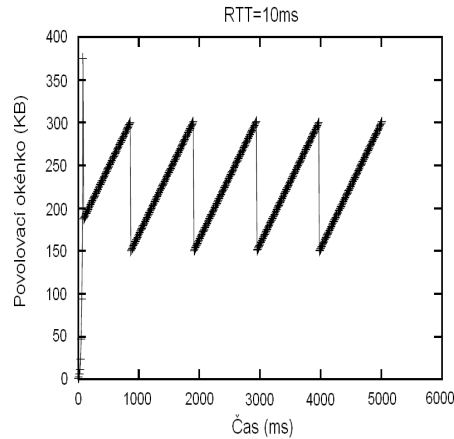
## 4.6 WEB 100

WEB100 je software založený na práci s vendory, umožňující maximální využití přidělené šířky pásma. Je vyladěn pro maximální efektivnost práce v síti, což většina software nenabízí. Umožňuje pružně reagovat na změny parametrů sítě. Měření parametrů, což nás bude nejvíce zajímat je založeno na tzv. Iperf. V principu se program snaží dostat síť do saturace, tedy plného nasycení přenášenými pakety a zjišťuje jaké je možné dosáhnout propustnosti na dané cestě. Detaily naleznete v literatuře[7]. Pracuje jako klient-server aplikace. Je nutné, aby byl server spuštěn jako první, kdy defaultně poslouchá na portu 12685.

Jestliže klient požaduje nové měření je vytvořeno TCP spojení pro přenos výsledků a nastavení měření. Okamžitě po přenosu nastavovacích parametrů, je vytvořeno nové měřicí spojení. Přenos měřicích paketů probíhá od klienta k serveru. Doba měření (rozsah) je nastavitelnou hodnotou stejně jako velikost přenosového okénka. Naměřené výsledky jsou spravovány na serveru a odeslány klientovi. Zobrazení výsledků probíhá na obou stanicích.

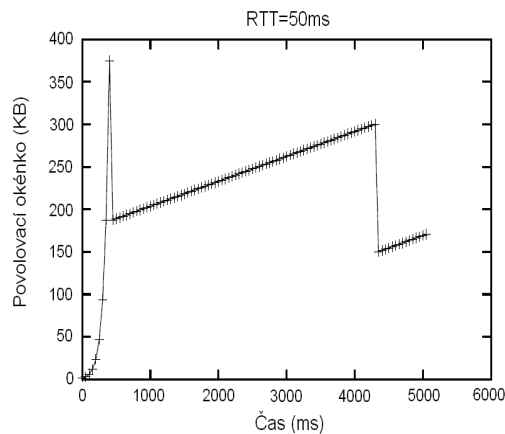
Měřicí aplikace pracují pouze s operační pamětí, tudíž ani na jedné straně není nutné zapisovat nějaká data na disk. Samotné měření pomocí TCP paketů je ovlivněno tzv. TCP pomalým startem ve své podstatě jde o zvyšování velikosti transportního okénka, v závislosti na propustnosti sítě. Na začátku měření je nutné přesně definovat konec pomalého startu transportního okénka,

pro snížení množství měřicích paketů, vysílaných do sítě. Při porovnání obou obrázků 4.1 a 4.2 je patrný vliv pomalého startu transportního okénka. Iperf nejprve vyhodnotí hodnotu průměrného RTT pro dané spojení a také maximální hodnotu datového segmentu TCP - MSS (maximální velikost segmentu – Maximum Segment Size). Hodnota RTT je zjišťována každých 20ms do chvíle než dostaneme správnou hodnotu, stejně tak je sledována hodnota povolovacího okénka. Zpoždění paketů má



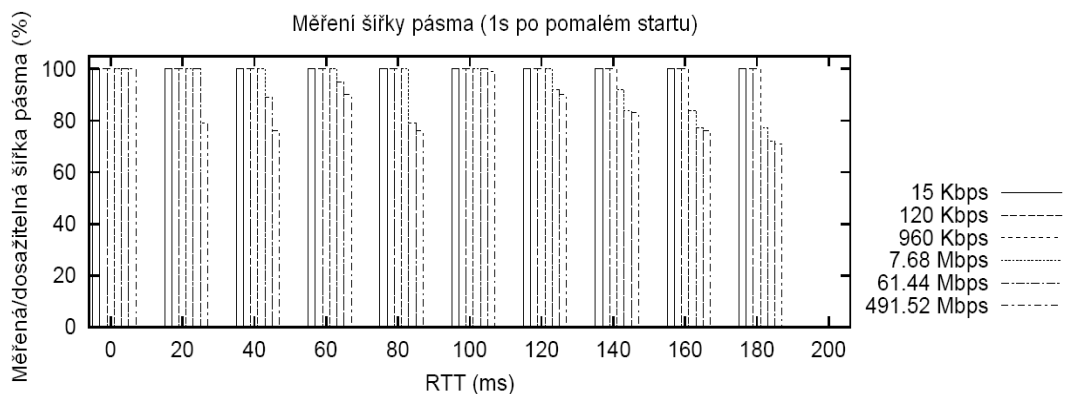
Obr. 4.1: Velikost povolovacího okénka v závislosti na čase, při  $RTT = 10ms$

značný vliv na rychlost navyšování velikosti povolovacího okénka. Na následujícím obrázku je vliv poměrně rozpoznatelný. 4.2 Po stanovení "konečné" správné hodnoty  $RTT$  je ustanovena hodnota okénka v intervalu  $2*RTT$ . Je nutné uložit starou a novou hodnotu povolovacího okénka. Jestliže se spojení bude stále nacházet ve



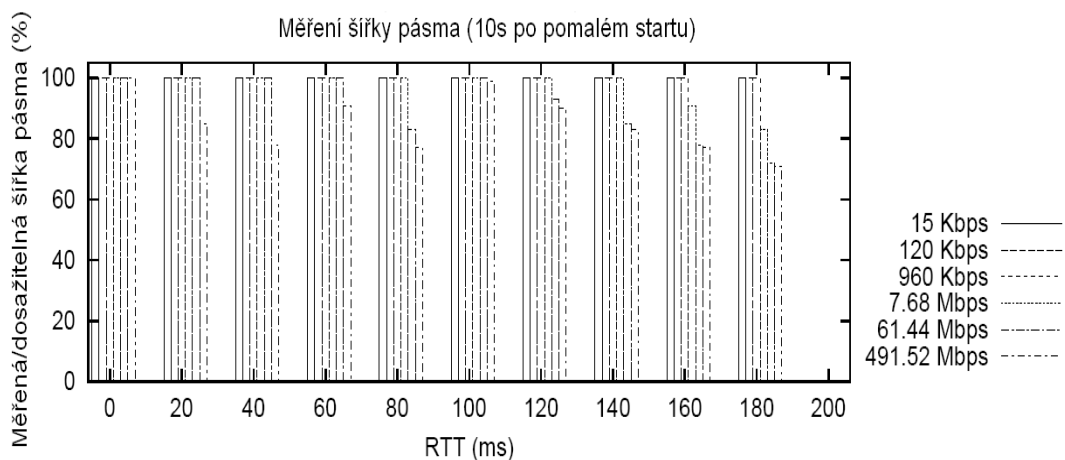
Obr. 4.2: Velikost povolovacího okénka v závislosti na čase, při  $RTT = 50ms$

pomalém startu TCP pravděpodobně zdvojnásobí velikost okna (minimálně však jedenkrát). V opačném případě, již mimo pomalý start povolovací okénko se zvětší na maximálně o  $2*MSS$ . Stará hodnota povolovacího okénka byla  $4*MSS$  a rozdíl oproti současné hodnotě povolovacího okénka je nejméně  $3*MSS$  v intervalu  $2*RTT$  potom se spojení nachází již mimo pomalý start. V měření popsané v [6] je možno najít i další výsledky vyplývající z měření pomocí Iperf a Web100. Poněkud širší náhled na vliv zpoždění  $RTT$  na dostupnost šířky pásma a také důsledky pomalého startu přináší následující obrázky 4.3 a 4.4.



Obr. 4.3: Poměr měřené a dosažitelné šířky pásma v závislosti na RTT pro přehled přenosů, 1s po pomalém startu

Na obrázku 4.3 je vidět jak odesílaná data se zvyšujícím se zpožděním RTT a objemem přenášených dat mají k dispozici ve výsledném poměru méně šířky pásma k přenosu uživatelských dat. Zaviněno je to právě časem realizace přenosu, který je v tomto případě 1s po pomalém startu, se všemi důsledky vlivu pomalého startu.



Obr. 4.4: Poměr měřené a dosažitelné šířky pásma v závislosti na RTT pro přehled přenosů, 10s po pomalém startu

Lepšího poměru měřené/dosažitelné šířky pásma dosahuje přenos "až" po 10s od pomalého startu, kdy jeho vliv již není natolik patrný.

## 5 BEZDRÁTOVÁ SÍŤ

Bezdrátová technologie Wifi je v poslední době poměrně rozšířená. Základem je protokol 802.11b/g. Využití takovýchto sítí je především tam, kde je vyžadována mobilita pracovních stanic a snadné sdílení, dostupnost a jednoduchost řešení.

Hlavním problémem těchto sítí je poměrně malá kapacita sítě sdíleného radiového kanálu. S tím také souvisí snížení výkonu přenosu dat v takovéto síti. Sdílené transportní prostředí a náhodný přístup k médiu, mají při každém přenosu jiné parametry, což má zásadní vliv na paketové zpoždění, ztrátovost a schopnost protokolu transportního maximálně využít dostupné prostředky. Je nutné tyto parametry podrobně sledovat a měřit.

### 5.1 Měření obecně rozdělit do tří částí

1. Měření parametrů fyzické vrstvy referenčního modelu ISO/OSI (Mezinárodní organizace pro normalizaci / propojení otevřených systémů – International Standards Organization / Open System Interconnection). Jde především o kvalitativní parametry fyzické vrstvy daného média v tomto případě bezdrátového spoje (SNR, RSSI, BER)
2. Měření parametrů vyšších přenosových vrstev. S využitím nástrojů pro měření přenosových parametrů transportních protokolů.
3. Měření hodnot v reálném čase. Při stanovení dostupnosti signálu potažmo spoje a jeho stability v čase. S reálným měřením se můžete seznámit v literatuře[9].

Add 1. Nepřímou metodou je možné změření parametrů fyzické vrstvy, díky spektrální analýze samotných rozhraní. Takovéto naměřené parametry si zpracovávají síťové prvky samostatně a výsledky jsou dostupné většinou přes konfigurační rozhraní.

Add 3. Je poměrně dost závislé na konkrétních podmínkách měření v dané situaci a samozřejmě faktorech ovlivňujících měření. Poloha stanice, intenzita signálu, počet stanic v síti a podobně.

Add 2. Je možné využít například protokolu transportní vrstvy TCP pro toto měření. Samotné měření je založeno na komunikaci klient-server, kdy nastavení parametrů a vlastností měření má na starosti server. Odesílá nastavení klientským stanicím. Před samotným přenosem se klientská stanice musí zaregistrovat na serveru po kterém následuje již zmíněné nastavení měřících parametrů v podobě stáhnutí

přenosového profilu. Přenosový profil obsahuje informace o množství datových přenosů  $N_t$  - number of transfers, které bude klient očekávat ze serveru. Množství dat  $D$  přenesených v každém přenosu a časový interval mezi koncem a začátkem dalšího přenosu  $T_{off}$ . Při měření více klientů ve stejném čase, klienti volí pro přístup k serveru náhodný interval. Jako měřicí program je použit program Tstat, který umožňuje měření parametrů TCP/IP provozu. Na základě pasivního sledování provozu v síti, jak přijímaných tak i vysílaných paketů určuje statistiky přenosu a vlastnosti sítě. Je schopen určit až 80 různých statistik včetně velikostí povolovacího okénka, počty opakovaných přenosů atd. Zdroj [8] uvádí podrobnější přehled o využití Tstat při měření parametrů přenosu mezi převážně mobilními stanicemi.

## 6 VLASTNÍ ŘEŠENÍ MĚŘENÍ

### 6.1 Měřené parametry

Po prostudování řady knih jsem se rozhodl realizovat měření tří základních veličin přenosu dat počítačovou sítí. Domnívám se, že mezi ty nejzákladnější patří šířka přenosového pásma, zpoždění paketů při přenosu dat a také ztrátovost paketů. Jak již v předchozím textu je vidět, šířka pásma se měří pomocí datového toku měřících paketů a jejich proměnlivém zpoždění, které vypovídá o dostupné šířce přenosového pásma (Iterační algoritmus měření A 3.1.2).

Pro toto měření je proto nutné stanovit jakého zpoždění dosahují přenášené pakety v rozdílných vysílacích intervalech. Určení posledního parametru, domnívám se parametru zásadního s patřičnou vypovídající hodnotou, je možné na základě opakování vyžádaných přenosů chybějících paketů v důsledku ztrát při jejich přenosu.

### 6.2 Protokol

Jako nejvhodnější protokol jsme zvolil protokol UDP. Je to nespolehlivý protokol, který nevyžaduje potvrzení správného přijetí vysílaných paketů. Jeho výhodou je poskytovaná přenosová rychlost, která se nepotvrzováním zvyšuje oproti spolehlivým protokolům jako je např. TCP.

Jako většina protokolů, tak i UDP využívá kontrolní součty a adresaci cílového portu, pro určení správné cílové aplikace.

bity	0-15	16-31
0	zdrojový port	cílový port
32	délka	CRC
64	čas odeslání packetu	
96	čas příjmu packetu	
128	číslo packetu	
160	data	

Obr. 6.1: Model UDP packetu

## 6.3 Návrh

Vlastní návrh postupu měření je založen na přenosu UDP paketů jak je vidět na obrázku 6.1. Jedná se o komunikaci klient-server v prostředí LAN za pomoci UDP paketů. Pro určení základních parametrů se vychází z údajů získaných z přenosu měřících paketů, které jsou odesílány ze serveru ke klientovy v daných intervalech  $\delta = 0 - 500ms$  a na straně klienta ukládány a dále zpracovávány.

Zvolené prostředí lokální sítě umožní lepší koordinaci spouštění obou aplikací a také i shromažďování naměřených hodnot. Při použití výše uvedeného UDP paketu se při průchodu sítě jistě zpozdí a právě záznamy v paketech o času vyslání a příjmu UDP paketů budou základními hodnotami pro vyhodnocení zpoždění vzniklých při přenosu dat. Odesílání paketů v různých intervalech může dopomoci k získání přesnější představy o frontách plnicích se při přenosu dat, režii nutné na spravování a výsledky zahrnout do celkového zpoždění paketů. Problém zpoždění nadále zůstává způsob synchronizace jednotlivých aplikací při měření. Tento fakt by měla vyřešit právě již zmiňovaná lokální síť.

Určení šířky pásma budu vycházet z Iteračního algoritmu měření A 3.1.2. Pro docílení daného vlivu na síť, budu muset připravit sérii datových paketů, které vystihnou datový tok, který je určujícím pro definici šířky pásma. Měřením vzniklých časových zpoždění budu moci určit, zda daný tok již přesahuje dostupnou šířku pásma anebo nikoli. Rozhodujícím bude schopnost algoritmu aplikace realizovat měření zpoždění paketů, ze kterého budu vycházet.

Ztrátovost paketů, předpokládám bude realizací dodatkovou s přihlédnutím k předcházejícím měřením.

Obě měřící aplikace budou vyvinuty v prostředí Visual Studio 2005 programovacím jazykem C#.

## 7 REALIZACE SW

### 7.1 Úpravy návrhu měření

V průběhu realizace SW aplikace došlo k několika problémům, které vedly ke změnám výše uvedeného návrhu měření. Změny jsou poměrně významné a proto je nutné je blíže rozvést.

