MASARYKOVA UNIVERZITA
FAKULTA INFORMATIKY

Příprava infrastruktury a informačních materiálů pro hru BW challenge

Bakalářská práce

Irena Hovorková

Brno, 2012

Prohlášení

Prohlašuji, že tato práce je mým původním autorským dílem, které jsem vypracovala samostatně. Všechny zdroje, prameny a literaturu, které jsem při vypracování používala nebo z nich čerpala, v práci řádně cituji uvedením úplného odkazu na příslušný zdroj.

Hovorková Irena

Vedoucí práce: RNDr. Tomáš Rebok, Ph.D.

Poděkování

Děkuji svému vedoucímu RNDr. Tomáši Rebokovi, Ph.D za cenné rady a připomínky k mé práci. Také bych ráda poděkovala panu Petru Kadlecovi za jeho trpělivost a ochotu poradit a panu Jakubovi Daňkovi za jeho důležité rady.

Své rodině děkuji za trpělivost při psaní této práce.

Shrnutí

Hlavním cílem této práce je příprava síťové a výpočetní infrastruktury pro hru Bandwidth challenge. Dále pak vytvoření všech nezbytných softwarových nástrojů včetně rezervačního systému pro přístup k reálné síťové infrastruktuře určené pro testování vytvořených protokolů. Zúčastněným týmům by měla poskytovat virtualizované prostředí pro vývoj.

Dále je součástí této práce příprava informačních materiálů ve formě letáku a webových stránek.

Klíčová slova

protokol TCP, hra, síťová infrastruktura, PXE, NFS, VMware, rezervační aplikace, Bandwidth challenge

Obsah

1 Úvod

Počítačové sítě jsou důležitou součástí našeho života. Poskytují uživateli možnost komunikace a sdílení nejrůznějších zdrojů (informace, hardwarové zdroje, programové vybavení). Existují již řadu let či spíše desetiletí, avšak s těmi původními nemají dnešní sítě mnoho společného. Jejich rozloha se zvětšila natolik, že přesahují hranice kontinentů. S tímto souvisí i počet uživatelů, který neustále roste už od dob vzniku prvních sítí. Komunikace přes počítačovou síť, ať už v podobě elektronické pošty, IP telefonie nebo kontaktu pomocí sociálních sítí, se již stala pro společnost naprostou samozřejmostí. Proto jsou počítačové sítě pro fungování dnešního světa nepostradatelné. Teprve pokud nejsou služby komunikačních sítí dostupné, si lidé uvědomí jejich opravdový význam.

Neustálý rozvoj počítačových sítí je umožněn vývojem nových síťových technologií. Síťové prvky, protokoly používané pro přenos i přenosová media jsou stále zdokonalovány. Oproti počátkům sítí se dnes data přenášejí vzduchem se stejnou samozřejmostí jako optickými vlákny. Linka s kapacitou 10 Gb/s již není ničím výjimečným, stejně tak získání informací od serveru na druhém konci světa do několika vteřin.

Pro přístup k datům sdílených v síti slouží uživatelům aplikace. Ta však ne-komunikuje přímo s počítačovou sítí, ale pouze s dalším prostředníkem, který už je síti blíže, a předává mu své požadavky. Podle tzv. OSI[1] modelu tvoří rozhraní mezi uživatelskými programy a nižšími vrstvami, které zajišťují samotný přenos dat, tzv. transportní vrstva. Tato vrstva je zodpovědná za doručení požadovaných informací správné aplikaci a zároveň využívá služeb síťové vrstvy, která má na starosti adresaci síťových prvků. Pomocí svých protokolů je transportní vrstva schopna ovlivnit, zda budou data přenášena spolehlivě bez ztrát, nebo naopak nespolehlivě, ale s minimálním zpožděním. Výběr protokolu záleží pouze na aplikaci a na tom, jaké služby vyžaduje.

1 Reference model of Open System Interconnection je popsán v druhé kapitole.

V průběhu let se podmínky, pro které byly protokoly vytvořeny, velmi změnily a původní implementace jim již nevyhovují. Jelikož se linky používané v dnešních sítích liší nejen kapacitou a vzdáleností uzlů, které propojují, ale především zpožděním, nejsou používané mechanismy např. pro řízení zahlcení dostatečně účinné. Z těchto důvodů musí být protokoly neustále zdokonalovány, aby byly schopny účinně pracovat. Jedním z takových protokolů je TCP.

Jak již bylo zmíněno, transportní vrstva je schopna zajistit spolehlivý přenos dat. Používá k tomu protokol TCP, jehož původní varianty však nedisponují potřebnou agresivitou. Ačkoliv je klasický protokol TCP schopen využít dostupné kapacity, neděje se tak dostatečně rychle, a tudíž není efektivní. V případě výpadku musí být schopen zotavit se v co nejkratším čase, což je označováno jako responsivita. Jelikož linky již dávno nejsou využívány v určitém okamžiku pouze jedním účastníkem, je na protokol kladen další požadavek – férovost. Ten představuje zajištění spolehlivého rozdělení kapacity sítě. Tyto základní požadavky není původní protokol TCP schopen v aktuálních sítích zajistit. V důsledku toho byl v průběhu let nadále zlepšován a rozvíjen.

Za dobu využívání TCP bylo vytvořeno mnoho variant, které více či méně úspěšně řešily problémy s efektivním využíváním dostupné šířky pásma. Nabízí se tedy téma jejich porovnání, odhalení nedostatků a případně vytvoření nové varianty. Tato bakalářská práce představuje možnost otestovat vlastnosti a chování variant TCP na reálné infrastruktuře netradiční formou. Prostřednictvím v této práci navrhované hry Bandwidth challenge dostanou týmy studentů či ostatních zájemců šanci vyzkoušet chování protokolu v praxi, vylepšit ho a zároveň porovnat své schopnosti s ostatními skupinami. Cílem práce tedy je připravit síťovou infrastrukturu a potřebný software pro tuto hru a zároveň zajistit její propagaci pomocí webu a informačních materiálů.

2 ISO/OSI model

Referenční model propojování otevřených systémů (*Reference model of Open System Interconnection*) je standard navržený organizací ISO. Byl vytvořen kvůli

zajištění schopnosti komunikace mezi síťovými zařízeními různých výrobců. V minulosti bylo možné vytvořit počítačovou síť složenou pouze ze síťových prvků stejného výrobce. Ta ovšem používala pouze proprietární protokoly, které nebyly kompatibilní s protokoly ostatních výrobců. Bylo tedy nutné vytvořit normu, podle níž by se mohla výroba nových prvků řídit. Touto normou je již zmíněný OSI model.

Model OSI je rozložen do sedmi vrstev. Každá z nich zajišťuje určitý úkol a komunikuje pouze se svými přímými sousedy. Vyšší vrstva vždy využívá služeb poskytovaných nižší vrstvou. Odesílaná a přijímaná data procházejí postupně všemi vrstvami od nejvyšší aplikační až po fyzickou.

- **Aplikační vrstva**

Poskytuje uživateli přístup k síťovým zdrojům prostřednictvím síťových aplikací, programů a protokolů. Nejznámějšími protokoly jsou např. HTTP (*Hypertext Transfer Protocol*) využívaný webovými prohlížeči, poštovní protokol SMTP (*Simple Mail Transfer Protocol*) nebo FTP (*File Transfer Protocol*).

- **Prezentační vrstva**

  Má na starosti překlad, šifrování a kompresi dat. Řeší odlišnosti ve vnitřní reprezentaci dat komunikujících stran.

- **Relační vrstva**

  Zajišťuje ustavení spojení – relací, mezi komunikujícími aplikacemi, jejich správu a ukončení. Umožňuje šifrované spojení a autentizaci pomocí protokolu SSL.

- **Transportní vrstva**

  Poskytuje vyšším vrstvám efektivnější přenosové služby, než jaké nabízí vrstva síťová. Její funkcí je především adresace a doručení dat v podobě segmentů nebo datagramů (v závislosti na použitém protokolu) mezi procesy.

