VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

MATEMATICKÉ MODELOVÁNÍ SÍŤOVÉHO PROVOZU SLUŽBY HTTP

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE JAN MIKLICA
AUTHOR

BRNO 2012

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELOKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

MATEMATICKÉ MODELOVÁNÍ SÍŤOVÉHO PROVOZU SLUŽBY HTTP

MATHEMATICAL MODELLING OF HTTP NETWORK TRAFFIC

BAKALÁŘSKÁ PRÁCE

BACHELOR´S THESIS

AUTOR PRÁCE — JAN MIKLICA

AUTHOR

VEDOUCÍ PRÁCE — doc. Ing. KAROL MOLNÁR, Ph.D.

SUPERVISOR

BRNO 2012

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky a komunikačních technologií

Ústav telekomunikací

Bakalářská práce

bakalářský studijní obor
Teleinformatika

Student: Jan Miklica ***ID:*** 119533
Ročník: 3 ***Akademický rok:*** 2011/2012

NÁZEV TÉMATU:

Matematické modelování síťového provozu služby HTTP

POKYNY PRO VYPRACOVÁNÍ:

Proveďte dlouhodobé zachycení provozu protokolu HTTP s minimálním trváním dvou dnů. Na základě vlastností zachyceného síťového provozu navrhněte metodu pro odvození parametrů pro klasický model síťového provozu (např. Markovský modulovaný, ON/OFF, IPP či Autoregresivní model). Porovnejte výsledné chování modelu se zachyceným provozem.
Prostudujte popis stochastického modelu síťového provozu a navrhněte postup pro odvození parametrů modelu. Odvoďte či odhadněte na základě navržených metod co nejpřesněji parametry pro stochastický model. Porovnejte chování stochastického modelu se zachyceným reálným provozem. Zdokumentujte nově získané poznatky.

DOPORUČENÁ LITERATURA:

[1] FROST, V., MELAMED, B. Traffic Modeling for Telecommunication Networks, IEEE Communications Magazine, 32(3), pp. 70-80, March, 1994
[2] FAPOJUWO, A., LEE, I.: Mathematical Modeling and Characterization of Wireless Network Traffic. Hauppauge: Nova Science Publishers, 2008, ISBN: 978-1604568691.
[3] PAPOULIS, A., PILLAJ, S.U.: Probabilities, Random Variables and Stochastic Process. New York: McGraw-Hill, 2002, ISBN:978-0071226615.

Termín zadání: 6.2.2012 ***Termín odevzdání:*** 31.5.2012

Vedoucí práce: doc. Ing. Karol Molnár, Ph.D.
Konzultanti bakalářské práce:

prof. Ing. Kamil Vrba, CSc.
Předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb

Abstrakt

Cílem této semestrální práce je zachytit síťový provoz na protokolu HTTP. Poté ze zachyceného provozu provést analýzu parametrů za pomocí vybraných modelů a porovnat je se skutečnými vlastnosti reálného síťového provozu. Tato bakalářská práce popisuje protokol HTTP a matematické modelování a k němu vytvořené simulační modely. V dokumentu je popsáno zachycování datového provozu na protokolu HTTP pomocí programu Wireshark. V další části jsou provedeny základní analýzy pomocí vybraných modelů. V závěru jsou popsány výsledky analýz vybraných modelu.

Klíčová slova

matematické modelování, síťový provoz, model, analýza, protokol

Abstract

The aim of this thesis is to capture network tradic on the HTTP protocol. Then from the captured tradic to analyze parameters using selected models and compare them with the achal properties of real network traffic. This bachelor´s thesis describes the HTTP protokol and mathematical modeling, and simulation models developed for it. The document described the capture of data tradic on HTTP using Wireshark. In the next part are performed elemental analysis of selected models. In conclusion the analysis results of the selected models are described.

Keywords

mathematical modeling, network tradic, model, analysis, protocol

Bibliografická citace mé práce:

MIKLICA, J. *Matematické modelování síťového provozu služby HTTP* . Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2012. 45 s. Vedoucí bakalářské práce doc. Ing. Karol Molnár, Ph.D..

Prohlášení

Prohlašuji, že svou bakalářskou práci na téma „Matematické modelování síťového provozu služby HTTP“ jsem vypracoval samostatně pod vedením vedoucího bakalářské 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 bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské 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

Poděkování:

Děkuji vedoucímu bakalářské práce panu doc. Ing. Karolu Molnárovi, Ph.D. za velmi užitečnou metodickou pomoc a cenné rady při zpracování práce.

V Brně dne ....

podpis autora

Obsah

Úvod

V počátcích vývoje telefonních či síťových technologií bylo vždy nutné, sestavit nějaký dostatečně vypovídající návrh pro fungování sítě- tzv. model. Tyto modely slouží pro simulaci požadovaného provozu a k pochopení fungování provozu. Nejprve pro simulaci byly nasazeny jednoduché modely, které předtím úspěšně simulovaly telefonní provoz. S postupem času modely nabíraly na přesnosti, ale také na složitosti. Principů, na kterých může fungovat síťový provoz, je velké množství. V podstatě může fungovat na každém modelu, ale ne každý model je spolehlivý a efektivní. Vývoj dospěl k vytvoření nejrozšířenějšího protokolu HTTP, který pracuje nad spolehlivým a zabezpečeným protokolem TCP. Přesto, protokol HTTP má své nedostatky. Navíc narůstá počet koncových uživatelů v síti a jsou kladeny větší požadavky na služby. Proto je tu snaha modelovat provoz za účelem opravení chyb či vymyšlení nového systému.

Tato práce se soustřeďuje na popsání vybraných modelů. Snaha v této práci je analyzovat zachycený provoz pomocí vybraných modelů a zjistit jak moc je odlišné modelování provozu od jeho skutečného chování.

1. Protokol http

HTTP (Hypertext Transfer Protocol) je aplikační protokol, využívaný převážně v internetu pro výměnu hypertextových dokumentů ve formátu HTML. Tento protokol je spolu s elektronickou poštou tím nejvíce používaným a zasloužil se o obrovský rozmach internetu v posledních letech. Protokol HTTP je postaven na principu požadavek – odpověď. Jakákoliv aktivita musí být vyvolána klientem. Komunikace se serverem probíhá přes TCP spojení (server většinou používá port č. 80). Úplný dotaz/odpověď musí mít specifikovánu metodu, URI (absolutní nebo relativní cesta k souboru nebo úplné URL dokumentu), verzi a hlavičky. V některých případech následuje po hlavičkách i tělo dotazu oddělené jedním prázdným řádkem. Existují 2 verze protokolu: 1.0 a 1.1. Server podporující protokol 1.0 vrátí odpověď a spojení ihned uzavře. Protokol 1.1 definuje tzv. perzistentní spojení, a proto servery podporující verzi 1.1 spojení neuzavřou hned, ale čekají chvíli na další příkazy. Klient může pokračovat v dotazování na ostatní prvky HTML stránky a spojení ukončit sám. [7]

1.1. Metody protokolu

- GET je nejpoužívanější metoda. Slouží k vyzvednutí objektu (HTML soubor, obrázek, cokoliv…) ze serveru. Odpověď je „kešovatelná“.
- POST. Pomocí této metody se dají v těle dopravit na server informace od uživatele (velmi často se POST používá pro odeslání rozsáhlejších dat z webových formulářů, pro upload souboru a podobně).
- HEAD se chová naprosto stejně jako GET, ale v odpovědi se nepřenáší tělo. Tento dotaz se hodí například ke zjištění, zda objekt existuje (při kontrole odkazů na stránce).
- PUT/DELETE vytvoří/smaže daný objekt ze serveru. Tyto metody se v praxi příliš nevyužívají.
- OPTIONS slouží ke zjištění informací o daném kontextu (nebo „*“ pro celý server). Klient může zjistit, které dotazy může na daný kontext zaslat.
- TRACE se používá ke sledování cesty celého dotazu. V těle odpovědi klient dostane pěkně seřazené všechny dotazy jednotlivých systémů, kterými požadavek procházel.
- CONNECT se spojí s uvedeným objektem před uvedený port. Používá se při průchodu skrze proxy pro ustanovení kanálu SSL. [12]

2. Matematické modelování síťového provozu

Matematické modelování síťového provozu se používá před návrhem každého typu sítě (telefonní, datové apod.) jako ověření pro splnění požadovaných parametrů. Charakter náhodných procesů, parametry systémů front, zpoždění, ztrátovost, jsou jen pár atributů z mnoha, které charakterizují síťový provoz. Z hlediska datového provozu protokolu HTTP je potřeba, aby provozní model byl schopen simulovat shlukový provoz (burstiness), zahlcení na síťových prvcích a přenos pomocí transportního protokolu TCP. Shlukový provoz (burstiness) lze charakterizovat jako náhodný, nárazový příchod paketů, na všech časových intervalech. Při nárazovém příchodu paketů vnikají problémy typu vytváření zahlcení či shluků, které vidíme jako mix krátkých a dlouhých mezi-příchozích časových úseků. Zahlcení na síťových prvcích charakterizujeme jako překročení kapacity paměti síťových prvků pro zpracování paketů. [2] [9]

V následujících kapitolách budou popsány některé modely síťového provozu od počátečních až po ty dnešní.

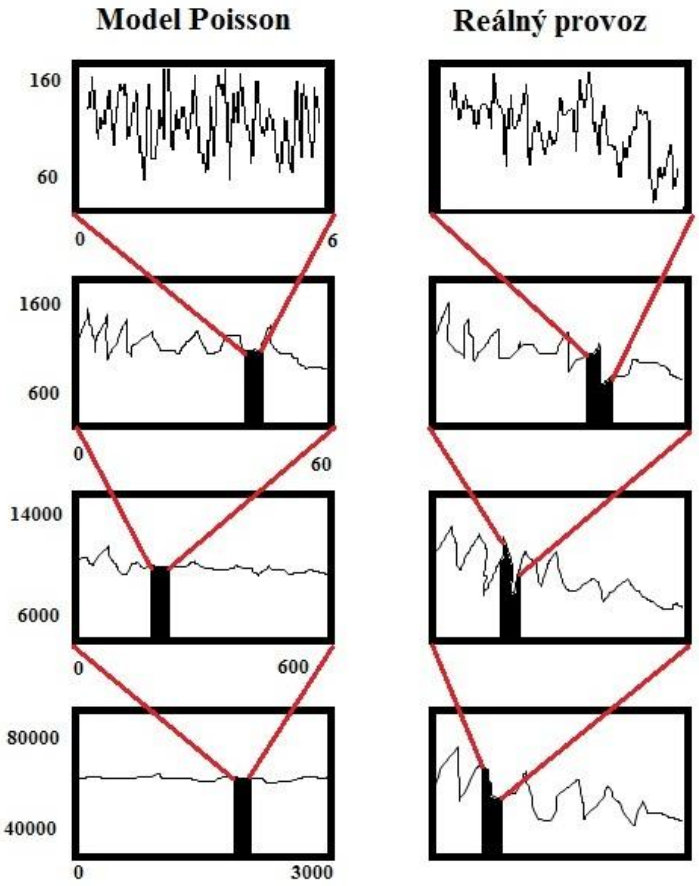
2.1. Poissonův distribuční model

Poissonův distribuční model je jeden z nejstarších provozních modelů, který byl navržen pro simulaci telefonní sítě. Řídí se 2 pravidly:[2]

- Příchody paketů A_n jsou nezávislé.
- Intervaly mezi příchozími pakety A_n jsou rozděleny exponenciálně s parametrem rychlosti:

$$\lambda : P\{\leq t\} = 1 - e - \lambda t \tag{2.1}$$

Model je založen na skutečnosti, že hovory v telefonní síti zpravidla nejsou realizovány nárazově. V datových sítích má provoz shlukový charakter, který tento model neumí vyjádřit. Pro lepší pochopení je zde názorná ukázka:



Obr. 2.1: Vyjádření shlukového provozu pomocí modelu Poisson

Z obrázku je zřejmé, že Poissonův model je schopen vyjádřit shlukový provoz pouze na krátkých časových intervalech, na delších se charakter skutečného a modelovaného provozu výrazně liší. Přesto se tento model stal základem, od kterého se odvíjely i některé novější modely [2]