1. Změna struktury UDP paketu.
2. Změna časování odesílání měřících paketů.
3. Řešení problému synchronizace stanic.

Add1. V návrhu jsem zmiňoval proměnlivou velikost měřících paketů. Proměnlivost byla účelná pro změnu datového toku pro měření dostupné šířky pásma. Ovšem v průběhu realizace se ukázalo, jako nutné využít maximální velikost paketů. S ohledem na fragmentaci na linkové vrstvě a několika testech reálných přenosů jsem zvolil velikost paketů 1472B. Využitelnou částí je prvních 28B, které obsahují potřebná data k měření parametrů a zbylých 1444B je nutná prázdná výplň (padding) logickými nulami. Strukturu paketu znázorňuje následující obrázek.7.1

bity	0-15	16 16-31	32-47	48-63
0	zdrojový port	cílový port	délka	CRC
64	čas odeslání packetu			
128	čas příjmu packetu			
192	číslo packetu	data	padding	
256 ⋮	padding			
11520	padding			

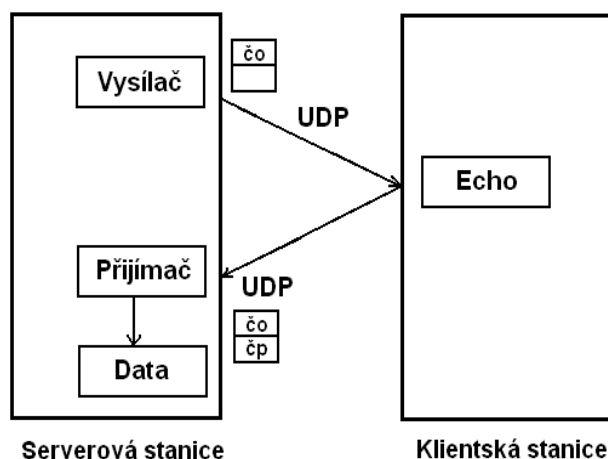
Obr. 7.1: Použitý UDP paket

Jak je na obrázku vidět parametry paketu se dramaticky změnilo oproti verzi návrhu. Protože časové záznamy ukládané do paketů jsou 18-ti místné číselné hodnoty bylo nutné zvětšit velikost pole pro čas odeslání packetu a také čas příjmu packetu na hodnotu 64 bitů. Naproti tomu se pole pro číslo packetu a uživatelská data, zmenšilo na polovinu. Samozřejmě pro dosažení maximální velikosti paketu se struktura rozšířila o pole výplně.

Add2. Drobnější změnou v návrhu je změna časování odesílání paketů. Pro zvýraznění stavu spojení, které je dostupné se ukázalo, že interval odesílání kratší než 40ms se již neprojeví v množství odeslaných měřících paketů => vliv na měřený spoj. Časový rozsah odesílání paketů jsem v aplikaci ponechal trochu větší než jsem naznačoval v návrhu a to 40-1000ms pro odesílání.

Add3. Nejnápadnější změnou, je schéma komunikace klient-server a celkové zpracování měřených dat. Návrh počítal s komunikací realizovanou serverem, kdy vysílané pakety budou zpracovávány na straně klienta, ovšem zmiňovaný problém synchronizace stanic se stal dosti závažným. Můj předpoklad synchronnosti těchto stanic v LAN síti byl chybný, proto jsem schéma komunikace musel změnit, aby tento problém při měření nenastal.

Nové schéma komunikace problém vyřešilo bez vážnějších komplikací. Vysílací stanicí i nadále zůstala serverová stanice, ale změna nastala na straně klientské stanice. Klientská stanice pakety přijme a ihned je odesílá nazpět serverové stanici, která vyhodnocuje naměřená data. Díky tomu, že měřící pakety jsou vysílány a zpracovávány na jedné stanici, problém synchronizace stanic odpadá.7.2



Obr. 7.2: Schéma komunikace klient-server

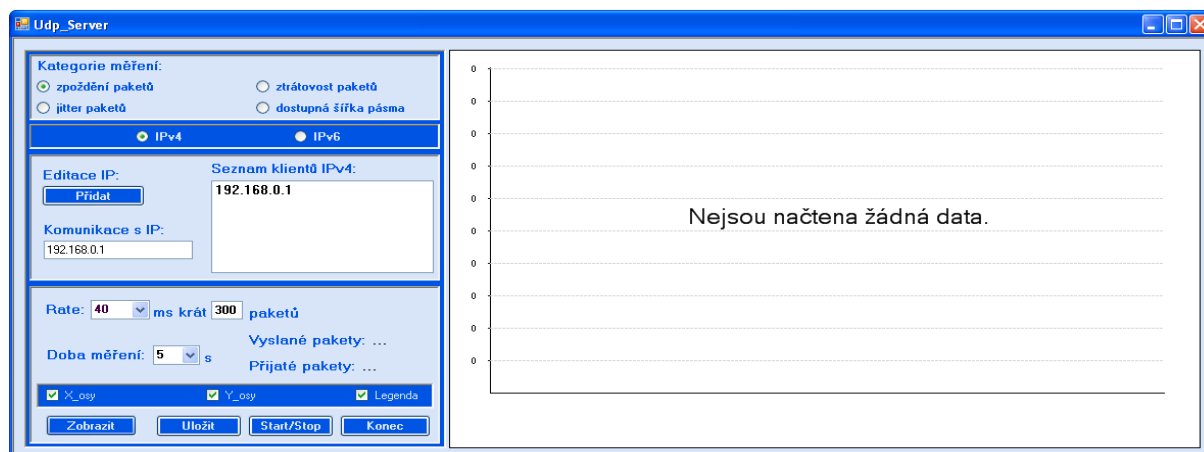
V obrázku pole *čo* představuje čas odeslání měřícího paketu a pole *čp* je čas příjmu měřícího paketu.

Výhodou této koncepce je nízké zatížení klientské stanice také paměťové nároky nedosahují ani 15MB RAM.

V ostatních ohledech jsem se snažil návrhy dodržet a postupovat dle principů popsaných v předcházejících částech tohoto dokumentu. Nadále se budu věnovat již samotným měřícím aplikacím a jejich vlastnostem a funkcím jako měřicímu nástroji. V následujícím textu je uveden popis realizovaných aplikací.

## 7.2 Serverová aplikace

Serverová aplikace se sama o sobě skládá z několika důležitých částí, které je nutné podrobně popsat pro přehlednost a detailní porozumění funkcím aplikace. Začnu samotnými parametry, které se nabízejí k měření. Obrázek 7.3 uvádí serverovou aplikaci v podobě ihned po startu.



Obr. 7.3: Serverová aplikace

### 7.2.1 Kategorie měření

Jak je již zřejmé ze zadání diplomové práce, měřením je za úkol zjistit několik základních parametrů datových přenosů počítačových sítí 7.4.

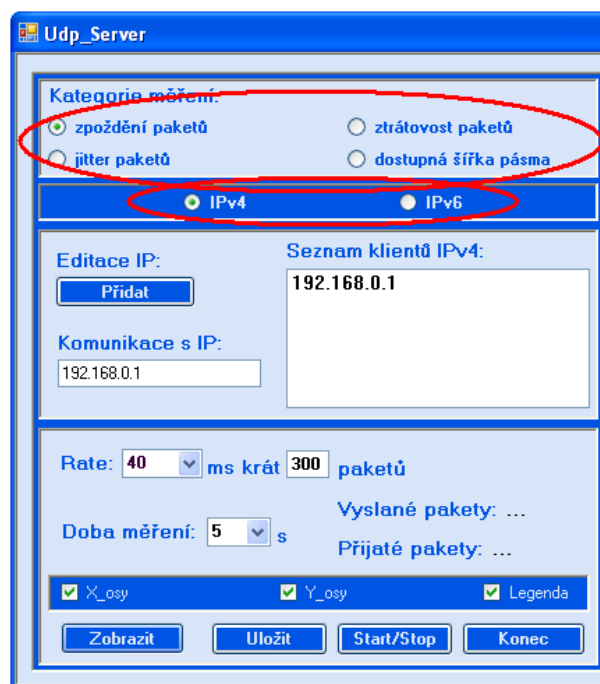
1. Zpoždění paketů.
2. Jitter paketů.
3. Ztrátovost paketů.
4. Dostupná šířka pásma.

Add1. Zpoždění paketů, které vzniká průchodem paketu počítačovou sítí je základním parametrem celého měření. Číselná hodnota zpoždění je získávána ze záznamů časů v přenášených paketech. Hodnota je dána rozdílem mezi časem odeslání paketu a čase příjmu paketu. Časové údaje jsou ukládány a kódovány do 64 bitových polí a po průchodu sítí také dekódovány vždy serverovou aplikací.

Výsledkem tohoto kódování-přenosu-dekódování a odečtení je výsledná hodnota RTT zpoždění paketu. Tyto hodnoty jsou dále zobrazovány v grafu a případně mohou být uloženy pro další zpracování ve formátu \*.csv souboru spolu s parametry

realizovaného měření, za jakých podmínek byly hodnoty naměřeny (počet paketů, doba trvání měření, IP adresa klientské stanice a další).

Add2. Další položkou z měřených parametrů je parametr jitter přenášených paketů. Tento údaj je v blízkém vztahu k výše zmíněnému zpoždění paketů a také z něho vychází. Jedná se o rozdíl zpoždění mezi paketem  $n$  a paketem  $n+1$ . V ideálním případě, kdy předpokládáme konstantní hodnotu zpoždění je tato hodnota rovna nule, ovšem jak již napovídá slůvko "ideálním", tomu v praxi tak není. Při reálném měření je dosahováno bloků konstantních hodnot, vždy když je zpoždění v daném intervalu neměnné. V grafu se taková situace projeví jako přímka nenulových hodnot trvajících dobu stálosti zpoždění.



Obr. 7.4: Kategorie měřených parametrů

Add3. Posledním parametrem, který je měřen souběžně se zpožděním paketů a také jitterem je ztrátovost přenášených paketů. Jedná se o procentuální vyjádření poměru vyslaných a přijatých paketů v danou dobu měření. Interval určující dobu, ve které se mohou do měření započítat opožděné pakety, je stanoven časem odesílání měřících paketů, tudíž 40-1000ms. Zobrazení naměřených hodnot je realizováno sloupcovým grafem a pro přehlednost je v legendě grafu uvedena průměrná hodnota ztrátovosti paketů v průběhu celého měření.

Tyto tři parametry jak jsem již naznačil jsou měřeny současně a také jejich zobrazování v grafu je zcela libovolné a nezávislé na ostatních. K měření těchto parametrů jsou použity pakety dle návrhu tedy pakety velikosti 28B pro zlepšení průchodnosti

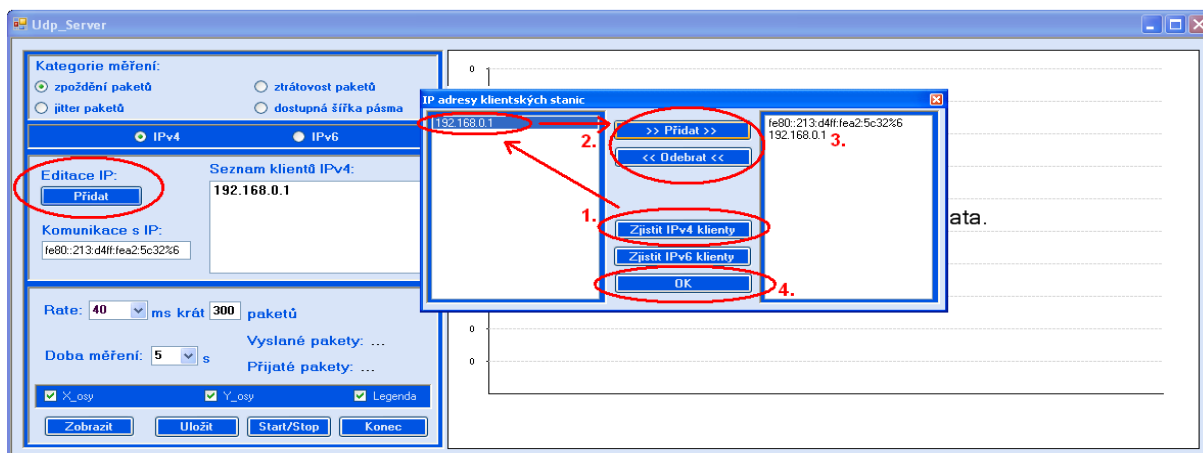
sítí. Poslední parametr tedy dostupná šířka pásma je specifikován datovým tokem, jak je popsáno v návrhu, resp. v části dokumentu předcházející samotnému návrhu.

Add4. Měření dostupné šířky pásma je specifickou záležitostí celé aplikace. Je odděleno od předchozího měření a na uživatelském nastavení je zcela nezávislé. Pro specifikování datového toku, je použito maximální velikosti měřících paketů a to 1472B. Měření je schopno rozlišit dostupnou šířku pásma v intervalu 1Mb/s až 100Mb/s. Jestliže bude dostupná šířka pásma vyšší bude zobrazena hodnotou 100Mb/s. Celková doba tohoto měření je maximálně 10s a stanoví aktuální dostupnou šířku pásma.

## 7.2.2 Adresy klientských stanic

Při návrhu řešení seznamu IP adres klientských stanic jsem vycházel z předpokladu, že obsluha serverové aplikace nezná IP adresy všech stanic na kterých jsou spuštěny klientské aplikace. Některé stanice, nemusí být aktivní jiné mohou být zcela nedostupné (odříznuté od sítě). Proto jsem dal obsluze možnost jak zjistit dostupné a aktivní klientské stanice. Pomocí multicastového paketu, který je vyslán serverovou aplikací do sítě, ke které je připojena. Veškerá editace a shromažďování IP adres klientských stanic začíná tlačítkem

”Přidat” uvozeného nadpisem ”Editace IP:” jak je vidět na obrázku 7.5.



Obr. 7.5: IP adresy klientských stanic

Zobrazené okno nabízí obsluhu aplikace vyslání dvou sady na sobě nezávislých multicastových paketů, které svými parametry osloví pouze stanice aktivní, schopné přijímat měřících paketů daných komunikačních protokolů (IPv4 a IPv6 - podrobnosti budou popsány později). Výhodou multicastového vysílání je, že nezatěžuje tolik přenosovou síť a pakety jsou určeny pouze těm stanicím, které jsou přidány

do dané multicastové skupiny stanic. Rozdíl v protokolu je v podstatě jediným odlišujícím faktorem v tomto vysílání paketů. Po stisknutí tlačítka dojde k odeslání příslušných paketů a klientské aplikace mají 5s na odpověď. Jestliže nepřijde odpověď do této doby, nebude jejich IP adresa zobrazena v levém seznamu. Jak jsem naznačil, pakety přijaté jako odpovědi na multicastové vysílání jsou zpracovány a IP adresy jejich výchozích stanic jsou zobrazeny v levém seznamu, kdy je na obsluhu, zda je vybere a přemístí je do pravého seznamu, tedy seznamu používaného k měření. Po stisknutí tlačítka "OK" se získané adresy nahrají do hlavního okna a obsluha si zvolí jakou klientskou aplikaci využít k měření. K výběru dojde dvoj-klikem na vybranou IP adresu klientské stanice. V případě, že obsluha serverové aplikace zná IP adresu klientské stanice, nemusí využívat vysílání multicastových paketů a může IP adresu vepsat přímo do pole "Komunikuji s IP" a zahájit měření.

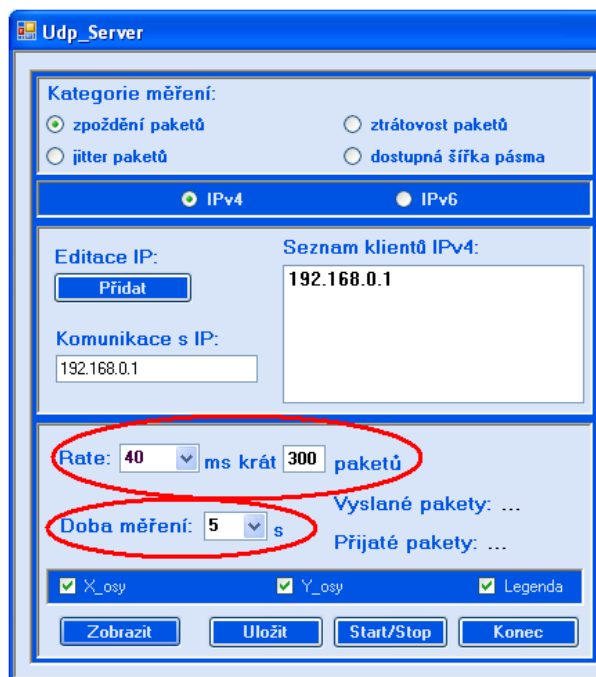
### 7.2.3 IPv4 vs. IPv6

V předcházející kapitole jsem se již toto rozšíření lehce naznačil. Díky protokolu IPv6 (internet protokol verze 6 – Internet Protocol Version 6) je měření možno realizovat i na sítích, kde se starší protokol IPv4 (internet protokol verze 4 – Internet Protocol Version 4) nevyskytuje, nebo není podporován. Rozšíření komplexnosti napomáhá obsáhnout širší okruh sítí a je snahou podchytit nástup tohoto protokolu na lokální síti. V samotném řešení aplikací si to vyžádalo nejenom úpravy přijímacích struktur (hlavně přijímacích vláken), ale především specifikaci portů příslušných vláken. Porty jsou nedílnou součástí komunikace v síti a je nutné jím věnovat nutnou pozornost. Každý z protokolů je proto specifikován portem jak nám nabízí následující tabulka 7.1.

### 7.2.4 Parametry měření

Jak je vidět na obrázku 7.6 tak měření samo osobě nabízí uživateli několik různých variant nastavení parametru měření.