- **Síťová vrstva**

  Směrováním paketů umožňuje jejich doručení mezi uzly, které nejsou přímo spojeny. Poskytuje jednoznačnou identifikaci zařízení v síti pomocí IP adresy.

- **Linková (spojová) vrstva**

  Přenáší data v podobě rámců mezi dvěma prvky propojenými přenosovým mediem. Adresace je založena na fyzických tzv. MAC adresách. Řídí přístup k mediu a zabývá se detekcí chyb.

- **Fyzická vrstva**

  Zajišťuje doručení bitů mezi příjemcem a odesílatelem. Rámce obsahující posloupnost bitů kóduje do elektromagnetického signálu. Obsahuje standardy pro specifikaci mechanických a elektrických vlastností síťových

zařízení.

Referenční model nespecifikuje žádné konkrétní protokoly, pouze vymezuje jednotlivé vrstvy a jejich úkoly. Jelikož je pro touto prací navrhovanou hru Bandwidth challenge nejdůležitější transportní vrstva, bude jejímu detailnějšímu popisu věnována následující sekce.

2.1 Transportní vrstva

Při průchodu dat například rozlehlou sítí (WAN[2]) je kvalita přenosu závislá především na službách, které nabízí jednotlivé podsítě prostřednictvím síťové vrstvy. Hlavním úkolem transportní vrstvy (L4) je poskytnout nadřazeným vrstvám kvalitnější přenosové služby, než jaké nabízí nižší vrstvy. K zajištění spolehlivého přenosu nad nespolehlivou službou využívá transportní vrstva řízení toku (*flow control*) a řízení chyb (*error control*). Pomocí řízení toku je regulován počet odesílaných paketů tak, aby nedošlo k zahlcení příjemce a zahazování těchto dat. Při průchodu sítí se mohou některé pakety poškodit nebo případně ztratit. Pomocí řízení chyb dochází ke kontrole integrity dat, a pokud jsou vyžadovány spolehlivé služby, k zajištění přeposílání ztracených dat.

Pro řízení chyb transportní vrstva využívá mechanismus potvrzování, který odhaluje ztrátu dat s využitím identifikace paketů sekvenčním číslem. Potvrzení (*acknowledgement*), ať už pozitivní, odesílané po doručení paketu, nebo negativní, oznamující nedoručení požadovaných dat v určitém časovém intervalu, vždy obsahuje číslo očekávaných dat. Opětovné odeslání pomocí mechanismů ARQ [1](*Automatic Repeat Request*) také přispívá k zajištění spolehlivosti přenosu.

Transportní vrstva přijímá data od aplikací, opatřuje je hlavičkou a předává síťové vrstvě (L3) k odeslání. Vytváří iluzi přímého transportu dat mezi komunikujícími procesy tzv. *proces-to-process delivery.* Mezilehlé uzly v síti totiž využívají k přenosu pouze nejnižší tři vrstvy OSI modelu – fyzickou, linkovou a síťovou. Hlavičky, připojené k datům v L4, tak zpracovává jen odpovídající vrstva na straně příjemce.

Každý paket v sobě nese informaci o zdrojové a cílové aplikaci – číslo portu. Je to 16 bitové číslo v rozsahu od nuly do 65535. Proces na zdrojovém nebo cílovém zařízení pak v počítačové síti identifikuje kombinace IP_adresa:port. Přidělování portů provádí světová organizace IANA (*Internet Assignment Numbers Authority*)[3].

Tato vrstva má také na starosti řízení spojení (*Connection control*). Spo-

2 Wide Area Network je počítačová síť rozkládající se na velkém geografickém území.

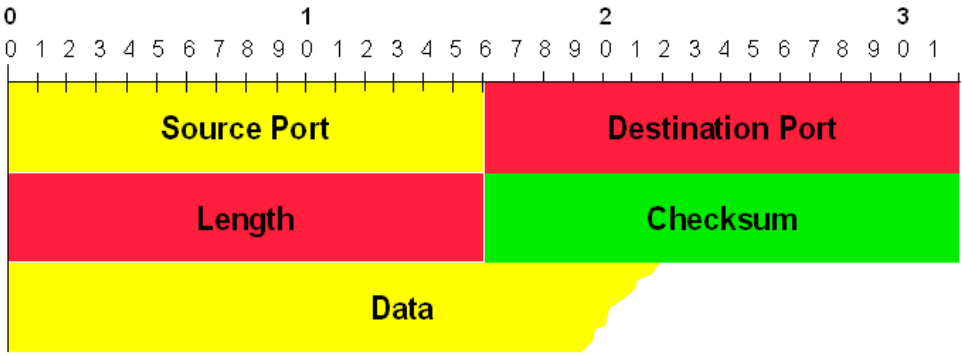
3 Organizace zabývající se přidělováním IP adres a AS čísel a portů aplikacím, dále řídí některé kořenové DNS servery.

jované služby jsou zajištěny pomocí spojení udržovaného po celou dobu přenosu, číslování paketů a potvrzování jejich doručení. K tomu je potřeba větší režie, a přenos je tedy pomalejší než u nespojovaných služeb. Ty ovšem nezabezpečují doručení všech dat nebo jejich správné pořadí.

Dalšími službami, které transportní vrstva poskytuje, jsou kvalita služby a řízení zahlcení sítě, jenž je detailně vysvětleno ve třetí kapitole o TCP protokolu.

2.1.1 Protokol UDP (*User Datagram Protocol*)

UDP přidává pouze minimální funkcionalitu oproti vrstvě síťové a není schopen poskytnout všechny služby, které se očekávají od transportní vrstvy. Tento jednoduchý protokol zajišťuje nespojované a nespolehlivé služby. Od nadřazené vrstvy získává data po blocích. Ta obaluje velmi malou hlavičkou, která kromě zdrojového a cílového portu aplikace, délky celého UDP paketu a kontrolního součtu neobsahuje jiné informace.



Obrázek 1: Hlavička UDP paketu.
http://www.tcp-ip-info.de/tcp-ip-schulung/img117.gif

Protože není při přenosu ustaveno žádné spojení a datagramy nejsou číslovány ani potvrzovány, vyžaduje tento protokol minimální režii. Zpracování UDP paketu tedy způsobuje pouze velmi malé zpoždění a proto je tento protokol vhodný např. pro *real-time*[4] přenosy. Dále ho využívají procesy, které jsou schopny zajistit spolehlivost pomocí vlastních mechanismů kontroly chyb a řízení toku.

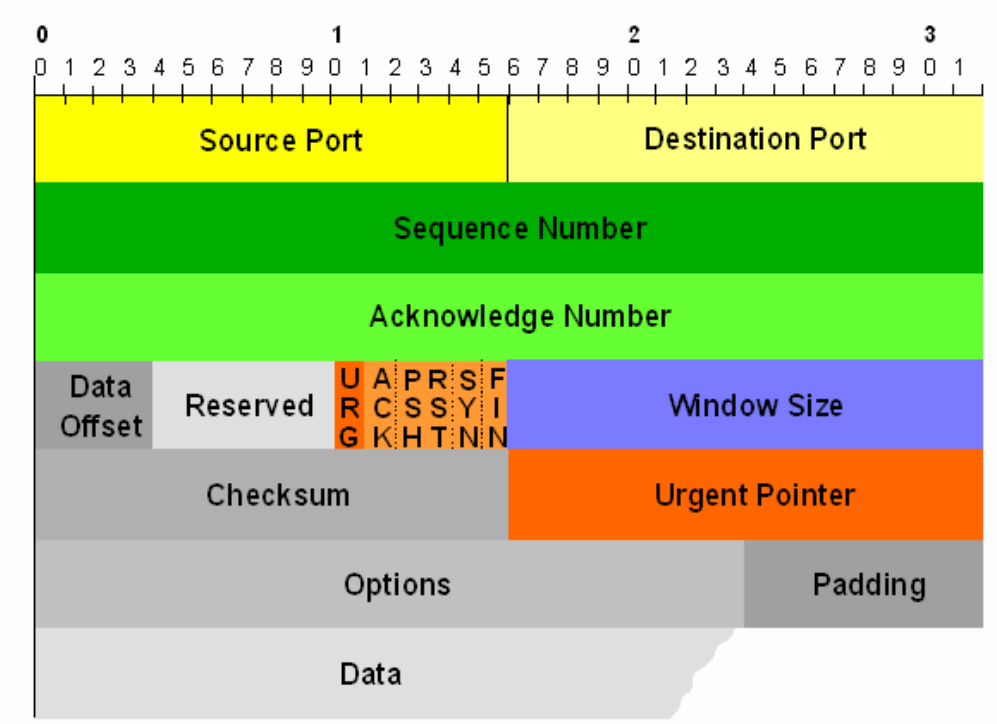
4 Přenos v reálném čase.