2.2. Paretův Distribuční model

Paretův distribuční model pracuje na principu Paretova procesu, který mezi-příchodové časy bere jako nezávislé a identicky rozložené. Definici Paretova procesu lze vyjádřit 3 funkcemi. Funkce pravděpodobnosti, že X je větší než x je vyjádřena vztahem: [4]

$$P(X > x) = (x/xm) - k \text{ pro všechny } x \geq xm \qquad (2.2)$$

Kde *k* je kladný parametr a *xm* je minimální hodnota proměnné *Xi*. Následující 2 funkce popisují pravděpodobnost distribuce a funkce hustoty. [4]

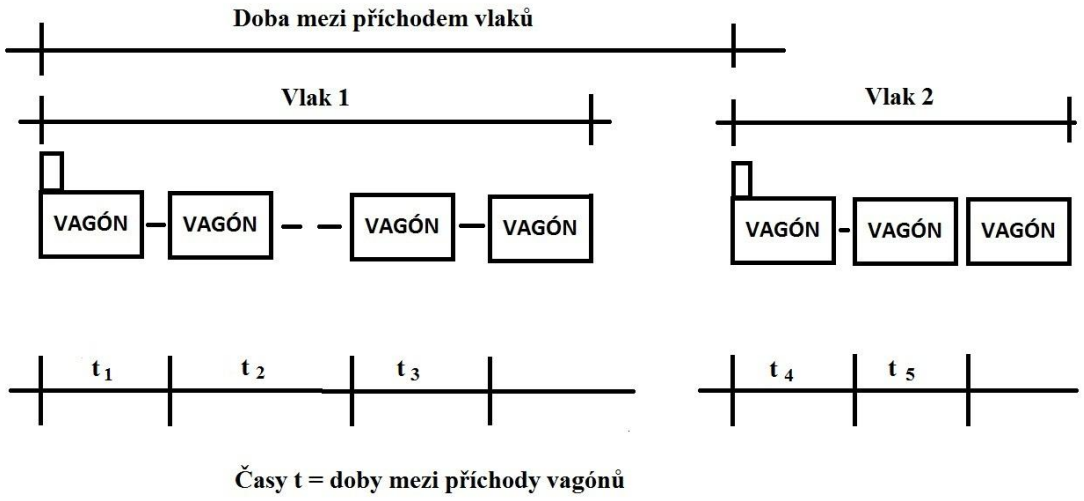
$$F(t) = 1 - (\alpha/t)\beta \; kde \; \alpha, \beta \geq 0 \; a \; t \geq \alpha \tag{2.3}$$

$$f(t) = \beta\alpha\beta t - \beta - 1 \tag{2.4}$$

Model se používá pro modelování sobě podobného provozu (self-similar), který bude popsán v kapitole 2.4. β a α jsou umísťovací a tvarovací parametry- pomocné parametry. Pareto distribuce je účinná ve vysokorychlostních sítích. Paretův model, podobně jako Poissonův model, se stal základem pro složitější modely. [4]

2.3. Model vlaku

Model vlaku je založen na principu seskupování paketů do tzv. vlaků (balíček paketů o určité velikosti), který se dělí na menší části tzv. vagóny.



Obr. 2.2: Seskupení paketů do tzv. vlaků a vagónů.

Tyto seskupené pakety cestují spolu stejnou cestou v síti od zdroje až k cíly. Model předpokládá, že pakety, které v krátkém časovém úseku přijedou těsně za sebou, jsou orientovány do stejného místa. Komunikace mezi zdrojem a cílem se skládá z řady zpráv tam a zpět. Tímto způsobem mohou vagóny sledovat stejnou trasu v síti. Model vlaku se vyznačuje následujícími parametry: [2] [11]

- Doba mezi příchodem vlaků.
- Doba mezi příchodem vagónů.
- Střední délka vagónu.
- Průměrná délka vlaku.

Doba mezi příchodem vlaků je doba rozdílu příchodu dvou paketových vlaků. Doba mezi příchodem vagónů je doba rozdílů příchodu vagónů v části vlaku. Střední délka vagónu a průměrné délky vlaků nejsou vždy stejné. Tyto velikosti jsou určeny technickými parametry sítě. Díky tomuto řešení Model vlaku nepřesahuje meze systému. Bohužel díky svému principu fungování je potřeba obtížné matematické analýzy systému. Nemá stanovené parametry pro určité typy protokolů. Model vlaku se nezabývá modelováním shlukového provozu. [2] [11]

2.4. Pojem sobě podobný (self-similar) provoz

Počáteční modely provozu vykazovaly problém zachytit provoz na velkém rozsahu časových měřítek. Datový provoz, na rozdíl od telefonního, vykazuje velké časové a prostorové variability. To se projevuje jako fraktální chování na všech časových úsecích. Tzn., že pakety chodí nárazové v různých délkách na všech časových úsecích. Také dochází k nežádoucí autokorelaci (korelaci mezi pakety téže řady). [11]

Sobě podobný provoz je charakterizován parametrem Hurst- tzv. míra sobě podobného provozu, která je definována v oblasti $0 \leq H \leq 1$. Tuto oblast rozdělujeme na $0 < H < ½$ oblast Short-Range Dependence (SRD) a na $1/2 < H < 1$ oblast Long-Range Dependence (LRD). V oblasti SRD nevznikají velké analytické problémy, protože autokorelace se rozkládají dostatečně rychle s agregací. Ovšem v oblasti LRD se autokorelace rozpadají pomalu s agregací. Vztahy níže popisují matematické vyjádření LRD. [11]

Vyjádření autokovarianční funkce pro $1/2 < H < 1$

$$\gamma(k) = \frac{\sigma 2}{2}[(k+1)2H - 2k2H + (k-1)2H] \text{ pro všechny } k \geq 1 \qquad (2.5)$$

Díky této funkci jsme schopni vyjádřit autokorelační funkci

$$r(k) = \gamma(k) / \sigma 2 \tag{2.6.}$$

Pak pro 0<*H*<1

$$r(k) \sim H(2H - 1)k2H - 2, k \to \infty \tag{2.7}$$

a pro 1/2<*H*<1

$$\sum_{k=-\infty}^{\infty} r(k) = \infty \quad [11] \tag{2.8}$$

2.5. Sobě podobné (self-similar) modely

Sobě podobné modely jsou založeny na matematických definicích uvedených v předchozí kapitole. Oblast SDR nepředstavuje problém pro zpracování dat. Ovšem pro oblast LDR dochází ke komplikované matematické analýze. V dalších kapitolách jsou popsány modely, které zahrnují problematiku self-similar- Frakční Brownův pohyb, Chaotické mapy a SWING. [11]

2.5.1. Frakční Brownův pohyb

Brownův Frakční pohyb (FBM) vychází s Gaussova procesu. Je to stochastický proces. Skládá se z náhodných proměnných, které jsou spojeny s každým bodem v rozsahu doby nebo prostoru. Tzn., že každá náhodná proměnná má normální distribuce. FBM je Gaussovský proces se spojitým časem. FBM je definován pro všechny pozitivní hodnoty času se středem na nule. FBM je dále definován autokorelacemi, Norros procesem a dalšími pomocnými funkcemi. Bohužel v praxi bylo obtížné u tohoto modelu určit parametr Hurst. Tento model je velice obtížné sestavovat z důvodu jeho složitosti. [6] [11]

2.5.2. Chaotické mapy

Chaotické mapy si lze představit jako formu průběžného stavu Markova řetězu. Diskrétní markovský řetězec, který vychází ze základního Markova řetězu, bude popsán v kapitole 2.6. Stavová proměnná, *x* je definována vztahem [11]

$$xn + 1 = f1\,(xn), xn > d \tag{2.9}$$

$$xn + 1 = f2\,(xn), xn < d \tag{2.10}$$

, kde *x* je v intervalu [0,1] a *d* představuje stavové hranice. Pro účely modelování ON / OFF zdrojů sítě je zde návrh, že pokud *x* je větší než *d*, zdroj je ve stavu ON. Když *x* je nižší než *d*, zdroj je ve stavu OFF. [11]

Pro modelování se používá dvojice chaotických map, jedna pro velikost TCP okna a jedna pro současný stav (ON / OFF). Předpokládá se, že každá iterace je RTT (doba odezvy). Je nutné stanovit délku trvání výchozího ON stavu, během kterého zdroj odešle informaci o dostupné velikosti TCP okna. Z těchto předpokladů, může být popsána sada funkcí pevně spojená s popisem vývoje velikosti TCP okna a *D* – doby šíření signálu od zdroje TCP spojení k cíli v sekundách. Tento omezený model nezohledňuje ztracené pakety. Zde se předpokládá, že tyto mechanismy představují další korelace. [9] [11]

V simulaci chaotické mapy se úspěšně vytváří spojení, které připomínají skutečné TCP spojení. Model nebyl otestován na reálném provozu. [9]

2.5.3. Model SWING

Tento jednoduchý model patří mezi nejnovější. SWING je založen na typu a náročnosti požadavků uživatelů v síti. Model zkoumá charakteristiku uživatelů a požadavky ústředen (RREs - Request-Response-Exchanges – výměna žádosti a odpovědí). Model automaticky extrahuje distribuci paketů pro uživatele, pro aplikace a pro vlastnosti sítě. Není zde žádné opatření pro analýzu vlastností sobě podobnosti, protože tyto vlastnosti vyplynou při generování provozu ze souhrnu ON-OFF zdrojů. SWING generuje reálný paketový provoz, který odpovídá základním modelům v síťovém emulačním prostředí (ModelNet). Model umožňuje uživateli měnit předpoklady o stavu sítě. [10] [11]

Model SWING můžeme charakterizovat pomocí 4 základních parametrů:

- Uživatel: Koncoví uživatelé určují charakteristiku komunikace různých aplikací. Důležité je, jak často je uživatel aktivní (doba mezi jednotlivými požadavky).

- Relace (sessions): Relace je požadavek provádět činnosti na vyšší úrovni např. – vyhledávání webových stránek či stahování souborů. Relace se můžou skládat z několika síťových připojení rozdělených do více destinací (stahování více obrázků z jednoho serveru, nebo stahování jednoho souboru z více serverů). Důležitý je počet a cíl jednotlivých spojů v rámci jedné relace.

- Spojení: Zde jsou zahrnuty vlastnosti připojení v rámci jedné relace. Např.: cíl, počet žádostí/odpovědí, čekací doba generování odpovědí, doba mezi požadavky a typ transportního protokolu (TCP nebo UDP). Charakteristické jsou jednotlivé odpovědi podle distribuce velikosti paketu.

- Síťová charakteristika: Zde může být zahrnuta ztrátovost, kapacity a latence u spojení mezi jednotlivými hosty. Zahrnuje obecný způsob, jakým aplikace uživatele popř. serveru mezi sebou komunikují. Důležitá charakteristika je tzv. podpis (signature). Například HTTP, P2P a SMTP, všechny tyto protokoly mají různé podpisy. Aby bylo možné úspěšně reprodukovat paketovou stopu, musíme vystihnout vhodné distribuce z původního trasování naplněním každého parametru, který má SWING stanovený ve svých pomocných tabulkách. V případě potřeby je možné individuálně nastavit distribuce hodnot pro tyto parametry pro extrapolaci na cílové prostředí. [10] [11]

Model SWING je schopen generovat shlukový provoz v široké časové škále. V reálně vytvořeném provozu je schopen v celém rozsahu Hurst ($0 \leq H \leq 1$) získat vhodné parametry pro generování stopy. Ovšem tento model je ještě ve vývoji a nese sebou i jistá omezení. Model byl dříve použit pro modelování http provozu. Při modelování bylo vyzkoušeno chování TCP protokolu. [10] [11]

2.6. Markovské modely

Markovské modely jsou založeny na Markovském systému řízení front, pravděpodobnostní matici, Markovým řetězcem a na předpokladu konečného počtu stavů. Markovský systém řízení front se definuje pomocí vstupního toku a obslužného systému. Vstupní tok je popisován vlastnostmi, jako je např. intenzita vstupu, která je definována vztahem [8]

$$\lambda = \frac{n}{T} \tag{2.11}$$

kde n je počet přicházejících požadavků za dobu T. Další vlastností u vstupního toku je Stacionarita. Stacionarita vyjadřuje stálost parametrů v čase. U nestacionárního procesu se parametry pro popis (typ a parametry distribuční funkce, atd.) mění s časem a to způsobuje velké komplikace.