Mezi nejvíce využívané doajista bude patřit celková doba měření a také počet vysílaných paketů. Přitom počet odeslaných paketů lze měnit právě nastavením hodnoty "Rate" a číslovky počtu paketů. Jak již bylo výše uvedeno, je hodnota pole "Rate" využívána v intervalu 40-1000ms. Hodnoty nižší nemají na měření vliv a hodnoty vyšší množství paketů snižují. Různou kombinací parametrů měření lze nastavit počet odeslaných paketů od jednotek až po tisíce paketů za sekundu. Jestliže se rozhodne obsluha serverové aplikace zjistit dostupnost dané klientské stanice, stačí nastavit dobu měření na 1s a množství paketů třeba na několik jednotek odeslaných jedenkrát za sekundu.



Obr. 7.6: Parametry měření

IP	Normální	Multicast
IPv4	31337	5050
IPv6	9050	7050

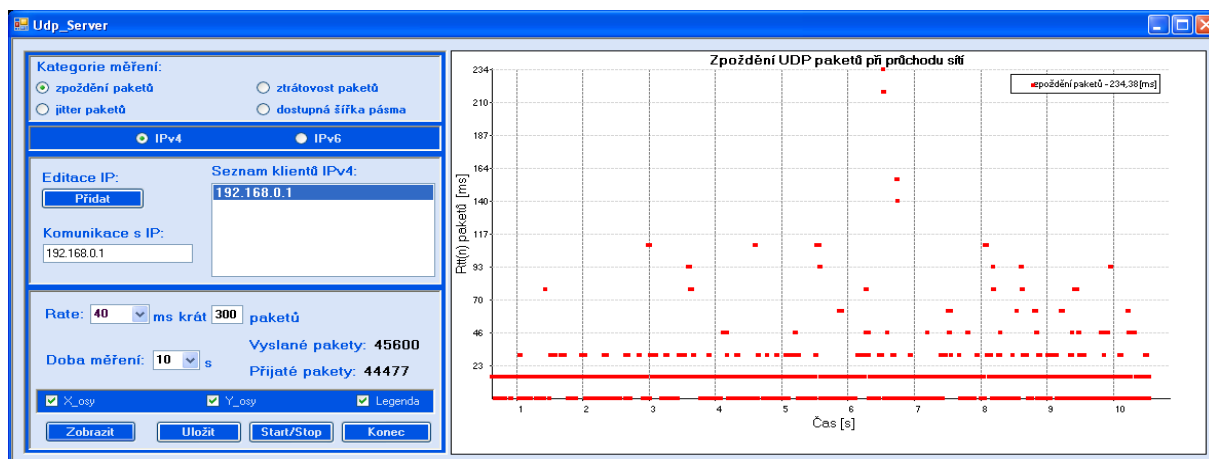
Tab. 7.1: Schéma IP portů

Vyslané množství paketu je datovým tokem vyslaným do sítě. Tento datový tok je roven počtu paketů, krát velikost paketů, která je pro měření zpoždění, jitteru a ztrátovosti konstantní a to 28B, krát 8 bitů pro vyjádření hodnoty v b/s. Pokud ovšem obsluha chce zjistit jak síť snáší větší zátěž, nastavení hodnoty "Rate" na 40ms a počtu paketů 4000 kusů je vhodným pro dosažení datového toku okolo 5Mb/s. Pro zjištění stavu spoje k dané klientské aplikaci bych zvolil kompromisní nastavení mezi těmito dvěma krajními stavy.

## 7.2.5 Zobrazování dat

Naměřené hodnoty zjišťovaných parametrů je také nutné nějakým způsobem zobrazovat. Za tímto účelem jsem napsal speciální funkční kód *MujGraf.cs*, který má za úkol data graficky znázornit. Zobrazování je řízeno interně z kódu funkce stejně jako i načítání naměřených hodnot. Nutnou podmínkou pro správnou funkci bylo ošetření vláken obsluhujících obnovování naměřených hodnot tak, aby nedocházelo k mezi-vláknovým chybám. Vzhledem k velkému množství přenášených paketu a

tím pádem i velkému množství naměřených a zobrazovaných hodnot (řádově tisíce). Zobrazovaná data jsou při intervalu větším než 30s poněkud komprimována, a zobrazení se stává pouze orientačním. Pro odhalení lokálních extrémů doporučuji, měření realizovat v intervalech do 10s, jak je vidět na obrázku 7.7.



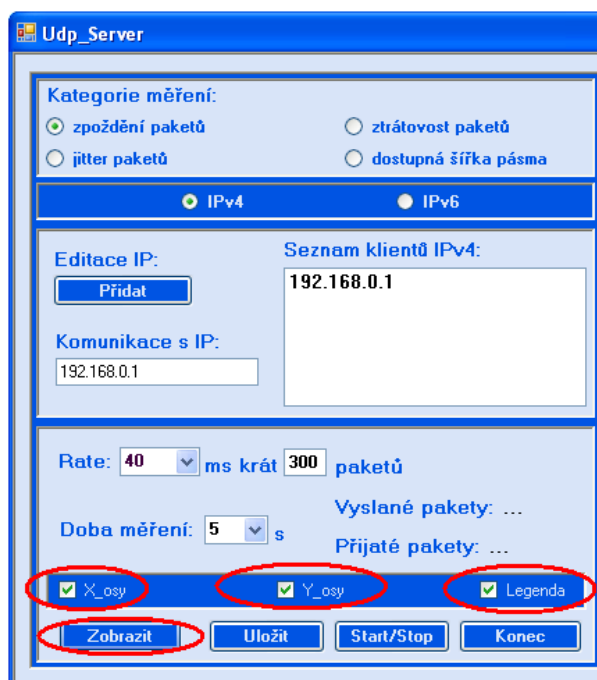
Obr. 7.7: Zpoždění paketu

Pro obsluhu je instalováno několik pomocných a zřehledňujících pomůcek, které jsou umístěny ve spodní části serverové aplikace 7.8. Jedná se o tlačítko "Zobrazit" a tři zaškrťovací okénka pro zobrazení/skrytí vedlejších os  $X$  a  $Y$ . Tlačítko má za úkol zobrazení aktuálních dat, která jsou načtena ve funkci grafu.

## 7.2.6 Funkce, spuštění a uložení

V této sekci se spíše než samotnou funkcí serverové aplikace, chci zabývat obsluhou a možnými problémy, které mohou nastat. V předešlém textu jsem popsal hlavní části aplikace a jejich funkce. Nyní bych se rád orientoval na samotný průběh měření a následné nakládání s naměřenými hodnotami. Na začátku je nutné si uvědomit co vlastně chceme měřit a podle toho zvolit danou kategorii měření. Neméně důležitá je volba IP protokolu a vyhledání klientských stanic pomocí tlačítka "Přidat" viz. 7.2.2.

Po zvolení správné IP adresy stanice a nastavení možných parametrů měření nám již nic nebrání ve spuštění měření ovládacím tlačítkem "Start/Stop" ve spodní části okna aplikace. Po zdárném doměření požadovaných hodnot, je možné data uložit na lokální disk stanice ve formátu \*.csv pro další zpracování. Spolu s měřenými hodnotami se uloží i základní informace o provedeném měření, jako například doba trvání měření, IP adresa klientské stanice a počet odeslaných a přijatých paketů. Měření je možno opakovat. V případě, že měření neproběhlo, doporučuji zkontrolovat IP adresu klientské stanice a nastavené parametry měření. Jestliže se naměřené hodnoty



Obr. 7.8: Grafické nástroje

nezobrazily, což se stává výjimečně, je vhodné využít funkce tlačítka "Zobrazit" a data vykreslit v grafu.

## 7.3 Klientská aplikace

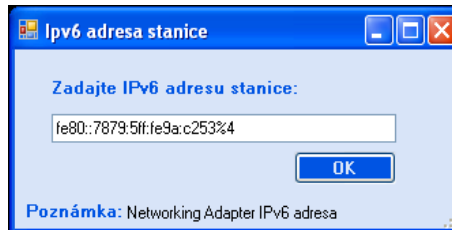
### 7.3.1 Funkce

Funkce již do značné míry vyplývá z obrázku 7.2 v kapitole "Serverová aplikace". Jak je vidět úkolem klientské aplikace je odesílání měřících paketů nazpět ke serverové stanici. Ještě před samotným měřením, je klientská aplikace ve stavu očekávání paketu multicastu, který je využíván ke zjišťování aktivity klientských stanic.

Klientské stanice jsou rozděleny na dva typy dle protokolu, který využívají.

1. Stanice protokolu IPv4.
2. Stanice protokolu IPv6.

První z nich se nikterak neliší od popsané funkce výše s upřesněním využívaného protokolu IPv4. Klientská stanice protokolu IPv6 je poněkud komplikovanější vzhledem k využívanému protokolu. Bohužel jsem nenašel způsob jakým načíst IPv6 adresu z operačního systému stanice, proto je při spuštění klientské aplikace protokolu IPv6 nutný zásah obsluhy. Obsluha musí zadat adresu stanice a potvrdit ji tlačítkem "OK" jak je vidět na následujícím obrázku 7.9.



Obr. 7.9: Formulář zadávání IPv6 adresy klientské stanice

Po zadání adresy (stanice IPv4 bez zadávání adresy) stanice přejde do běhu na pozadí operačního systému. Její vytížení operační paměti je cca 15MB RAM (náročnost serverové aplikace je asi 20MB RAM). Při spuštění klientské aplikace je nutné dávat pozor na síťová rozhraní, která jsou aktivní. Protože klientská stanice protokolu IPv4 získává svoji adresu z operačního systému stanice, jen nutné dodržet podmínku jednoho aktivního síťového spoje a to i pro aplikaci serverovou. V opačném případě nedojde k odeslání odpovědi na multicastové pakety a klientská aplikace zůstane v režimu čekání na multicastový paket!

## 7.4 Podmínky použití

Pro správnou činnost obou měřících aplikací je nutné, aby na stanicích byla nainstalována knihovna síťových podpůrných programů .NETFramework minimálně ve verzi 2. Správná činnost aplikací je podmíněna aktivitou pouze jednoho síťového spoje. V opačném případě selže multicastové vysílání a nedojde ke správné inicializaci měřících aplikací. Klientská aplikace využívající protokolu IPv6 vyžaduje ruční zadání adresy stanice. Aplikace jsou spustitelné v operačním systému Windows XP a Windows Vista.

## 7.5 Vývojové diagramy aplikací

Nyní se dostáváme k samotnému programovému řešení obou aplikací a jejich struktuře a běhu cyklů a procesů. Nejlépe samotnou funkci programu vystihují vývojové diagramy. Proto v následujícím textu bude vždy uvedena konkrétní ilustrace vývojového diagramu spolu s komentářem pro lepší orientaci.

### 7.5.1 Serverová aplikace

Na začátek se soustředím na kostru programu a dále rozvedu jednotlivé typy měřících funkcí. Jádrem serverové aplikace je následující struktura procesů zachycená na

obrázku 7.10. Samozřejmě jako všechny aplikace tak i tato při svém spuštění nutně vyžaduje počáteční nastavení výchozích hodnot proměnných veličin, deklaraci hodnot řídicích časovačů, přijímacích vláken a jiné. V dalším kroku již nadchází čas pro výběr požadovaných měření a inicializace měření. Následuje příjem měřicích paketů a jejich zpracovávání dle kategorie měření. Výsledek je zobrazen pomocí grafu a naměřené hodnoty je možné uložit do souboru, pro další zpracovávání. Aplikace poté přechází do své výchozí polohy pro další měření.

### 7.5.2 Zpoždění paketů

Funkce měření zpoždění paketů je základem pro všechna zbylá měření. Opět na počátku své činnosti funkce inicializuje nastavení proměnných hodnot a časovačů řídicích měření. Následně dojde ke spuštění těchto řídicích procesů a funkce přejde do přijímacího režimu, kdy očekává měřicí pakety ve svém přijímacím vláknu běžícím na pozadí uživatelského okna. Po přijetí určitého množství dat, dojde k uložení hodnot do funkce grafu, která již sama zajistí jejich zobrazení 7.11.

### 7.5.3 Jitter paketů

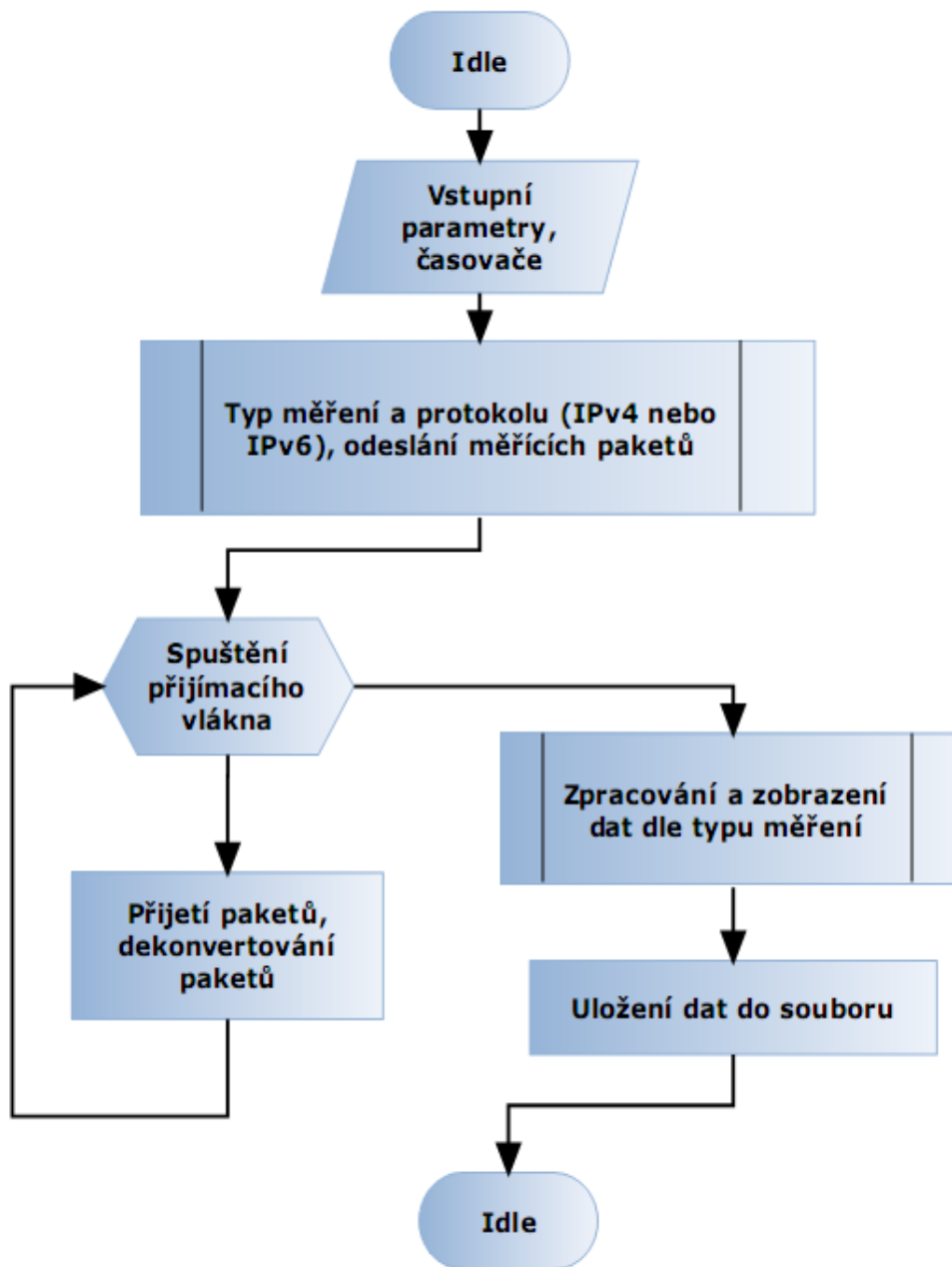
Vývojový diagram funkce jitteru je dosti podobný vývojovému diagramu zpoždění paketů. Minimální rozdíl nastává při ukládání dat, kdy je nutné ještě před samotným zobrazením dat určit hodnotu jitteru. Ostatní procesy jsou stejné. Vývojový diagram je znázorněn na obrázku 7.12.

