VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

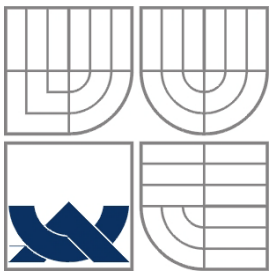
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

IDENTIFIKACE POČÍTAČE NA ZÁKLADĚ ČASOVÝCH ZNAČEK PAKETŮ

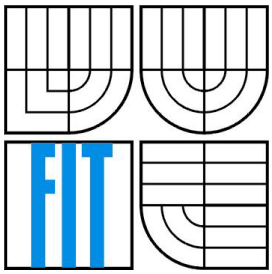
BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE JAN NOVOTNÝ
AUTHOR

BRNO 2012



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

IDENTIFIKACE POČÍTAČE NA ZÁKLADĚ ČASOVÝCH ZNAČEK PAKETŮ

COMPUTER IDENTIFICATION BASED ON PACKET TIMESTAMPS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — JAN NOVOTNÝ
AUTHOR

VEDOUCÍ PRÁCE — ING. JAN KAŠTIL
SUPERVISOR

BRNO 2012

Abstrakt

Tato práce pojednává o technice identifikace počítače na základě časových značek paketů. V práci je dále popsán algoritmus pro výpočet zkreslení hodin počítače, na kterém se celá technika identifikace počítače zakládá. Ukážeme využití této techniky k identifikaci několika počítačů v reálné počítačové síti, k identifikaci zařízení, na kterém běží překlad adres NAT a k identifikaci počítačů nacházejících se za NATem. Nakonec ukážeme, jak lze určit počet počítačů nacházejících se za NATem.

Abstract

This work describes perspective identification technique called computer identification based on packet timestamps. This work also describes an algorithm to calculate skew of computer clock, on which the entire computer identification technique is based on. We will use this technique to identify a number of computers in the real computer network, the device that is running NAT and computers located behind NAT. Finally we show, how to determine the number of computers located behind NAT.

Klíčová slova

Identifikace počítače, časové značky, hodiny počítače, zkreslení hodin, frekvence hodin, NAT

Keywords

Computer identification, timestamps, computer clock, clock skew, clock frequency, NAT

Citace

Novotný Jan: Identifikace počítače na základě časových značek paketů, bakalářská práce, Brno, FIT VUT v Brně, 2012

Identifikace počítače na základě časových značek paketů

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Jana Kaštila. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

……
Jan Novotný
Datum (16.5.2012)

Poděkování

Chtěl bych poděkovat svému vedoucímu, panu Ing. Janu Kaštilovi, za jeho vůli a ochotu diskutovat se mnou všechny problémy, na které jsem při vypracovávání narazil.

Obsah

1 Úvod

V dnešní době počítačů a Internetu se do této oblasti přesunula i kriminalita a to tzv. internetová kriminalita. Proto je třeba případného internetového pachatele jednoznačně identifikovat, případně sledovat jeho pohyb na internetu i v případě, kdy pachatel změní fyzické místo jeho připojení k Internetu.

Česká republika je členem Evropské unie a ta 14. prosince 2005 přijala v platnost zákon o uchovávání dat o telefonních hovorech, krátkých textových zprávách a datových (internetových) spojeních (data retention directive). V České republice byla schválena vyhláška č. 485/2005 dne 7. prosince 2005. Podle vyhlášky, schválené ministerstvem informatiky a vnitra, jsou provozovatelé veřejných komunikačních sítí povinni uchovávat několik měsíců údaje o elektronické komunikaci – údaje typu časů, IP adres, emailových adres a podobně, o které pak mohou žádat „oprávněné orgány“. Rozsah údajů je o něco širší než požaduje Evropská unie. Vyhláška ukládá uchovávat údaje po dobu šesti měsíců. Tato vyhláška byla ovšem dne 22. března 2011 zrušena Ústavním soudem.

Počítač lze na internetu identifikovat pomocí několika identifikátorů, jako např. jeho IP adresy nebo jeho MAC adresy. IP adresa ale není zcela jednoznačný identifikátor v případě, že pachatel změní svůj přístupový bod k internetu (např. využíváním veřejných přístupových bodů), čímž se může změnit i jeho IP adresa, nebo zastíní svou IP a MAC adresu použitím proxy serveru, nebo se nachází za NATem. Pokud bychom zastupovali určitý správní orgán, tak bychom mohli o tyto údaje zažádat provozovatele veřejné komunikační sítě (providera). Tento způsob by ale fungoval pouze za předpokladu, že by provozovatel dané záznamy opravdu vlastnil a pachatel by byl připojen z některé členské země Evropské unie, nehledě na dobu, za kterou by se záznamy podařilo získat. Vzhledem k tomu, že chceme pachatele identifikovat co nejrychleji, musíme najít jiný způsob.

Proto se zaměříme na hardware počítače a to konkrétně na jeho počítačové hodiny a jejich zkreslení (clock skew). Pomocí hodnot generovaných hodinami počítače dokážeme zkreslení hodin vypočítat. Tyto hodnoty hodin získáme z pole časových značek (razítek) transportního protokolu TCP. Z toho tedy plyne, že snímané zařízení (počítač) pachatele musí mít přístup k Internetu.

Dříve již byly představeny techniky pro zjišťování operačního systému a dalších informací [1,2,3,5], ale díky technice nazvané remote physical device fingerprinting [7] představené v roce 2005 (Kohno, Broido, Claffy), můžeme sledovat zařízení, které může být vzdálené jakkoli daleko od měřícího zařízení a bez toho aniž by o tom snímané zařízení vědělo nebo jakkoliv spolupracovalo s měřícím zařízením.

Práce je dále členěna následovně. Ve druhé kapitole popíšeme základní vlastnosti hodin a také popíšeme, jak fungují hardwarové a systémové hodiny počítače. Ve třetí části se blíže zaměříme na jednu ze základních vlastností hodin a to konkrétně na zkreslení hodin. Popíšeme výpočet frekvence hodin a výpočet zkreslení hodin. Ve čtvrté kapitole stručně popíšeme transportní protokol TCP, jeho povinné a volitelné položky a blíže se zaměříme na volitelnou položku časového razítka. V páté kapitole uvedeme, jak lze spočítat zkreslení hodin v reálné počítačové síti pomocí časových značek. V šesté kapitole popíšeme implementovaný program pro výpočet zkreslení hodin počítače. V předposlední sedmé kapitole uvedeme a zhodnotíme testy, které byly provedeny pomocí implementovaného programu. V závěrečné části budeme diskutovat možná rozšíření této práce.

2 Hodiny

V této sekci popíšeme některé základní vlastnosti hodin, které popisuje [8] (Paxson) a [7] (Moon, Skelly, Towsley), blíže popíšeme, jaké máme k dispozici zdroje času v počítači a jaké jsou jejich omezení, výhody a nevýhody.

2.1 Základní pojmy

Proto, abychom správně pochopili, jak technika identifikace počítače funguje, je důležité porozumět základním vlastnostem hodin. Hodiny, které budeme značit **C**, jsou navrženy tak, aby reprezentovaly část doby, která uběhla od určité počáteční doby **i[C]**. Jako **C(t)** budeme označovat hodiny **C** udávající čas **t**.

Jednou ze základních vlastností je **perioda** hodin **r[C]**. Jedná se o nejmenší jednotku, o kterou jsou hodiny aktualizovány. Tuto hodnotu nazýváme *tick*. Například hodiny s periodou 10 ms jsou stále navyšovány, ale každých 10 ms se nám zobrazí jejich hodnota tedy *tick*.

Rozdíl (offset) hodin **off[C]**, je rozdíl mezi časem udávaným hodinami **C(t)** a přesným skutečným časem $\mathbf{t_r}$ definovaným národním standardem, kde $\mathbf{t} \geq 0$:

$$off[C] = (C(t) - t_r) \tag{1}$$

Pokud máme dvoje různé hodiny $\mathbf{C_a}$ a $\mathbf{C_b}$, udávající čas $\mathbf{C_a(t)}$ a $\mathbf{C_b(t)}$, potom rozdíl mezi těmito hodinami $\mathbf{off[C_aC_b]}$, kde $\mathbf{t} \geq 0$ definujeme jako:

$$off[C_aC_b] = C_a(t) - C_b(t) \tag{2}$$

Přesnost (accuracy) hodin je, jak blízko se blíží absolutní hodnota offsetu hodin nule.

Frekvence hodin **f[C]** je inverzní funkce k **periodě** hodin **r[C]**. Hodiny s rozlišením 10 ms jsou navrženy tak, aby běžely s frekvencí 100 Hz. Frekvence hodin $\mathbf{C_a}$ udávající čas $\mathbf{C_a(t)}$, je první derivace hodin v čase, kde $\mathbf{t} \geq 0$:

$$f[C_a] = C_a'(t) \tag{3}$$

Zkreslení (skew) hodin **s[C]** je první derivace rozdílu hodin a skutečného správného času, kde $\mathbf{t} \geq 0$:

$$s[C] = (C_a(t) - t_r)' \tag{4}$$

Zkreslení $\mathbf{s[C_a]}$ hodin $\mathbf{C_a}$ ve vztahu k hodinám $\mathbf{C_b}$ se vypočítá podle vzorce (5) jako derivace rozdílu mezi hodinami $\mathbf{C_a}$ a $\mathbf{C_b}$, kde $\mathbf{t} \geq 0$. Tento výraz můžeme dále rozvést, čímž dostaneme, že zkreslení hodin je rozdíl frekvencí jednotlivých hodin $\mathbf{C_a}$ a $\mathbf{C_b}$.

$$s[C_a] = (C_a(t) - C_b(t))' = (C_a'(t) - C_b'(t)) \tag{5}$$

Drift hodin **d[C]** je druhá derivace hodin **C** v čase (6), kde $\mathbf{t} \geq 0$.

$$d[C] = C''(t) \tag{6}$$

Drift $\mathbf{d[C_a]}$ hodin $\mathbf{C_a}$ ve vztahu k hodinám $\mathbf{C_b}$, kde $\mathbf{t} \geq 0$ se vypočítá jako druhá derivace rozdílu hodin $\mathbf{C_a}$ a $\mathbf{C_b}$:

$$d[C_a] = (C_a(t) - C_b(t))'' = (C_a''(t) - C_b''(t)) \tag{7}$$

Poměr hodin (clock ratio) $\mathbf{p[C_aC_b]}$ udává poměr mezi frekvencemi jednotlivých hodin $\mathbf{C_a}$ a $\mathbf{C_b}$, kde $\mathbf{t} \geq 0$. Pokud tento vztah vyjádříme, dostaneme následující vzorec:

$$p[C_aC_b] = C_a'(t) / C_b'(t) \tag{8}$$

Nechť $\mathbf{C_a}$ a $\mathbf{C_b}$ mají konstantní frekvence, $\boldsymbol{\alpha}$ je poměr hodin a $\boldsymbol{\delta}$ je zkreslení hodin $\mathbf{C_b}$ ve vztahu k hodinám $\mathbf{C_a}$, respektive:

$$\alpha = C_b'(t) / C_a'(t) \tag{9}$$
$$\delta = C_b'(t) - C_a'(t) \tag{10}$$

Potom vztah mezi poměrem hodin $\boldsymbol{\alpha}$ a zkreslením hodin $\boldsymbol{\delta}$ je:

$$\delta = C_b'(t) - C_a'(t) = \alpha C_a'(t) - C_a'(t) = (\alpha - 1)C_a'(t) \tag{11}$$

Moon, Skelly a Towsley [7] v jejich práci uvedli, že v okamžiku, kdy hodiny $\mathbf{C_a}$ a $\mathbf{C_b}$ běží s odlišnými frekvencemi, doba změřená hodinami $\mathbf{C_a}$ bude odlišná od doby změřené hodinami $\mathbf{C_b}$. Pokud budou mít hodiny stejnou frekvenci, ale budou mít různý rozdíl (offset) změřená doba hodinami $\mathbf{C_a}$ bude shodná s dobou změřenou hodinami $\mathbf{C_b}$. V případě, že hodiny budou mít nenulové zkreslení, doba změřená hodinami $\mathbf{C_a}$ nebude shodná s dobou změřenou hodinami $\mathbf{C_b}$.

2.2 Hardwarové hodiny počítače

Každý osobní počítač obsahuje dvoje hodiny, které běží nezávisle na sobě. Vestavěné hardwarové hodiny a systémové softwarové hodiny.

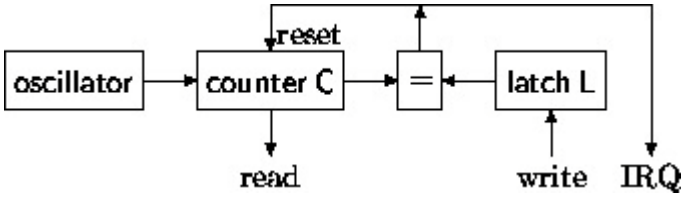
Hardwarové hodiny jsou založeny na zařízení CMOS, které spotřebuje velmi málo energie. Když je počítač vypnutý, běží na baterii. Když se počítač opět zapne, softwarové hodiny se spustí a nastaví, s přesností do jedné sekundy, svůj čas podle času vestavěných hardwarových hodin. Ačkoli jsou hodiny synchronizovány při spuštění PC, mohou dále běžet s velmi rozdílnými hodnotami a budou se pravděpodobně vzhledem k sobě předbíhat nebo zpožďovat, když je počítač spuštěn.

Hardwarové hodiny se aktualizují jednou za sekundu, a nemohou udávat přesnější jednotky než sekundy. Z tohoto důvodu je není možné číst nebo nastavit do lepšího rozlišení. Přesnost hardwarových hodin je určena kvalitou jejich oscilátorů (32,768 kHz krystalu). Tyto krystaly jsou citlivé na změny teploty a jiné faktory, a často nejsou kalibrovány ve výrobě. I za nejlepších podmínek, nejsou tyto oscilátory stabilnější než 1 ppm (0,1 sekundy za den). Ve skutečném provozu se většina hardwarových hodin předbíhá nebo ztrácí v čase v rozmezí 1 až 15 sekund za den. I když jsou hardwarové hodiny obvykle značně lepší než softwarové hodiny, tak i přesto nejsou hardwarové hodiny přesným měřičem času.