Obslužný systém je charakterizován intenzitou obsluhy. Matematické vyjádření intenzity obsluhy je [8]

$$\mu = \frac{m}{T} \tag{2.12}$$

kde m je počet obsloužených požadavků a T je interval sledování. [8]

$$\frac{1}{\mu} \tag{2.13}$$

Vztah 2.13 je vyjádření střední hodnoty intenzity obsluhy. Pravděpodobnostní matice popisuje následující události, které mohou nastat za Δt: [8]

- přichází nový požadavek,
- bude ukončena obsluha požadavku v obslužném kanálu,
- bude ukončena obsluha a současně bude přijat i nový požadavek,
- nedojde k žádné změně. [8]

Pravděpodobnostní matice je definována v maticovém tvaru konkrétně pro systém M/M/1/0: [8]

$$P = \begin{matrix} 1 - \lambda * \Delta t & \lambda * \Delta t \\ \mu\lambda * \Delta t & 1 - \mu\lambda * \Delta t \end{matrix} \tag{2.14}$$

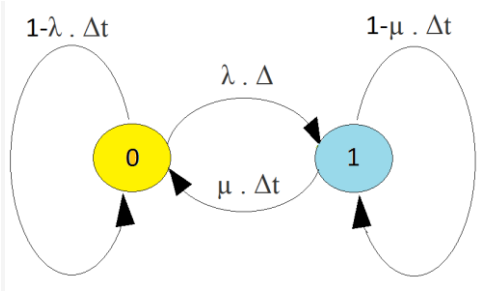
Přesnost, ale i složitost, Markova modelu se lineárně zvyšuje s použitým počtem stavů. Markovské modely předpokládají, že následující stav je závislý pouze na aktuálním stavu a nikoli na stavech předchozích. Jinými slovy to tedy znamená, že pravděpodobnost následujícího stavu, vyjádřeného nějakou náhodnou proměnou $Xn+1$, závisí pouze na současném stavu, ve kterém se právě nacházíme, vyjádřeným proměnou Xn a nikoli na žádném dalším stavu Xi, kde $i<n$. [8]

Soubor náhodných proměnných vztahujících se k odlišným stavům Xn je nazýván jako Diskrétní Markovský řetězec (Discrete Markov Chain- DCM). Za předpokladu, že zkoumaný stavový přechod v systému je vyjádřen pouze celočíselnými hodnotami 0, 1, 2, .. , n, pak Markovský řetězec (Markov Chain- MC) je nespojitý v čase a náhodná proměnná X sleduje geometrické rozložení. V opačném případě je Markovský řetězec spojitý v čase a rozložení je exponenciální. Na principu

Markovských modelů bylo vytvořeno několik variant, které budou následně popsány. [4]

2.6.1. Semi-Markovský-model

U tohoto modelu mezi-stavový přechod sleduje libovolné rozložení pravděpodobnosti a časové rozložení mezi stavovými přechody může být ignorováno. Stavové přechody jsou modelovány jako nespojité entity s ohledem na čas (viz obrázek 2.3). Díky tomuto předpokladu můžeme tento model považovat za Diskrétní Markovský řetězec. [4]



Obr. 2.3: Stavový automat: 0 – žádný požadavek, 1 – požadavek. Tento model platí pouze pro systém M/M//1/0

2.6.2. Markovský-modulovaný Poissonovský model

Původně byl Markovský-modulovaný Poissonovský model (MMPM) navržen pro smíšený provoz přenosu hlasu a dat. MMPM se snaží zachytit datový provoz za pomocí Markovských-modulovaných Poissonovských procesů s využitím Markovského řetězce. Markovský řetězec se skládá ze dvou stavů. Každý stav má přiřazené intenzity λ a průměrné doby pobytu požadavku v systému r. Může být vyjádřen také jako 4-n-tice *(λ1, λ2, r1, r2)*. Pokud chceme vypozorovat tyto parametry, musíme je odvodit z reálného síťového provozu. Parametry jsou vybrány pro popis následujících charakteristik: [11]

- Průměrná rychlost příchodů.
- Krátkodobý rozptyl od střední hodnoty počtu příchodů.
- Dlouhodobý rozptyl od střední hodnoty počtu příchodů.
- Počet příchodů v krátkém časovém úseku. [11]

Model může být dokonce rozšířen o více než 2 stavy. V tomto případě musíme použít rozšířenou matici řádu 2 (N +1), kde N je počet stavů (buffer size). Pokud je model navržen správně, pak není problém nastavit počet stavů na potřebnou hodnotu pomocí analytických prostředků. [11]

Ukázalo se, že model v minulosti vykazoval dobré výsledky při simulaci. Na druhou stranu se model spíše hodí pro hlasový přenos než pro datový, protože simulace založené na základu MMPM vykazovaly velice podobné problémy jako základní Poissonovské modely- špatné modelování shlukového provozu. [11]

2.6.3. Markovský modulovaný model proudění

Markovský modulovaný model proudění (Markov Modulated Fluid Model – MMFM) lze využít např. pro simulaci ATM přepínače. Jiné modely oproti MMFM berou příchod každé buňky jako samostatnou akci. MMFM charakterizuje příchod buněk jako průtok. Průtok lze charakterizovat jako příchod určitého počtu buněk (se stanoveným rozsahem) za časový interval. Události zde vznikají pouze, když se změní průtok příchodu buněk. Změny v průtoku buněk jsou méně časté než příchody jednotlivých buněk. Model proudění následkem toho využívá menší výpočetní výkon a paměť zdrojů, v porovnání s ostatními simulačními modely. [4]

Základním rysem modelu proudění je charakterizovat provoz na síti jako nepřetržitý proud vstupů s konečnou rychlostí datového toku. Příchozí datová rychlost je reprezentována jako stream s konečnou rychlostí. Vzhledem k jednoduchosti metody charakterizace provozu, model proudění je analyticky povolný. Jako každý jiný Markovský modulovaný proces, Markovský modulovaný model proudění (MMFM) používá základní Markovský řetězec, který určuje rychlost zdrojů. V každém okamžiku aktuální stav podkladových Markovských řetězců určuje průtok na vstupu. [4]

2.6.4. ON/OFF model

Tento model popisuje provoz entit mezi spojovou a aplikační vrstvou OSI modelu. Model se nejčastěji používá pro analýzu struktury IP provozu. Tento model používá pouze 2 stavy – ON/OFF. Doba trvání stavů ON/OFF se nazývá přechodový čas (transition time) a většinou bývá popsán exponenciálním rozdělením. [4]

Při realizaci modelu se používá N zdrojů. Tyto zdroje musí být statisticky identické a přitom na sobě nezávislé. Fronta o velikosti M je obsluhována konstantní rychlostí C. Parametr L charakterizuje zdroje při stavu ON. Parametr L udává průměrný počet paketů, které jsou při ON stavu generovány. Parametr S vyjadřuje maximální počet vygenerovaných paketů zdroje ve stavu ON. Poslední parametr r udává průměrný počet zdrojů tvořící frontu. Poměr mezi průměrným počtem zdrojů r a maximálním počtem vygenerovaných paketů S ve fázi zdroje ON se nazývá Rovnováha pravděpodobností a vypočítá se: [4]

$$\gamma = r / S. \tag{2.15}$$

Zdroj může být modelován jako dvoustavový Markovský řetězec, ovšem průměrný počet generovaných paketů ze zdroje musí být mnohem větší než 1, tedy L>>1. [4]

Přechod zdroje ze stavu OFF do stavu ON a naopak, se vypočítá:

t_1 (přechod ze stavu OFF do stavu ON): $\gamma S / (L(1-y))$ (2.16)

t_2 (přechod ze stavu ON do stavu OFF): S/L [4] (2.17)

2.6.5. IPP model

Tento model se nazývá Přerušený Poissonovský proces (Interrupted Poisson process, IPP). IPP využívá stejně jako ON/OFF dva stavy také nazývané ON/OFF. Uvažujme IPP v nespojitém čase, pak ve stavu ON přicházejí pakety do každého časového slotu. Přicházející pakety sledují Bernoulliho rozložení, které může být vyjádřeno za pomocí následující pravděpodobnostní funkce: [4]

$$P(X = 1) = p,\ P(X = 0) = 1 - p. \text{ [5]} \tag{2.18}$$

,kde p je pravděpodobnost a náhodné veličiny X obvykle vyjádřené hodnotami 0 až 1.

IPP je velice podobný ON/OFF modelu s tím rozdílem, že při stavu OFF zde nepřicházejí žádné pakety – není zde žádný provoz. [4]

2.7. Autoregressivní model

Autoregressivní Model patři do skupiny lineárních modelů, které snaží předpovídat aktuální výstup systému označený jako y_n, který je závislý na předchozích výstupech systému označené jako yk kde $k < n$ a na aktuálních vstupech v systému označené jako x_n a na předchozích vstupech u systému označené jako x_k kde $k < n$. [4]

Autoregressivní model řádu p, který se označuje jako AR (p), má následující tvar: [4]

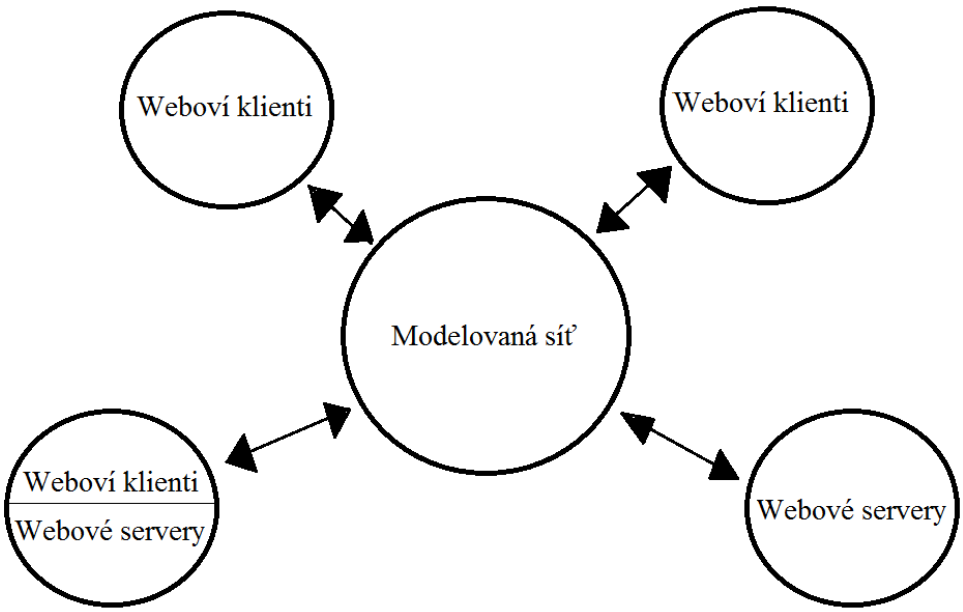
$$Xt = R1\,Xt - 1 + R2\,Xt - 2 + \ldots + Rp\,Xt - p + Wt \qquad (2.19)$$

kde W_t je bílý šum, R_i jsou reálná čísla a X_t jsou předepsané korelace náhodných čísel. Auto-korelační funkce AR (p) procesu se skládá z tlumených sinusových vln v závislosti na tom, zda kořeny řešení modelu jsou reálné, nebo imaginární. [4]

Existují i jiné varianty Autoregrresivních modelů. Varianta označovaná jako Auto-regrresivních model je závislá pouze na předchozích výstupech systému, nebo varianta označovaná jako model s pohyblivým průměrem (Moving Average Model-MAM), který je závislý na předchozích vstupech. Další je označován jako Diskrétní autoregrresivní model řádu p, označovaný jako DAR (p), který je podobný Autoregressivnímu modelu řádu p. DAR vytváří diskrétní veličiny se stejnými pravděpodobnostními distribucemi jako u Autoregressivního modelu. [4]