3 TCP (*Transmission Control Protocol*)

Jak již bylo zmíněno, hra Bandwidth challenge, která je předmětem této práce, má sloužit především k vývoji a testování protokolu TCP. Proto jsou v této kapitole popsány mechanismy, jakými tento protokol zajišťuje své služby.

Na rozdíl od UDP poskytuje protokol TCP maximální funkcionalitu, jakou očekáváme od L4. Snaží se zajistit spolehlivý přenos dat a chovat se korektně vůči příjemci a síti s využitím mechanismů řízení toku a zahlcení. K řízení toku se využívá informace uvedená v hlavičce každého segmentu. Jedná se o velikost okna příjemce (*Recipient window*, rwnd), což je počet paketů, které ještě může přijmout bez toho, aby musel nějaké zahodit. Řízení zahlcení využívá tzv. okno zahlcení (*Congestion window*). To reprezentuje počet segmentů, které mohou být odeslány do sítě, aniž by došlo k zahlcení. Finální objem opravdu odeslaných dat se pak rovná minimu z těchto dvou oken.

Před samotným přenosem musí být ustaveno spojení. Při plně duplexním (*full-duplex*) spojení jej jedna strana iniciuje, ale podmínkou ustavení je jeho potvrzení druhou stranou. Mechanismus označovaný jako třícestný *handshake* (*threeway handshake*) slouží právě k tomuto účelu. Po přenesení všech dat musí být zrušení spojení opět odsouhlaseno oběma komunikujícími stranami.



Obrázek 2: Hlavička TCP paketu.
http://www.tcp-ip-info.de/tcp-ip-schulung/img106.gif

Hlavička přidaná protokolem je mnohem složitější než u UDP (2.1.1). Obsahuje navíc sekvenční číslo segmentu, číslo potvrzovaného segmentu (*Acknowledgement number*), rezervovaná pole, řídící data, velikost okna, urgentní data a volby. Velikost okna používá především mechanismus pro řízení toku, vysvětlený v kapitole 3.2.

Nadřazené vrstvě nabízí přenos dat po bytech, čímž vzniká iluze proudu bytů. Reálně jsou byty ukládány v paměti (*bufferu*) a po nashromáždění určitého množství jsou odeslány ve formě segmentu. tomto segmentu. Maximální objem uživatelských dat v jednom segmentu (*Maximum segment size*, MSS) definuje implementace TCP nebo operační systém.

3.1 Řízení chyb

Protokol TCP zajišťuje spolehlivé služby, proto je nutné kontrolování kvality a pořadí, v jakém příjemce segmenty obdržel, zda byly vůbec doručeny a jestli nebyla přijata duplicitní data. K tomu slouží několik mechanismů. Prvním z nich je kontrolní součet obsažený v hlavičce každého segmentu. Pokud se jeho hodnota neshoduje se součtem vypočítaným na straně příjemce, segment se zahazuje. Pro upozornění na ztrátu dat slouží další z mechanismů - metoda pozitivního potvrzování. Jelikož není potvrzovaný každý doručený segment, je využito tzv. kumulativní potvrzování. Odesílatel má nastaven časový úsek (*timeout*), během kterého očekává doručení potvrzovacího paketu. Jeho délka se většinou rovná dvojnásobné době cesty segmentu k příjemci a zpět, tzv. *Round Trip Time* (RTT). Kvůli vyhlazování abnormalit se tato hodnota ještě upravuje.

Přeposílání ztracených dat reprezentuje poslední mechanismus. Je založen na metodě *Go-Back-N Automatic Repeat Request* s tím rozdílem, že na straně příjemce funguje *buffer* pro segmenty, které přišly v nesprávném pořadí (*out-of-order*).

3.2 Řízení toku (*Flow control*)

Metoda, která se snaží zabránit zahlcení příjemce. Dojde k němu, pokud není k dispozici dostatek paměti pro přijetí segmentů, které jsou již na cestě. Ty jsou zahazovány a musí být později odeslány znovu. Proto příjemce vkládá do potvrzovacích paketů informaci o velikosti jeho volné paměti. Odesílatel pak reaguje na snižující se rwnd zpomalením odesílaní dat do sítě.

3.3 Řízení zahlcení (*Congestion control*)

Řízení zahlcení slouží k přizpůsobení rychlosti odesílání dat tak, aby nedošlo k překročení kapacity sítě. Směrodatná se stává pouze nejmenší volná kapacita na trase. K zahlcení dochází, protože síťové prvky, které data zpracovávají při průchodu sítí, mají vstupní a výstupní fronty. Pokud se tyto fronty naplní, nadbytečná data se zahazují. K naplnění dojde v případě, že zpracování je pomalejší než rychlost, jakou data přicházejí, nebo naopak rychlejší než odesílání již zpracovaných.

Rychlost vysílání je regulována velikostí okna zahlcení na straně odesílatele. Reprezentuje množství dat, které smí být odesláno do sítě, aniž by došlo k zahlcení. Jeho velikost se upravuje podle přímých informací ze sítě (například ATM sítě), anebo pokud nejsou dostupné (klasické IP sítě), velikost se odhaduje. AIMD (*Additive Increase Multiplicative Decrease*) představuje základní přístup pro odhad v klasických verzích TCP.

Existují dva různé přístupy k řešení zahlcení. Jedním je proaktivní přístup, který se snaží zahlcení předcházet. A druhý, tzv. reaktivní řeší situaci, kdy k zahlcení opravdu došlo. Aby mohla komunikace dále probíhat, je nutné snížit rychlost vysílání.

3.4 Varianty TCP

Postupem času byly vyvinuty různé varianty TCP. Většinou se liší pouze v algoritmech pro řízení zahlcení, respektive odhadování cwnd. Jejich společným rysem je snaha o co nejrychlejší nárůst v rámci počáteční fáze a udržení co nejvyšší rychlosti po co nejdelší dobu ve fázi druhé. Způsob, jakým těchto vlastností dosahují, je rozděluje do několika skupin. Především záleží na tom, zda jsou směrovače v počítačové síti schopny zasílat odesílateli zprávu o stavu sítě. Pokud ano, mohou být využity tzv. router-based algoritmy pro řízení zahlcení, které jsou například součástí verzí VCP a XCP. Bohužel není možné implementovat tuto funkci do všech routerů světových sítí. Proto se využívají algoritmy ze skupiny *end-*

to-end. Odhad velikosti okna zde záleží na sledované proměnné. Může se jednat o:

- **Loss-based congestion control**

  Zahlcení je rozpoznáno na základě ztráty paketu. Okno zahlcení se zvětšuje s doručením každého potvrzení. Pokud však dojde ke ztrátě, okno zahlcení se zmenší. K rozšíření okna bohužel dochází i při blížícím se zahlcení, protože tyto varianty nejsou schopny zajistit tzv. *RTT-fairness* (férové rozdělení kapacity mezi proudy s různým RTT). Dochází tedy k oscilaci mezi plným využitím kapacity a stavem pod ním, což zvyšuje ztrátovost paketů. Příkladem tohoto druhu jsou klasické varianty Tahoe a Reno, popsané v kapitole 3.5.1, a novější varianty HSTCP, STCP, BIC, CUBIC a další.