### 7.5.4 Ztrátovost paketů

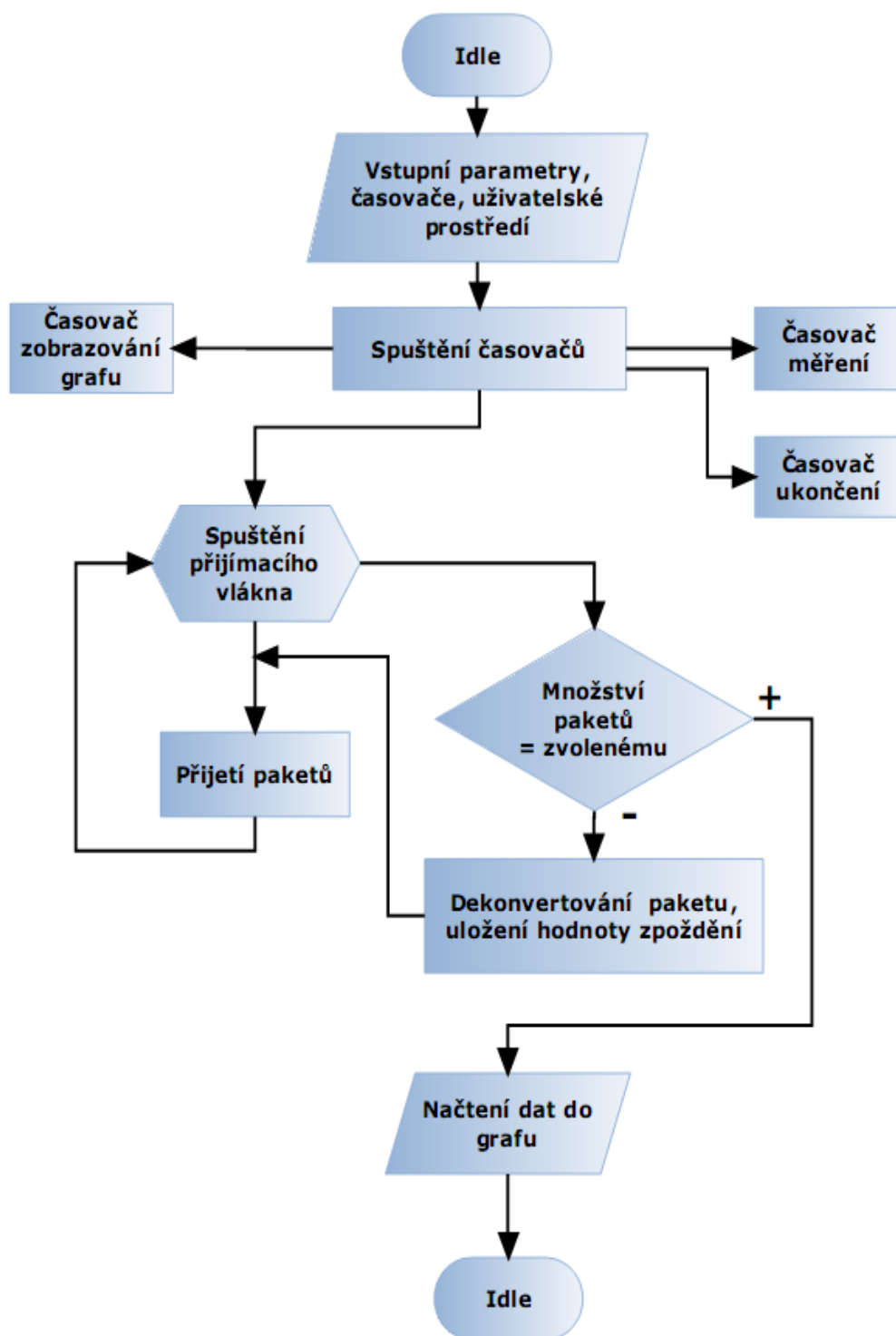
Ztrátovost je posledním z parametrů, který je měřen současně s předcházejícími dvěma. Vývojový diagram je díky tomu také velice podobný vývojovým diagramům předcházejících veličin. Rozdíl je patrný pouze u ukládání naměřených dat, kdy je nutné specifikovat hodnoty ztrátovosti dle definice a poté je uložit ke vykreslení grafem. Vývojový diagram na obrázku 7.13.

### 7.5.5 Časovače řídicí běh měření a zobrazování

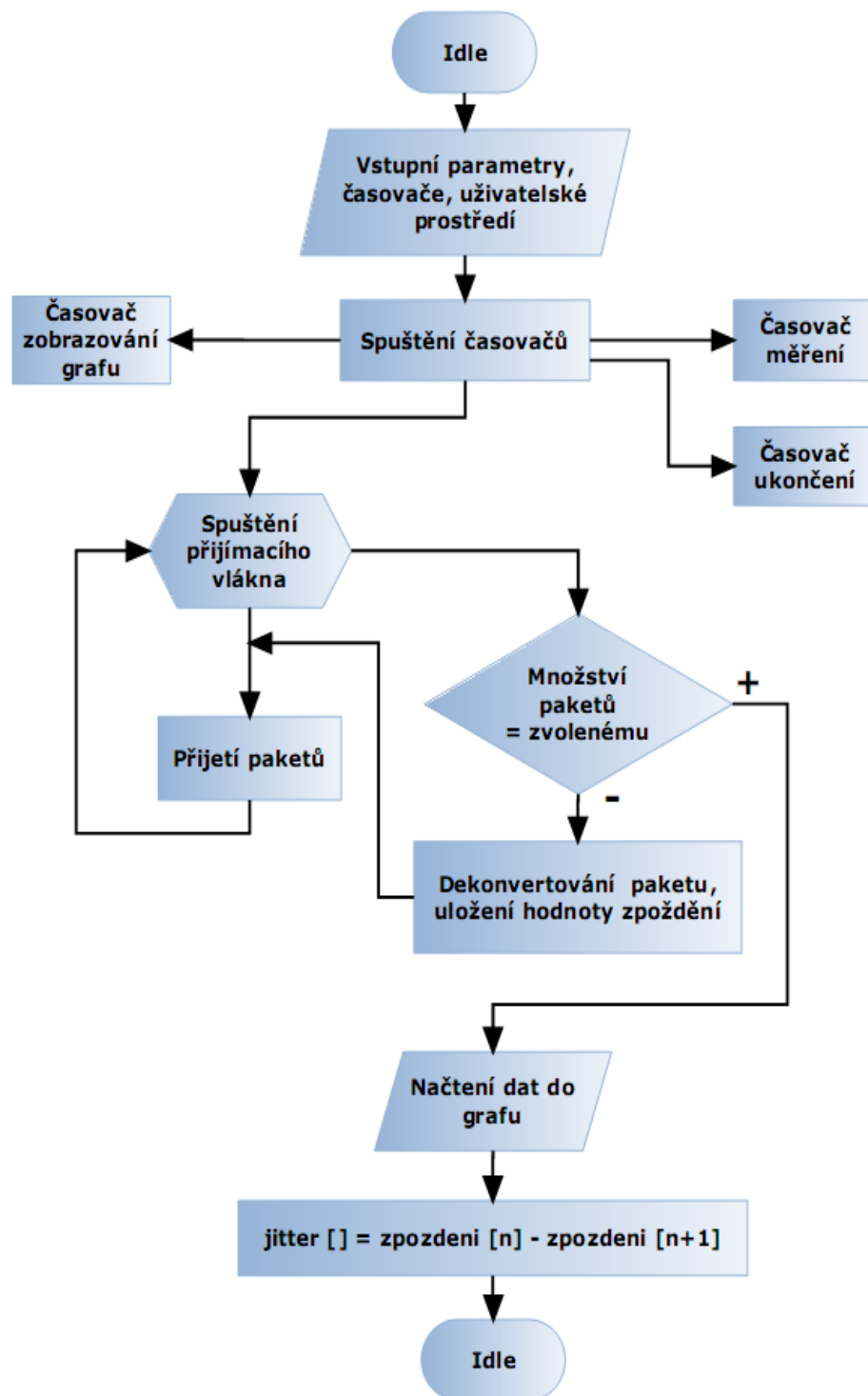
Časovače hrají v běhu programu velice důležitou roli. Program serverové aplikace obsahuje celkem tři časovače, konkrétně je to časovač měření, časovač zobrazování grafu a časovač ukončení. Hlavním je časovač měření, který má za úkol vysílání měřicích paketů a také časovač ukončení, který všechny zmiňované časovače zastavuje. Poslední časovač je časovačem funkce *MujGraf.cs* a zobrazuje měřené veličiny na základě definovaného typu grafu.



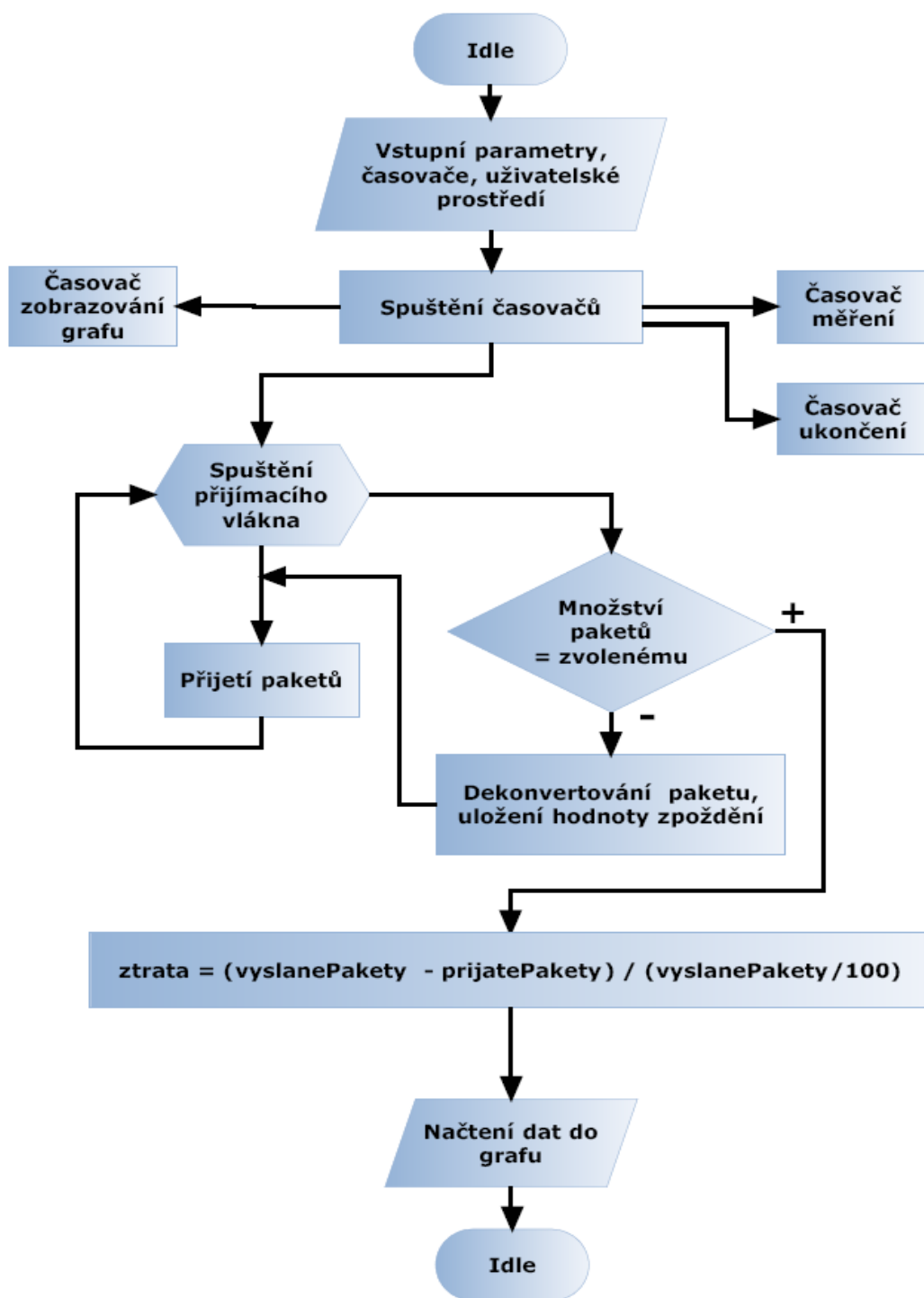
Obr. 7.10: Vývojový diagram serverové aplikace



Obr. 7.11: Vývojový diagram funkce zpoždění paketu



Obr. 7.12: Vývojový diagram funkce jitter paketu



Obr. 7.13: Vývojový diagram funkce ztrátovost paketu

Typ grafu je určen uživatelem na základě kategorie měření při inicializaci měření. Funkci časovačů znázorňuje jejich vývojový diagram na obrázku 7.14.

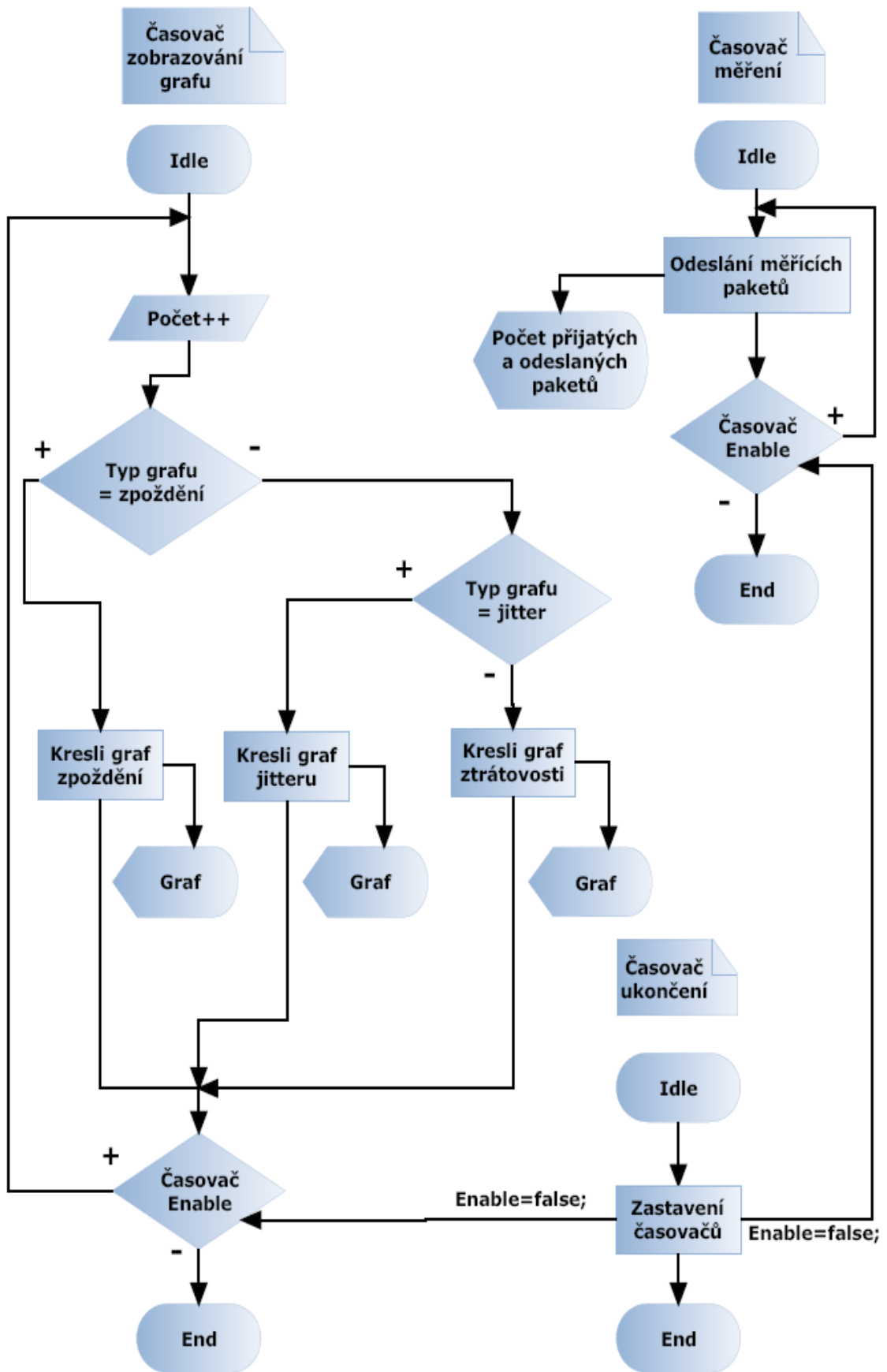
### 7.5.6 Dostupná šířka pásma

Jak bylo v dřívějším textu popsáno veličina šířky pásma je v aplikaci samostatným měření a má svá specifika jak naznačuje vývojový diagram 7.15 této funkce.