2.8. Stochastický model

Tyto novější modely byly navrženy pro generování syntetického provozu http. Webový provoz se vyjadřuje jako sekvence TCP spojení, kde jsou stránky a jiné objekty jednotlivých spojení popsané jako hodnoty náhodných proměnných. Tyto modely se skládají ze shromažďování statistických modelů, které určují vlastnosti náhodných proměnných. [4]



Obr. 2.4: Základní architektura Stochastického modelu

Středový obláček – Modelovaná síť představuje velké množství směrovačů, protokolů apod. Stanice (klienti či servery) jsou mezi sebou spojeny za pomocí TCP. Každé TCP spojení nese jeden či více požadavků nebo odpovědí vyměněné mezi klientem a HTTP serverem. Každá výměna se skládá ze žádostí či odezvy z klienta na server nebo ze serveru na klienta. TCP spojení přenášející více než jeden HTTP požadavků a odpovědí při výměně se nazývají trvalé připojení. [3]

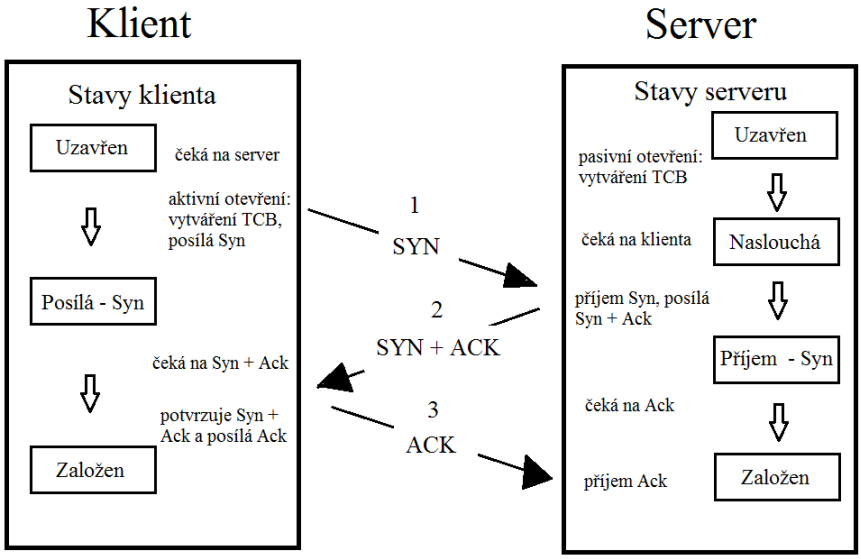
Připojení jsou seřazeny podle času zahájení a *i* značí *i-té* připojení. Stochastické modely počítají s těmito proměnnými hodnotami: [3]

- t_i: Doba mezi příchody paketů od začátku spojení *i* na straně klienta a na počátku spojení *i + 1* na straně serveru.
- R_i: Doba odezvy - čas mezi vysláním segmentu a doručením potvrzení ACK ve směru z klienta na server.
- r_i: Stejné jako R_i, ale ve směru ze serveru na klienta.
- B_i: Rychlost daná úzkým místem v síti ve směru z klienta na server.
- b_i: Stejné jako B_i, ale v opačném směru.
- L_i: pravděpodobnost ztráty paketů na ve směru z klienta na server.
- l_i: Stejné jako L_i, ale v opačném směru.
- p_i: Počet požadavků na stránku..
- $m_{i,j}$, $j = 1,\ldots P_i$: Počet vyměněných požadavků/odpovědí za *j* stránek při připojení *i*, celkový počet výměn požadavků/odpovědí je $n_i = m_{i,j} + \ldots + m_{i,pi}$.

- $F_{i,l}$,l = 1,. . . n_i: Velikosti požadavků ve směru z klienta na server.
- $f_{i,l}$, l = 1,. . . n_i: Stejné jako $F_{i,l}$, ale v opačném směru.
- $g_{(p)\,i,\,j}$, j = 1,. . . p_i - 1: p_i> 1, doba mezi příchodem posledního paketu, který nese poslední odpověď na klienta na stránce j a mezi vygenerováním prvního paketu, který přenáší první požadavek na klienta na stránce j + 1.
- $g_{(e)\,i,\,j,\,k}$, k = 1,. . . , $m_{i,\,j}$ - 1: Pro $m_{i,\,j}$ > 1, doba mezi příchodem posledního paketu,který nese odpověď na klienta pro soubor k a mezi vygenerováním prvního paketu, který nese první požadavek na klientský soubor k + 1.
- $D_{i,l}$, l = 1,. . . n_i: Serverové zpoždění, doba mezi příchodem posledního paketu, který nese požadavek na server a mezi vygenerováním prvního paketu, který nese odpověď na server. [3]

Popis měření zdrojových proměnných z časových razítek příchozích paketů na lince a z odpovídajících TCP / IP hlaviček, s výjimkou L_i, l_i, *Bi a bi:* [3]

- t_i: Doba mezi příchodem paketu SYN předcházejícího klienta a mezi paketem SYN stávajícího klienta na pozorovaném odkazu.
- R_i: Čas mezi klientským paketem SYN a mezi serverovým paketem SYN + ACK.
- r_i: Čas mezi odesláním serverového paketu SYN + ACK a přijetí klientského paketu ACK, který doplňuje třícestný handshake.



Obr. 2.5: Třícestný handshake

- p_i: Určeno z prahování parametru $g_{(e)\,i,\,j,\,k}$. Počet výměn požadavek / odpověď je definována jako přenos datových paketů obsahujících požadavek z klienta na následující přenos datových paketů obsahujících odpovědi ze serveru.
- M_{ij}, j = 1,. . . p_i: určuje práh parametru $g_{(e)\,i,\,j,\,k}$.

- $f_i, 1, f_i, 2, \ldots, f_{i,ni}$: Vypočteno ze sekvenčních čísel.
- $F_i, 1, F_i, 2, \ldots, F_{i,ni}$: Vypočteno ze sekvenčních čísel.
- $g_{(p)\,i,\,j,}\ j = 1, \ldots p_i - 1$. Doba mezi příchodem posledního datového paketu ze serveru při předchozí výměně a mezi dalším datovým paketem z aktuálního klienta z jiné stránky nejvyšší úrovně, mínus parametr r_i na straně klienta.
- $g_{(e)\,i,\,j,\,k,}\ k = 1, \ldots, mi, j - 1$: Doba mezi příchodem posledního datového paketu ze serveru při předchozí výměně a mezi dalším datovým paketem z aktuálního klienta ze stejné stránky nejvyšší úrovně, mínus parametr r_i na straně klienta.
- $D_i, 1, D_i, 2, \ldots, D_I,$ Doba mezi posledním datovým paketem na straně klienta a mezi prvním datovým paketem na straně serveru, mínus parametr R_i na straně serveru. [3]

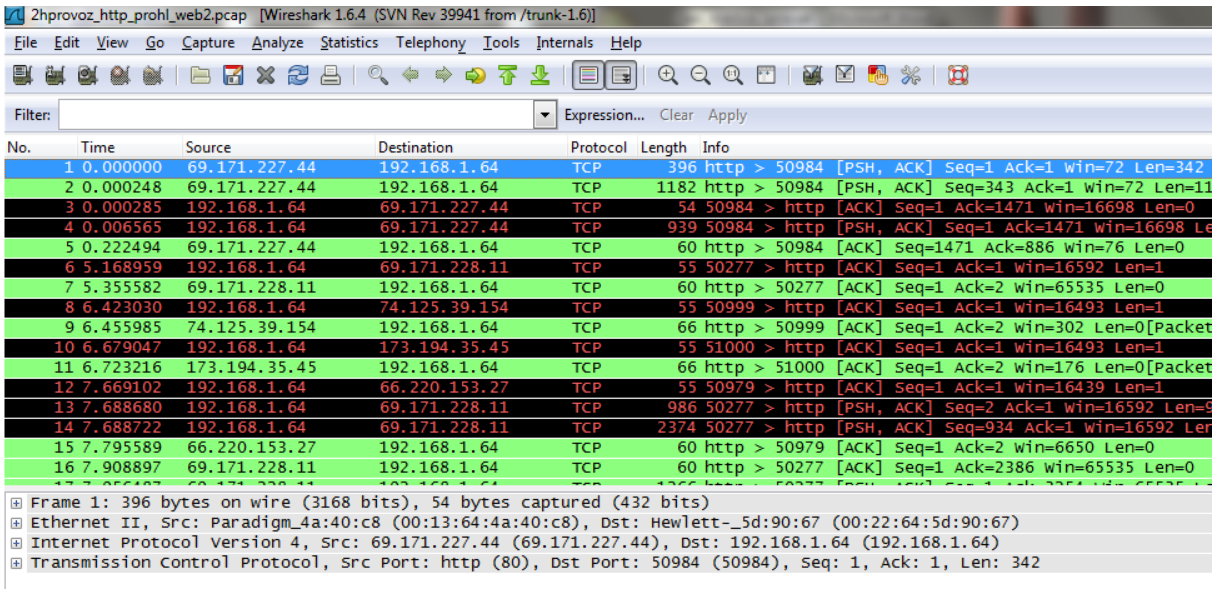
Většina proměnných pro spojení jsou také modelovány jako frakční součet-rozdíl, nebo FSD- modely časových řad. Označení y_i je pro FSD časové řady. Obecná marginální distribuční funkce y_i slouží k transformaci proměnné z_i, časové řady s normálním marginálem s průměrem 0 a rozdílností 1. U z_i se předpokládá, že sleduje Gaussovské časové řady s parametry d a θ [Stoch.Mod.str.9]. [3]

3. Zachytávání provozu

Kvůli odvození parametrů pro jednotlivé modely je nutné nejdříve zachytit provoz protokolu HTTP. Pro zachytávání byl použit program Wireshark verze 1.6.4. Ve filtru bylo nutné nastavit zachytávání na portu č. 80 (příkaz do filtru *tcp port http*), na kterém pracuje protokol HTTP. Zachytávání bylo provedeno na straně klienta, kde byla sledována jen jedna stanice. Tyto zachycená data se použijí pro analýzu vybraných modelů.

4. Základní analýza

Program Wireshark poskytuje 7 základních parametrů, které jsou rozděleny do sloupců. Číslo paketu (No) ukazuje pořadí zachyceného paketu. Čas (Time) udává, v jakém čase od zahájení zachytávání byl paket zachycen. Zdroj (Source) a cíl (Destination) udává v jakém směru paket přišel- z klienta na server nebo obráceně. Protokol (Protocol) udává, na jakém paket pracuje (TCP apod.). Délka (Lenght) udává velikost paketu v bytech. Informace (Info) poskytují bližší informace o paketu a liší se v závislosti na protokolu.



Obr. 4.1: Panel programu Wireshark

Testovaný provoz byl zachytáván 4 hodiny a 4 minuty. Bylo staženo celkem 146 636 paketů o celkové velikosti 112 300 846 bajtů. Průměrná rychlost paketů byla 10 za sekundu a průměrná rychlost bajtů za sekundu byla 7681,5.

Na zachyceném provozu se mohou objevit nežádoucí pakety, které vykazují chybu jako duplikace ACK, špatný kontrolní součet apod. Tyto pakety jsou ve Wiresharku odlišovány černou barvou s červeným písmem (viz obr. 4.1). V tomto případě stačilo do filtru zadat příkaz *tcp.analysis.flags* a objeví se jen chybové pakety. Poté, v panelu editovat, bylo zadáno ignorovat všechny zobrazené pakety. Může se stát, že Wireshark po příkazu ignorovat zobrazené pakety, zobrazí další chybové pakety, které předtím nezobrazoval. Proces se opakuje, dokud Wireshark přestane zobrazovat

chybové pakety. Poté byl zrušen zadaný filtr *tcp.analysis.flags*. Následně zůstaly potřebné pakety pro základní analýzu- na obr. 4.1 zobrazeny zelenou barvou (barva vyznačena konkrétně pro protokol HTTP a jemu podřízené). Zbylo 89 755 paketů o velikosti 61 205 612 bajtů. Pakety byly vyfiltrovány z důvodů neužitečné informace v panelu informace a navíc mnoho paketů vykazovalo duplikaci jiného paketu. Tyto faktory mohou ovlivnit základní analýzu.