- **Delay-based congestion control**

  TCP varianty této skupiny reagují na zpoždění ve frontách routerů. Blížící se zahlcení je totiž signalizováno prodlužujícím se RTT, protože fronty na směrovačích jsou již skoro naplněny a doba čekání na zpracování paketu se tím pádem prodlužuje. Problémem je však provoz na zpětné cestě, který snižuje propustnost na cestě dopředné. Jedná se o tzv. *Reverse traffic problem*. „*Delay-based* algoritmy požadují určitý počet paketů ve frontách routerů, aby mohly zajistit průměrnou propustnost okolo maximální hodnoty. Proto by velikost bufferů na routerech měla být větší než tento určený počet paketů. V opačném případě může docházet k výrazným ztrátám paketů v síti.“[1]

  Patří sem verze Vegas a na něm založený FAST blíže popsané v sekci 3.5.2.

- **Loss-delay-based congestion control**

  Členové této skupiny kombinují principy z obou předchozích tak, že jeden z principů je někdy upřednostňován jako primární a druhý jako tzv. *Fallback*. Zůstal požadavek na velikost bufferů jako ve skupině *delay-based* a také přetrvává problém některých variant protokolu s propustností ovlivněnou zpětným provozem. Do této kategorie se řadí varianty CTCP, Illinois, EEFAST a HCC (sekce 3.5.3).

3.4.1 Tradiční TCP - Tahoe a Reno

Kolem roku 1986 začala síť Internet trpět problémy se zahlcením. První implementací protokolu TCP, která je řešila, byla Tahoe, prezentovaná roku 1988. Jejím autorem byl Van Jacobson. Později byla nahrazena verzí Reno, která byla schopna zkrátit čas obnovy po zahlcení.

Tyto dvě verze používají klasický princip AIMD, tedy mechanismus určený k předcházení zahlcení v síti. Jeho součástí jsou tři algoritmy:

- ***Slow start***

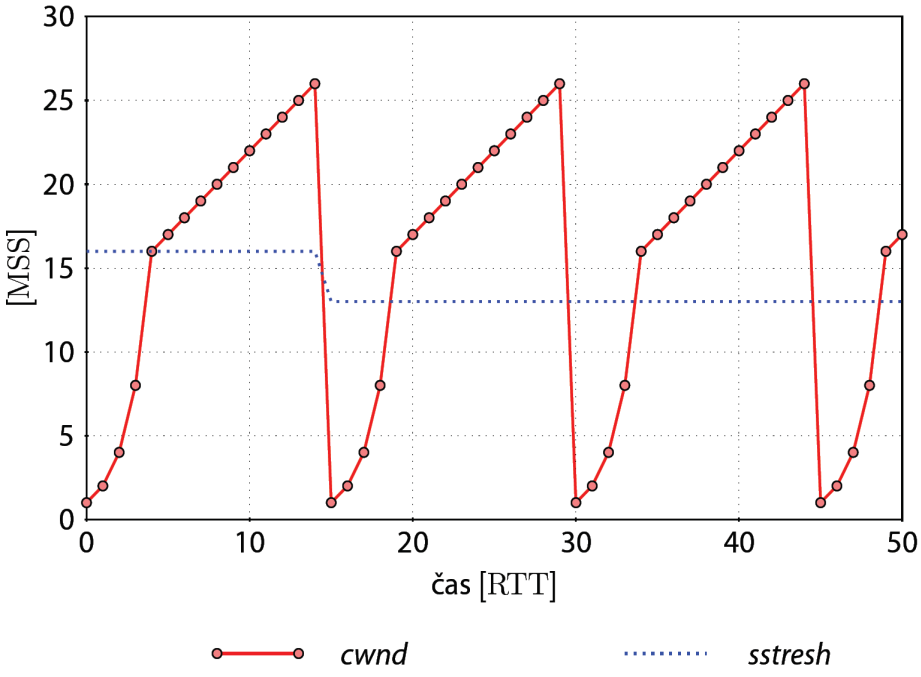
Na počátku komunikace nejsou známy žádné informace o síti. Odeslání obsahu příliš velkého okna by síť zahltilo. Proto se jeho velikost postupně zvyšuje po každém doručeném potvrzení, což vede k exponenciálnímu růstu provozu. Při dosažení velikosti okna příjemce (rwnd) nebo při ztrátě paketu, je polovina hodnoty (u klasických TCP variant) cwnd uložena do proměnné sstresh (*Slow start Treshold*). U varianty Tahoe se při dalším startu tato hodnota bere jako hranice pro ukončení slow start fáze. Oproti tomu varianta Reno fázi Slow-start po výpadku úplně vypustí, což se označuje jako *fast recovery* (rychlá obnova). Okno cwnd má tedy rovnou hodnotu stressh.

- ***Additive increase***

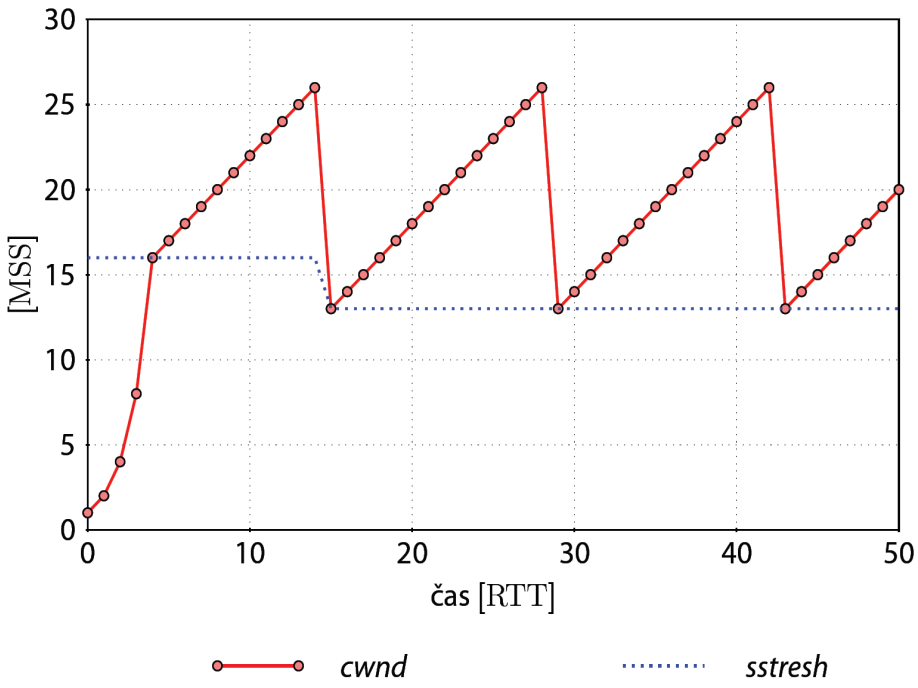
  Navazuje na *slow start* fázi a snižuje rychlý počáteční růst objemu odeslaných dat. Snaží se udržet co nejvyšší rychlost po co nejdelší možný čas.

- ***Multiplicative decrease***

  Pokud dojde ke ztrátě paketu, která indikuje zahlcení, rychlost odesílání je výrazně snížena. To umožňuje zajistit férovost mezi více TCP proudy.



Obrázek3: TCP Tahoe
https://is.muni.cz/auth/el/1433/jaro2011/PB156/um/lecture5.pdf



Obrázek 3: TCP Reno
https://is.muni.cz/auth/el/1433/jaro2011/PB156/um/lecture5.pdf

Čekání na vypršení časového limitu (*timeout*) u odesílatele je v sítích velmi drahé. Pokud příjemce potvrzuje každý doručený segment, při ztrátě následujících dat se odesílá znovu stejné potvrzení. Varianta Reno nabízí funkci *fast retransmission* (rychlé přeposlání) kontrolující duplicitu potvrzení. Pokud narazí na tři shodná, je požadovaný segment okamžitě odeslán znovu.