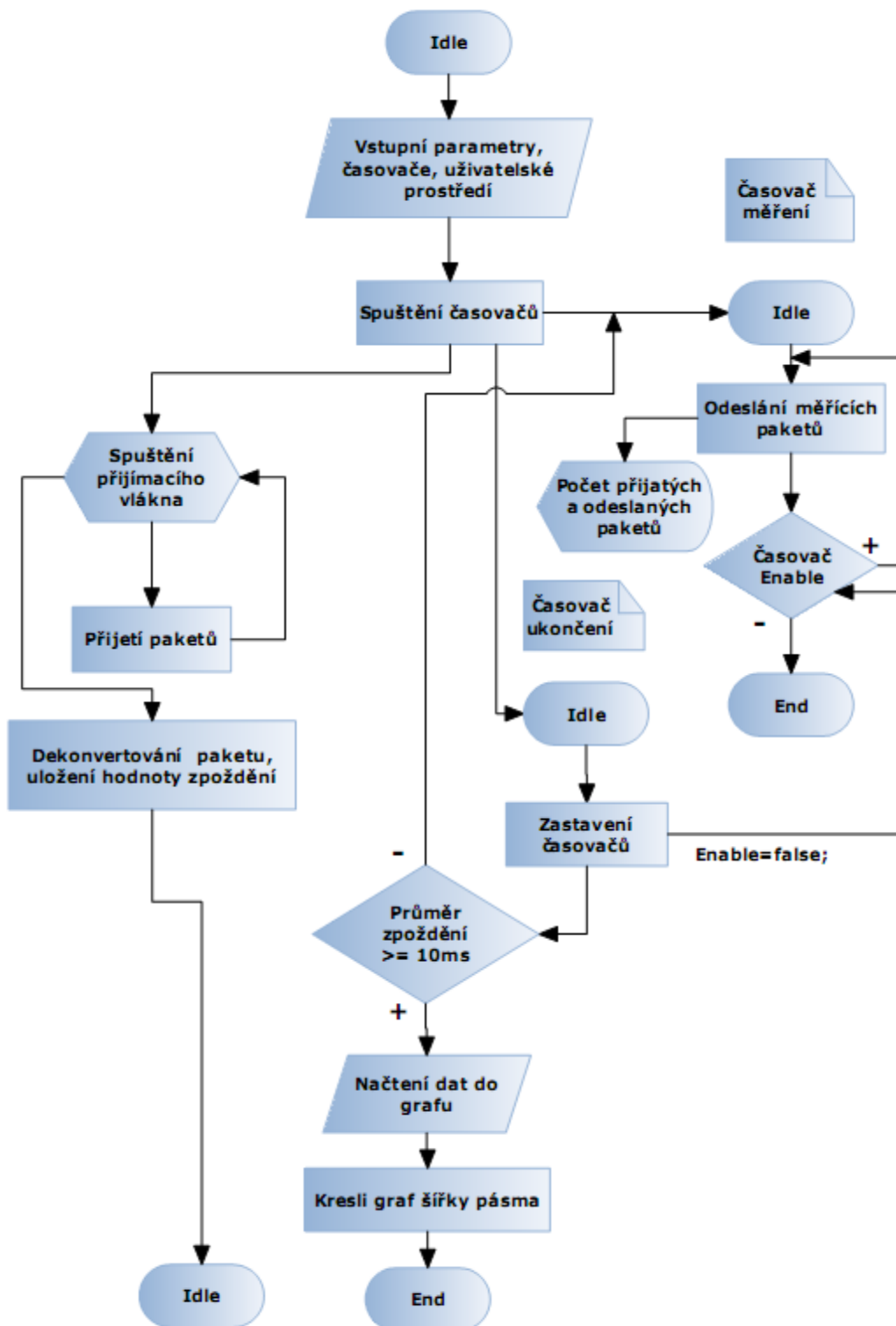
Začátek této funkce je stejný jako všechny předešlé. Dochází k nastavení počátečních parametrů a hodnot časovače. Hlavním rozdílem je schéma řízení vyhodnocování přijatých dat. Data jsou vyhodnocována po ukončení měření trvajícím 1s. Při splnění podmínky vzrůstajícího zpoždění je stanovena dostupná šířka pásma rovna odeslanému datovému toku při předchozím vysílání měřících paketů. V opačném případě, je datový tok měřících paketů navýšen zhruba dvojnásobně a měření je opakováno.

### 7.5.7 Klientská aplikace

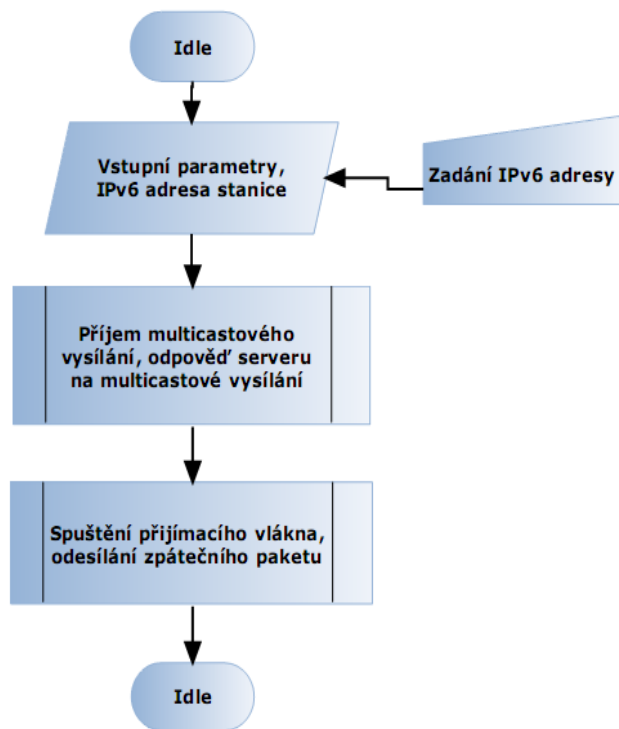
Vývojový diagram klientské aplikace je poměrně jednoduchý. Rozhodl jsem se uvést pouze verzi protokolu IPv6, která alespoň trochu komunikuje s obsluhou. Na začátku své činnosti vyžaduje zásah obsluhy v podobě zadání IPv6 adresy klientské stanice. Následuje příjem měřících paketů a zpětné odeslání na stranu serveru. Obrázek 7.16 tuto funkci vystihuje.



Obr. 7.14: Vývojový diagram funkce časovačů serverové aplikace



Obr. 7.15: Vývojový diagram funkce dostupné šířky pásma

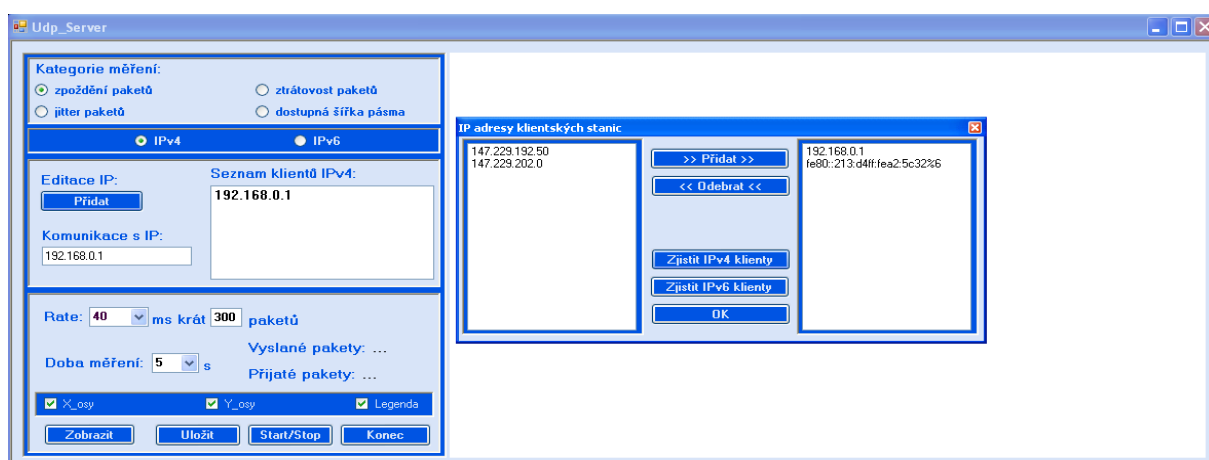


Obr. 7.16: Vývojový diagram funkce klientské aplikace

## 8 MĚŘENÍ PARAMETRŮ SÍTĚ

### 8.1 Měření parametrů sítě protokolem IPv4

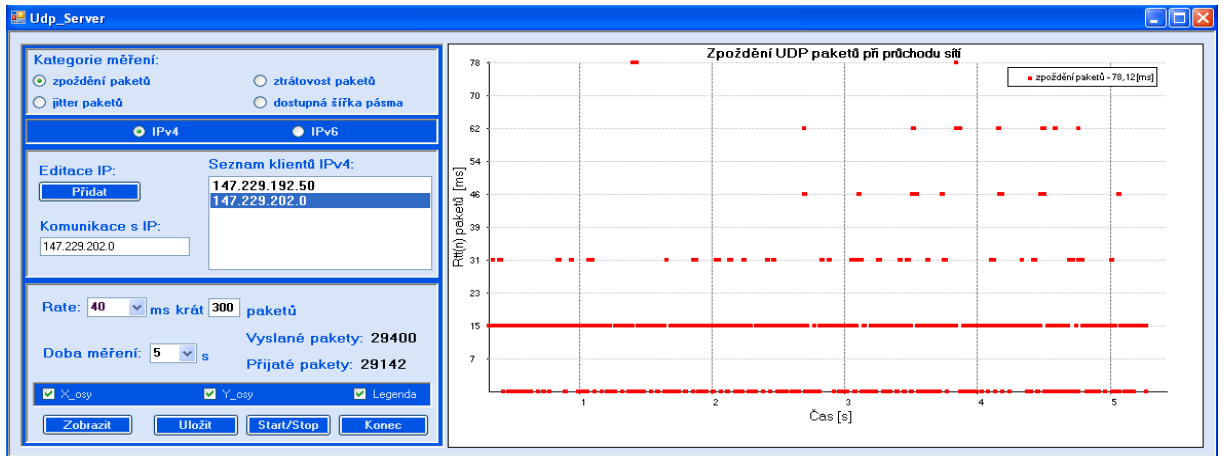
Cílem této sekce je představit práci SW v reálných podmínkách větší sítě. Po sérii testů v laboratoři ústavu telekomunikací jsem se rozhodl měření realizovat na kolejní síti KaM VUT v Brně. Měření bylo realizováno v rámci jednoho bloku kolejí List, kde se nacházela stanice serveru a jedna klientská stanice (číslo 1.) a druhá stanice byla umístěna na Purkyňových kolejích (číslo 2.). Před samotným měřením je vhodné si zavolat všechny klientské stanice a zjistit jejich adresy 8.1 pomocí multicastového vysílání.



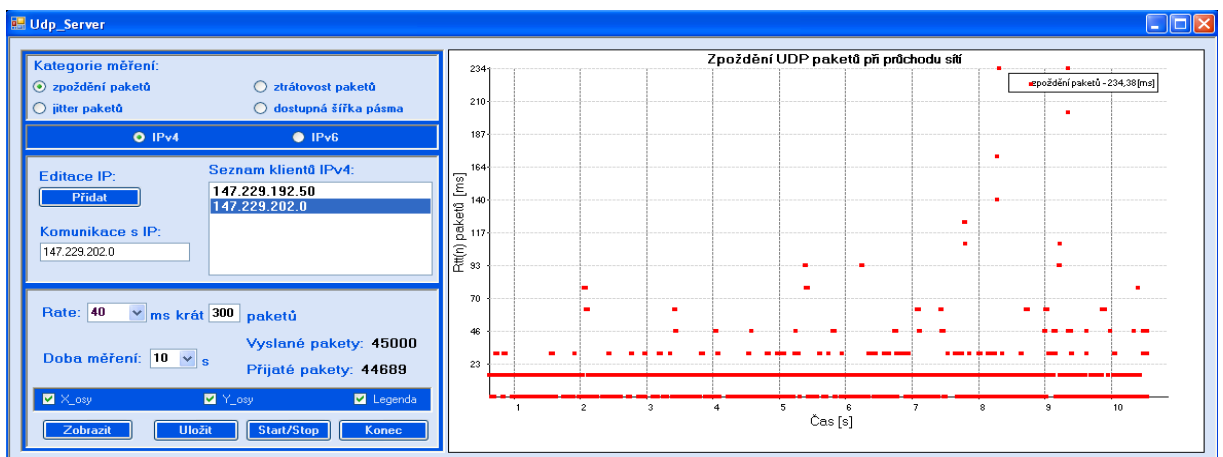
Obr. 8.1: Ověření dostupnosti klientských stanic

Měření obsáhla několik časových rozsahů a byli využity obě verze IP protokolu. V následující sérii obrázků si postupně projdeme zobrazovaná měřená data. Nejprve se zaměříme na měření protokolem IPv4.

Jestliže se nám povedlo zjistit adresy klientských stanic, už nám nic nebrání zvolit si konkrétní stanici (dvoj-klikem na adresu ve výpisu) a nastavit parametry samotného měření (dobu měření, počet paketů atd.). Nastavené parametry měření jsou takové, aby nikterak nepřetěžovali síť, ale jsou takové, aby nám nastínily v jakém stavu se síť nachází. Zvoleným nastavením doby měření 5s, "Rate" je roven 40ms a počet paketů je nastaven na hodnotu 300 paketů se průměrné množství paketů odeslaných za 1s pohybuje okolo 6000 paketů. Realizovaný datový tok odpovídá zhruba 1,5Mb/s, což není nijak závratné číslo pro datovou síť. Zvolená doba měření je záměrně do 10s, protože při zobrazování řádově tisíců hodnot se hodnoty komprimují a zobrazení se stává dosti nepřehledným. Tohoto nastavení, nebo nastavení velice podobného jsem se držel celé měření, protože je považuji za dostatečně výmluvné pro zjištění stavu datového spoje.



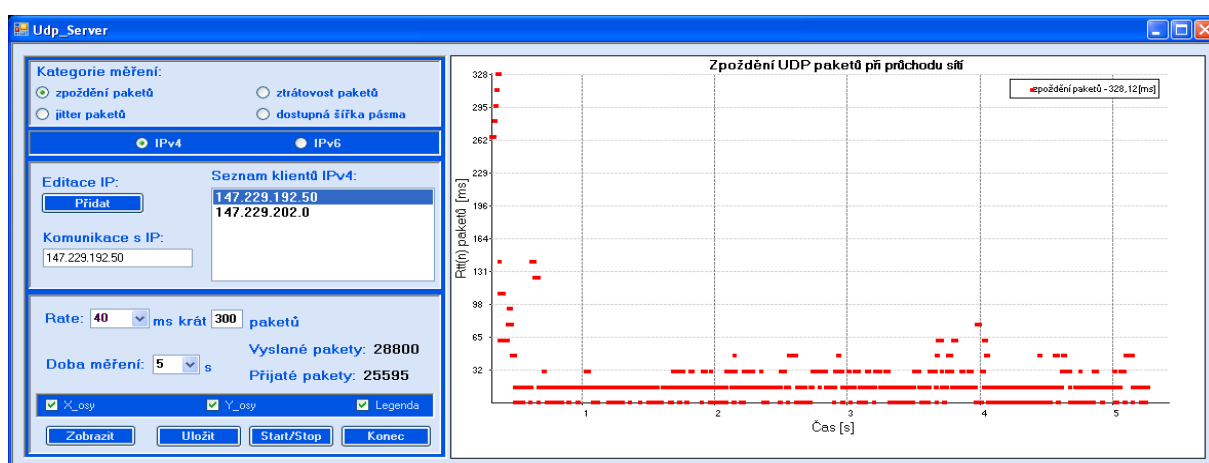
Obr. 8.2: Zpoždění paketů stanice číslo 1. po dobu 5s



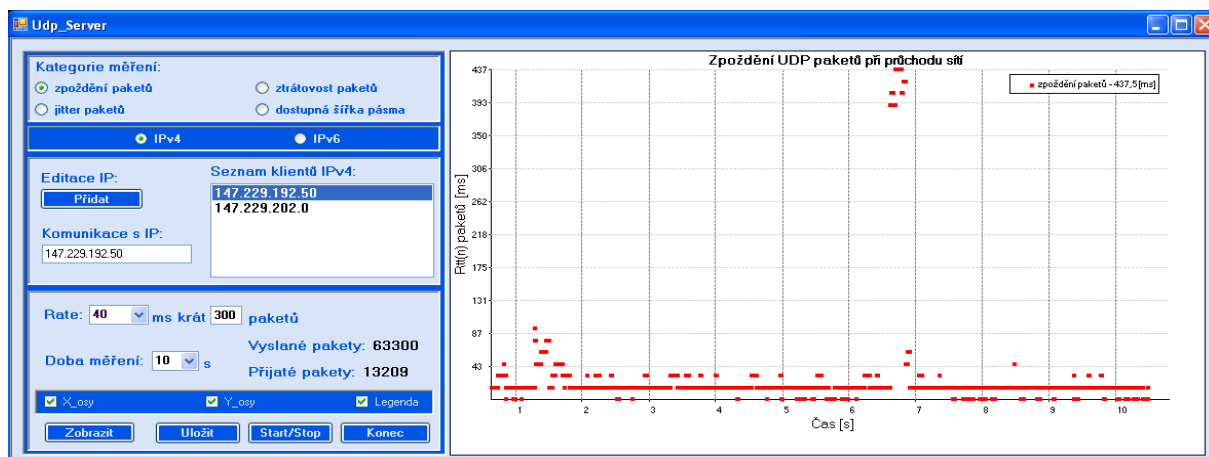
Obr. 8.3: Zpoždění paketů stanice číslo 1. po dobu 10s

Prvním měřením je znázorněno zpoždění paketů mezi serverovou aplikací a stanicí bližší (číslo 1.) jak ilustruje obrázek 8.2 a 8.3. Častý výskyt některých hodnot při zobrazovaném množství splývá v přímku na dané úrovni zpoždění. Příkladem může být hodnoty 15ms, která se vyskytuje poměrně často.