2.3 Systémové hodiny počítače

Systémové (virtuální neboli softwarové) hodiny představují reálný čas a běží pouze po dobu, kdy je počítač zapnutý. Systémové hodiny představují 24-hodinový časovač, který nemá reálný koncept dní, ale ve skutečnosti udává čas a datum. Systémové hodiny nemají tvar tradičních hodin, minut a sekund. Operační systém, který je závislý na systémových hodinách, převádí hodnotu načítanou čítačem na hodiny, minuty a sekundy.

Typická struktura hardwaru systémových hodin je na obrázku 2.1. Počítačové hodiny obsahují nějaký typ referenčního oscilátoru stabilizovaného křemíkovým krystalem nebo nějakým jiným typem krystalu. Oscilační frekvence je většinou upravena pomocí před-děličky na vhodnou frekvenci jako např. 1 MHz nebo 100 Hz. Oscilátor generuje hodinové impulsy, které jsou čítány v čítači C. Pokaždé, když čítač dosáhne programovatelné hranice L (načítá se L impulsů) je generováno přerušení nebo jinými slovy také tzv. „tick“. Procesor reaguje na přerušení zvýšením hodnoty softwarového čítače a resetováním hardwarového čítače C. Systémové hodiny jsou tedy založeny na pravidelných hardwarových přerušeních, díky nimž jádro udržuje aktuální čas při každém přerušení.



Obr. 2.1: Hardwarová struktura systémových hodin.

Pokud jde o datum, operační systém čte datum prostřednictvím BIOSu během inicializace počítače.

Tyto softwarové hodiny jsou užitečné, ale mají několik omezení. Softwarové hodiny jsou špatný časoměřič. Jejich přesnost je omezena stabilitou žádostí o přerušení. Jakákoli změna v požadavku na přerušení způsobí, že hodiny budou napřed nebo pozadu. V tomto důsledku, pokud necháme počítač zapnutý po dlouhou dobu, mohou být softwarové hodiny o velké množství času pozadu – možná o minutu nebo více pro každý den. Pokud měřící zařízení dokáže zjistit hodnoty ze systémových hodin **C** sledovaného zařízení v několika bodech v čase, měřící zařízení tak dokáže zjistit zkreslení těchto hodin **s[C]** sledovaného zařízení.

Můžeme tedy říci, že softwarové ani hardwarové hodiny nejsou vhodné pro přesné měření času. Existují ještě virtuální hodiny generující časová razítka v protokolu TCP, ale ty budou blíže popsány v kapitole o TCP protokolu.

3 Zkreslení hodin

V této kapitole popíšeme, jak lze vypočítat zkreslení hodin. Moon, Skelly a Towsley [7] v jejich práci uvedli, že pokud poměr hodin (clock ratio) mezi hodinami $\mathbf{C_a}$ a $\mathbf{C_b}$ bude větší nebo menší než 1, rozdíl (offset) hodin se pak bude zvětšovat nebo zmenšovat, v průběhu celého měření. Pro odstranění tohoto vlivu na měření rozdílu hodin je potřeba transformovat měření rozdílu hodin tak, aby bylo měření konzistentní s jedinými hodinami. Jinými slovy, aby doba změřená hodinami $\mathbf{C_a}$ s frekvencí $\mathbf{f_1}$ byla shodná (konzistentní) s dobou změřenou hodinami $\mathbf{C_b}$ s frekvencí $\mathbf{f_2}$. Proto musíme určit jedny z těchto hodin jako ukazatel přesného skutečného času, respektive určit k jakým referenčním hodinám se bude naše měření vztahovat. Určíme si tedy jedny z těchto hodin jako ukazatel přesného skutečného času např. $\mathbf{C_b}$. Moon, Skelly a Towsley [7] uvedli, že frekvence těchto skutečných přesných hodin odpovídá určité konstantě např. $\mathbf{C_b'(t) = 1}$ Hz, čímž při dosazení do vztahu udávající poměr frekvencí mezi dvěma hodinami (9) dostaneme $\boldsymbol{\alpha} = \mathbf{1 / C_a'}$.

V sekci 2.1 jsme uvedli, že zkreslení hodin (clock skew) je rozdíl frekvencí mezi dvěma hodinami neboli také první derivace rozdílu mezi dvěma hodinami. Pokud tedy máme k dispozici dva zdroje hodin $\mathbf{C_a}$ a $\mathbf{C_b}$ udávající čas $\mathbf{C_a(t)}$ a $\mathbf{C_b(t)}$, tak jsme schopni mezi těmito hodinami vypočítat rozdíl a jeho následnou derivací zkreslení hodin. Výše jsme uvedli, že musíme zvolit referenční hodiny, které budou sloužit jako zdroj skutečného přesného času a ke kterým bude měření konzistentní. Těmito hodinami si zvolíme hodiny $\mathbf{C_b}$. Dále také z předchozí části víme, že frekvence těchto hodin se rovná určité konstantě např. $\mathbf{C_b'(t) = 1}$ Hz. Pokud toto aplikujeme na vztah $\boldsymbol{\alpha} = \mathbf{C_a' / C_b'}$ udávající poměr frekvencí mezi hodinami dostaneme $\boldsymbol{\alpha} = \mathbf{C_a'}$. Jinými slovy frekvenci námi zvolených referenčních hodin $\mathbf{C_b}$ můžeme zanedbat.

3.1 Výpočet frekvence hodin

Proto, abychom mohli vypočítat zkreslení hodin, musíme nejprve znát frekvenci hodin $\mathbf{C_a}$. Jak jsme uvedli v sekci 2.1 frekvence hodin je první derivace funkce hodin v čase $\mathbf{C_a'(t)}$. Funkce hodin je lineární funkce ve tvaru $y = ax + b$, kde y odpovídají hodnoty hodin $\mathbf{C_a(t)}$ a x odpovídají hodnoty hodin $\mathbf{C_b(t)}$. První derivací této funkce tedy dostaneme tvar $y' = a$, kde a [Hz] udává frekvenci $\mathbf{C_a'(t)}$ hodin $\mathbf{C_a}$. Vypočítanou frekvenci hodin $\mathbf{C_a}$ použijeme níže pro výpočet zkreslení hodin.

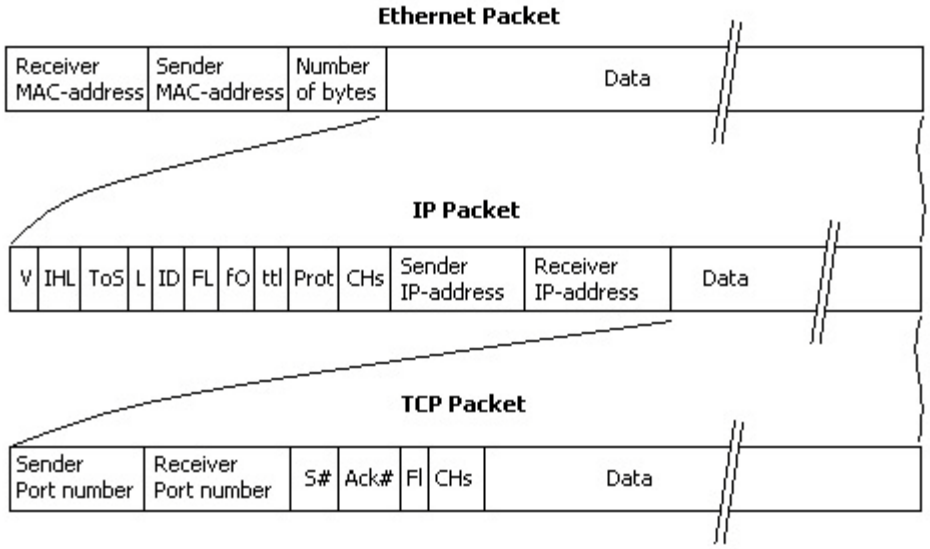
3.2 Výpočet zkreslení hodin

Pokud tedy máme k dispozici hodnoty hodin $\mathbf{C_a(t)}$ a $\mathbf{C_b(t)}$, tak jsme schopni vypočítat zkreslení mezi těmito hodinami.

Abychom však mohli vypočítat rozdíl mezi těmito hodinami a následně zkreslení hodin, musíme hodnoty hodin převést na společnou frekvenci. Jako referenční hodiny jsme si zvolili hodiny $\mathbf{C_b}$, proto musíme hodnoty hodin $\mathbf{C_a(t)}$ převézt na frekvenci hodin $\mathbf{C_b}$. Hodnoty získané z hodin $\mathbf{C_a(t)}$ podělíme jejich frekvencí $\mathbf{C_a'(t)}$. Postup pro zjištění frekvence hodin jsme popsali v předchozí sekci 3.1. Nyní máme k dispozici hodnoty hodin o stejné frekvenci a můžeme tak vypočítat rozdíl mezi nimi (2). Vypočítaný rozdíl hodin derivujeme, čímž vypočítáme zkreslení hodin (5).

4 TCP protokol

V této kapitole stručně popíšeme transportní protokol TCP, jeho povinné a volitelné položky. TCP protokol je jedním ze základních protokolů sady protokolů Internetu, konkrétně představuje transportní vrstvu. Základní jednotkou přenosu v protokolu TCP je TCP segment (viz Obr. 4.2) neboli paket (viz Obr. 4.1). TCP segment se vkládá do IP datagramu. IP datagram se vkládá do linkového rámce. Protokol TCP je spojovanou službou (connection oriented), tj. službou která mezi dvěma aplikacemi naváže spojení – vytvoří na dobu spojení virtuální okruh. Tento okruh je plně duplexní (data se přenášejí současně na sobě nezávisle oběma směry). Přenášené bajty jsou číslovány. Ztracená nebo poškozená data jsou znovu vyžádána. Integrita přenášených dat je zabezpečena kontrolním součtem. Jinými slovy aplikace používající protokol TCP nemusí kontrolovat, zdali náhodou nebyla nějaká data během přenosu ztracena nebo díky chybě přenosu modifikována o to se postará protokol TCP. Toto zabezpečení je účinné pouze proti poruchám technických prostředků. Neklade si za cíl zabezpečit data proti inteligentním útočníkům, kteří mohou data modifikovat a současně také přepočítat kontrolní součet. Ochranou přenášených dat proti takovýmto cíleným útokům se v rodině protokolů TCP/IP zabývají protokoly vyšší vrstvy (SSL, S/MIME). Protokol garantuje spolehlivé doručování a doručování ve správném pořadí. TCP také rozlišuje data pro vícenásobné, současně běžící aplikace (např. webový server a emailový server) běžící na stejném počítači.



Obr. 4.1: Struktura paketu [4].

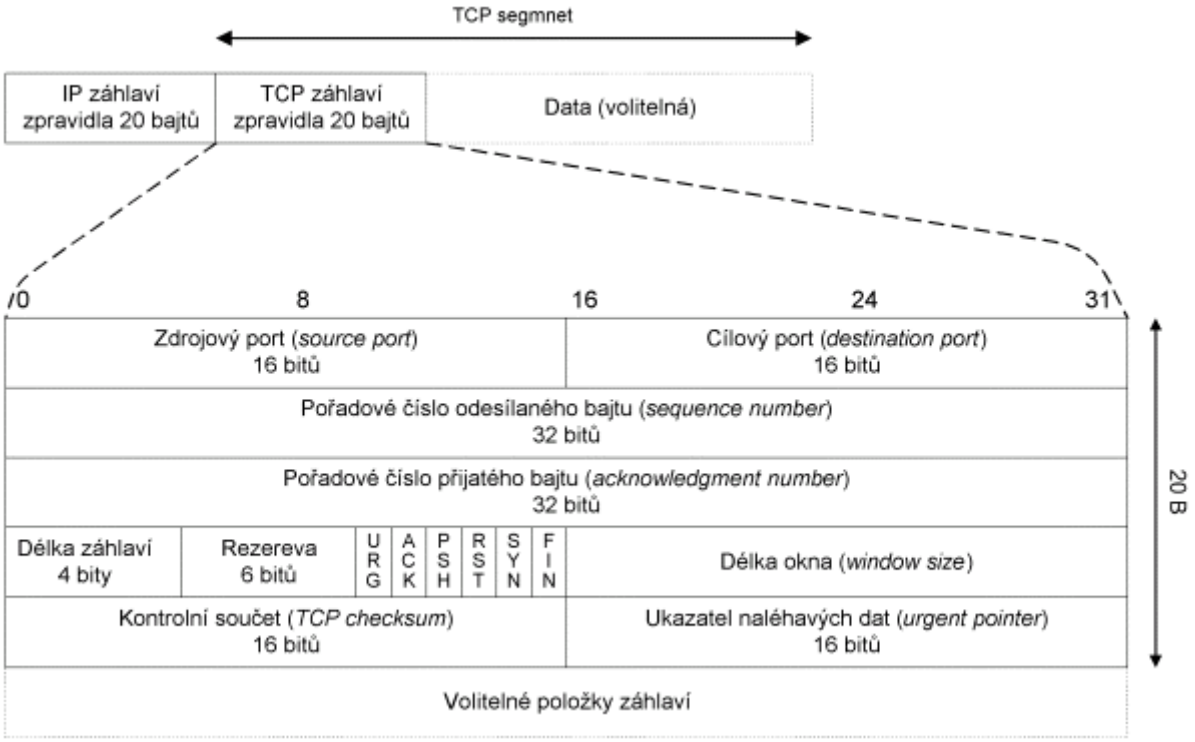
Konce spojení ("odesílatel" a „adresát") jsou určeny tzv. číslem portu. Toto číslo je dvojbajtové, takže může nabývat hodnot 0 až 65535. U čísel portů se často vyjadřuje okolnost, že se jedná o porty protokolu TCP tím, že se za číslo portu napíše lomítko a název protokolu (číslo portu/tcp). Pro protokol UDP je jiná sada portů než pro protokol TCP (též 0 až 65535), tj. např. na portu 53/tcp může být spuštěna jiná aplikace (služba), než na portu 53/udp.