3.4.2 Vegas a FAST

Verze Vegas[2] byla představena roku 1994 a patří do skupiny protokolů přistupujících k řízení zahlcení proaktivně. Sledováním změn v RTT přizpůsobuje rychlost odesílání tak, aby ve frontách routerů zůstával malý nenulový počet paketů. TCP Vegas tak dosahuje lepší propustnosti a menší ztrátovosti paketů než verze Reno.

Implementace *Fast AQM Scalable (FAST) TCP[3]* je odvozena od verze Vegas, ale umožňuje dosáhnout vyšší síťové propustnosti a využití plné kapacity sítě. Jelikož její způsob výpočtu okna zahlcení umožňuje mnohem přesnější reakce na stav sítě, respektive stav front na síťových prvcích, potřebuje mnohem kratší čas pro obnovu po výpadku. Další vlastností je její větší stabilita. Rozsah okna se totiž nemění lineárně jako v případě Vegas.

Pro výpočet okna zahlcení využívá minimální pozorované RTT (T_{min}) a průměrné RTT(T').

Dále je potřeba Tq (odhad zpoždění front u RTT) a f'(B), která nabývá hodnot 8, 20 nebo 200 pro šířky pásma menší než10Mb/s, 10Mb/s – 100Mb/s a větší než

100Mb/s. Tyto hodnoty je možné měnit pomocí *sysctl()*[5].

Gama je konstanta s hodnotou mezi nulou a jedničkou. Pokud dojde k zahlcení, velikost cwnd se zmenší na polovinu.

3.4.3 HCC (*Hybrid congestion control*)

Řadí se mezi verze protokolu TCP implementující *loss-delay-base congestion control.* Primárním principem pro řízení zahlcení je sledování zpoždění. Protože sekundárním indikátorem je ztrátovost, není potřeba větší velikost bufferů jako u FAST. Problémy při provozu na zpětné cestě jsou eliminovány měřením pouze jednocestného zpoždění.

HCC[4] při *delay-based* řízení zahlcení průběžně počítá velikost tzv. předpokládaného okna. To poté porovnává s aktuálním oknem a na základě velikosti rozdílu těchto dvou hodnot (Δw) dále rozhoduje o zvětšení nebo zmenšení okna. Aby nedocházelo k příliš velkým výkyvům ve velikostech změn, které způsobují snížení výkonu protokolu, před provedením aktualizace je Δw porovnána s proměnnou I_{max}.

$$I_{max} = \alpha_i \gamma$$

α_i reprezentuje počet paketů ve frontách routerů při rovnovážném stavu.

Pokud je |Δw| větší než I_{max}, je změna zmenšena na hodnotu proměnné I_{max}.

Okno zahlcení se mění po jednom RTT zvětšením pouze o polovinu Δw.

Při ztrátě paketu přichází na řadu sekundární princip řízení zahlcení. Dojde k uložení velikosti aktuálního okna do proměnné *loss-refference* (zároveň w_i^{max}) a okno se zmenší na polovinu. Poté následuje tzv. fáze zotavení, při níž je velikost změny rovna polovině rozdílu mezi *loss-reference* a aktuální velikostí okna(w_i^{min}).

$$w_i^{tar} = (w_i^{max} - w_i^{min}) / 2$$

Velikost cwnd se mění stejným způsobem jako v primárním algoritmu. Jestliže nedojde ke ztrátě paketu a velikost okna zahlcení je větší než hodnota *loss-refference*, řízení zahlcení opět přebírá *delay-based* algoritmus.

Následující kapitola popisuje konkrétní cíl práce, tedy hru Bandwidth challenge. Jsou zde vysvětleny použité technologie a prezentovány vytvořené softwarové prostředky.

5 http://linux.die.net/man/8/sysctl

4 Hra Bandwidth Challenge

Hra Bandwidth challenge, jejíž příprava je předmětem této práce, nabízí účastníkům možnost otestovat existující verze protokolu TCP, vylepšit je nebo navrhnout svůj vlastní protokol a ověřit si jeho účinnost. Cílem je samozřejmě vytvoření co nejefektivnějšího řešení řízení zahlcení, které je velkou slabinou TCP na linkách s velkým zpožděním.

Pro vývoj bude mít každý tým k dispozici dva virtuální stroje. Samotné testování pak bude probíhat na fyzických serverech, které jsou připojeny k vyhrazené lince. Ta bude využívána pouze skupinou, která si ji zarezervuje. Virtualizované prostředí společně s rezervačním systémem zaručují všem týmům stejné startovací podmínky a jistotu, že jejich testy nebudou ovlivněny pokusy ostatních protihráčů.

Součástí hry bandwidth challenge jsou informační materiály ve formě letáku a webových stránek s rezervační aplikací.

4.1 SCINET Network Challenge

SC Conference je mezinárodní konference zabývající se výkonnými počítačovými sítěmi, uchováním dat a analýzou. V roce 2000 byla na této konferenci představena hra Network challenge[6]. SCinet team vytvořil velmi výkonnou počítačovou síť s cílem dát možnost aplikacím, které mají velmi vysoké požadavky na výkon sítě, předvést svoji funkčnost. Oslovili členy komunity, aby podali návrhy na demonstraci svých aplikací. Bylo vybráno 11 z nich, kterým byla síť poskytnuta. Odborná porota po předvedení výsledků testování vybrala ty nejlepší a jejich tvůrci získali finanční odměnu. Tato hra je každoročně součástí konference.

Bandwidth challenge má s touto hrou společný základ. Zájemcům je poskytnuta síť, na které se snaží dosáhnout co nejlepších výsledků v porovnání s ostatními účastníky. Zadání je však mnohem specifičtější. Výzkum a testování by se mělo týkat pouze protokolu TCP. Od Network challenge se Bandwidth challenge liší především v nákladech na síťovou infrastrukturu a samozřejmě také výkonem.

6 http://www.sc2000.org/

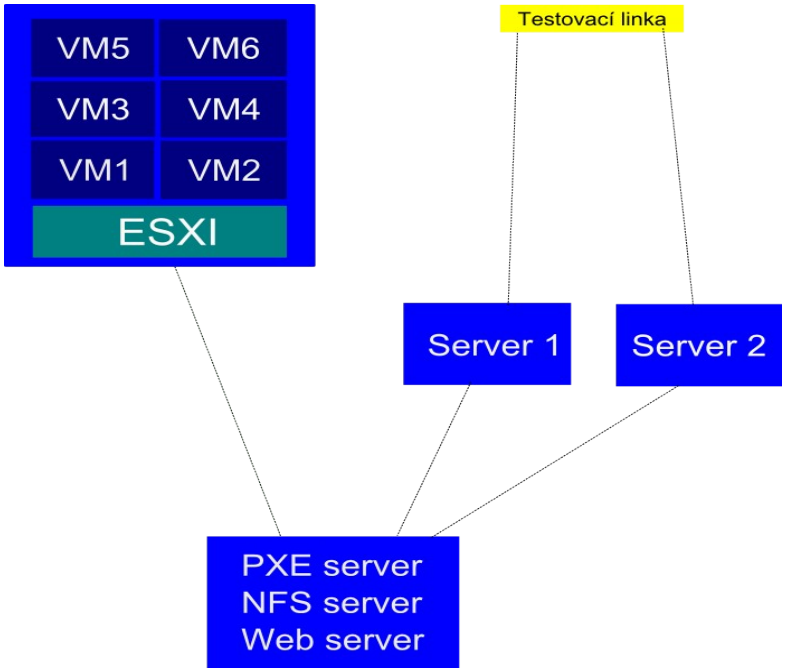
4.2 Síťová infrastruktura

Celé technické zázemí této hry je velmi jednoduché a je znázorněno na obrázku číslo 3. Pro účely testování je vyhrazena 10Gb linka mezi Prahou a Brnem. Na této trase jsou umístěny rušičky. Délka linky totiž není dostatečná, aby se problémy na rozlehlých sítích plně projevily. Pro přístup k této lince slouží pouze dva servery. Každý je připojen k jednomu konci, ale oba jsou fyzicky umístěny v Brně.