Vidíme, že zpoždění nedosahuje nikterak překvapivých a závratných hodnot. Hodnoty se pohybují řádově v desítkách milisekund což vzhledem k provozu na kolejích není nic výjimečného. V druhé dvojici obrázků 8.4 a 8.5 je již vidět nárůst zpoždění, v závislosti na větší vzdálenosti stanic. U některých měření může docházet při inicializaci spoje k špičkovému navýšení hodnoty zpoždění paketů jak je vidět například na obrázku 8.4.

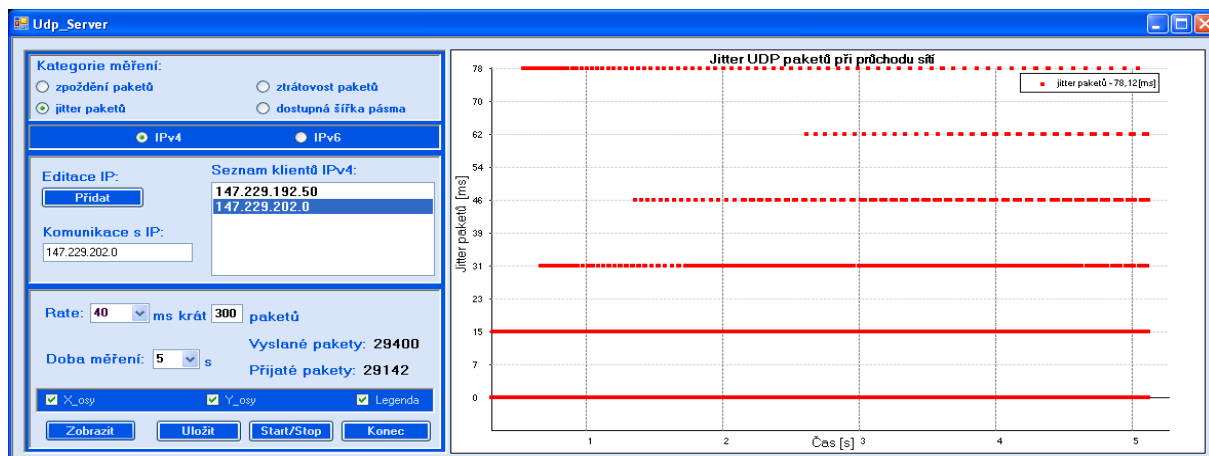


Obr. 8.4: Zpoždění paketů stanice číslo 2. po dobu 5s



Obr. 8.5: Zpoždění paketů stanice číslo 2. po dobu 10s

Pro přehlednost uvádím grafy jitteru a poté i ztrátovosti obou stanic. Graf jitteru nám naznačuje dynamiku s jakou docházelo v průběhu měření ke změnám zpoždění paketů. Napomáhá k odhalení špičkových výchylek zpoždění paketů, kdy se zobrazí jako jen několik osamocených hodnot. Ovšem v realizovaných měřeních se dynamika změn zpoždění projeví hlavně na začátku, kdy dochází k ustálení přenosu paketů. Zhuštění bodů na počátku grafu tomu napovídá. Obrázky jsou zobrazeny zde 8.6, 8.6, 8.8 a 8.9.

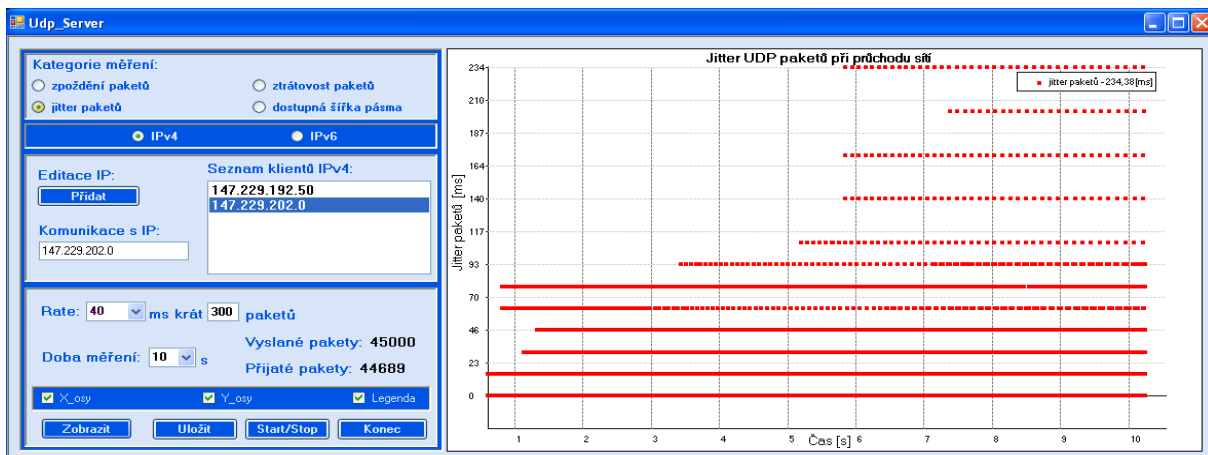


Obr. 8.6: Jitter paketů stanice číslo 1. po dobu 5s

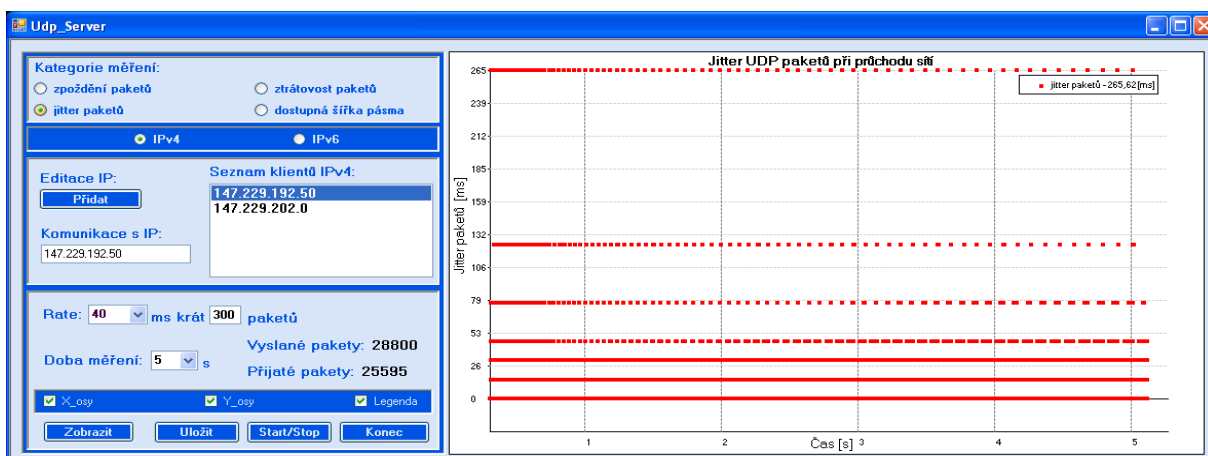
O zajímavější výsledky se nám postarají grafy ztrátovosti přenášených paketů.

Realizované měření naznačuje vysokou ztrátovost paketů při inicializaci vysílání měřených paketů. Směrovače se adaptují na nově vzniklé spojení, zjišťují si topologii sítě na jakých portech se nachází konkrétní síť atd.

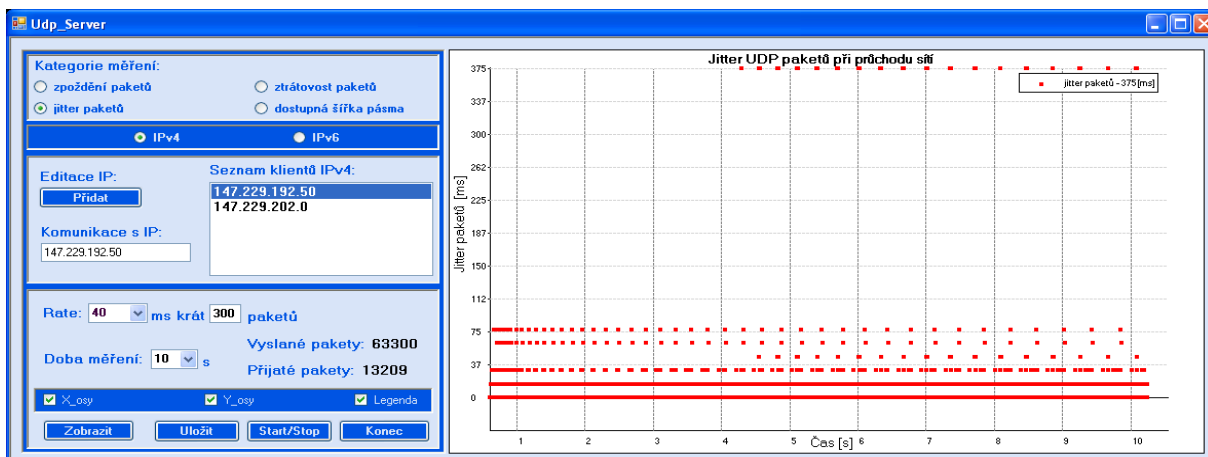
V této době může docházet k zahazování paketů, v důsledku krátkodobého vytížení



Obr. 8.7: Jitter paketů stanice číslo 1. po dobu 10s

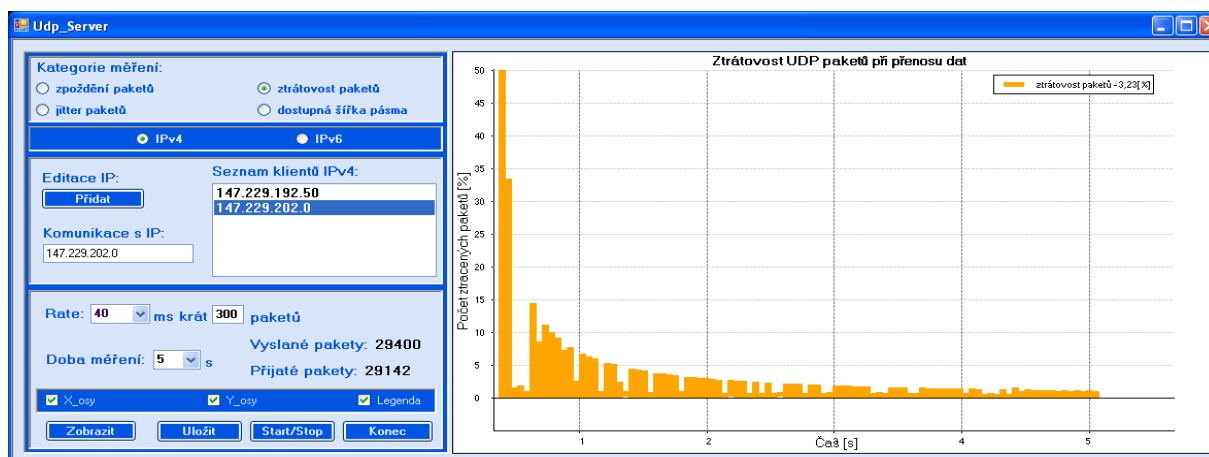


Obr. 8.8: Jitter paketů stanice číslo 2. po dobu 5s

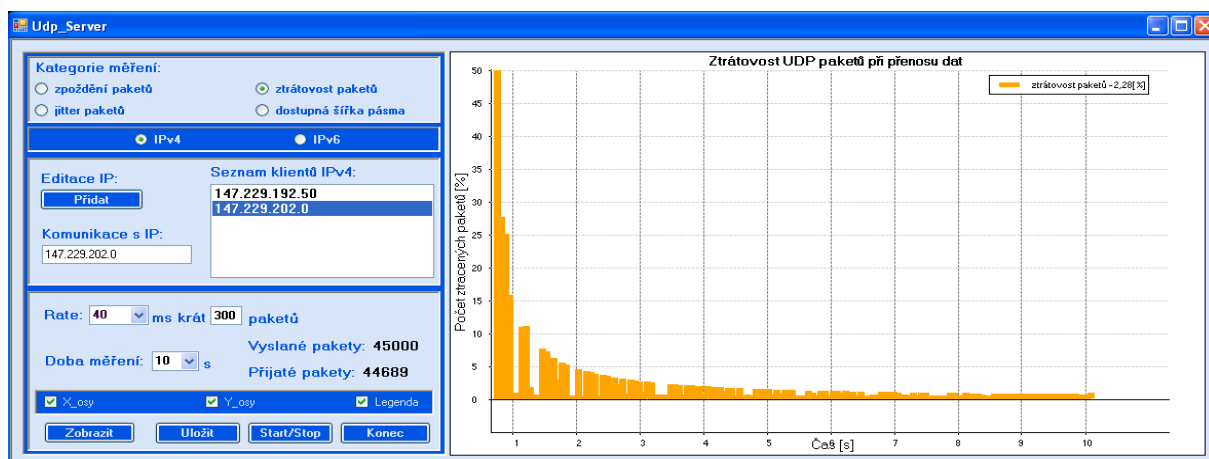


Obr. 8.9: Jitter paketů stanice číslo 2. po dobu 10s

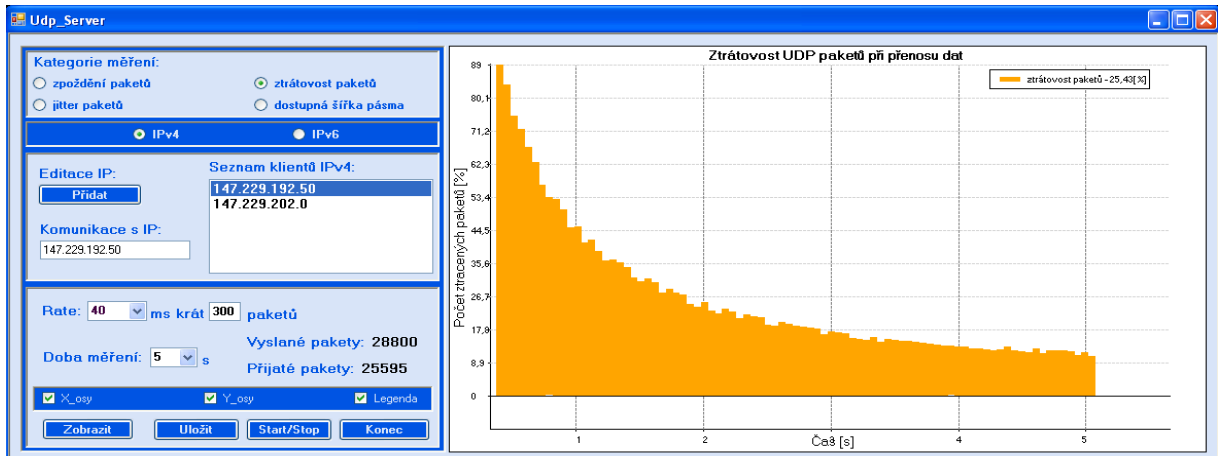
směrovače obzvláště jestliže je přenášeno velké množství paketů. V průběhu měření dochází k ustálení hodnoty ztrátovosti paketů na hodnoty jednotek procenta, zhruba do 4% odeslaných paketů. Tato charakteristika je typická pro většinu spojení a realizovaných měření. Při měření stanice číslo 2. byla hodnota ztrátovosti vždy o dost větší než při prvním měření. Tento fakt je možno vysvětlit vyšším počtem směrovacích prvků při průchodu sítí. Obě situace charakterizují následující obrázky.