Cílová aplikace (služba) je v Internetu adresována (jednoznačně určena) IP adresou, číslem portu daného transportního protokolu a použitým transportním protokolem (TCP, UDP nebo jiným). Protokol IP dopraví IP datagram na konkrétní počítač. Na tomto počítači běží jednotlivé aplikace (služby). Podle čísla cílového portu transportního protokolu operační systém pozná, které aplikaci má TCP segment doručit.

4.1 TCP segment

V této sekci popíšeme strukturu TCP segmentu, kterou zobrazuje obrázek 4.2. Segment TCP obsahuje několik povinných položek, které blíže popíšeme níže. Celková velikost těchto povinných položek je 20 bajtů.

Zdrojový port (source port) je port odesílatele TCP segmentu, **cílový port** (destination port) je portem adresáta TCP segmentu. Pětice: zdrojový port, cílový port, zdrojová IP-adresa, cílová IP-adresa a protokol (TCP) jednoznačně identifikuje v daném okamžiku spojení (síťový tok) v Internetu. **Pořadové číslo odesílaného bajtu** (sequence number) je pořadové číslo prvního bajtu TCP segmentu v toku dat od odesílatele k příjemci (TCP segment nese bajty od **pořadového čísla odesílaného bajtu** až do délky segmentu). Tok dat v opačném směru má samostatné (jiné) číslování svých dat. Jelikož pořadové číslo odesílaného bajtu je 32 bitů dlouhé, tak po dosažení hodnoty 2^{32} - 1 nabude cyklicky opět hodnoty nula. Číslování obecně nezačíná od nuly (ani od nějaké určené konstanty), ale číslování by mělo začínat od náhodně zvoleného čísla. Vždy když je nastaven příznak **SYN**, tak operační systém odesílatele začíná znovu číslovat, tj. vygeneruje startovací pořadové číslo odesílaného bajtu, tzv. ISN (Initial Sequence Number). Naopak **pořadové číslo přijatého bajtu** (acknowledgment number) vyjadřuje číslo následujícího bajtu, který je příjemce připraven přijmout, tj. příjemce potvrzuje, že správně přijal vše až do **pořadového čísla přijatého bajtu** mínus jedna. **Délka záhlaví** (data offset) vyjadřuje délku záhlaví TCP segmentu v násobcích 32 bitů (4 bajtů). **Délka okna** (window size) vyjadřuje přírůstek **pořadového čísla přijatého bajtu**, který bude příjemcem ještě akceptován. **Ukazatel naléhavých dat** (urgent pointer) může být nastaven pouze v případě, že je nastaven příznak **URG**. Přičte-li se tento ukazatel k **pořadovému číslu odesílaného bajtu**, pak ukazuje na konec úseku naléhavých dat. Odesílatel si přeje, aby příjemce tato naléhavá data přednostně zpracoval.



Obr. 4.2: Struktura TCP segmentu [4].

V poli příznaků mohou být nastaveny následující příznaky:

- **URG** – TCP segment nese naléhavá data.
- **ACK** – TCP segment má platné pole **pořadové číslo přijatého bajtu** (nastaven ve všech segmentech, kromě prvního segmentu, kterým klient navazuje spojení).
- **PSH** – Zpravidla se používá k signalizaci, že TCP segment nese aplikační data, příjemce má tato data předávat aplikaci. Použití tohoto příznaku není ustáleno.
- **RST –** Odmítnutí TCP spojení.
- **SYN** – Odesílatel začíná s novou sekvencí číslování, tj. TCP segment nese pořadové číslo prvního odesílaného bajtu (ISN).
- **FIN** – Odesílatel ukončil odesílání dat. Pokud bychom použili přirovnání k práci se souborem, pak příznak FIN odpovídá konci souboru (EOF). Přijetí TCP segmentu s příznakem FIN, ale neznamená, že v opačném směru není dále možný přenos dat. Jelikož protokol TCP vytváří plně duplexní spojení, tak příznak FIN způsobí jen uzavření přenosu dat v jednom směru. V tomto směru už dále nebudou odesílány TCP segmenty obsahující příznak PSH (nepočítaje v případné opakovaní přenosu).

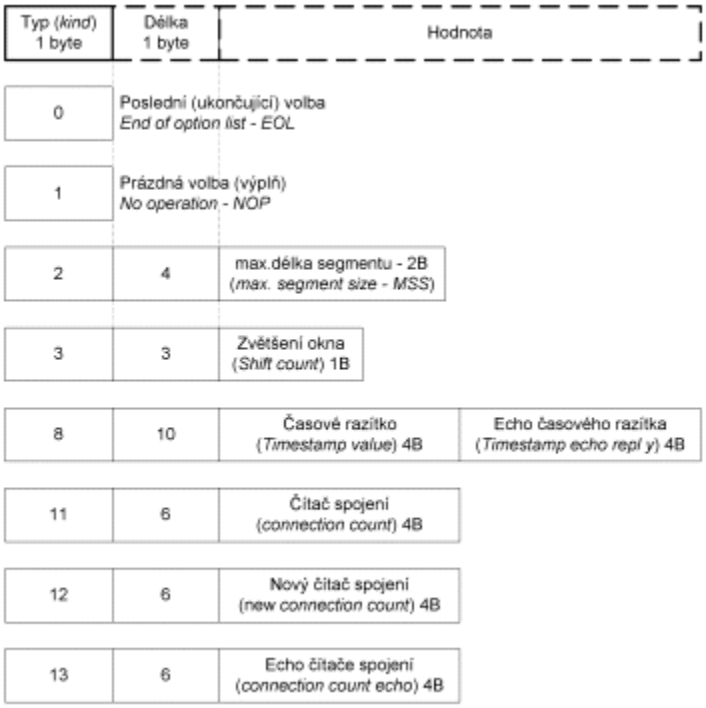
Poslední povinnou položkou TCP segmentu je **kontrolní součet**. Z hlediska zabezpečení integrity přenášených dat je důležitý kontrolní součet v záhlaví TCP segmentu počítaný i z přenášených dat. Tento kontrolní součet se počítá nejen ze samotného TCP segmentu, ale i z některých položek IP-záhlaví. Kontrolní součet vyžaduje sudý počet bajtů, proto v případě lichého počtu se data fiktivně doplní jedním bajtem na konci.

4.2 Volitelné položky TCP záhlaví

Povinné položky TCP záhlaví tvoří 20 bajtů. Za povinnými položkami následují volitelné položky. Volitelná položka se skládá z typu volitelné položky, délky volitelné položky a hodnoty. Délka TCP záhlaví musí být dělitelná čtyřmi. V případě, že délka záhlaví by nebyla dělitelná čtyřmi, pak se záhlaví doplňuje prázdnou volitelnou položkou NOP.

Jelikož pole délka záhlaví je pouze 4 bity dlouhé, tak toto pole může nabývat maximálně hodnoty $1111_2 = 15_{10}$. Délka záhlaví se udává v násobcích čtyř, tudíž záhlaví může být dlouhé maximálně 15x4 = 60 bajtů. Povinné položky zaberou 20 bajtů, takže na volitelné zbývá nejvýše 40 bajtů. Některé volby TCP záhlaví včetně jejich struktury jsou uvedeny na obrázku 4.3.

TCP záhlaví může obsahovat několik volitelných položek, ale v této práci se zaměříme pouze na volitelnou položku označovanou jako **časové razítko** (timestamp).



Obr. 4.3: Některé volitelné položky TCP záhlaví [4].

4.3 Časové razítko

Časové razítko je hodnota udávající počet načítaných tiků hodin počítače. Používá se především pro výpočet RTT (Round-Trip Time), dále se využívá pro ochranu proti přetečení pořadového čísla paketu PAWS (Protection against Wrapped Sequence Numbers) a také je možné pomocí časových značek vypočítat systémový čas a čas spuštění systému počítače (boot time).

Timestamp option obsahuje dvě čtyř-bajtové pole (viz Obr. 4.4) TSval (timestamp value) a TSecr (timestamp echo reply). Pro jednoduchost, je spojené časové razítko a odpověď časového razítka do jednoho pole TCP timestamp option. Hodnota pole TSval obsahuje aktuální počet tiků hodin počítače, které generují časové razítko při odeslání paketu. Hodnota pole TSecr je vyplněna pouze v případě, že je nastaven bit **ACK** v hlavičce TCP. Pokud je tato podmínka splněna pole TSecr (neboli echo časového razítka) obsahuje hodnotu pole TSval, která přišla v předchozím paketu, na který reaguje acknowledge paket (viz Obr. 4.5). Pokud není nastaven **ACK** bit v hlavičce TCP tak hodnota TSecr musí být nula.

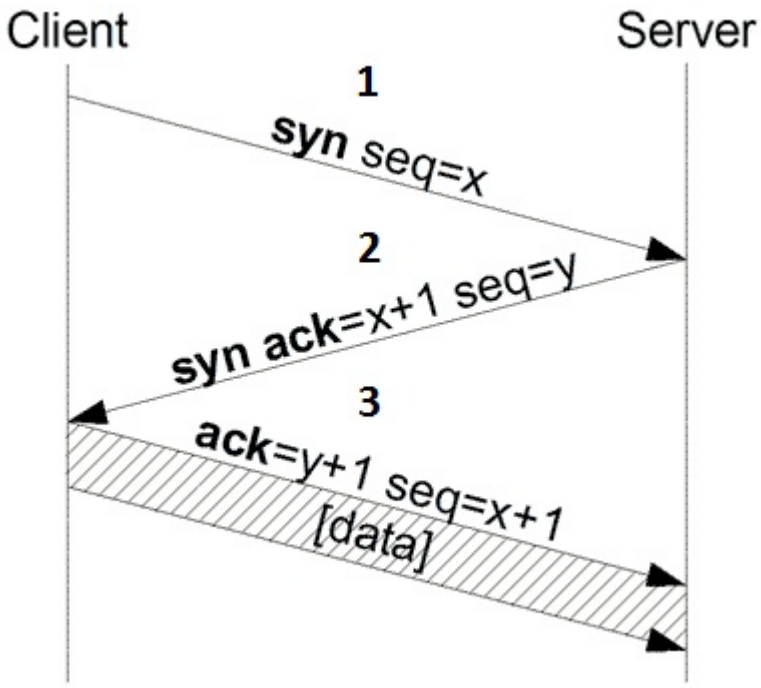
```
TCP Timestamps Option (TSopt):
   Kind: 8
   Length: 10 bytes
    +-------+-------+---------------------+---------------------+
    |Kind=8 |  10   |   TS Value (TSval)  |TS Echo Reply (TSecr)|
    +-------+-------+---------------------+---------------------+
        1B      1B              4B                  4B
```

Obr. 4.4: Struktura volitelné položky časového razítka [9].

TCP je symetrický protokol, který umožňuje odesílání dat kdykoli v obou směrech, a proto k odezvě časového razítka může dojít v obou směrech. RFC 1323 [9] specifikuje TCP timestamp option TCP protokolu a udává, že časová razítka budou vždy zasílány i přijímány v obou směrech.

```
TCP  A                                                TCP B

                <A,TSval=1,TSecr=120> ------>

     <---- <ACK(A),TSval=127,TSecr=1>

                <B,TSval=5,TSecr=127> ------>

     <---- <ACK(B),TSval=131,TSecr=5>
```

Obr. 4.5: Příklad časových značek v síťovém toku [9].



Obr. 4.6: Navázání komunikace mezi klientem a serverem prostřednictvím protokolu TCP.

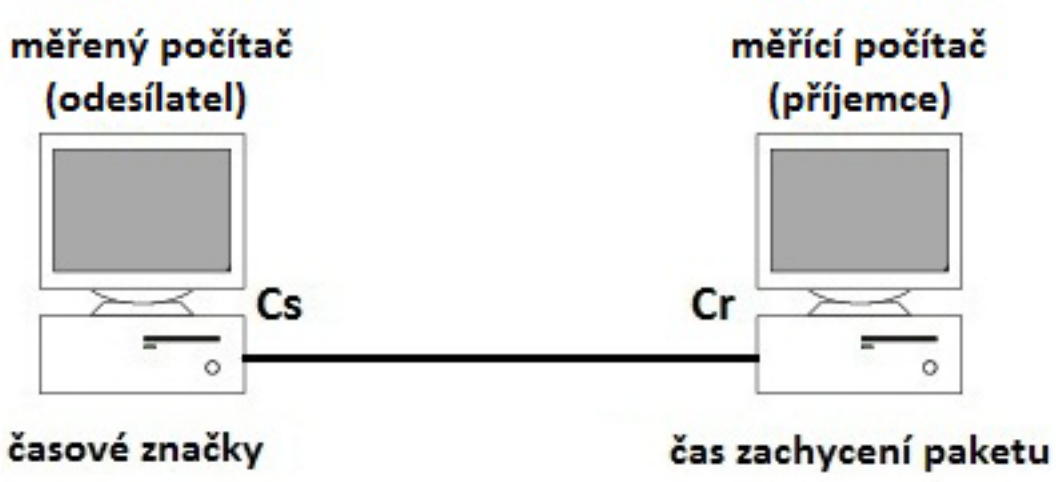
V protokolu TCP je možné odesílat časové razítka v počátečním TCP segmentu, který musí mít nastaven bit **SYN**, ale nesmí mít nastaven bit **ACK**. Dále je možné poslat časové razítko (TSopt) pouze tehdy, kdy obdržel volbu časového razítka (TSopt) v úvodním TCP segmentu s nastaveným **SYN** bitem pro připojení. Jak je tedy vidět z obrázku 4.6 můžeme položku časového razítka vložit do paketu číslo 1, kde klient navazuje spojení se serverem. Pokud splníme tuto podmínku, server i klient budou vkládat položku časového razítka do všech paketů daného spojení.