Další nutnou součást představuje server s technologií VMware, na kterém fungují virtuální stroje pro všechny zúčastněné týmy. Podrobněji je tato technologie vysvětlena v kapitole 4.2.1. Virtuální stroje slouží týmům pro výzkum. Pokud potřebují mít přístup k dedikované lince z důvodu testování, jsou tyto stroje nahrány na fyzické servery. Použitá metoda se nazývá migrace virtuálního serveru na fyzický (*Virtual to physical machine migration[5]*). Jelikož jsou fyzické servery bez jakéhokoliv operačního systému, je pro jejich automatický boot z počítačové sítě je vyžívána technologie PXE boot.

Tento systém poskytuje možnost pracovat na vývoji, i pokud zrovna linka není k dispozici. Především nemůže dojít k tomu, že by po týmu, který pracoval s fyzickými servery, zbyla nějaká data nebo nastavení. Pro přenos virtuálního stroje na fyzický je tedy nutný ještě jeden server s potřebnými aplikacemi, který prakticky řídí celou síťovou infrastrukturu této hry. Pomocí skriptu ovládá přenos strojů a zároveň obsahuje web server, který zprostředkovává přístup k rezervační aplikaci umístěné na webových stránkách.

Nastavení a obsah tohoto serveru je popsáno v dalších kapitolách. Zároveň příloha obsahuje DVD s jeho připraveným obrazem (*image*). Po drobných úpravách popsaných v kapitole 4.5 a zajištění potřebné síťové infrastruktury, je možné celou tuto hru uskutečnit.

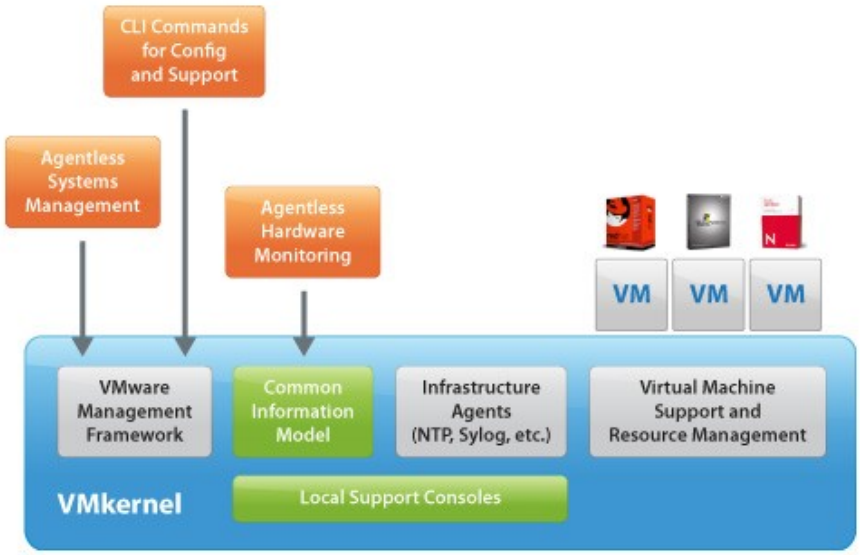


Obrázek 3: Nákres topologie hry.

4.2.1 VMware

Tato technologie umožňuje vytvořit na jednom fyzickém počítači několik virtuálních strojů (obrázek 3). Virtuální stroj (*virtual machine*, VM) představuje izolovaný softwarový (programový) kontejner, který má svůj vlastní operační systém, CPU (*Central Processor Unit*), operační paměť, pevný disk atd., stejně jako klasický počítač. Všechny jeho součásti jsou však pouze programové. To vede ke značné úspoře nákladů na potřebný hardware a zároveň i úspoře místa. Ani instalovaný operační systém není schopen poznat rozdíl mezi virtuálním a fyzickým strojem. Jedná se o velmi rozšířenou metodu hojně využívanou v komerční sféře.

Pro účely hry plně vyhovuje neplacená architektura ESXi. Tento software se instaluje přímo na server bez jakéhokoliv operačního systému. Jeho součástí je především virtualizační jádro (vmkernel), zobrazené na nákresu struktury ESXi na obrázku 4



Obrázek4: Schéma architektury ESXi.
http://www.vmware.com/files/images/screens_esxi/dgrm_esxi-migration-esxi-architecture.jpg

Při vytváření nových virtuálních strojů má správce možnost nastavit jejich hardwarovou konfiguraci. Musí tedy zvolit velikost CPU, pevného disku, typ síťové karty a tak podobně. Při přenosu virtuálního stroje na fyzický mohou vzniknout problémy způsobené rozdílným hardwarem. V praxi to znamená, že operační systém nainstalovaný na virtuálním stroji bude očekávat jiný hardware, než jaký bude na fyzickém stroji k dispozici. To může způsobit závady, které se velmi těžko hledají. Proto je na místě nastavit konfiguraci VM tak, aby odpovídala hardwaru na fyzických serverech.

Při přípravě infrastruktury pro hru Bandwidth challenge je nutné zohlednit, že každý tým potřebuje dva virtuální stroje odpovídající fyzickým serverům připojeným k vyhrazené lince. Tyto VM budou sloužit k vývoji a testování. V časovém úseku, který si skupiny zarezervují, budou kopie jejich virtuálních stojů fungovat na fyzických serverech. Přenos je automaticky prováděn skriptem popsaném v sekci 4.2.6

4.2.2 Fyzické servery

Jak již bylo řečeno, pro přístup k vyhrazené lince slouží dva servery. Jelikož je potřeba předejít zkreslování dat z testování a využití maximálního výkonu těchto strojů, není na nich předinstalován žádný software. Operační systém je zaváděn až v okamžiku začátku rezervace linky. Jednotlivé obrazy virtuálních strojů, které servery potřebují pro svůj start, se připravují na serveru v počítačové síti vytvořené pro tuto hru. K tomu, aby je mohly servery použít, slouží tzv. PXE boot, popsaný v sekci 4.2.3. Fyzické stroje tedy musí být připojeny k této síti a zároveň k vyhrazené

lince, čímž vzniká požadavek na dvě síťové karty.

Aby server požádal o obraz, ze kterého nabootuje, musí dojít k jeho restartu případně vypnutí a opětovnému zapnutí. Z důvodu automatizace hry je nutné ovládat tyto stroje vzdáleně. Restart lze spustit pomocí ssh[7] příkazu:

```
ssh root@ipadresa 'reboot'.
```

Může se však stát, že stroj nebude dostupný nebo nebude na příkaz reagovat (v případě, že dojde k závažným chybám při testování a stroj „zamrzne“).

Spolehlivějším způsobem je využití tzv. *Power Distribution Unit* (PDU). Jedná se o speciální elektrické zásuvky s IP adresou využívané především v serverových místnostech. Umožňují individuální nastavení zásuvek. Vzdáleným přístupem je tedy možné serverům vypnout přívod elektrického proudu. Server se tak spolehlivě odstaví a při opětovném zapnutí elektrické energie vyhledá v počítačové síti PXE server.

4.2.3 PXE server

Technologie PXE[3] (Preboot eXecution Environment) umožňuje start počítače s využitím počítačové sítě nezávisle na instalovaném operačním systému (na počítači nemusí být žádný) a mediích pro ukládání dat (pevný disk, DVD atd.). Dnešní síťové karty již standardně umožňují zavedení operačního systému z počítačové sítě. Často se tato technologie používá pro automatické instalace systému do počítačů, například bez pevného disku.

PXE server obsahuje obraz disku s operačním systémem, který propaguje do počítačové sítě pomocí TFTP. Jelikož je při startu klientského počítače využita počítačová síť, je nutnou podmínkou pro fungování PXE serveru DHCP.