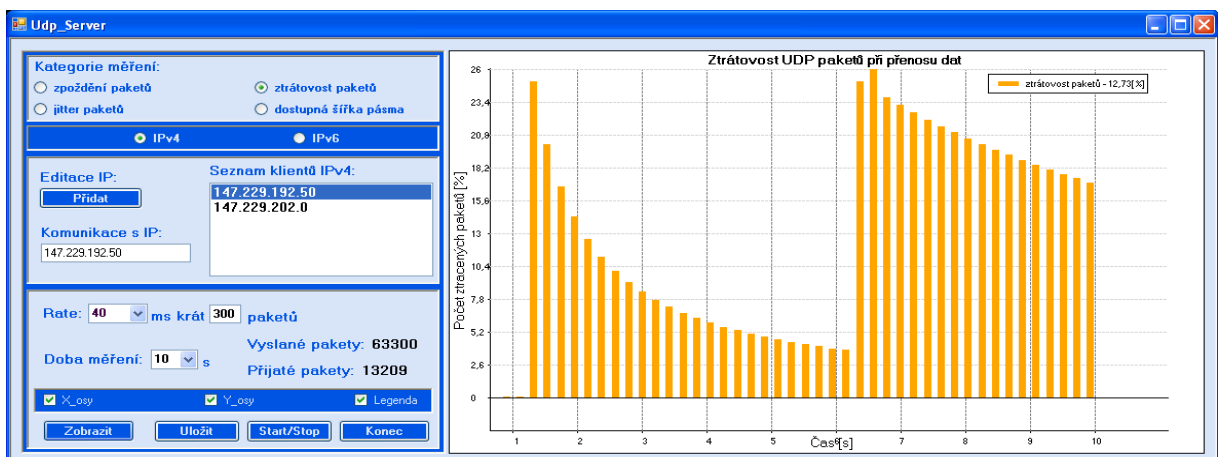
Obr. 8.10: Ztrátovost paketů stanice číslo 1. po dobu 5s



Obr. 8.11: Ztrátovost paketů stanice číslo 1. po dobu 10s



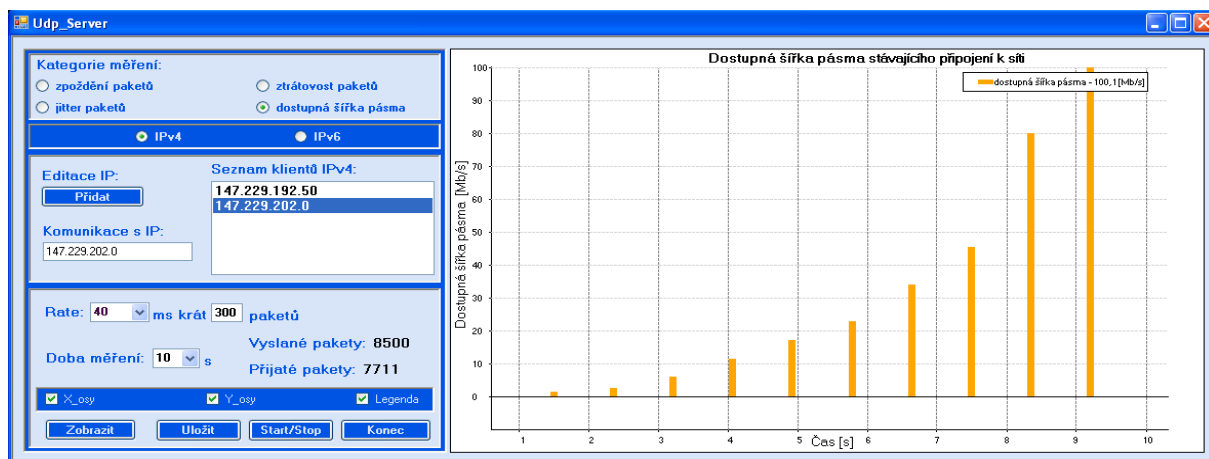
Obr. 8.12: Ztrátovost paketů stanice číslo 2. po dobu 5s



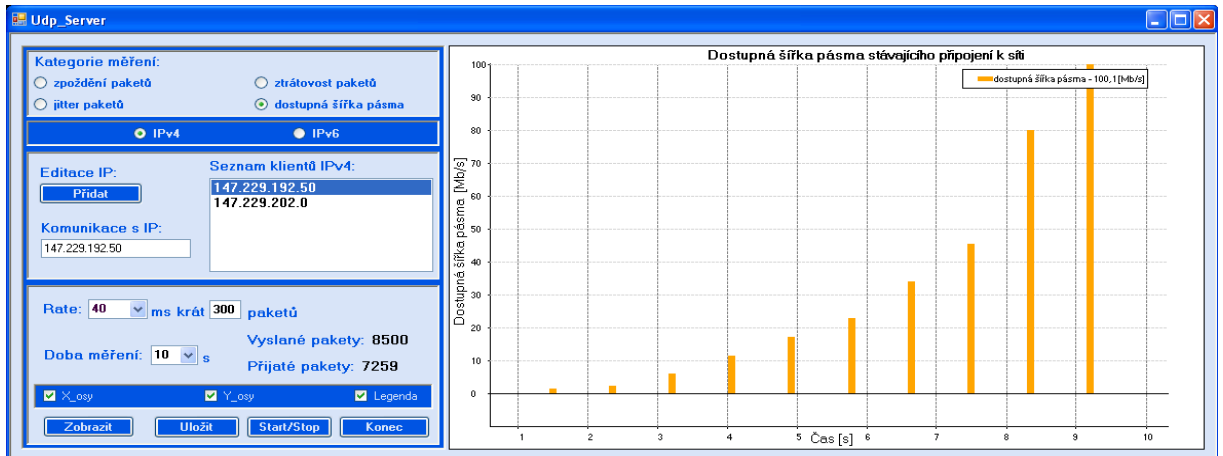
Obr. 8.13: Ztrátovost paketů stanice číslo 2. po dobu 10s

Posledním realizovaným měřením je měření aktuální dostupné šířky pásma. Toto měření považuji za jedno z nejnáročnějších vzhledem k vyhodnocování naměřených dat. Vyhodnocení je dle stanovené definice přímo závislé na schopnosti aplikace vyhodnotit změnu zpoždění přenášených paketů. Samotné měření se skládá z několika kroků a je nutné je správně vyhodnotit. Od hodnoty šířky pásma nad 20Mb/s jsme zvolil krok měření 10Mb/s, což při takovéto rychlosti sítě považuji za adekvátní. Od hodnot nad 40Mb/s je krok čtyřnásobný respektive první krok je 40Mb/s a další již jen 20Mb/s protože je již dosaženo hodnoty 100Mb/s. Protože měření bylo realizováno na poměrně rychle sítí dosáhla měřená šířka pásma maximální hodnoty a to 100Mb/s.

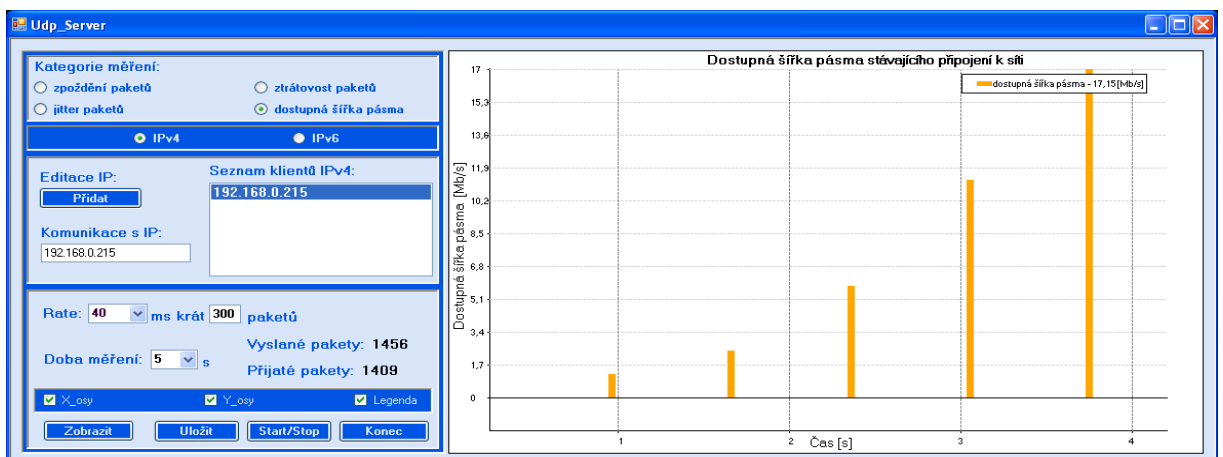
Hodnoty nad 80Mb/s jsou vzhledem k zvolenému kroku měření pouze orientační. Pro ilustraci jsem proměřil jinou klientskou stanicí dostupnou přes Wifi spoj na lokální síti. Všechna tři měření dostupné šířky pásma, jsou zobrazena na následujících obrázcích 8.14, 8.15 a 8.16.



Obr. 8.14: Dostupná šířka pásma stanice číslo 1.



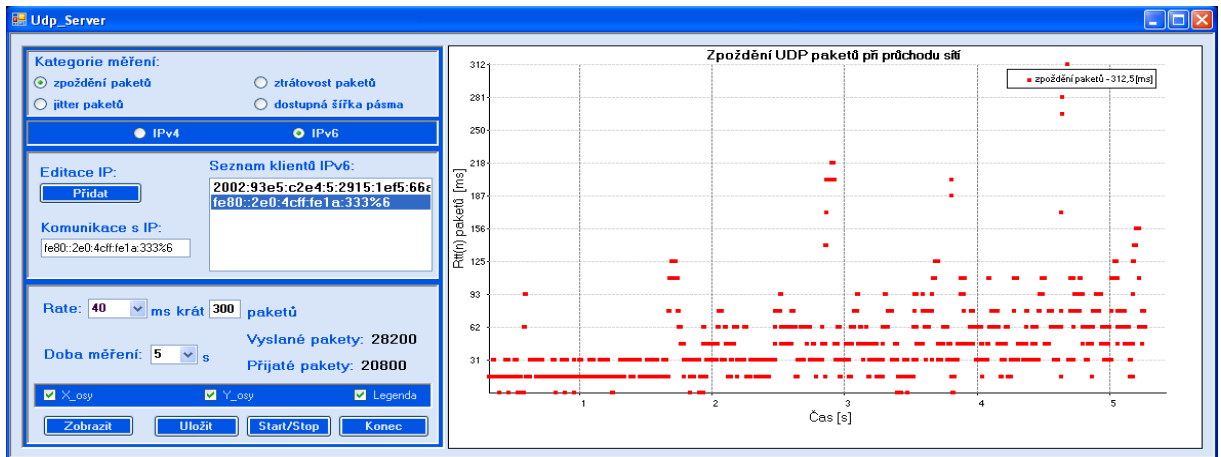
Obr. 8.15: Dostupná šířka pásma stanice číslo 2.



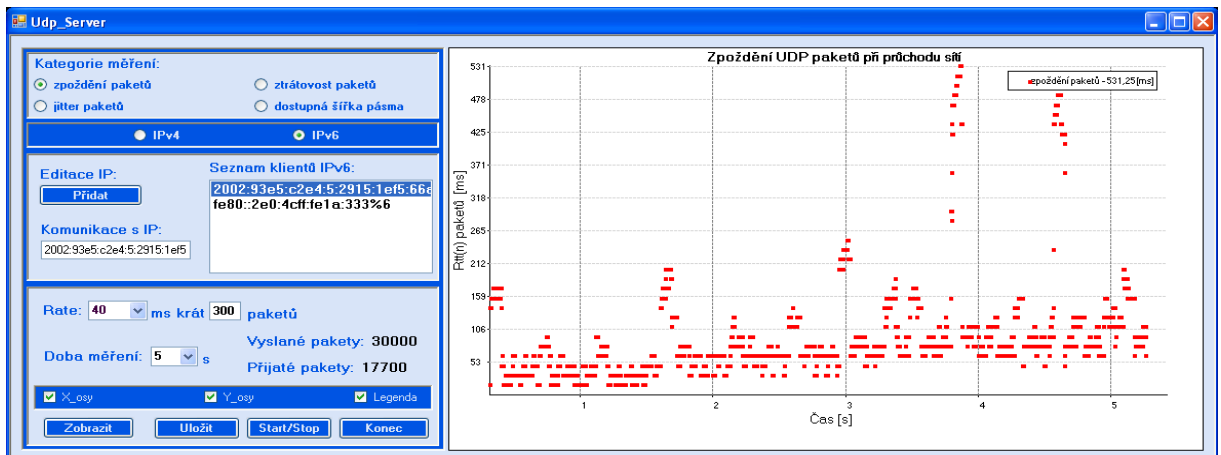
Obr. 8.16: Dostupná šířka pásma stanice přes WIFI spoj

## 8.2 Měření parametrů sítě protokolem IPv6

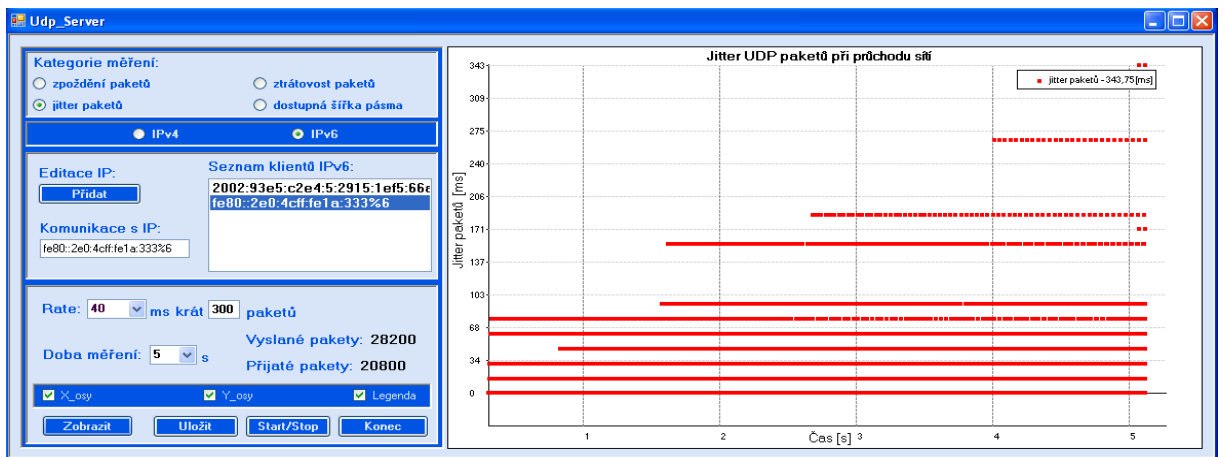
Při měření parametrů sítě pomocí protokolu IPv6 se naměřené hodnoty poněkud lišily od předcházejícího měření. Hodnoty zpoždění byli oproti měření protokolem IPv4 minimálně dvojnásobné nebo i vyšší. Vysvětlení se nabízí větší náročnost tohoto protokolu na směrování a režii s tím spojenou. Ostatní sledované parametry se od toho také odvíjejí, ať už jitter nebo ztrátovost přenášených paketů. Měření šířky pásma změna protokolu nepoznamenala a hodnoty naměřené různými protokoly jsou takřka shodné. Zkrácený přehled měření poskytnou následující obrázky měřených parametrů.