Pro identifikaci počítače je nejdůležitější, aby položka TCP timestamp option byla uvedena v síťovém toku dat. Jinými slovy, aby každý paket obsahující TCP segment v daném síťovém toku dat obsahoval 32-bitové časové razítko vygenerované tvůrcem paketu. RFC 1323 [9] neudává, odkud se tato hodnota musí vzít, ale udává, že by se časové razítko mělo vzít z „virtuálních hodin“, které „alespoň přibližně udávají skutečný čas“. RFC 1323 [9] PAWS algoritmus stanovuje (sekce 4.2.2), že

rozlišení (perioda) těchto virtuálních hodin má být mezi jednou milisekundou a jednou sekundou. Tyto virtuální hodiny nazýváme jako *TCP timestamps option clock* nebo také jako *TSopt clock* C_{tcp}. Pokud měřící zařízení dokáže zjistit hodnoty z TCP timestamps option clock **C** sledovaného zařízení v několika bodech v čase, potom jsme schopni zjistit zkreslení těchto hodin **s[C]** sledovaného zařízení.

5 Určení zkreslení hodin

V sekci 3.2 jsme uvedli, jak vypočítat zkreslení hodin (clock skew) obecně. V této kapitole uvedeme, jak vypočítat zkreslení hodin v praxi a to konkrétně v reálné počítačové síti mezi dvěma počítači. Tyto počítače označíme jako, příjemce (měřící počítač, ze kterého provádíme měření) a odesílatel (měřený počítač, u kterého chceme zjistit zkreslení hodin). Každý z těchto počítačů obsahuje své vlastní počítačové hodiny, které označíme jako $\mathbf{C_s}$ hodiny odesílatele a $\mathbf{C_r}$ jako hodiny příjemce (viz Obr. 5.1), kde hodnoty hodin odesílatele reprezentují časové značky protokolu TCP $\mathbf{C_s(t)}$, níže tyto hodnoty budeme označovat jako $\mathbf{v_i}$. Hodnoty hodin příjemce $\mathbf{C_r(t)}$ zase udávají aktuální dobu přijetí paketu od odesílatele, níže tyto hodnoty budeme označovat jako $\mathbf{x_i}$. Moon, Skelly a Towsley [7] jako referenční hodiny, které budou sloužit jako zdroj skutečného přesného času a ke kterým bude měření konzistentní, zvolili hodiny příjemce $\mathbf{C_r}$. Ve třetí kapitole jsme uvedli, že frekvence těchto hodin příjemce odpovídá určité konstantě např. $\mathbf{C_r'(t) = 1}$, kterou můžeme zanedbat. Pokud toto aplikujeme na vztah (9) udávající poměr frekvencí mezi dvěma hodinami dostaneme $\boldsymbol{\alpha} = \mathbf{C_s'} / \mathbf{1}$, tedy $\boldsymbol{\alpha} = \mathbf{C_s'}$.

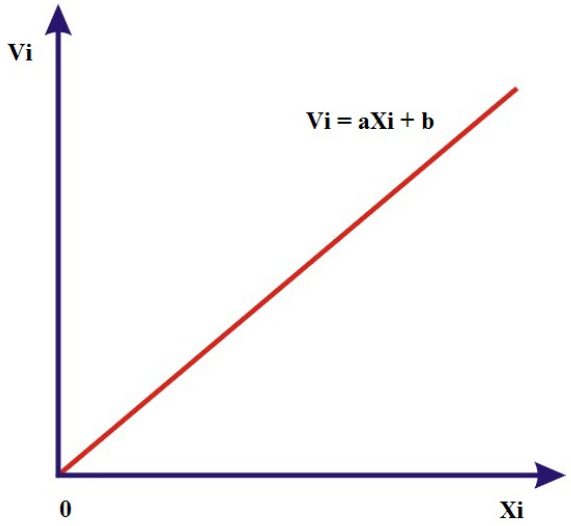


Obr. 5.1: Měřený počítač – Měřící počítač.

5.1 Výpočet frekvence hodin

Ve třetí kapitole jsme uvedli, že proto, abychom mohli vypočítat zkreslení hodin, musíme nejprve znát frekvenci hodin odesílatele. Dále jsme také v sekci 3.1 uvedli, že frekvence hodin je první derivace funkce hodin v čase $\mathbf{C_s'(t)}$. Funkce hodin je lineární funkce ve tvaru $\mathbf{v_i = ax_i + b}$, jak znázorňuje obrázek 5.2, kde $\mathbf{v_i}$ odpovídají hodnoty časových razítek protokolu TCP $\mathbf{v_i}$ (T_i) a $\mathbf{x_i}$ odpovídají hodnoty aktuální doby přijetí paketu $\mathbf{x_i}$ (t_i). První derivací této funkce tedy dostaneme tvar $\mathbf{v_i'} = \mathbf{a}$, kde **a** [Hz] udává frekvenci $\mathbf{C_s'(t)}$ hodin odesílatele $\mathbf{C_s}$. Pokud tedy hodnotami $\mathbf{x_i}$ a $\mathbf{v_i}$ proložíme přímku (12) (derivujeme funkci $\mathbf{v_i = ax_i + b}$), získáme tak směrnici této vzniklé přímky tedy frekvenci hodin odesílatele $\mathbf{C_s'(t)}$. Tuto frekvenci odesílatele si označíme jako $\mathbf{f_s}$.

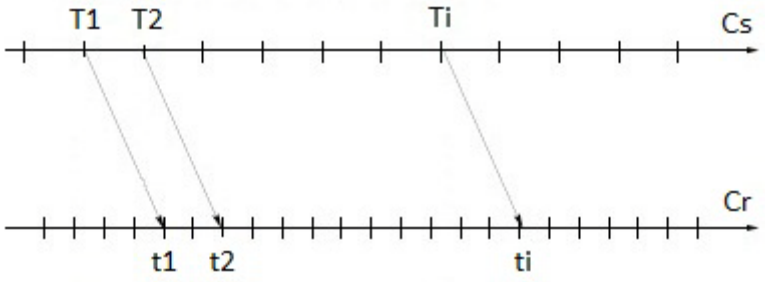
$$f_s = v_i' = (ax_i + b)' \tag{12}$$



Obr. 5.2: Znázornění funkce hodin.

5.2 Výpočet zkreslení hodin

Obrázek 5.3 znázorňuje, jak hodiny odesílatele $\mathbf{C_s}$ generují časová razítka $\mathbf{T_i}$. Tato jednotlivá časová razítka vkládá odesílatel do paketů, konkrétně do volitelné položky TCP segmentu, které následně odešle příjemci. Příjemce si pomocí jeho hodin $\mathbf{C_r}$ zaznamená dobu přijetí paketu jako hodnotu $\mathbf{t_i}$.



Obr. 5.3: Znázornění hodin příjemce a odesílatele [6].

- $\mathbf{C_s}$: hodiny odesílatele (měřeného zařízení).
- $\mathbf{C_r}$: hodiny příjemce (měřícího zařízení).
- **N**: počet paketů, které přijme příjemce od odesílatele.
- **|N|**: počet paketů obsahující časové značky, které přijme příjemce od odesílatele.
- $\mathbf{t_i}$: udává čas v sekundách, kdy příjemce obdržel i-tý paket od odesílatele, kde $i \in \{1,\ldots, |N|\}$.
- $\mathbf{T_i}$: udává časové razítko protokolu TCP vložené odesílatelem do i-tého paketu, kde $i \in \{1,\ldots, |N|\}$.

Kohno, Broido a Claffy [7] představili následující postup pro výpočet zkreslení hodin pomocí časových značek paketů protokolu TCP. Vypočítaná hodnota $\mathbf{x_i}$ [s] z rovnice (13) udává časový údaj od počátku měření (od času příchodu prvního paketu $\mathbf{t_1}$) po zaznamenání času příchodu i-tého paketu $\mathbf{t_i}$.

$$x_i = t_i - t_1 : i \in \{1,\ldots, |N|\} \qquad (13)$$

Hodnota $\mathbf{v_i}$ vypočítaná pomocí rovnice (14) udává počet tiků hodin odesílatele o frekvenci hodin odesílatele od počátku měření (od příchodu prvního paketu udávajícího počet tiků hodin odesílatele $\mathbf{T_1}$) po zaznamenání i-tého paketu udávajícího počet tiků hodin odesílatele $\mathbf{T_i}$.

$$v_i = T_i - T_1 : i \in \{1,\ldots, |N|\} \tag{14}$$

Počet tiků hodin $\mathbf{v_i}$ z předchozí rovnice (14) udává počet tiků hodin odesílatele o frekvenci hodin odesílatele. Pro správný výpočet zkreslení hodin je nutné vypočítat počet tiků hodin odesílatele o frekvenci hodin odpovídající hodinám příjemce. Výše jsme ovšem uvedli, že frekvenci hodin příjemce můžeme zanedbat, proto pouze podělíme počet tiků hodin odesílatele $\mathbf{v_i}$ frekvencí hodin odesílatele $\mathbf{f_s}$ (15). Výpočet frekvence hodin odesílatele $\mathbf{f_s}$ je uveden v sekci 5.1. Hodnota $\mathbf{w_i}$ tedy udává počet tiků hodin odesílatele odpovídající frekvenci hodinám příjemce.

$$w_i = v_i / f_s : i \in \{1,\ldots, |N|\} \tag{15}$$

Pro výpočet zkreslení hodin je třeba vypočítat rozdíl mezi hodinami odesílatele a hodinami příjemce. K tomu použijeme vzorec (16), kde pomocí rozdílu hodnot $\mathbf{w_i}$ udávajících počet tiků hodin odesílatele a hodnot $\mathbf{x_i}$ udávajících časy zachycení paketů hodin příjemce vypočítáme rozdíl (offset) mezi těmito hodinami $\mathbf{y_i}$.

$$y_i = w_i - x_i : i \in \{1,\ldots, |N|\} \tag{16}$$

Postupným výpočtem předešlých vzorců (13) (14) (15) a (16) jsme získali množinu σ (17), která obsahuje hodnoty $\mathbf{x_i}$ reprezentující časové údaje a hodnoty $\mathbf{y_i}$ reprezentující rozdíl mezi hodinami odesílatele a hodinami příjemce. Pokud těmito hodnotami proložíme přímku, respektive vypočítáme první derivaci rozdílu hodin v čase a tím tak získáme směrnici **a** vzniklé přímky pomocí vzorce (18), dostaneme tak zkreslení hodin (clock skew) odesílatele, kde **a** [-] odpovídá zkreslení hodin.

$$\sigma = \{(x_i, y_i) : i \in \{1,\ldots, |N|\}\} \tag{17}$$

$$y_i' = (ax_i + b)' \tag{18}$$

Vypočítané zkreslení hodin je tedy unikátní identifikátor měřeného zařízení (počítače).

6 Popis programu

Implementovaný program *clockskew* umožňuje uživateli zjistit zkreslení hodin počítače. Program je možné spustit v několika módech:

1. Mód 1 – programu se předají dva vstupní parametry, z nichž první udává název rozhraní, na kterém budeme zaznamenávat síťovou komunikaci, a druhý parametr udává počet paketů, který chceme zachytit. Zaznamenání síťové komunikace a její následné uložení do souboru zprostředkovává program tcpdump [11]. Zaznamenaná síťová komunikace je dále zpracována a postup zpracování je popsán níže. Výsledné zkreslení hodin se počítá ze všech síťových toků pro danou zdrojovou IP adresu dohromady.
2. Mód 2 - funguje stejně jako mód 1 s tím rozdílem, že se programu navíc předá další vstupní přepínač. Výsledné zkreslení hodin se potom počítá pro každý síťový tok zvlášť.
3. Mód 3 - umožňuje, při spuštění programu, jako vstupní parametr zadat jméno již zaznamenaného souboru v PCAP formátu (např. pomocí programu tcpdump) obsahujícího síťovou komunikaci a z tohoto souboru pak následně vypočítat zkreslení hodin pro všechny síťové toky pro danou zdrojovou IP adresu dohromady.
4. Mód 4 - funguje stejně jako mód 3, kde se navíc programu předá další vstupní přepínač. Výsledné zkreslení hodin se potom počítá pro každý síťový tok zvlášť.
5. Mód 5 - slouží pro výpočet zkreslení hodin ze zadaného počtu paketů. První parametr udává zaznamenaný soubor v PCAP formátu obsahující síťovou komunikaci. Druhý parametr udává počet paketů, ze kterého chceme zkreslení hodin vypočítat.

Program využívá knihovnu netbench [10] pro zpracování PCAP souboru. Z každého paketu obsahujícího časové značky protokolu TCP, jsou získány údaje časových značek, konkrétně položka TSval. Tato hodnota je společně s cílovou a zdrojovou IP adresou, s cílovým a zdrojovým TCP portem a s časem přijetí daného paketu uložena do seznamu. Seznam se dále zpracovává podle příslušného síťového toku, kde se pomocí hodnoty časového razítka TCP protokolu a času přijetí paketu vypočítá zkreslení hodin počítače. Zkreslení hodin počítače se vypočítává pomocí algoritmu, který je uveden v sekci 5.2. Po zpracování všech síťových toků je vypsána výsledná tabulka ve tvaru, který je zobrazen dle tabulky 6.1.