- **DHCP (Dynamic Host Configuration Protokol)**

  Zajišťuje automatické přidělování parametrů nutných pro síťovou komunikaci, tedy IP adresu, masku sítě, adresu brány a adresu DNS (*Domain Name System*) serveru. Klienti v počítačové sítí požádají o potřebné informace DHCP server a ten jim je obratem odešle. Zároveň si uchovává záznamy o přidělených adresách, aby nedošlo k duplicitě. Adresy mohou být přidělovány staticky (podle fyzické adresy je přidělena konkrétní IP adresa) nebo je k dispozici určitý rozsah adres, ze kterých DHCP vybírá. Pokud počítačová síť, v níž je PXE server umístěn, nemá vlastní DHCP server, je nutné ho nainstalovat.

V případě bootování pomocí PXE slouží DHCP server k propagaci informace o poloze PXE serveru respektive TFTP serveru. Je tedy nutné nastavit DHCP serveru adresu TFTP serveru a název souboru s binárním kódem PXE serveru (pxelinux.0).

Při startu počítač vyšle do sítě zprávu sloužící ke zjištění přítomnosti DHCP serveru

7 Program umožňuje přihlášení na vzdálený stroj a provádění příkazů pomocí šifrované komunikace. Více na http://unixhelp.ed.ac.uk/CGI/man-cgi?ssh+1.

v síti (DHCPDISCOVER) s příznakem PXE. Pokud má DHCP server povoleno zasílání informací počítačům žádajícím o bootování z počítačové sítě, odešle zpět potřebné parametry. Dále už klientský počítač komunikuje přímo s PXE serverem. Ten mu přidělí odpovídající obraz disku, který je pak přenášen sítí s využitím protokolu TFTP.

- **TFTP (*Trivial File Transfer Protocol*)**

  Je velmi jednoduchý protokol určený pro přenos souborů využívající služeb protokolu UDP. Jelikož UDP poskytuje pouze nespolehlivé služby (jak bylo vysvětleno v sekci 2.1.1), musí obsahovat vlastní mechanismy pro řízení spojení. TFTP vyžaduje potvrzení pro každý blok dat odeslaný do sítě. Na síti je tedy vždy pouze jeden paket. Z toho důvodu je přenos velmi pomalý.

  Pro přípravu bootování ze sítě je nutné nainstalovat TFTP server a nastavit cestu k adresáři, ve kterém budou umístěna poskytovaná data.

Konfigurační soubory PXE serveru dovolují rozdílné nastavení pro jednotlivé počítače. Na základě fyzické adresy počítače poskytne PXE server pouze obraz určený tomuto počítači (soubor musí mít název shodný s řetězcem reprezentujícím fyzickou adresu). Pokud žádný soubor nevyhovuje, je použito nastavení v konfiguračním souboru default. Soubor pxelinux.0, umístěný v adresáři TFTP serveru společně s obrazem (obrazy) operačního systému, přesměruje klientský počítač k adresáři s konfiguračními soubory (pxelinux.cfg opět v adresáři tftp serveru). Priorita konfiguračních souborů je znázorněna v následujícím výpisu.

```
PXE entry point found (we hope) at 9AE5:00D6
   My IP address seems to be C0A80146 192.168.1.70
   FTFTP prefix:
   Trying to load: pxelinux.cfg/01-00-14-22-a1-53-85
   Trying to load: pxelinux.cfg/C0A80146
   Trying to load: pxelinux.cfg/C0A8014
   Trying to load: pxelinux.cfg/C0A801
   Trying to load: pxelinux.cfg/C0A80
   Trying to load: pxelinux.cfg/C0A8
   Trying to load: pxelinux.cfg/C0A
   Trying to load: pxelinux.cfg/C0
   Trying to load: pxelinux.cfg/C
   Trying to load: pxelinux.cfg/default
```

Vzhledem k tomu, že hra Bandwidth challenge využívá PXE boot pro dva servery, je vhodné nastavit PXE server právě podle fyzických adres serverů.

4.2.4 NFS server

Po „bootování“ počítače prostřednictvím technologie PXE počítač obsahuje pouze operační systém. Pokud je ovšem potřeba přenést i nějaká konkrétní data, je vhodné využít tzv. NFS (*Network File System*) server umožňující vzdálený přístup k adresářům. Cesty ke sdíleným adresářům a IP adresy počítačů, které k nim mohou

přistupovat, jsou součástí konfiguračního souboru exports v adresáři /etc.

Pro fungování hry Bandwidth challenge je NFS server velmi podstatný. Jak již bylo řečeno, soutěžní týmy budou mít k dispozici 2 virtuální stroje, jejichž kopie budou v čase, který si zaregistrují, přeneseny na fyzické servery s přístupem k vyhrazené lince. Je však nutné přenést nejen kopii jejich operačního systému, ale i poskytnout data. Právě k tomu je využíván NFS server.

4.2.5 WEB server

Součástí této bakalářské práce je zajištění propagace hry pomocí informačních materiálů. Hlavním prostředkem ke zviditelnění této hry jsou webové stránky, které zároveň obsahují rezervační aplikaci pro rozdělení přístupů k testovací lince.

Funkcí web serveru je doručení stránek klientům, kteří o ně žádají pomocí protokolu HTTP (*Hypertext Transfer Protoco*l). Součástí těchto stránek odesílané spolu s HTML (*Hypertext Markup Language*) dokumenty mohou být i obrázky, video nebo například skripty. Odpověď klientovy obsahuje i stavové kódy, které udávají stav vyřízení požadavku. Velmi známý je například stavový kód 404, značící, že požadovaný soubor nebyl nalezen.

Obsah, který server odesílá, může získat staticky (soubory jsou odeslány ve stavu, v jakém je web server nalezl), nebo dynamicky, kdy jsou na základě požadavku klienta zpracována potřebná data a jsou odeslána ve formě HTML dokumentů. Pro vytváření dynamického obsahu existuje mnoho různých technologií (PHP, .Net, Perl). V případě propagačních stránek hry Bandwidth challenge byl použit jazyk JSP (Java Server Pages) v kombinaci s Apache Tomcat web serverem.

- **Apache Tomcat**

  Je open source[8] software, který implementuje technologie Java Servlet, Java Server Pages a zároveň poskytuje http server založený na programovacím jazyku JAVA. Byl vyvinut ve společnosti Apache Software Foundation v projektu Jakarta. Oproti komerčním verzím je výrazně jednodušší. Což se projevuje v menší náročnosti na výpočetní výkon a menší robustnosti.

Apache Tomcat byl vybrán na základě poskytovaných technologií a jeho malé náročnosti. Je součástí obrazu serveru připraveného na přiloženém DVD i s webovými stránkami.

4.2.6 Generování testovacích dat

Celá hra je založena na principu přenosu informací datovou linkou. Vzhledem k tomu, že nejkratší možná doba rezervace je 10 minut kvůli spolehlivému zaznamenání rozdílů ve výkonu jednotlivých verzí protokolu TCP, je nutné zajistit dostatečné množství dat pro přenos. Vyhrazená linka má relativně velkou kapacitu, a

8 Software s otevřeným zdrojovým kódem (veřejně dostupným).

proto hraje roli i rychlost jakou budou data generována.

Kapacita operační paměti serverů není dostatečně velká, aby mohla poskytnout celý balík potřebných dat. Odpovídající úložným prostorem disponuje pevný disk, ale rychlost čtení by omezovala provoz na lince. Proto se jako jediná vhodná metoda pro zajištění dostatečného množství dat jeví průběžné softwarové generování v operační paměti. K tomuto účelu byla vytvořena knihovna, která hráčům poskytuje potřebná data a zároveň nabízí funkce k jejich kontrole a měření doby přenosu dat.

Aby bylo porovnání výsledků jednotlivých týmů spravedlivé, je nutné zajistit všem hráčům stejné podmínky. Proto byla navržena knihovna, která přímo definuje funkce pro generování a porovnávání dat a zároveň je schopna vrátit informace nutné k vyhodnocení testování.