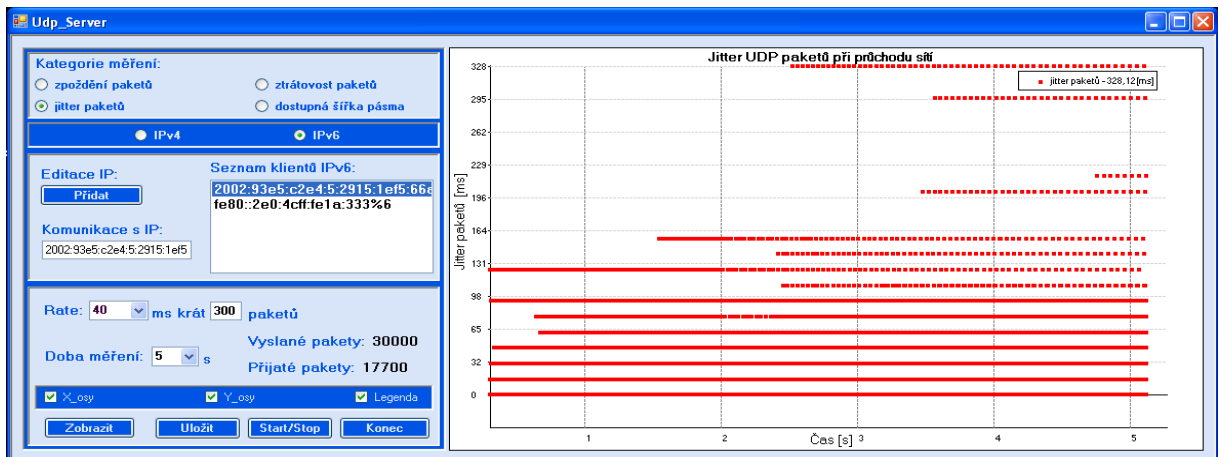
Obr. 8.17: Zpoždění paketů stanice číslo 1. po dobu 5s



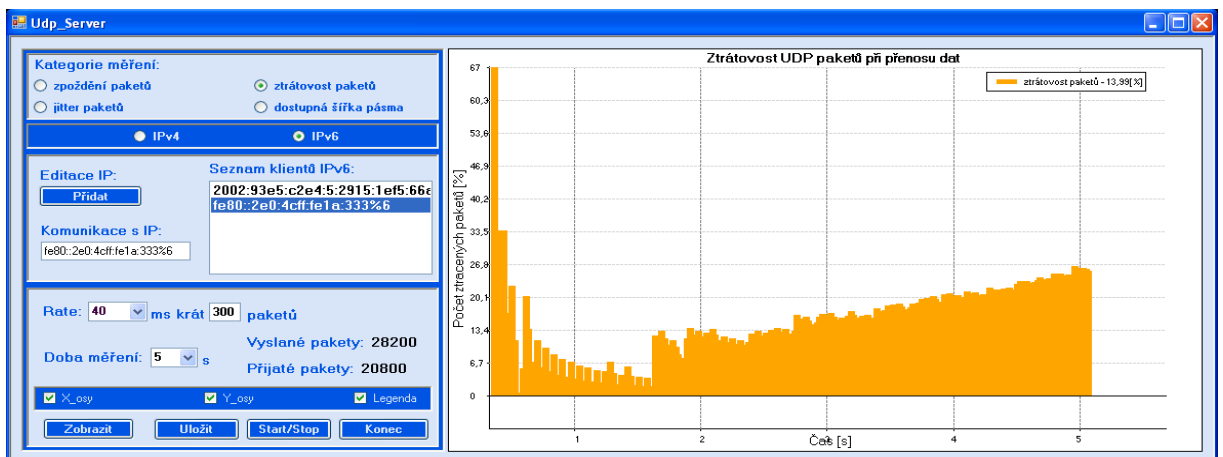
Obr. 8.18: Zpoždění paketů stanice číslo 2. po dobu 5s



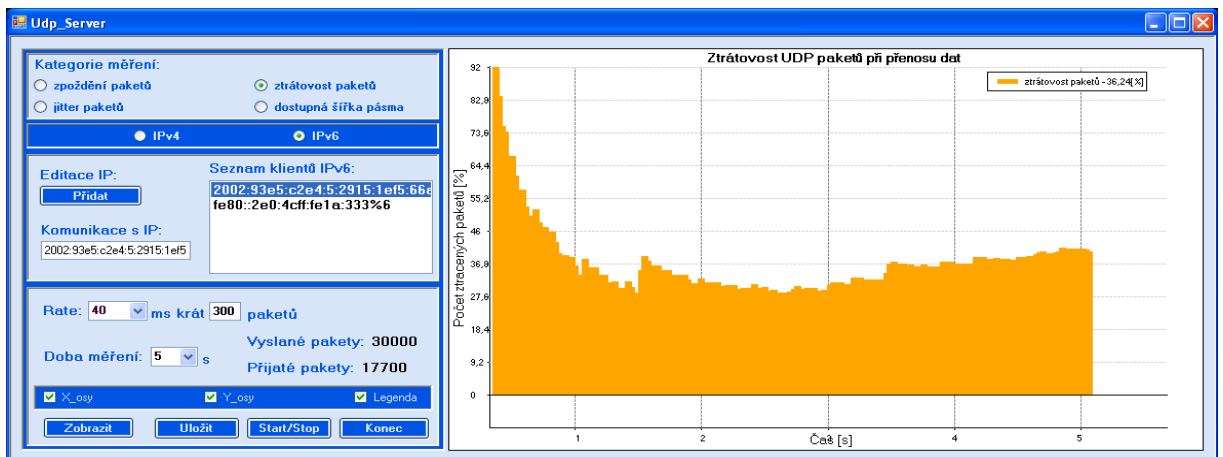
Obr. 8.19: Jitter paketů stanice číslo 1. po dobu 5s



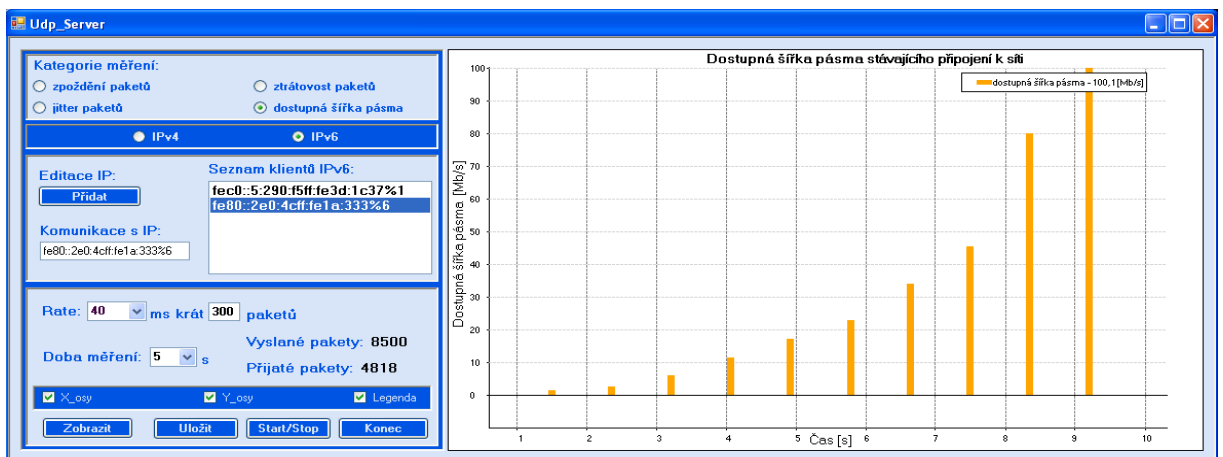
Obr. 8.20: Jitter paketů stanice číslo 2. po dobu 5s



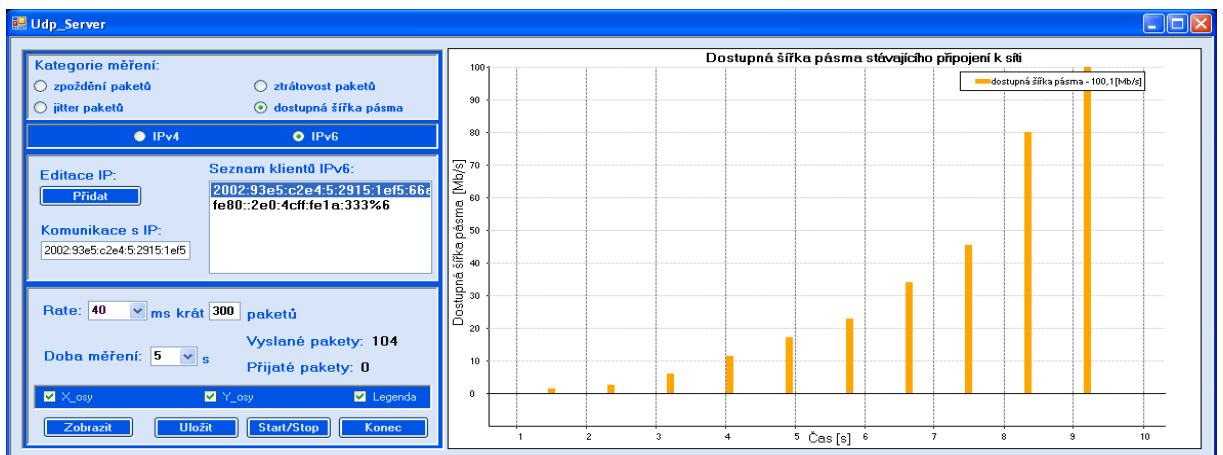
Obr. 8.21: Ztrátovost paketů stanice číslo 1. po dobu 5s



Obr. 8.22: Ztrátovost paketů stanice číslo 2. po dobu 5s



Obr. 8.23: Dostupná šířka pásma stanice číslo 1.



Obr. 8.24: Dostupná šířka pásma stanice číslo 2.

## 9 ZÁVĚR

Pro dosažení stanovených cílů diplomové práce bylo zapotřebí, abych nejprve prostudoval řadu publikací zabývajících se problematikou základních parametrů datových sítí a způsobu jejich měření. Některé základní metody měření jsem uvedl pro ilustraci v tomto projektu. Ze získaných znalostí jsem provedl návrh vlastního řešení problému měření těchto nejzákladnějších parametrů datových přenosů počítačovými sítěmi. Při návrhu jsem se zaměřil především na zpoždění paketů, jitter paketů, ztrátovost paketu a dostupnou šířku pásma daného síťového spoje.

Realizace návrhu měření byla pouze dílčím úkolem projektu. Hlavním cílem diplomové práce bylo realizovat SW aplikaci, na základě zmíněného návrhu, která umožní dané parametry sítě nejenom měřit, ale naměřené hodnoty zobrazit v grafu a umožnit jejich uložení na lokální disk pro pozdější zpracování. Navržené řešení se po čase ukázalo jako nedostačující a bylo nutné jej patřičně rozšířit. Návrh počítal s komunikací typu klient-server, která má výhodu v shromáždění dat na jednom místě a umožňuje lépe měřit větší množství klientských stanic. Přenos měřících paketů měl být realizován ze strany klientské stanice ovšem problém synchronizace se stal závažnějším, než jsem původně předpokládal a proto jsem schéma komunikace upravil tak, aby měřící pakety byli vysílány a také přijímány na serverové stanici. Klientská stanice je "pouze" přeposílá nazpět serveru.

Abych dosáhl širší možnosti využití realizované aplikace rozšířil jsem podporu komunikačního prostředí ze standardního protokolu IPv4 také o protokol IPv6. Hlavním rozdílem těchto dvou protokolů je adresace koncových uzlů sítě. Protokol IPv6 nabízí velikost pole adresy 128b a umožňuje mnohem širší adresaci koncových uzlů, naproti tomu protokol IPv4 má pouze 32b adresu koncového uzlu sítě. Dostávám se k hlavní části realizovaného SW, kterou je serverová aplikace, která veškeré měření řídí a obsluhuje.

Před samotným začátkem měření, je vhodné zjistit jaké klientské aplikace jsou spuštěné a tím pádem připravené k měření parametrů sítě. Za tímto účelem, jsem do aplikací instaloval systém zjišťování dostupnosti aktivních klientských stanic pomocí multicastových paketů vyslaných do sítě. Ovšem, jestliže je známa konkrétní adresa klientské stanice, je možné ji přímo použít k vlastnímu měření, aniž obsluha využije multicastového vysílání paketů.

Realizovaný SW jsem vyzkoušel v učebně UTKO a měření jsem realizoval na síti KaM VUT v Brně. Měření obsáhla dvě klientské stanice a jednu stanicí serverovou. Pro měření bylo využito obou zmiňovaných protokolů. Detailně popis měření je uveden v dokumentu v kapitole Měření parametrů sítě - 8, kde jsem se snažil upozornit na zobrazovaná měření a naměřené hodnoty parametrů sítě. Hodnoty se ve směs shodovaly s teoretickým předpokladem. Na závěr jsem otestoval funkčnost aplikací

v prostředí WLAN.

Posledním úkolem, bylo naměřené hodnoty uložit do souboru pro pozdější možné zpracování. Formát uložení dat byl změněn z původního uložení do souboru *\*.txt* na praktičtější formát a to *\*.csv* pro jeho nespornou výhodu importu dat například do MS Excel. Pro správnou funkci obou aplikací je nutné, aby na stanicích byl nainstalován balík podpůrných síťových programů .NET Framework. Sice není nezbytně nutné pro samotnou aplikaci, ale využití multicastového vysílání si klade za podmínku pouze jednoho aktivního síťového připojení.

Celý projekt SW aplikace jsem realizoval ve vývojovém prostředí MS Visual Studio 2005 programovacím jazykem C#, který považuji za nejvhodnější pro programování zaměřené na síťové aplikace.

## LITERATURA

- [1] Cisco Systems. *Cisco IOS 12.0 Quality of Service* [online]. 1999, poslední aktualizace 12. 11. 2006 [cit. 17. 11. 2007]. Dostupné z URL: <[http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito\\_doc/qos.htm](http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/qos.htm)>.
- [2] JAIN M. *IEEE/ACM Transactions on networking* [online]. AUG 2003, vol. 11, iss. 4, page(s): 537 - 549 [cit. 17. 11. 2007]. Dostupné z URL: <<http://portal.acm.org/citation.cfm?id=941374.941378>>.
- [3] KLAŠKA, L. *Jak změřit základní kvalitativní parametry sítě a program NetDoppler společnosti WildPackets* [online]. 6.12.2006, [cit. 17. 11. 2007]. Dostupné z URL: <<http://www.svetsiti.cz/view.asp?rubrika=Tipy&clanekID=118>>.  
[nms.lcs.mit.edu/6.899/bolot.pdf](http://nms.lcs.mit.edu/6.899/bolot.pdf)
- [4] BOLOT, J. *End-to-End Packet Delay and Loss Behavior in the Internet* [online]. 1993, poslední aktualizace 7. 7. 1997 [cit. 17. 11. 2007]. Dostupné z URL: <[nms.lcs.mit.edu/6.899/bolot.pdf](http://nms.lcs.mit.edu/6.899/bolot.pdf)>.
- [5] COCCETTI, F. *Bandwidth measurements and router queues* [online]. 2002, poslední aktualizace 20.9.2002 [cit. 17. 11. 2007]. Dostupné z URL: <[ipm.mib.infn.it/bandwidth-measurements-and-router-queues.pdf](http://ipm.mib.infn.it/bandwidth-measurements-and-router-queues.pdf)>.
- [6] TIRUMALA, A. *Measuring end-to-end bandwidth with Iperf using Web100* [online]. 2005, [cit. 17. 11. 2007]. Dostupné z URL: <[www.csm.ornl.gov/dunigan/pam.pdf](http://www.csm.ornl.gov/dunigan/pam.pdf)>.
- [7] *Web100* [online]. 2007, poslední aktualizace 15. 11. 2007 [cit. 17. 11. 2007]. Dostupné z URL: <<http://www.web100.org/>>.
- [8] FRANCESCHINIS, M. *Measuring TCP over WiFi: A Real Case* [online]. 2006, poslední aktualizace 18. 3. 2006 [cit. 17. 11. 2007]. Dostupné z URL: <[www.tlc-networks.polito.it/mellia/papers/winmee.pdf](http://www.tlc-networks.polito.it/mellia/papers/winmee.pdf)>.
- [9] KOCUR, Z. *Měření v bezdrátových sítích* [online]. 200ž, poslední aktualizace 2. 7. 2006 [cit. 17. 11. 2007]. Dostupné z URL: <[www.comtel.cz/files/download.php?id=2819](http://www.comtel.cz/files/download.php?id=2819)>.

# SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

ACL seznam oprávněného přístupu – Access Control List

BER bitová chybovost – Bit Error Rate

CAR angažovaná míra přístupu – Committed Access Rate

CBWFQ klasifikovaná přímá váhovací fronta – Class-Based Weighted Fair Queuing

CQ účastnická fronta – Custom Queuing

FTP souborový transportní protokol – File Transfer Protocol

HP Hewlett-Packard

ICMP kontrolní protokol zpráv internetu – Internet Control Message Protocol

IP internet protokol – Internet Protocol

IPv4 internet protokol verze 4 – Internet Protocol Version 4

IPv6 internet protokol verze 6 – Internet Protocol Version 6

ISO/OSI Mezinárodní organizace pro normalizaci / propojení otevřených systémů  
– International Standards Organization / Open System Interconnection

ISP poskytovatel služby Internet – Internet Service Provider

LAN místní počítačová síť – Local Area Network

MSS maximální velikost segmentu – Maximum Segment Size

NBAR síťové aplikace rozpoznávání – Network-Based Application Recognition

NLANR The National Laboratory for Applied Network Research

PQ prioritní fronta – Priority Queuing

QoS kvalita služeb – Quality of Services

RAM paměť s libovolným (náhodným) přístupem – Random-Access Memory

RSSI indikace intenzity přijímaného signálu – Received Signal Strength Indication

RTT oboustranné zpoždění – Round Trip Time

SLoPS opakované přenášení dat – Self-loading Periodic Streams

SNR odstup signál/šum – Signal-to-noise ratio

SW programové vybavení počítače – Software

TCP Transmission Control Protocol

TCP/IP transportní protokol v prostředí Internetu – Transmission Control Protocol/Internet Protocol

UDP User Datagram Protocol

WFQ přímá váhová fronta – Weighted Fair Queuing

WRED prvotní náhodná detekce – Weighted Early Random Detect