Při spuštění programu v módu 2 a v módu 4 je navíc s výstupní tabulkou vypsána i tabulka udávající počet počítačů nacházejících se za danou IP adresou (na dané IP adrese pravděpodobně probíhá překlad adres NAT [14]), která je zobrazena na obrázku 6.1. Program musí být spuštěn při spuštění v módu 1 a v módu 2 na měřícím zařízení (ne na měřeném zařízení).

zdrojová IP adresa:zdrojový TCP port	cílová IP adresa:cílový TCP port	zkreslení hodin počítače [PPM]
192.168.0.105:80	192.168.0.110:60901	71.4497
192.168.0.106:80	192.168.0.110:51461	50.6558
192.168.0.109:80	192.168.0.110:60014	165.4479
192.168.0.112:80	192.168.0.110:58662	89.1852
192.168.0.101:80	192.168.0.110:60113	97.7972

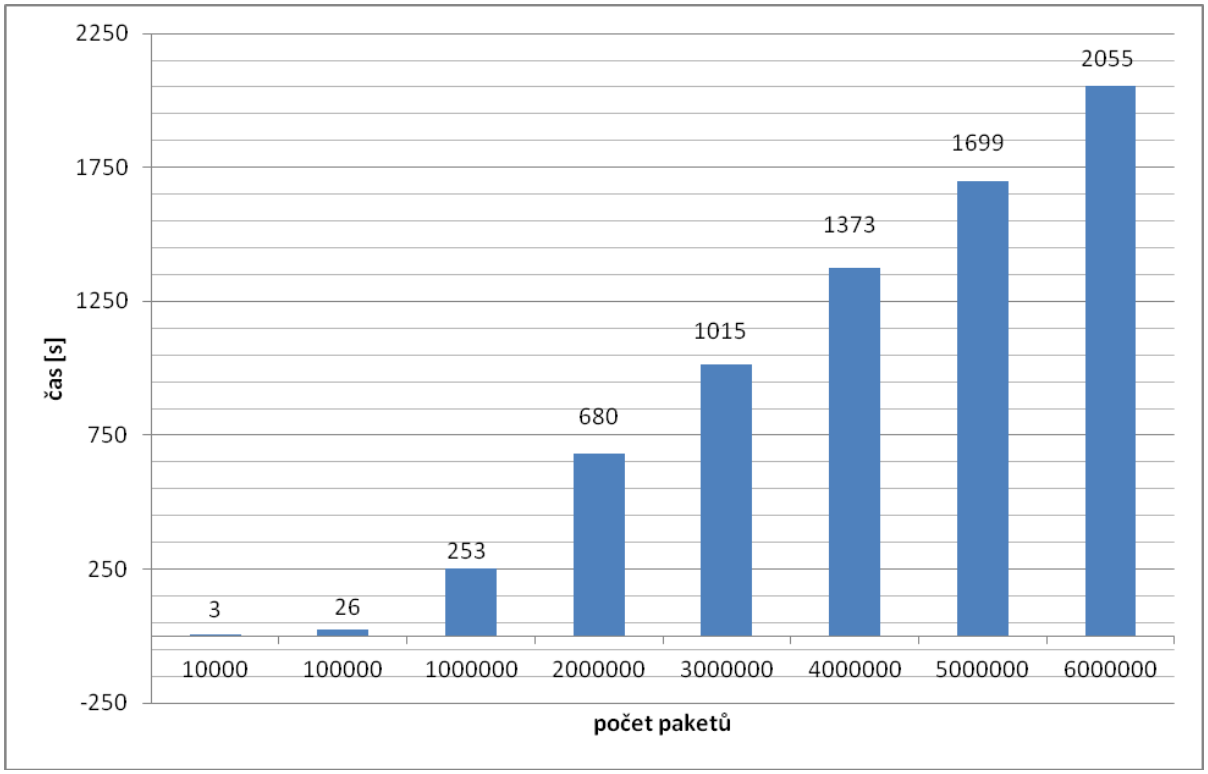
Tab. 6.1: Výstup programu clockskew.

```
Results...

Netflow:        192.168.0.105:445 | 192.168.0.2:44316   Clock skew:     75.3972 PPM
Netflow:        192.168.0.2:44316 | 192.168.0.105:445   Clock skew:     -0.0365 PPM
Netflow:        192.168.0.2:56836 | 192.168.0.105:139   Clock skew:     -23.3927 PPM
Netflow:        192.168.0.2:56837 | 192.168.0.105:139   Clock skew:     7.6491 PPM
Netflow:        192.168.0.2:56838 | 192.168.0.105:139   Clock skew:     None


Counting number hosts...

Source IP address:      192.168.0.105   Number hosts:   1
Source IP address:      192.168.0.2     Number hosts:   3
```

Obr. 6.1: Výstup programu clockskew.

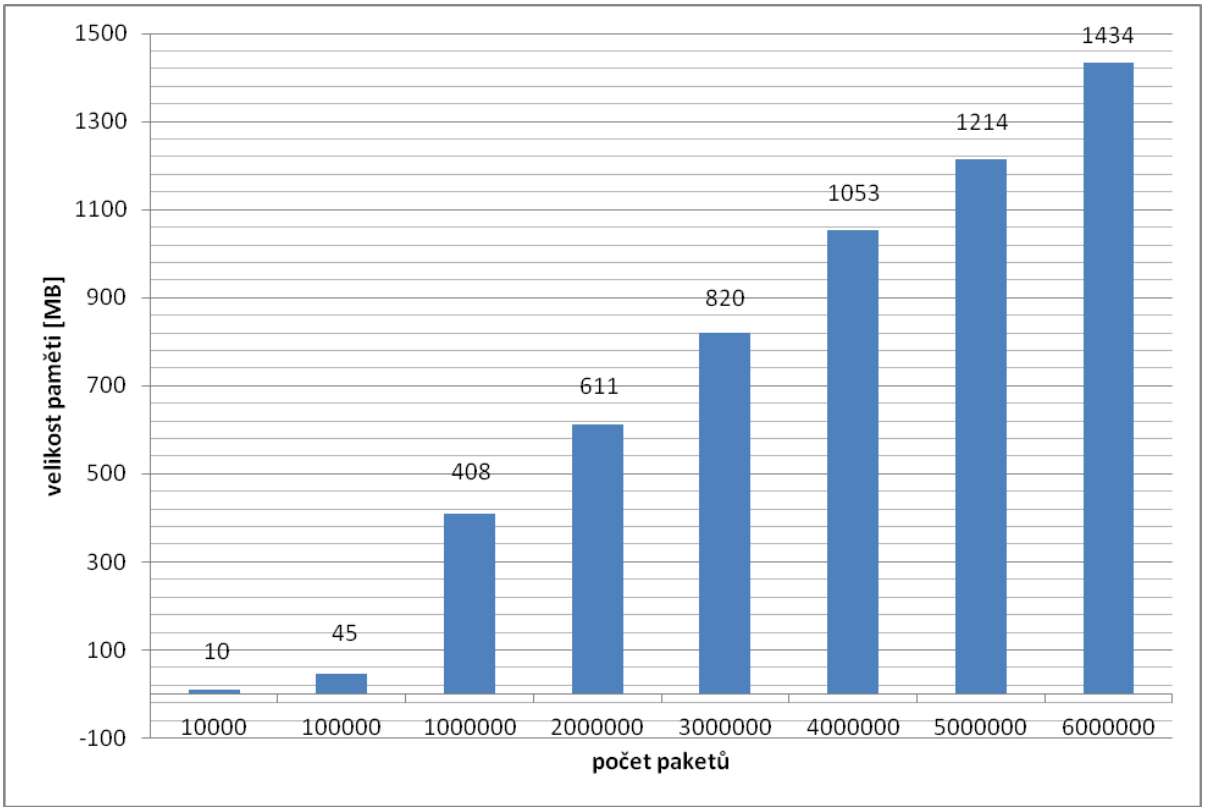
Je potřeba zmínit, že v tabulce budou zobrazeny i ty síťové toky, které směřují z měřícího počítače nebo z počítače na kterém byl zaznamenán síťový tok. Pro tyto toky, ale bude program vracet chybné výsledky, neboť čas přijetí paketu nebude odpovídat času přijetí paketu, ale času odeslání paketu a navíc paket bude obsahovat časová razítka TCP protokolu generovaná hodinami na měřícím počítači. Výsledné zkreslení hodin počítače vypočítané pro takový síťový tok nebude správné. Pokud bude mít měřící počítač IP adresu např. 192.168.0.110 tak potom tabulka 6.1 zobrazuje správné výsledky. Takto způsobenou chybu je možné odstranit již při zaznamenávání síťové komunikace a to tím způsobem, že budeme zaznamenávat pouze příchozí síťovou komunikaci na měřící zařízení.

Program má lineární časovou složitost (viz Graf. 6.1), kde čas vykonávání programu závisí na velikosti vstupních dat (na množství paketů).



Graf. 6.1: Graf zobrazující časovou složitost v závislosti na velikosti vstupu.

Problémem může být paměťová složitost, která je lineární (viz Graf. 6.2) a závisí na velikosti vstupních dat (na množství paketů).



Graf. 6.2: Graf zobrazující paměťovou složitost v závislosti na velikosti vstupu.

7 Praktické výsledky

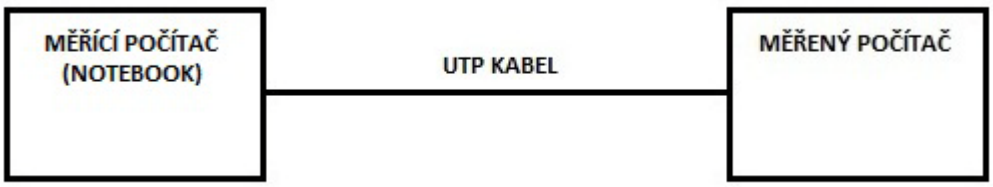
V této sekci budou popsány testy, které byly provedeny pomocí implementovaného programu *clockskew* popsaného v kapitole šest, a dále budou uvedeny a vyhodnoceny dosažené výsledky provedených testů.

7.1 Identifikace počítačů v síti

Pro identifikaci byly použity čtyři počítače, kde tři počítače sloužily jako měřené počítače a jeden počítač jako měřící počítač. V sekci 7.1.1 bylo měřeno zkreslení hodin počítače z jednoho síťového toku na jeden měřený počítač a v sekci 7.1.2 pomocí dvou síťových toků na jeden měřený počítač. Každé měření bylo pro všechny měřené počítače provedeno tři-krát. Účelem testů bylo vypočítat zkreslení hodin měřených (identifikovaných) počítačů, zjistit zda je možné pomocí vypočítaného zkreslení hodin od sebe měřené počítače jednoznačně odlišit a zjistit, jak zkreslení hodin ovlivní měření při daném množství použitých síťových toků.

7.1.1 Pomocí jednoho síťového toku

Jako první test byla provedena, pomocí výše zmíněného programu *clockskew* popsaného v kapitole šest, identifikace několika počítačů v síti pomocí jednoho síťového toku. Měřící počítač, ze kterého bylo prováděné měření, byl notebook s operačním systémem Unix, konkrétně s Ubuntu 11.04. Měřící počítač byl připojen vždy k jednomu z měřených (identifikovaných) počítačů několika způsoby, které budou popsány níže.



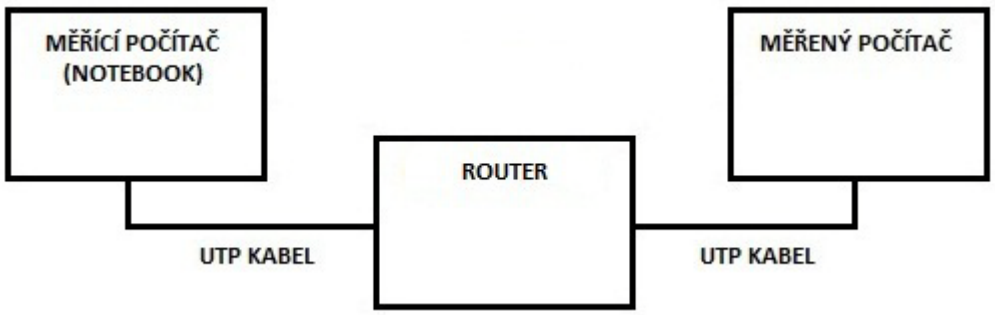
Obr. 7.1: Přímé zapojení (měřící počítač – měřený počítač).

Způsob testování byl následující. Na měřeném počítači byl spuštěn webový server (apache). Pomocí webového prohlížeče na měřícím počítači došlo k připojení na měřený počítač, kde byl k dispozici ke stažení vytvořený referenční soubor, který byl používán v průběhu testování. Tento referenční soubor byl pomocí webového prohlížeče stahován na měřící počítač. Zároveň byl spuštěn vytvořený program *clockskew* v módu 2, který z komunikace mezi počítači (jeden síťový tok), vytvořené v důsledku stahování souboru, vypočítal zkreslení hodin měřeného počítače. Tento způsob testování byl pro všechny tři zapojení a měřené počítače stejný.

měřený počítač	počet paketů obsahující časové značky	frekvence hodin měřeného počítače [Hz]	zkreslení hodin [PPM]	maximální odchylka pro jeden síťový tok [%]
PC – 1	1 122 349	100	75.3972	0.6
PC – 1	1 122 351	100	75.4180	
PC – 1	1 114 707	100	75.0086	
PC – 2	1 122 348	100	50.9337	0.3
PC – 2	1 122 349	100	50.8147	
PC – 2	1 122 345	100	50.8548	
PC – 3	1 169 352	10	153.6755	7.5
PC – 3	1 171 083	10	166.1203	
PC – 3	1 171 083	10	157.5429	

Tab. 7.1: Výsledky zkreslení hodin pro přímé zapojení.

Tabulka 7.1 zobrazuje naměřené výsledky pro přímé zapojení (viz Obr. 7.1), kdy byl měřící počítač propojen s měřeným počítačem přímo síťovým kabelem. Z výsledků je patrné, že pro PC – 1 a PC – 2 byly naměřeny téměř neměnné hodnoty, kde se maximální odchylka pohybovala v řádech desetin procent. Ovšem hodnoty naměřené pro PC – 3 se měnily s maximální odchylkou 7,5 %. Vysoká maximální odchylka byla pravděpodobně způsobená nízkou frekvencí hodin PC – 3, která byla 10 Hz. V sekci o přesnosti měření (7.2) bude tento problém popsán blíže.