Po odfiltrování nežádoucích paketových hlaviček se provoz rozdělil do dvou směrů. První je směr ze serveru na klienta a druhý je z klienta na server. Filtr se nastavoval podle IP adres u zdroje a cíle. IP adresa klienta byla 192.164.1.6, ostatní byly adresy různých serverů. Podle tohoto může být bez problému nastaven filtr na rozdělení směru. Ve sloupci zdroj byly označeny 3 pakety s IP 192.164.1.6 a na označené pakety byl aplikován filtr *ip.src == 192.168.1.6*. Poté na obrazovce zůstanou pakety ve směru klienta na server, které se uloží do pomocného souboru. Pro opačný směr ve sloupci cíl byl aplikován filtr *ip.dst == 192.168.1.6* a zobrazené pakety byly opět uloženy do jiného pomocného souboru. Nyní jsou pakety rozděleny do dvou směrů. Tento postup je určen pro analýzu komunikace z jedné stanice a neudává souhrnné informace o všech uživatelích služby.

Po rozdělení do dvou směrů je nutné, aby všechny pakety měly ve svém pátém sloupci napsáno TCP, protože pod tímto protokolem v sedmém sloupci informace (Info) jsou pro analýzu provozu potřebné údaje. Během zachytávání provozu se objevily pakety, které mají v pátém sloupci protokol (Protocol) např. protokol HTTP. Pro převedení do TCP je nutné najít paket s obsahem protokol HTTP a ve spodní části Wiresharku (pod 7 hlavními sloupci), kde jsou rozepsané podrobnější informace jako rámec, Ethernet II apod. (viz. obr. 4.1), se zakáže ukazování protokolu HTTP a veškeré zobrazení paketů, které mělo ve sloupci protokol HTTP se přepne do protokolu TCP.

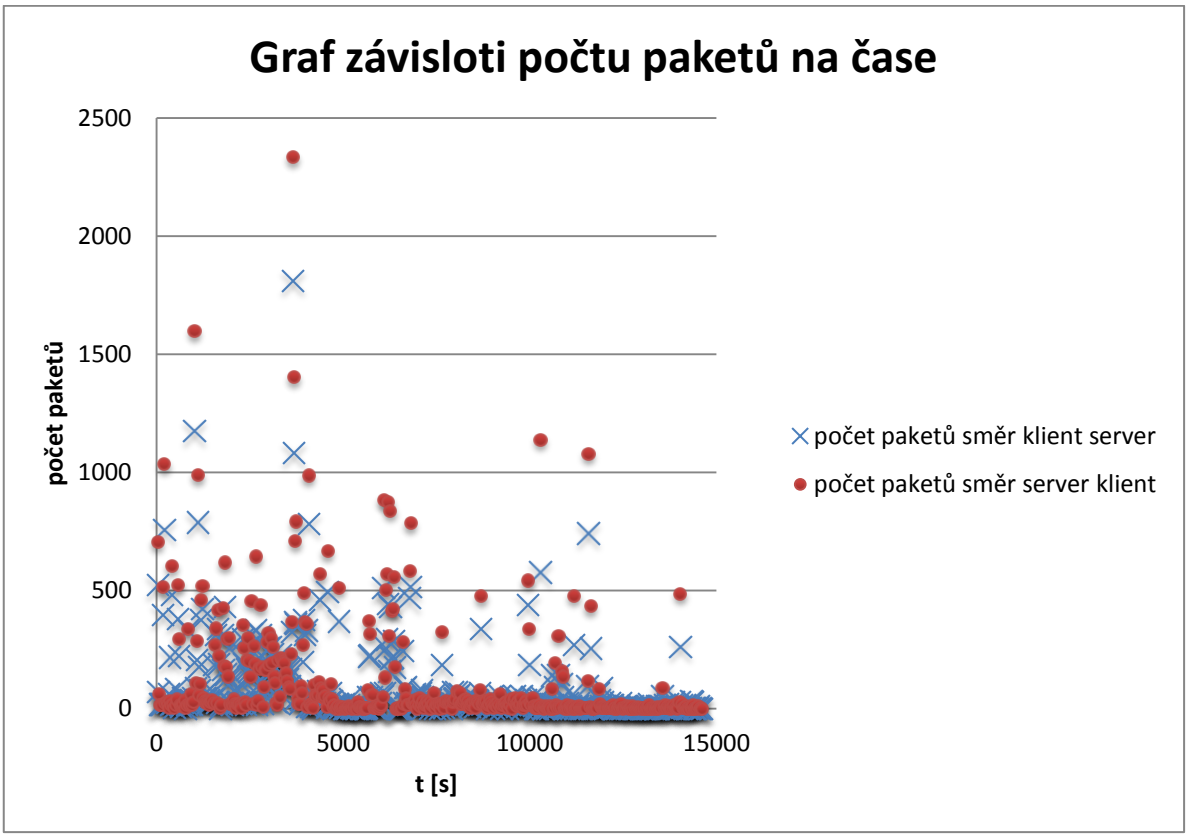
. V jednotlivých směrech je nutné pro základní analýzu vyčlenit 30 sekundové intervaly a v každém intervalu spočítat počet paketů, počet bajtů, počet nových spojení a průměrné trvání spojení otevřených během daného intervalu. Pro výpočty je nejprve nutné, zachycená data z Wiresharku exportovat do formátů csv souborů. Pokud jsou data exportovány v jiném z nabízených formátů, tak dochází ke špatnému přepisu. Tyto soubory csv se importují do programu Microsoft Office Excel a jsou opět uloženy jako

formát Excel.csv. V tomto formátu jsou přepsána data ze sedmi základních parametrů z Wiresharku. Tyto předchozí kroky je nutné provést pro oba směry. Tyto dva upravené soubory byly přesunuty do složky Python- výpočetní program pro analýzu, kde je připravený výpočetní soubor. V příkazovém řádku je nutno přejít do složky Python. Poté je nutné v příkazovém řádku zadat příkaz *python název_výpočetního_souboru.py název_upraveného_souboru_s_daty.csv > název_výstupního_souboru.csv.* Výstupem je soubor ve formátu csv, který je možno otevřít v programu Excel. V tomto výstupním souboru každý řádek představuje interval o délce 30 sekund. Tyto řádky jsou seřazeny chronologicky. Výstupní soubor obsahuje 4 sloupce. První sloupec představuje počet paketů, druhý velikost dat v bajtech, třetí počet nových spojení a čtvrtý představuje průměrnou délku trvání spojení. Jako poslední úprava se provede nahrazení všech teček za čárky v programu Pspad. Pokud by tento krok nebyl proveden, mohl by nastat problém nesourodnosti dat.

Program Prython pracuje se sedmi základními parametry, které jsou exportované z Wiresharku. Nejprve je nutné rozdělit data na intervaly o délce 30 sekund. Pro tento úkol je využit parametr čas, který obsahuje čas zachycení paketu. Po překročení časového rozdílu 30 sekund se vytvoří další řádek pro zapisování čtyř požadovaných výpočtů. Počet paketů pro jednotlivý interval je spočítán jako součet řádků, které danému intervalu náleží. Každý řádek představuje jeden paket. Počet bajtů v intervalu se spočítá jako součet všech hodnot ve sloupci délka. Tyto kroky jsou stejné pro oba směry. Počet nového spojení se započítá, pokud se ve sloupci informace objeví údaj [SYN] nebo [SYN, ACK]. Pro směr ze serveru na klienta se spojení započítá pro údaj [SYN, ACK] a pro směr z klienta na server údaj [SYN]. Zároveň je nutné, zapisovat číslo portů u těchto údajů, které se nachází také ve sloupci informace a zapisovat čas ve sloupci čas. Pro průměrnou délku trvání spojení program Python vyhledává ve sloupci informace údaj [FIN, ACK] a pomocí čísla portů přiřadí spojení k údaji [SYN, ACK] či [SYN] se shodným číslem portu. Díky rozdílu času u údaje [SYN, ACK] či [SYN] a u údaje času [FIN, ACK] je zjištěna délka trvání spojení. Po té je vypočítán průměr. Může se stát, že délka trvání spojení přesáhne několik intervalů. V tomto případě je délka trvání spojení započítána do intervalu kde se objeví [FIN, ACK]. Pokud nastane případ, že údaj [SYN, ACK] či [SYN] a [FIN, ACK] nenajde příslušnou dvojici, spojení není započteno. Obvykle k tomu dochází na začátku a konci zachytávání. Výjimečně je tomu

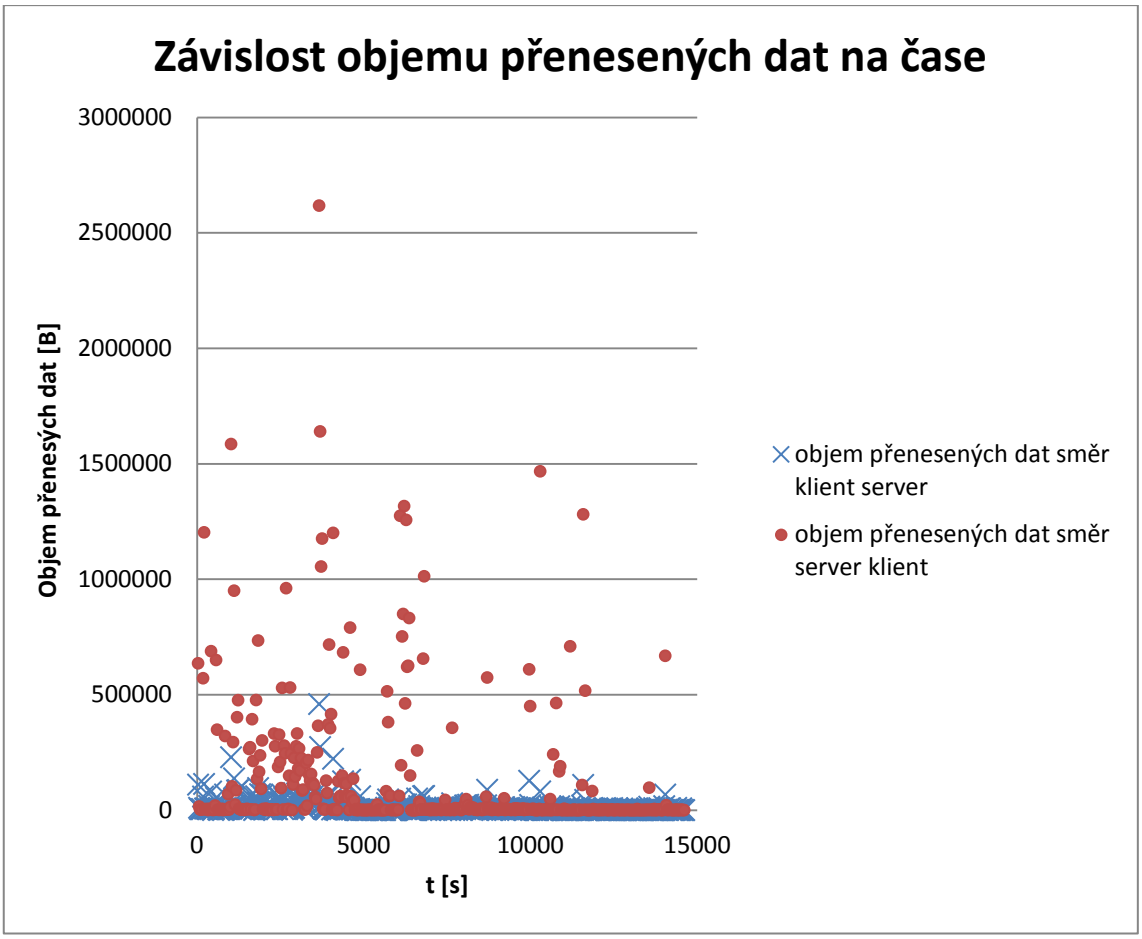
jinak. Příslušný kód výpočetního souboru je v přílohách s komentářem. Název výpočetního souboru je stats.py (viz Seznam příloh).