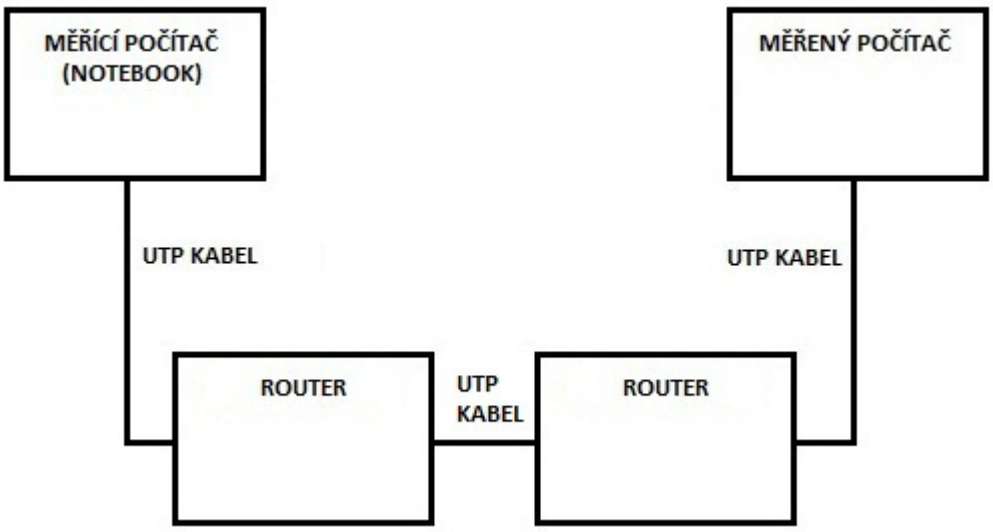


Obr. 7.2: Zapojení s jedním routerem (měřící počítač – router – měřený počítač).

měřený počítač	počet paketů obsahující časové značky	frekvence hodin měřeného počítače [Hz]	zkreslení hodin [PPM]	maximální odchylka pro jeden síťový tok [%]
PC – 1	1 171 080	100	75.4674	0.5
PC – 1	1 171 083	100	75.7975	
PC – 1	1 162 664	100	75.8023	
PC – 2	1 122 347	100	50.4108	0.2
PC – 2	1 122 343	100	50.3118	
PC – 2	1 171 082	100	50.3811	
PC – 3	1 171 080	10	116.4217	33.7
PC – 3	1 171 083	10	147.6430	
PC – 3	1 171 083	10	175.4581	

Tab. 7.2: Výsledky zkreslení hodin pro zapojení s jedním routerem.

Tabulka 7.2 zobrazuje výsledky měření pro zapojení s jedním routerem, kdy byl měřící počítač propojen s měřeným počítačem prostřednictvím aktivního síťového prvku (viz Obr. 7.2). Pro PC – 1 a PC – 2 byly opět naměřeny téměř neměnné hodnoty, kde se maximální odchylka pohybovala v řádech desetin procent. Pro PC – 1 a PC – 2 se ukázalo, že router nemá vliv na výsledné zkreslení hodin. Hodnoty naměřené pro PC – 3 měly maximální odchylku 33,7 %. Velikost odchylky u PC – 3 souvisí s frekvencí hodin (viz sekce 7.2).



Obr. 7.3: Zapojení se dvěma routery (měřící počítač – router – router – měřený počítač).

měřený počítač	počet paketů obsahující časové značky	frekvence hodin měřeného počítače [Hz]	zkreslení hodin [PPM]	maximální odchylka pro jeden síťový tok [%]
PC – 1	1 171 083	100	71.4497	1.9
PC – 1	1 171 083	100	72.8160	
PC – 1	1 171 081	100	71.4562	
PC – 2	1 171 081	100	50.6558	0.4
PC – 2	1 171 082	100	50.4794	
PC – 2	1 171 084	100	50.6559	
PC – 3	1 171 081	10	196.1338	17.5
PC – 3	1 170 592	10	165.4479	
PC – 3	1 171 083	10	161.9558	

Tab. 7.3: Výsledky zkreslení hodin pro zapojení se dvěma routery.

Tabulka 7.3 zobrazuje naměřené výsledky pro zapojení se dvěma routery, kdy byl měřící počítač propojen s měřeným počítačem prostřednictvím dvou aktivních síťových prvků (viz Obr. 7.3). Pro PC – 1 bylo naměřeno zkreslení hodin s maximální odchylkou 1,9 %, což je oproti předchozím měřením o něco vyšší hodnota. Předpokládáme, že příčina zvýšení odchylky může být způsobena větším zatížením routerů. Pro PC – 2 byly naměřeny opět téměř neměnné hodnoty zkreslení hodin, kde se maximální odchylka pohybovala v řádech desetin procent. Hodnoty zkreslení hodin pro PC – 3 byly naměřeny s maximální odchylkou 17,5 %. Velikost odchylky u PC – 3 souvisí s frekvencí hodin (viz sekce 7.2).

Pokud porovnáme výsledky ze všech testů pro jednotlivé počítače a zapojení dohromady (viz Tab. 7.4) zjistíme, že pro PC – 1 se zkreslení hodin pro jednotlivé testy měnilo s maximální odchylkou 5,8 %. Vysokou odchylku u PC – 1 způsobil test pro zapojení se dvěma routery. Pro PC – 2 se zkreslení hodin měnilo s maximální odchylkou 1,3 % a ve všech testech bylo naměřeno jen s minimální odchylkou. Zkreslení hodin pro PC – 3 bylo naměřeno s celkovou maximální odchylkou 40,7 %, která byla způsobena nízkou frekvencí hodin (viz sekce 7.2).

měřený počítač	frekvence hodin měřeného počítače [Hz]	zkreslení hodin [PPM] pro přímé zapojení	zkreslení hodin [PPM] pro zapojení s jedním routerem	zkreslení hodin [PPM] pro zapojení se dvěma routery	maximální odchylka pro jeden síťový tok [%]
PC – 1	100	75.3972	75.4674	71.4497	5.8
PC – 1	100	75.4180	75.7975	72.8160	
PC – 1	100	75.0086	75.8023	71.4562	
PC – 2	100	50.9337	50.4108	50.6558	1.3
PC – 2	100	50.8147	50.3118	50.4794	
PC – 2	100	50.8548	50.3811	50.6559	
PC – 3	10	153.6755	116.4217	196.1338	40.7
PC – 3	10	166.1203	147.6430	165.4479	
PC – 3	10	157.5429	175.4581	161.9558	

Tab. 7.4: Výsledky ze všech měření a jejich maximální odchylky.

Z výsledků zkreslení hodin pro jednotlivé měřené počítače při použití jednoho síťového toku plyne, že lze mezi sebou tyto tři počítače jednoznačně identifikovat.

7.1.2 Pomocí dvou síťových toků

V předchozí sekci byly provedeny testy k identifikaci měřených počítačů pomocí jednoho síťového toku. V této sekci byly provedeny testy k identifikaci měřených počítačů pomocí dvou síťových toků. K měření byl používán měřící počítač uvedený v sekci 7.1.1. Měřené počítače byly s měřícím počítačem propojeny stejným způsobem jako v předchozí sekci 7.1.1.

K výpočtu zkreslení hodin měřeného počítače bylo využito dvou síťových toků, z nichž první z nich byl vytvořen následovně. Na měřeném počítači byl spuštěn webový server (apache). Pomocí webového prohlížeče na měřícím počítači došlo k připojení na měřený počítač, kde byl k dispozici ke stažení nový referenční soubor (jiný než v sekci 7.1.1) o velikosti 4,27 GB, který byl používán v průběhu testování. Tento referenční soubor byl pomocí webového prohlížeče stahován na měřící počítač. Druhý síťový tok byl vytvořen pomocí programu iperf [13]. Na měřícím počítači byl spuštěn program iperf v režimu server. Měřené počítače byly pomocí programu iperf v režimu klient připojovány k serveru vytvořenému na měřícím počítači. Zároveň byl spuštěn vytvořený program *clockskew*, který z komunikace mezi počítači (dva síťové toky), vytvořené v důsledku stahování souboru a programu iperf, vypočítal zkreslení hodin měřeného počítače. Program *clockskew* byl spuštěn v módu 1 proto, aby zkreslení hodin měřeného počítače počítal dohromady ze všech síťových toků směřujících z měřeného počítače k měřícímu počítači. Tento způsob testování byl u všech třech zapojení stejný.

měřený počítač	počet paketů obsahující časové značky	frekvence hodin měřeného počítače [Hz]	zkreslení hodin [PPM]	maximální odchylka pro dva síťové toky [%]	maximální odchylka pro jeden síťový tok [%]
PC – 1	4 000 000	100	74.3934	2.1	0.6
PC – 1	4 000 000	100	75.7883		
PC – 1	4 000 000	100	75.9766		
PC – 2	4 000 000	100	54.0920	0.7	0.3
PC – 2	4 000 000	100	54.3779		
PC – 2	4 000 000	100	54.4446		
PC – 3	4 000 000	10	89.7260	7.9	7.5
PC – 3	4 000 000	10	94.3974		
PC – 3	4 000 000	10	97.3786		

Tab. 7.5: Výsledky zkreslení hodin pro přímé zapojení.

Tabulka 7.5 zobrazuje naměřené výsledky pro přímé zapojení, kdy byl měřící počítač propojen s měřeným počítačem přímo síťovým kabelem (viz Obr. 7.1). Pro PC – 1 bylo naměřeno zkreslení hodin pro dva síťové toky s maximální odchylkou 2,1 %, ale maximální odchylka pro jeden síťový tok byla 0,6 %. U PC – 2 byla naměřena odchylka zkreslení hodin pro dva síťové toky téměř stejná jako u jednoho síťového toku a i pro PC – 3 byla maximální odchylka zkreslení hodin pro dva síťové toky téměř stejná, jako maximální odchylka pro jeden síťový tok. U PC – 3 došlo vlivem

výpočtu zkreslení hodin z většího množství paketů ke snížení zkreslení hodin oproti měření zkreslení hodin v předchozí sekci.

měřený počítač	počet paketů obsahující časové značky	frekvence hodin měřeného počítače [Hz]	zkreslení hodin [PPM]	maximální odchylka pro dva síťové toky [%]	maximální odchylka pro jeden síťový tok [%]
PC – 1	4 000 000	100	75.7158	0.2	0.5
PC – 1	4 000 000	100	75.5831		
PC – 1	4 000 000	100	75.4482		
PC – 2	4 000 000	100	46.7862	0.2	0.2
PC – 2	4 000 000	100	46.7115		
PC – 2	4 000 000	100	46.7020		
PC – 3	4 000 000	10	96.7460	10.3	33.7
PC – 3	4 000 000	10	98.5783		
PC – 3	4 000 000	10	107.7515		

Tab. 7.6: Výsledky zkreslení hodin pro zapojení s jedním routerem.

Tabulka 7.6 zobrazuje výsledky měření pro zapojení s jedním routerem, kdy byl měřící počítač propojen s aktivním síťovým prvkem a ten byl propojen s měřeným počítačem (viz Obr. 7.2). Pro PC – 1 bylo naměřeno zkreslení hodin pro dva síťové toky s maximální odchylkou téměř stejnou jako pro jeden síťový tok. Pro PC – 2 byla naměřena odchylka zkreslení hodin pro dva síťové toky shodně jako pro jeden síťový tok. Pro poslední testovaný počítač PC – 3 byla naměřena maximální odchylka zkreslení hodin pro dva síťové toky několika násobně nižší, než maximální odchylka pro jeden síťový tok. Tento rozdíl způsobilo měření zkreslení hodin ze dvou síťových toků a následným výpočtem zkreslení hodin ze čtyři-krát většího množství paketů než v případě měření zkreslení hodin z jednoho síťového toku. Vliv množství paketů na měření zkreslení hodin bude blíže vysvětlen v sekci 7.2.

měřený počítač	počet paketů obsahující časové značky	frekvence hodin měřeného počítače [Hz]	zkreslení hodin [PPM]	maximální odchylka pro dva síťové toky [%]	maximální odchylka pro jeden síťový tok [%]
PC – 1	4 000 000	100	72.3566	1.0	1.9
PC – 1	4 000 000	100	73.0706		
PC – 1	4 000 000	100	73.0651		
PC – 2	4 000 000	100	46.4011	1.3	0.4
PC – 2	4 000 000	100	46.9137		
PC – 2	4 000 000	100	47.0118		
PC – 3	4 000 000	10	92.1455	8.7	17.5
PC – 3	4 000 000	10	95.2371		
PC – 3	4 000 000	10	100.8539		

Tab. 7.7: Výsledky zkreslení hodin pro zapojení se dvěma routery.

Tabulka 7.7 zobrazuje naměřené výsledky pro zapojení se dvěma routery, kdy byl měřící počítač propojen s měřeným počítačem prostřednictvím dvou aktivních síťových prvků (viz Obr. 7.3). Pro PC – 1 bylo naměřeno zkreslení hodin s maximální odchylkou pro dva síťové toky 1,0 %, oproti předchozímu měření pro jeden síťový tok, kdy byla hodnota maximální odchylky 1,9 %. Pro PC – 2 byly naměřeny o více jak trojnásobek vyšší hodnoty zkreslení hodin pro dva síťové toky, než pro jeden síťový tok. K této odchylce mohlo dojít vlivem zatížení routeru v průběhu měření. Hodnoty zkreslení hodin pro PC – 3 byly naměřeny s maximální odchylkou pro dva síťové toky téměř o polovinu menší, než pro jeden síťový tok. Tato změna souvisí s množstvím zachycených paketů viz sekce 7.2.

měřený počítač	frekvence hodin měřeného počítače [Hz]	zkreslení hodin [PPM] pro přímé zapojení	zkreslení hodin [PPM] pro zapojení s jedním routerem	zkreslení hodin [PPM] pro zapojení se dvěma routery	maximální odchylka pro dva síťové toky [%]	maximální odchylka pro jeden síťový tok [%]
PC – 1	100	74.3934	75.7158	72.3566	4.8	5.8
PC – 1	100	75.7883	75.5831	73.0706		
PC – 1	100	75.9766	75.4482	73.0651		
PC – 2	100	54.0920	46.7862	46.4011	14.8	1.3
PC – 2	100	54.3779	46.7115	46.9137		
PC – 2	100	54.4446	46.7020	47.0118		
PC – 3	10	89.7260	96.7460	92.1455	16.8	40.7
PC – 3	10	94.3974	98.5783	95.2371		
PC – 3	10	97.3786	107.7515	100.8539		

Tab. 7.8: Výsledky ze všech měření a jejich maximální odchylky.