Data z výstupních souborů, byly vyneseny do grafů v závislosti na čase. V grafech jsou rozlišeny 2 bodové křivky. Každá představuje jiný směr. Každý bod představuje vypočítané hodnoty (záleží na grafu) závislé na časovém úseku o délce 30 sekund, ve kterém jsou zachyceny.



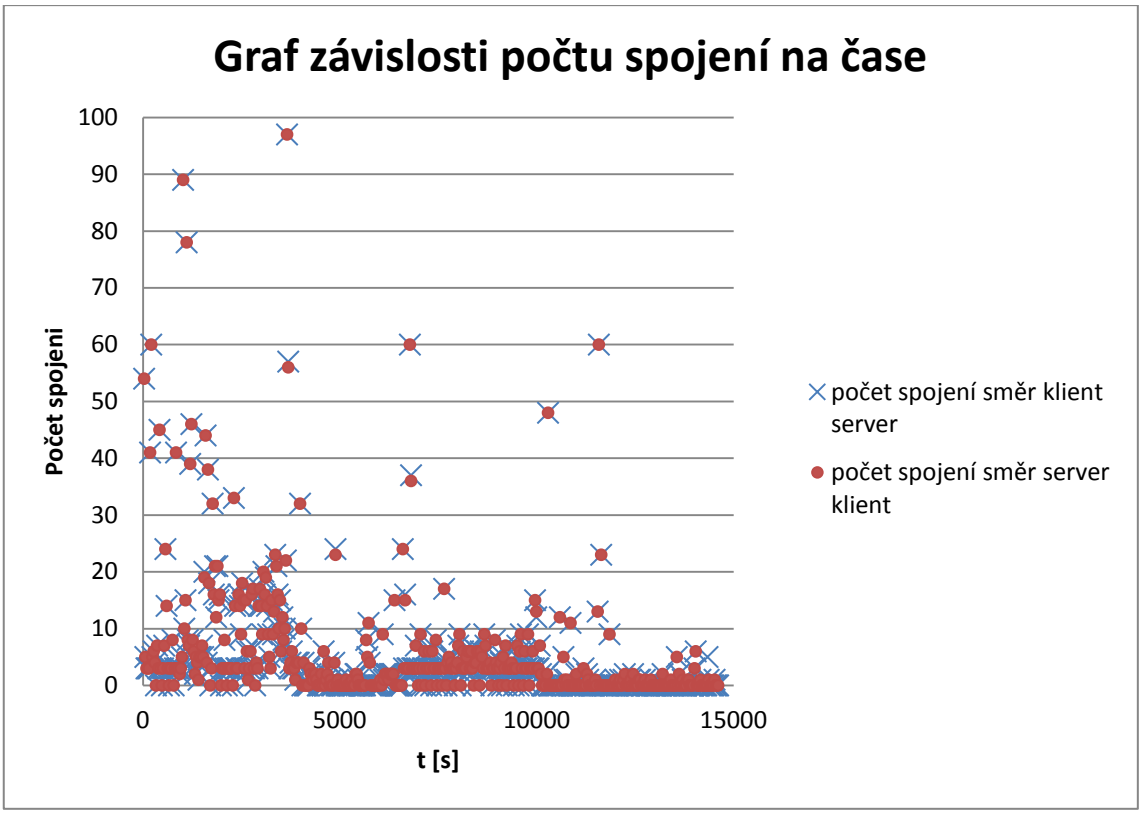
Obr. 4.2: Graf závislosti počtu paketů na čase.

V obrázku 4.2 jsou jednotlivé body na sobě nezávislé. Z grafu vyplívá, že pokud byla zvýšená aktivita na straně serveru, došlo i ke zvýšené aktivitě na straně klienta. Z celkového počtu paketů bylo přeneseno 56 % směr server klient a 44 % pro opačný směr. Celkový počet paketů během 30 sekundových intervalů se pohyboval kolem 100 a méně- okolo 80 % hodnot. Zbylých 20 % hodnot přesáhlo hodnotu 100. Intervaly ve kterých se objevil počet paketů 500 a výše byly ojedinělé:



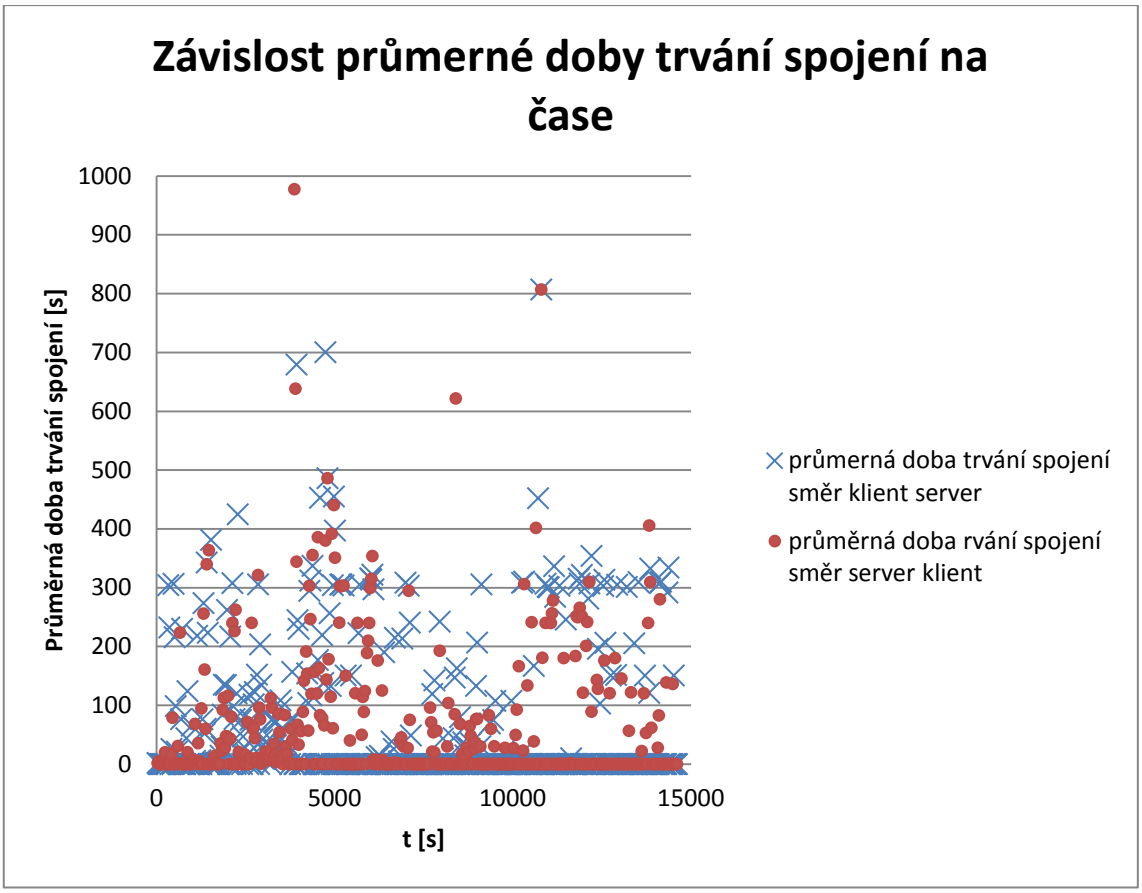
Obr. 4.3: Graf závislosti objemu přenesených dat na čase.

V obrázku 4.2 jsou jednotlivé body na sobě nezávislé. Z celkového objemu dat bylo přeneseno okolo 88 % směr server klient a 12 % objemu dat v opačném směru. Celkový počet přeneseného objemu dat během 30 sekundových intervalů se pohyboval v rozmezí 0 až 3 kB okolo 36 %. V rozmezí 3 až 20 kB bylo přeneseno 47 % hodnot. Rozmezí 20 kB a výše přesáhlo 17 % hodnot.



Obr. 4.4: Graf závislosti počtu spojení na čase.

V obrázku 4.4 jsou jednotlivé body na sobě nezávislé. Ze směru klient server nebo server klient byl počet spojení na 99 % identický. V menším počtu spojení např. pod 20 bylo množství spojení na straně klienta a serveru totožné. Ve vyšších hodnotách se počet spojení může lišit s velmi malou odchylkou. Z celkového počtu spojení byl rozsah 0 až 15 spojení okolo 91 % hodnot. Pro počet spojení 15 a vyšší vykazovalo 9 % hodnot.



Obr. 4.5: Graf závislosti průměrné doby trvání spojení na čase.

V obrázku 4.5 z celkové doby trvání spojení bylo spojení udržováno na straně klient server 53 % a na straně server klient 47 %. Průměrná dobra trvání spojení se pohybovala v rozmezí 1 až 30 sekund pro 70 % hodnot. Pro 30 % hodnot přesáhla průměrná doba trvání spojení 30 sekund.

5. Analýza pomocí Markovského modelu

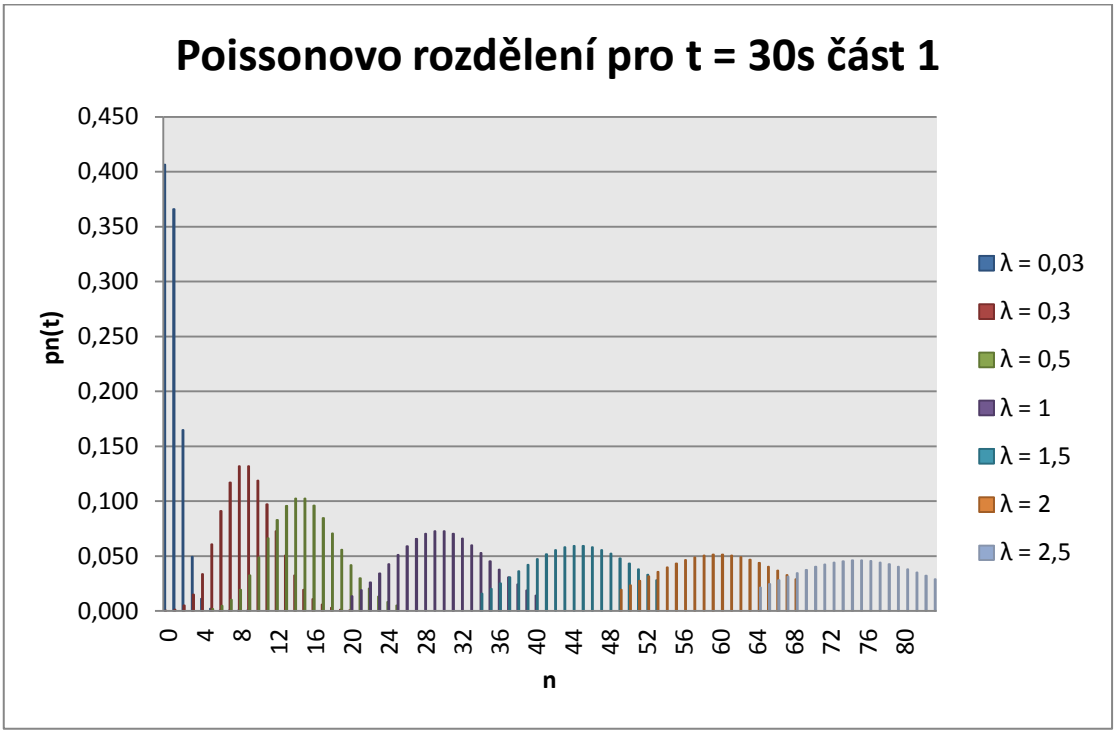
Pro modelování provozu pomocí Markovského modelu byla zvolena závislost počtu paketů na čase. Metoda výpočtu pro oba směry provozu je stejná. Pro modelování byl vybrán směr s klienta na server. Závislost počtu paketů na čase byla analyzována v předchozí kapitole. Model je založen předpovědi příchodu počtu paketů.