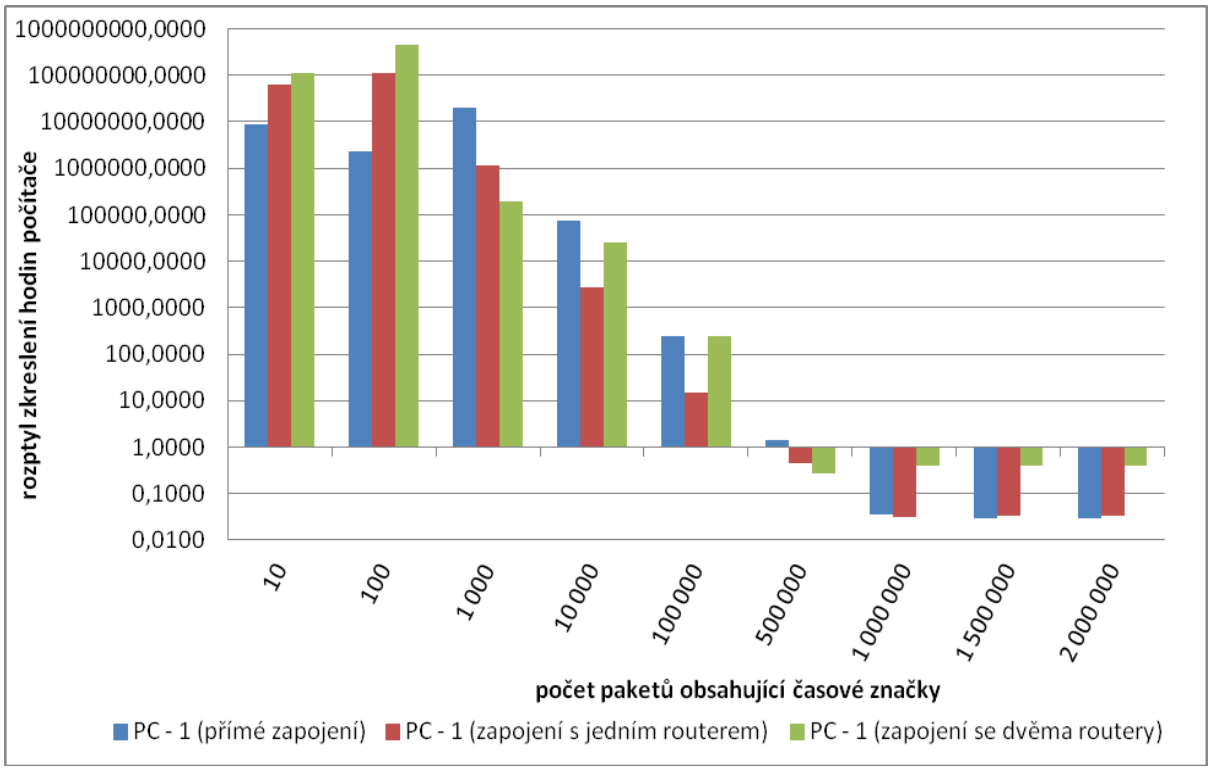
Pokud opět porovnáme výsledky ze všech testů pro jednotlivé počítače a zapojení dohromady (viz Tab. 7.8) zjistíme, že pro PC – 1 se zkreslení hodin pro jednotlivé testy měnilo s maximální odchylkou 4,8 % pro dva síťové toky a to je o 1 % méně, než pro jeden síťový tok. Pro PC – 2 se zkreslení hodin měnilo s maximální odchylkou 14,8 % pro dva síťové toky a to je velký rozdíl oproti maximální odchylce pro jeden síťový tok s hodnotou 1,3 %. Pro PC – 3 byla naměřena maximální odchylka zkreslení hodin pro dva síťové toky 16,8 % a to je výrazné zlepšení oproti jednomu síťovému toku s hodnotou maximální odchylky 40,7 %.

Při porovnání výsledků z měření zkreslení hodin pro jeden síťový tok s výsledky měření zkreslení hodin pro dva síťové toky plyne, že lze tyto tři počítače jednoznačně od sebe rozlišit. Použitím dvou síťových toků bylo dosaženo lepších výsledků ve dvou ze tří případů. Použitím dvou síťových toků bylo zkreslení hodin počítáno z téměř čtyři-krát většího množství paketů, než při výpočtu zkreslení hodin z jednoho síťového toku. Větší množství paketů mělo vliv především na PC – 3, kde se výrazně zmenšila maximální odchylka. V následující sekci se pokusíme určit přesnost měření a určit tak kolik je potřeba minimálního počtu paketů k výpočtu stabilní hodnoty zkreslení hodin počítače.

7.2 Přesnost měření

V předchozí sekci bylo zjištěno, že pomocí zkreslení hodin počítače lze od sebe jednoznačně rozlišit několik počítačů v počítačové síti. V této sekci se zaměříme na přesnost měření zkreslení hodin počítače. Cílem testu bylo zjistit, jak množství zachycených paketů, ze kterých počítáme zkreslení hodin počítače, ovlivní výsledný výpočet zkreslení hodin počítače.

Jednotlivé měřené počítače (PC – 1, PC – 2 a PC – 3) byly postupně zapojeny dle zapojení zobrazených na obrázcích Obr. 7.1, Obr. 7.2 a Obr. 7.3. Na měřeném počítači byl spuštěn webový server (apache). Pomocí webového prohlížeče na měřícím počítači došlo k připojení na měřený počítač, kde byl k dispozici ke stažení vytvořený nový mnohem větší referenční soubor (oproti referenčním souborům ze sekce 7.1) o velikosti několika GB, který byl používán v průběhu měření přesnosti zkreslení hodin počítače. Tento referenční soubor byl pomocí webového prohlížeče stahován na měřící počítač. Síťová komunikace vytvořená v důsledku stahování souboru byla zaznamenávána pomocí programu tcpdump [11] a ukládána ve formátu PCAP. Tyto soubory obsahující síťovou komunikaci byly předány programu *clockskew*, který byl spuštěn v módu 5 tak, aby vypočítal zkreslení hodin počítače při zadání určitého množství paketů, ze kterého chceme zkreslení hodin vypočítat. Z naměřených hodnot zkreslení hodin počítače byl vypočítán rozptyl (střední kvadratická odchylka).

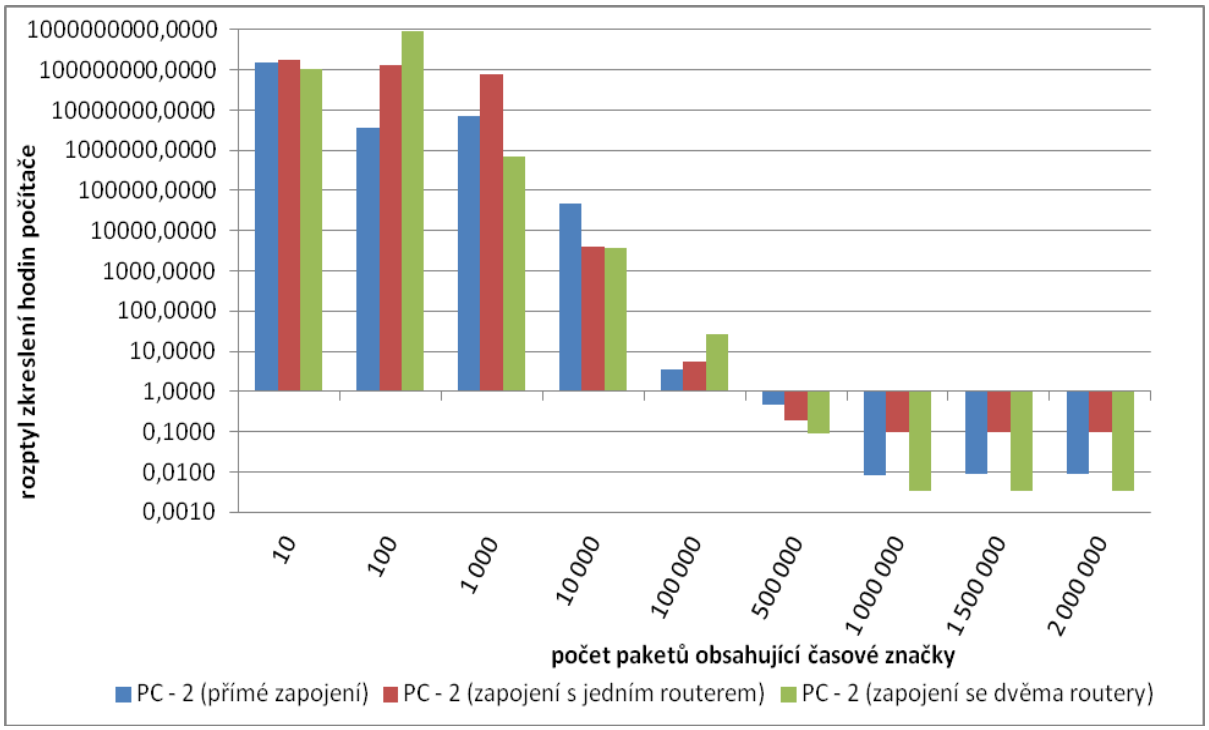


Graf. 7.1: Graf zobrazující závislost rozptylu zkreslení hodin počítače na počtu paketů obsahující časové značky pro PC – 1 (100 Hz).

První test byl proveden pro měřený počítač s označením PC – 1. Hodiny tohoto počítače běžely s frekvencí 100 Hz. Graf 7.1 zobrazuje rozptyl zkreslení hodin počítače měnící se v závislosti na počtu paketů obsahujících časové značky pro PC – 1. Jak můžeme vidět, tak rozptyl zkreslení hodin počítače se stabilizoval až po zaznamenání více jak milionu paketů obsahujících časové

značky. Rozptyl zkreslení hodin počítače se již s větším počtem paketů neměnil, tedy ani zkreslení hodin počítače.

Druhý test byl proveden pro měřený počítač s označením PC – 2. Hodiny počítače běžely s frekvencí 100 Hz. Graf 7.2 zobrazuje rozptyl zkreslení hodin počítače měnící se v závislosti na počtu paketů obsahujících časové značky pro PC – 2. Rozptyl zkreslení hodin počítače se u PC – 2 stabilizoval (stejně jako u PC – 1) až po zaznamenání více jak milionu paketů obsahujících časové značky. Rozptyl zkreslení hodin počítače se již s větším počtem paketů neměnil stejně tak se neměnilo ani zkreslení hodin počítače.

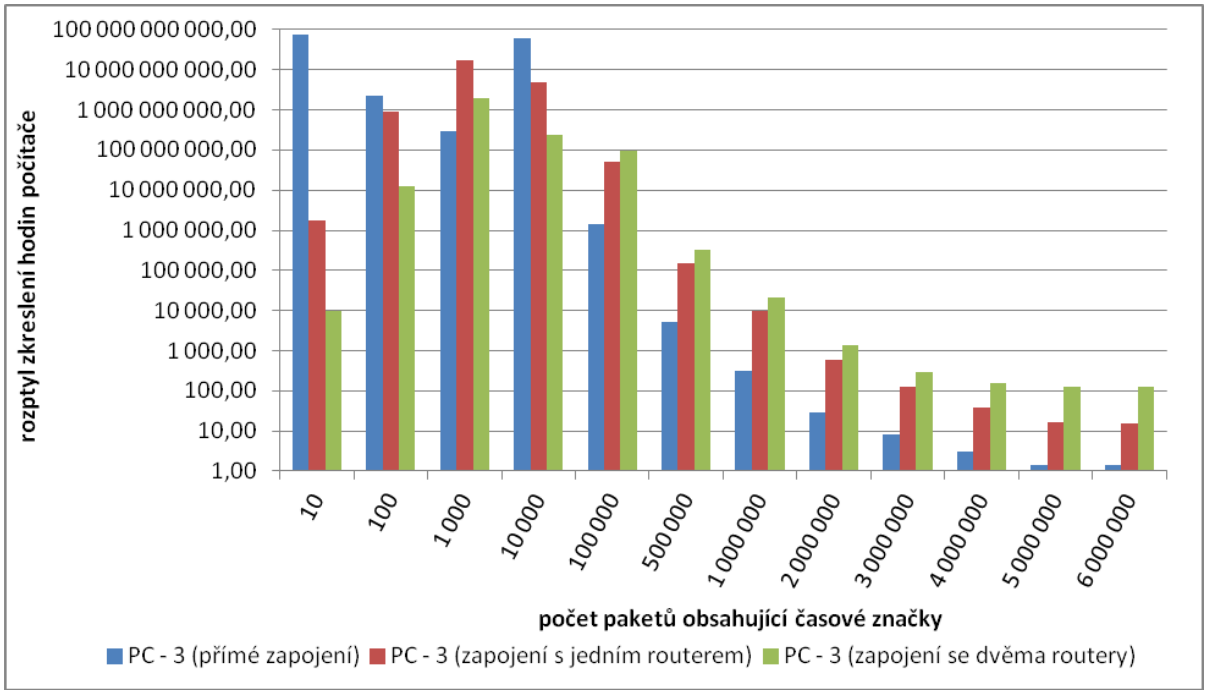


Graf. 7.2: Graf zobrazující závislost rozptylu zkreslení hodin počítače na počtu paketů obsahující časové značky pro PC – 2 (100 Hz).