Model se řídí podle

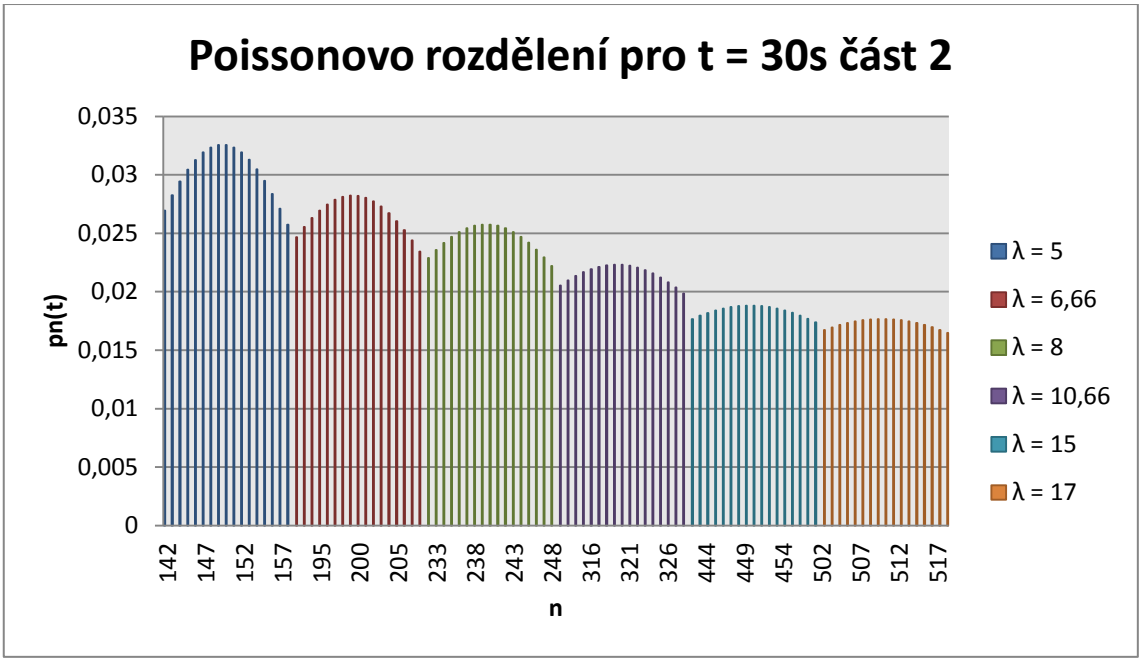
$$pn(t) = \frac{(\lambda t)\hat{}n}{n!} e^{-\lambda t} \tag{5.1}$$

kde (v čitateli je *n* v exponentu) *n* = 0,1,2,.. *n*, což je vztah pro Poissonovo rozdělení. Ve vztahu *n* vyjadřuje počet paketů, *t* časový interval, ve kterém byly pakety zachyceny, λ vyjadřuje intenzitu provozu vypočítanou ze vztahu 2.11 a $p_n(t)$ vyjadřuje pravděpodobnost výskytu počtu paketů v daném časovém intervalu. [8]

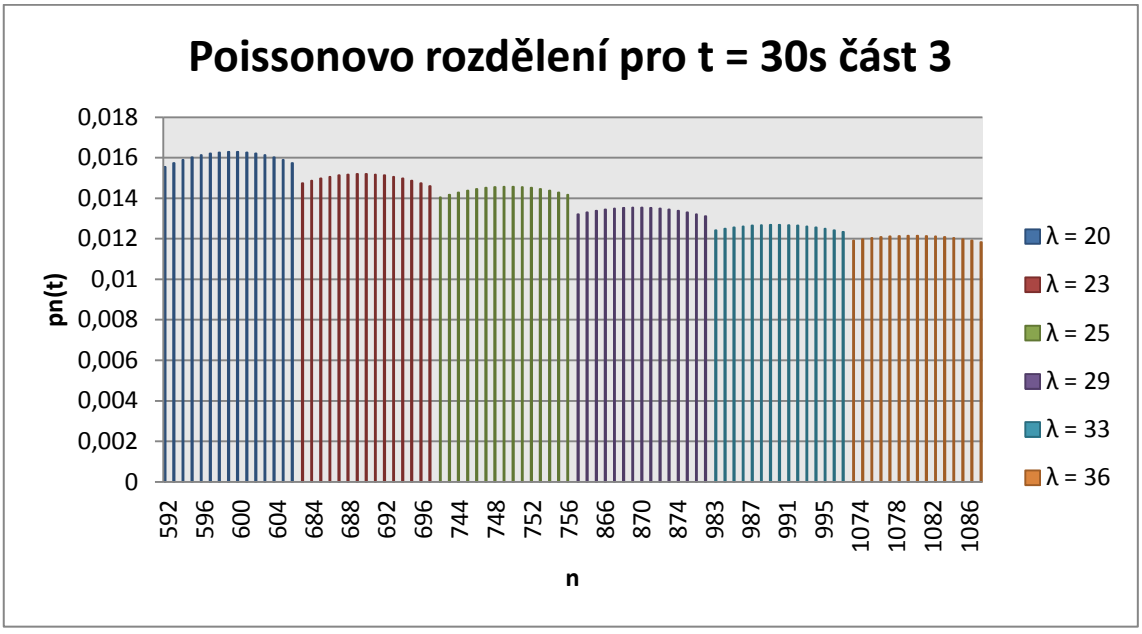
Pro model je nutné vytvořit několik funkcí λ pro dostatečné pokrytí v definičním oboru- definiční obor vyjadřuje počet paketů. Pro každou zvolenou intenzitu je nutné zvolit alespoň 15 hodnot pro *n*, které vykazují pro danou intenzitu největší pravděpodobnost.



Obr. 5.1: Poissonovo rozdělení pro t = 30s část 1. [8]



Obr. 5.2: Poissonovo rozdělení pro t = 30s část 2. [8]

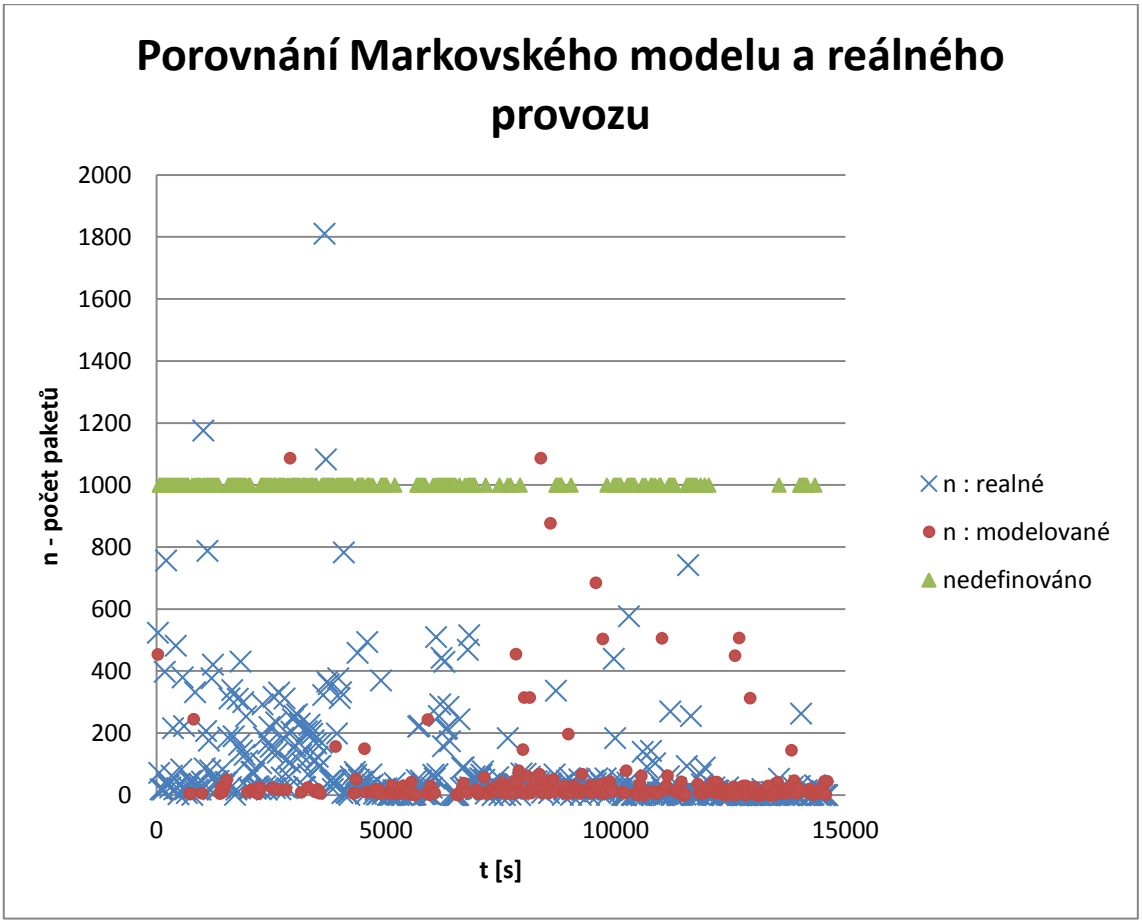


Obr. 5.3: Poissonovo rozdělení pro t = 30s část 3. [8]

Pro lepší přehlednost byly funkce rozděleny do tří grafů. V obrázcích 5.1, 5.2 a 5.3 představují $p_n(t)$ pomocné vymodelované funkce, které slouží k přiřazení vymodelovaných hodnot n. V případě pevně daného intervalu se projeví diskrétní

charakter. V základní analýze byl provoz rozdělen na intervaly o délce 30s. Z tohoto důvodu Poissonovo rozdělení pro t = 30s. [8]

Z reálného provozu byly spočítány λ předpovědi. Intenzity předpovědi byly vypočítány jako průměr pěti předchozích intenzit. Vypočítané intenzity předpovědi byly dosazeny do $p_n(t)$ Poissonova rozdělení a k nim byl dosazen n reálný počet paketů, který odpovídal časovému intervalu pro intenzitu předpovědi. Vypočítaná $p_n(t)$ byla porovnána s vymodelovanými $p_n(t)$. Díky tomuto porovnání byla zjištěna modelovaná hodnota n.



Obr 5.4: Srovnání reálných a modelovaných hodnot n pomocí Markovského modelu.

Model vykazoval přesnost hlavně na nižších hodnotách počtu paketů. Model byl schopen vyjádřit okolo 56 % hodnot. Pokud se objevili nárazy velice převyšující předchozí počet paketů- model selhal. Hodnota 1000 pro zelené body nedefinováno, byla zvolena jen pro lepší přehled. Ve skutečnosti u nedefinovaných bodů byla hodnota

$p_n(t)$ velmi malé číslo a z tohoto důvodu nebylo možné určit modelované n. Model může být zdokonalen a to následujícími způsoby:

- Predikci λ odhadovat z více než pěti předchozích λ.
- Rozšířit funkce $p_n(t)$ o více než 15 hodnot n.
- Vymodelovat více $p_n(t)$, které mezi sebou mají minimální rozdíl.

Pokud by reálný provoz vykazoval lineární charakteristiku příchodu počtu paketů, byl by přesný. Ovšem provoz na protokolu HTTP vykazuje nárazovou charakteristiku příchodů počtů paketů.

6. Analýza pomocí Stochastického modelu

Stochastický model je schopen modelovat několik parametru, které byly popsány v kapitole 2.8. Tato práce se zaměří na parametr R_i. Parametr R_i vyjadřuje čas mezi klientským [SYN] a mezi [SYN, ACK] na straně serveru. Pro zjištění reálných hodnot R_i se použije podobný postup jako u základní analýzy dat. Pro výpočet je použit program Python, který vymezí 30 sekundové intervaly, ve kterých budou zjištěny časy R_i a v každém intervalu spočítán jejich průměr. Pro výpočet je nutné, aby exportovaná data z Wiresharku nebyla rozdělená podle směru, ale naopak, aby byly oba dva směry sloučené, protože zprávy [SYN] chodí pouze ve směru klient server a zprávy [SYN, ACK] pouze ve směru server klient. Náležící dvojice [SYN] a [SYN,ACK] se poznají podle stejného čísla portu ve sloupci informace a podle opačných zdrojů a cílů u IP adres (zpráva [SYN] má IP adresu např. A → B a [SYN,ACK] má IP adresu B → A). Zdrojový kód programu Python je v přílohách společně s popsaným komentářem (liší se od zdrojového kódu, který je určený pro základní analýzu dat). Model je založen na předpovědi velikosti času R_i.

Výsledná modelovaná hodnota R_i se vypočítá podle Gaussovských časových řad

$$zi = \sqrt{1-\theta}\, si + \sqrt{\theta}\, ni \text{ [3]} \tag{6.1}$$

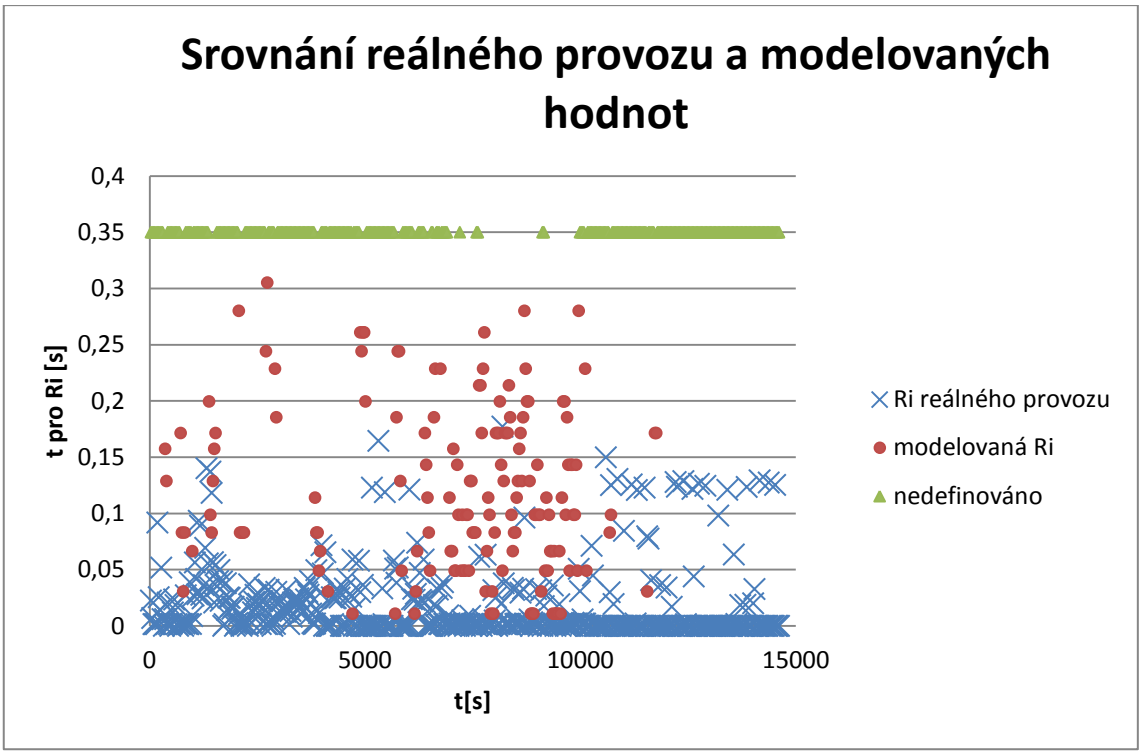
kde s_i je frakční ARIMA proměnlivá podle parametru d, n_i je Gaussovský bílý šum a parametr θ je pomocný proměnlivý parametr podle parametru p, který je noví počet spojení. Pomocí parametr $d = 0{,}32$ je zvolen podle práce [3] a vypočítá s_i ze vztahu [3]

$$si = ni + ni - 1 \text{ [1]} \tag{6.2}$$

kde n_i je Gaussovský bílý šum, který se vypočítá ze vztahu

$$ni = \frac{1-d}{2}. \text{ [1]} \tag{6.3}$$

Parametr θ se vypočítá podle vztahu $logit2(\theta(\rho)) = -0.053 + 0.396\ log_2(\rho)$. Parametr náleží $0 \leq \theta \leq 1$. Za p se dosadí hodnota predikce počtu spojení, která se spočítá jako průměr předchozích pěti hodnot součtu počtu ze směru klient server. [3]



Obr 6.1: Srovnání reálného provozu s modelovanými hodnotami R_i pomocí Stochastického modelu.

Stochastický model byl schopen modelovat okolo 30 % hodnot. Hodnota 0,35 pro nedefinováno je zvolena pro lepší přehled. Z nedefinovaných hodnot okolo 95 % nesplnilo podmínku $0 \leq \theta \leq 1$. Počet spojení p byl pro modelování příliš vysoký. Ideální počet spojení pro modelování se pohyboval v rozmezí od 3 do 7. Pro zbylých 5 % hodnot vycházela modelovaná z_i vykazovala záporné číslo. Je to dáno zjednodušeným matematickým vyjádřením s_i a bílého šumu n_i s pevně daným parametrem d. Pokud by byl počet spojení v ideálních hodnotách, model by vykazoval schopnou predikci. Testované úseky by se musely zkrátit, aby se eliminoval vyšší počet spojení.

Závěr

Cílem této práce bylo zachytit síťový provoz a analyzovat pomocí vybraných modelu. V praktické části byl zachycen provoz na protokolu HTTP pomocí programu Wireshark. Po zachycení a úpravě provozu byla provedena základní analýza reálného provozu, která sloužila pro přípravu na analýzu pomocí vybraných modelů. Pro první analýzu byl vybrán Markovský model. U Markovského modelu se podařilo vymodelovat 56 % hodnot. U Markovského modelu byla procentuální úspěšnost ovlivněna nárazovými příchody paketů. U stochastického modelu se podařilo vymodelovat 30 % hodnot, ale hodnoty se svou velikostí více blížili hodnotám reálného provozu oproti Markovskému modelu. U stochastického modelu byla procentuální úspěšnost ovlivněna počtem spojení. V počtu nárazových (vyšších) příchodů paketů mohl být počet spojení ideální. Naopak v nižších počtech paketů mohl být počet spojení nevyhovující. Počet spojení vykazoval náhodný charakter.

Důležitými parametry jsou požadavky na straně klienta a webové stránky, které jsou navštěvovány. Z těchto důvodů má síťoví provoz HTTP náhodný charakter. Proto je modelování HTTP provozu obtížné. Přesto je důležité se pokoušet o upřesnění vyjádření matematického modelování pro protokol HTTP vhledem k narůstajícím požadavkům sítě.

Seznam použité literatury

[1] ANDERSON, David, BOWEI, Xi, CLEVELAND, William S. *Multifractal and Gaussina Fractional Sum-Difference Models for Internet Traffic* [online]. [cit. 29.4.2012] Dostupné na www: http://fodava.gatech.edu/files/reports/FODAVA-10-25.pdf.

[2] BECCHI, Michela. *From Poisson Processes to Self-Similarity: a Survey of Network Traffic Models* [online]. 2006 [cit. 2011-10-20] Dostupné na www: http://www1.cse.wustl.edu/~jain/cse567-06/ftp/traffic_models1/index.html

[3] CAO, Jin, CLEVELAND, William S., GAO, Yuan, JEFFAY, Kevin, F. SMITH Donelson, WEIGLE, Michele. *Stochastic Models for Generating Synthetic HTTP Source Traffic* [online]. [cit. 2011-11-26] Dostupné na www: http://www.stat.purdue.edu/~wsc/papers/packmime.http.pdf

[4] CHANDRASEKARAN, Balakrishnan. *Survey of Network Traffic Models* [online]. 2006 [cit. 2011-11-02] Dostupné na www: http://www1.cse.wustl.edu/~jain/cse567-06/ftp/traffic_models3/index.html

[5] Creative Commons. *Bernoulli Distribution* [online]. [cit. 2011-11-12] Dostupné na www : http://en.wikipedia.org/wiki/Bernoulli_distribution

[6] Creative Commons, *Gaussian process* [online]. [cit. 2011-11-12] Dostupné na www: http://en.wikipedia.org/wiki/Gaussian_process

[7] Creative Commons, *Hypertext Transfer Protocol* [online]. 14. 11. 2011 [cit. 2011-10-19].Dostupné na www: http://cs.wikipedia.org/wiki/Hypertext_Transfer_Protocol

[8] MOLNÁR, Karol. *Moderní síťové technologie 2011* [online]. [cit. 2011-10-29] Dostupné na www : http://www.utko.feec.vutbr.cz/~molnar/mmos/fronty.pdf

[9] MOLNÁR, Karol. *Řízení toku dat a Řízení provozu u protokolu TCP* [online]. [cit. 2011-10-20] Dostupné na www: http://www.utko.feec.vutbr.cz/~molnar/index.php?stranka=mmos

[10] VISHWANATH, Kashi Venkatesh a VAHDAT, Amin. *Swing: Realistic and Responsive Network Traffic Generation* [online]. [cit. 2011-11-11] Dostupné na www : http://cseweb.ucsd.edu/~kvishwanath/papers/swington.pdf

[11] WILSON, Michael. *A historical view of network traffic models*. [online] 2006 [cit. 2011-10-20] Dostupné na www: http://www.cse.wustl.edu/~jain/cse567-06/ftp/traffic_models2/index.html

[12] ZAPLETAL, Lukáš. *Protokol HTTP 1.1 pod lupou* [online]. 27. 3. 2001 [cit.2011-10-19]. Dostupné na WWW: http://www.root.cz/clanky/protokol-http-1-1-pod-lupou/

Seznam symbolů, veličin a zkratek

ACK Acknolegment – potvrzení

AR Autogressiv Model – Autogressivní model

ATM Asynchronus Transfer Mode – asynchronní přenosový mód

DAR Discrete Autoregressiv Model – diskrétní Autoregressivní model

DMC Discrete Markov Chain – diskrétní Markovský řetězec

FBM Franctional Brownian Motion – frakční Brownův pohyb

FIN Finish – dokončit

HTML HyperText Markup Language – značkovací jazyk pro hypertext

HTTP HyperText Transfer Protocol – hypertextový přenosový protokol

IP Internet Protocol – internetový protokol

IPP Interrupted Poisson Proces – přerušený Poissonův proces

LRD Long Range Dependence – závislost na delším rozsahu

MAM Moving Avereage Model – posuvný průměrový model

MC Markov Chain – Markovský řetězec

MMFM Markov Modulated Fluid Model – Markovský modulovaný model proudění

MMPM Markov Modulated Poissen Model – Markovský modulovaný Poissnovský model

OSI Open Systém Interconnection – otevřený systém propojení

P2P Peer to Peer – klient klient (doslova rovný s rovným)

RREs Request, Response, Exchanges – žádosti, odpovědi, výměny

SMTP Single Mail Transfer Protocol – jednoduchý přenosový protokol pro poštu

SRD Short Range Dependence – závislost na kratčím rozsahu

SSL Secure Sockets Layer – zabezpečená zásuvná vrstva

SYN Synchronize – synchronizace

TCP Transmission Control Protocol – přenosový řídící protokol

UDP User Datagram Protocol – uživatelský diagramový protokol

URI Uniform Resource Identifier – jednotný identifikátor zdroje

URL – Uniform Resource Locator – jednotný lokátor zdroje

Seznam příloh

Přílohy na cd

Celek.pcap – stažený provoz (soubor se otvírá pomocí programu Wireshark)

Filtr.pcap – stažený provoz bez nadbytečných paketů

Markovský.xlsx – soubor s pomocnými výpočty pro Markovský model včetně grafů

Pracovní.xlsx – soubor s pomocnými výpočty pro základní analýzu včetně grafů

Stats.py – zdrojový kód pro výpočet základní analýzy (soubor se otvírá pomocí Wordpadu nebo pomocí poznámkového bloku)

Stats2.py – zdrojový kód pro výpočet času R_i

Stochastický.xlsx – soubor s pomocnými výpočty pro Stochastický model včetně grafů