Poslední test byl proveden pro měřený počítač s označením PC – 3. Tento počítač byl oproti předchozím měřeným počítačům specifický svojí frekvencí hodin, která byla 10 Hz. Graf 7.3 zobrazuje rozptyl zkreslení hodin počítače měnící se v závislosti na počtu paketů obsahujících časové značky pro PC – 3. U tohoto počítače byl při prvním měření, kdy byl zaznamenán pouze milion paketů, naměřen příliš velký rozptyl zkreslení hodin počítače v řádech desetitisíců. Proto byl test proveden opakovaně, kdy bylo zaznamenáno větší množství paketů obsahujících časové značky (přes šest milionů paketů). Rozptyl zkreslení hodin počítače se stabilizoval až po zaznamenání pěti milionů paketů obsahujících časové značky. Stejně tak se stabilizovalo i zkreslení hodin počítače. Konkrétně u zapojení se dvěma routery se nepodařilo snížit rozptyl pod hodnotu 100. Tato odchylka byla pravděpodobně způsobena zpožděním jednoho z routerů.

Z dosažených výsledků můžeme konstatovat, že pro počítače s frekvencí hodin 100 Hz je potřeba počítat zkreslení hodin počítače alespoň minimálně z jednoho milionu paketů obsahujících časové značky. Pro počítač s frekvencí hodin 10 Hz je třeba počítat zkreslení hodin počítače alespoň minimálně z pěti milionů paketů obsahujících časové značky. Pokud by měl měřený počítač ještě nižší frekvenci než 10 Hz, potom by bylo potřeba mnohem více paketů obsahujících časové značky, než je pět milionů. Tento jev je dán frekvencí hodin počítače. Hodiny s vyšší frekvencí vygenerují za jeden časový úsek více hodnot, než hodiny s nižší frekvencí za stejný časový úsek.

Je třeba zmínit, že testy byly provedeny za ideálních podmínek, kdy aktivní síťové prvky nebyly přetížené a nedocházelo tak k zahazování paketů. Pokud by došlo k zahazování paketů, začal by se měnit rozdíl (offset) hodin a to by vedlo ke změně zkreslení hodin počítače. Proto nemůžeme přesně určit, kolik je třeba zaznamenat paketů obsahujících časové značky, neboť výpočet zkreslení hodin počítače je přímo závislý na chování sítě.



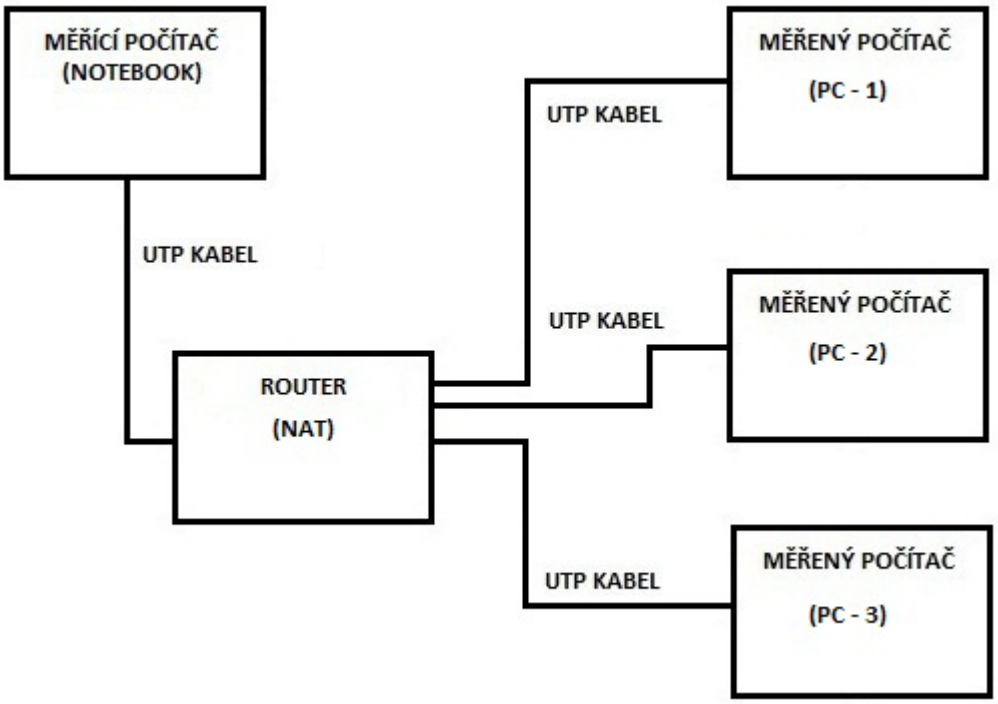
Graf. 7.3: Graf zobrazující závislost rozptylu zkreslení hodin počítače na počtu paketů obsahující časové značky pro PC – 3 (10 Hz).

7.3 Identifikace počítače za NATem

V této sekci se zaměříme na identifikaci počítače nacházejícího se za NATem [14]. NATem rozumíme aktivní síťový prvek (např. router), na kterém běží překlad adres (NAT). Cílem testu bylo zjistit, zda je možné pomocí vytvořeného programu *clockskew* identifikovat tři počítače, které se nachází za NATem.

Měřené počítače byly zapojeny dle zapojení, které je zobrazené na obrázku 7.4. Na měřícím počítači byl spuštěn program iperf [13] v módu server. Měřené počítače byly pomocí programu iperf v módu klient připojovány k serveru běžícím na měřícím počítači. Zároveň byl spuštěn program *clockskew* v módu 2, který zaznamenával síťové toky směřující k měřícímu počítači. Z těchto síťových toků pak program vypočítal zkreslení hodin měřených počítačů.

Tabulka 7.9 zobrazuje výsledky zkreslení hodin pro počítače nacházející se za NATem. Pro všechny měřené počítače se maximální odchylka zkreslení hodin počítače pohybovala v řádech desetin procent. Na základě dosažených výsledků můžeme říci, že lze jednoznačně identifikovat počítače nacházející se za NATem. Na samotný výpočet zkreslení hodin počítače nemá vliv, zda se měřený počítač nachází za NATem či nikoli. NAT je ovšem aktivní sítový prvek a při zatížení bude mít vliv na výsledné zkreslení hodin počítače (v tomto případě NAT zatížen nebyl).



Obr. 7.4: Zapojení s počítači za NATem (měřící počítač – NAT – měřený počítač).

měřený počítač	počet paketů obsahující časové značky	frekvence hodin měřeného počítače [Hz]	zkreslení hodin [PPM]	maximální odchylka [%]
PC – 1	6 000 000	100	73.1281	0.4
PC – 1	6 000 000	100	73.4163	
PC – 1	6 000 000	100	73.3654	
PC – 2	6 000 000	100	46.7502	0.5
PC – 2	6 000 000	100	46.9584	
PC – 2	6 000 000	100	46.8843	
PC – 3	6 000 000	10	88.9862	0.3
PC – 3	6 000 000	10	88.9642	
PC – 3	6 000 000	10	88.7497	

Tab. 7.9: Výsledky zkreslení hodin počítačů nacházejících se za NATem.

7.3.1 Zjištění počtu počítačů za NATem

V předchozí sekci bylo ukázáno, že lze identifikovat počítače nacházející se za NATem. V této sekci bude popsáno a bude proveden test, jak lze pomocí programu *clockskew* zjistit počet počítačů nacházejících se za NATem.

Dříve již byla představena technika pro zjištění počtu počítačů nacházejících se za NATem [12] (Steven M. Bellovin). Tato technika využívá ke zjištění počtu počítačů nacházejících se za NATem IP ID. Nevýhodou této techniky je, že proto, aby mohla určit počet počítačů nacházejících se za NATem, tak musí být počítače nacházející se za NATem aktivní všechny najednou v době měření. Aktivním počítačem se rozumí počítač, který vysílá síťový tok. Při použití programu *clockskew* není

nutné, aby byly počítače nacházející se za NATem aktivní všechny najednou. Počítače v průběhu měření mohou přecházet ze stavu aktivního do stavu neaktivního.

Při zjišťování počtu počítačů nacházejících se za NATem byly počítače zapojeny dle zapojení, které je zobrazené na obrázku 7.4. Na měřícím počítači byl spuštěn program iperf [13] v módu server. Měřené počítače byly pomocí programu iperf v módu klient postupně připojovány k serveru běžícím na měřícím počítači tak, aby byl vždy aktivní pouze jeden měřený počítač. Pro každý počítač byl test proveden tři-krát. Zároveň byl spuštěn program *clockskew* v módu 2, který zaznamenával síťové toky směřující k měřícímu počítači. Z těchto síťových toků pak program vypočítal zkreslení hodin měřených počítačů.

IP adresa měřeného počítače	IP adresa routeru (LAN)	IP adresa routeru (WAN)	IP adresa měřícího počítače	zdrojový TCP port	cílový TCP port	zkreslení hodin [PPM]
192.168.2.188	192.168.2.1	192.168.0.120	192.168.0.108	59665	5001	46.7502
192.168.2.188	192.168.2.1	192.168.0.120	192.168.0.108	59665	5001	46.9584
192.168.2.188	192.168.2.1	192.168.0.120	192.168.0.108	59665	5001	46.8843
192.168.2.196	192.168.2.1	192.168.0.120	192.168.0.108	59667	5001	73.1281
192.168.2.196	192.168.2.1	192.168.0.120	192.168.0.108	59667	5001	73.4163
192.168.2.196	192.168.2.1	192.168.0.120	192.168.0.108	59667	5001	73.3654
192.168.2.108	192.168.2.1	192.168.0.120	192.168.0.108	59669	5001	88.9642
192.168.2.108	192.168.2.1	192.168.0.120	192.168.0.108	59669	5001	88.7497
192.168.2.108	192.168.2.1	192.168.0.120	192.168.0.108	59669	5001	88.9862

Tab. 7.10: Výsledky zkreslení hodin počítačů nacházejících se za NATem.

Tabulka 7.10 zobrazuje výsledky zkreslení hodin měřených počítačů nacházejících se za NATem. Z výsledků můžeme vidět, že za NATem (IP adresa 192.168.0.120) se nachází právě tři počítače. Hodnoty zkreslení hodin měřených počítačů identifikují jednoznačně dané počítače, aniž bychom znali jejich IP adresy. Z výsledné tabulky pak lze jednoduše spočítat počet počítačů nacházejících se za NATem. Samotný výpočet počtu počítačů nacházejících se za NATem provádí program *clockskew* spuštěný v módu 2 a 4, kde součástí výstupu je i tabulka obsahující IP adresu a počet počítačů, které se za touto IP adresou nachází. Touto metodou lze také dokázat, že na dané IP adrese probíhá překlad adres NAT. Pokud by se ovšem za NATem nacházel další NAT, za kterým by se nacházel určitý počet počítačů, tak je program *clockskew* schopen určit celkový počet počítačů nacházejících se za prvním a druhým NATem dohromady, ale není možné určit, kolik počítačů se nachází pouze za druhým NATem.

8 Závěr

Identifikace počítače na základě časových značek paketů je jednou z možností jak lze identifikovat počítač v počítačové síti. Pro menší počítačové sítě lze jednoznačně počítače mezi sebou odlišit. Podařilo se identifikovat a rozlišit od sebe tři počítače při různých zapojeních. Dále bylo ukázáno, že lze pomocí této techniky identifikovat počítače nacházející se za NATem a určit tak jejich počet.

Tato práce skýtá několik možností na rozšíření. K výpočtu zkreslení hodin použít časové značky jiného protokolu než protokolu TCP např. ICMP, HTTP. Provést testování na větší počítačové síti, která by obsahovala řádově větší množství počítačů. Otestovat jaký vliv může mít na zkreslení hodin napájení ze sítě nebo z baterie v případě měření zkreslení hodin u notebooků. Zjistit vliv teploty na zkreslení hodin počítače.

Literatura

[1] Nmap free security scanner. 2011. [cit. 2011-12-22]. Dostupné z: http://nmap.org

[2] X probe - active OS fingerprinting tool. 2011. [cit. 2011-12-22]. Dostupné z: http://xprobe.sourceforge.net

[3] P0f is a versatile passive OS fingerprinting and masquerade detection utility. 2011. [cit. 2011-12-22]. Dostupné z: http://freecode.com/projects/p0f

[4] DOSTÁLEK, Libor a Alena KABELOVÁ. *Velký průvodce protokoly TCP/IP a systémem DNS*. 2. aktualizované vydání. Praha: Computer Press®, 2000. ISBN 80-7226-323-4.

[5] VEYSSET, F., O. COURTAY a O. HEEN. *New tool and technique for remote operating system fingerprinting*. 2002.

[6] MOON, Sue B., Paul SKELLY a Don TOWSLEY. *Estimation and Removal of Clock Skew from Network Delay Measurements*. Amherst, 1998. Technical Report 98-43. Department of Computer Science University of Massachusetts.

[7] KOHNO, Tadayoshi, Andre BROIDO a K. C. CLAFFY. *Remote physical device fingerprinting*. San Diego, May 2005. IEEE Symposium on Security and Privacy 2005. Department of Computer Science & Engineering, University of California.

[8] PAXSON, Vern. *On Calibrating Measurements of Packet Transit Times*. Berkeley, 1998. University of California.

[9] JACOBSON, V., R. BRADEN a D. BORMAN. *TCP extensions for high performance*. RFC 1323, May 1992.

[10] PUS, Viktor, Jiri TOBOLA, Vlastimil KOSAR, Jan KASTIL a Jan KORENEK. Netbench: Framework for Evaluation of Packet Processing Algorithms. *Symposium On Architecture For Networking And Communications Systems*. Los Alamitos, CA, USA: IEEE Computer Society, 2011, s. 95-96. ISSN 978-0-7695-4521-9.

[11] Tcpdump [online]. 4.2.1 / 1.2.1. 2012-01-01 [cit. 2012-03-21]. Dostupné z: http://www.tcpdump.org

[12] BELLOVIN, Steven M. *A Technique for Counting NATted Hosts*: *In Proceedings of the Second Internet Measurement Workshop*. Marseille, France, 2002. 267-272.

[13] Iperf [online]. 2.0.5. [cit. 2012-03-31]. Dostupné z: http://sourceforge.net/projects/iperf/

[14] EGEVANG, K. a P. FRANCIS. *The IP Network Address Translator (NAT)*. RFC 1631, May 1994.

Seznam příloh

Příloha 1. DVD s testovacími daty a zdrojovými kódy.