Výkon variant protokolu TCP může být podroben dvěma různým zkouškám. První z nich měří čas potřebný pro přenesení bloku dat určité délky. Logicky je lepší ta verze, která spotřebuje menší množství času. Druhým způsobem je generování a přenášení dat po určitou dobu a následné porovnání doručeného množství. Oba tyto postupy potřebují vlastní algoritmus pro kontrolu doručených dat.

Výběr mezi těmito dvěma variantami je prováděn při inicializaci funkce. Zároveň se musí definovat maximální hodnota (čas nebo objem dat) podle typů porovnávání, řetězec využitý jako semínko pro generátor náhodných čísel, jeho délka a velikost bloku dat se kterým budeme manipulovat. Jednotlivé funkce jsou pak napsány tak, že kontrolují nebo generují data po blocích této délky.

Kontrola dat je založena na typu testování. V obou případech však vrací hodnotu 1 pokud se doručená a odeslaná (data vygenerovaná na straně příjemce se stejným semínkem generátoru) data liší (byla cestou pozměněna). Dále jsou ošetřeny chyby při generování dat nebo snaha otestovat blok jiné velikosti než pro jakou byla knihovna iniciována. V případě, že bude měřena doba, za kterou data projdou sítí, je nutné aby bylo přeneseno více bloků než jeden. První totiž spouští počítání času (na straně příjemce) a časovač tak má hodnotu nula.

Knihovna i s podrobnými komentáři je součástí stroje přiloženého na DVD.

Jelikož se tímto typem generování dat spotřebovává výpočetní výkon CPU, je možné využít procesor grafické karty (GPU, *Graphic Processing Unit*). Tuto schopnost nabízí pouze některé grafické karty výrobce NVIDIA s architekturou CUDA (*Computer Unified Device Architecture*).

- **CUDA**

  Technologii CUDA představila společnost NVIDIA v roce 2006. Její konkurencí je ATI stream společnosti AMD. Tato architektura zprostředkovává přístup k výpočetnímu výkonu procesoru grafické karty a umožňuje na ní spouštět programy.

Protože chceme zajistit co nejvýkonnější testovací prostředí na jednoduché síťové infrastruktuře, je vhodné, aby fyzické servery připojené k vyhrazené lince byly schopny generovat data co možná nejrychleji. K tomu je možné použít právě

grafické karty s potřebnou technologií.

4.3 Princip ovládání síťové infrastruktury

Je velmi žádoucí, aby hra navrhovaná v této bakalářské práci fungovala pouze s minimální účastí administrátora. Proto byl vytvořen skript, který ovládá veškeré potřebné operace pro zajištění funkcionality síťové infrastruktury. V této sekci jsou popsány konkrétní příkazy použité v tomto skriptu a jejich význam. Samotný skript s komentáři je součástí přiloženého DVD.

Skript je provázán s rezervační aplikací pomocí souboru s rezervovanými časy. Protože čas ukončení rezervace je ošetřen v aplikaci, soubor obsahuje pouze údaj o začátku budoucích rezervací a označení skupiny, která ji vytvořila. Označení je důležité pro identifikaci jednotlivých virtuálních strojů. Při jejich tvorbě je potřeba toto zohlednit nebo případně skript upravit.

Každý virtuální stroj identifikuje jeho jméno, ale především tzv. VM ID. Pro získání výpisu všech strojů s jejich identifikačním číslem slouží příkaz

```
vim-cmd vmsvc/getallvms
```

provedený na ESXi. Protože skript potřebuje ovládat ESXi vzdáleně, je použito ssh, které umožňuje provést příkazy na vzdáleném stroji. Podmínkou plynulého běhu skriptu jsou však ssh klíče, které nevyžadují heslo. Příkaz pak vypadá následovně:

```
ssh root@IPadresaESXi 'vim-cmd vmsvc/getallvms'.
```

Pro vyfiltrování potřebných vmid skript využívá porovnání jmen strojů. Proto je důležité, aby byla vytvořena z názvů skupin, konkrétně ve tvaru: nazevskupiny1, nazevskupiny2. Pomocí správného identifikačního čísla může být stroj vzdáleně ovládán, např. vypnutí nebo zapnutí.

Všechny virtuální stroje mají v ESXi uložené obrazy disku. Jedná se o soubory s příponou .vmdk (jmenostroje.vmdk a jmenostroje-flat.vmdk). Oba tyto soubory jsou nutné pro vytvoření kopie stroje. Jelikož potřebujeme získat jeho kopii na PXE server, musíme tedy přenést odpovídající obraz disku. Aby mohl být disk zkopírován, musí nejprve dojít k vypnutí virtuálního stroje. S využitím identifikačního čísla a vzdáleného přístupu k ESXi pomocí ssh skript stroj po provedení příkazu

```
ssh root@IPadresaESXi 'vim-cmd vmsvc/power.off VMID'
```

vypne.

K samotnému přenosu souborů byl zvolen program rsync. Protože kopírování strojů bude prováděno při každé rezervaci linky, není nutné znovu přenášet všechna data. Rsync přenáší pouze rozdíl dat zdrojového a cílového adresáře, což výrazně zkracuje dobu vytváření kopie.

Po dokončení přenosu obrazu disku je stroj opět zapnut příkazem

```
ssh root@IPadresaESXi 'vim-cmd vmsvc/power.on VMID'.
```

Tím končí veškerá komunikace s ESXi a následující operace už jsou prováděny pouze na serveru.

Dalším krokem je příprava sdílení dat s využitím NFS. Jak bylo vysvětleno v sekci 4.2.4, fyzický server bude po startu obsahovat pouze operační systém. Potřebná data není potřeba kopírovat přímo na servery. Plně postačí sdílení přes počítačovou síť.

Přenesený obraz disku musí být nejdříve extrahován do adresáře. VMware nabízí k tomuto účelu službu VMware Disk Mount (vmware-mount). S jeho pomocí je obsah disku rozbalen do adresáře /mnt/point příkazem:

```
vmware-mount jmenostroje.vmdk 2 /mnt/point
```

Pro přípravu PXE serveru musí být změněn soubor fstab, který obsahuje všechny připojované diskové oddíly. Požadovaná konfigurace je již připravena. Skript ji proto pouze překopíruje do virtuálního stroje. Dále je potřeba do připraveného systému připojit soubory /dev /proc a /sysfs:

```
mount -o bind /dev /mnt/point
mount -t proc none /mnt/point
mount -t sysfs none /mnt/point.
```

Při přenosu virtuálního stroje na fyzický je úmyslně pozměněno jádro (kernel) operačního systému tak, aby po startu počítač hledal v síti NFS server, který mu poskytne potřebná data. K tomu je nutné nejdříve změnit kořenový adresář příkazem

```
chroot /mnt/point.
```

V tuto chvíli už skript pracuje přímo s virtuálním strojem a využitím příkazu sed, který nalezne v souboru daný řetězec, nahradí ho jiným, upraví konfiguraci kernelu.

```
sed -i 's/BOOT=local/BOOT=nfs/' /etc/initramfs-
tools/initramfs.conf
```

Když je úprava kernelu dokončena, skript příkazem uname -r zjistí jeho aktuální verzi. Poté nechá vytvořit nový obraz systému, ale se stejným označením (kvůli nastavení NFS serveru).

```
/usr/sbin/mkinitramfs -o /boot/initrd.img-$VERZE
```

K dokončení přípravy PXE serveru chybí pouze vytvoření kopie nového obrazu virtuálního stroje v adresáři TFTP serveru. Nejdříve je však ukončen chroot (příkazem exit), čímž se kořenový adresář vrátí zpět.

Tímto je tedy připraveno vše potřebné pro boot fyzických serverů z počítačové sítě. Posledním krokem je restart, respektive vypnutí a opětovné zapnutí, těchto serverů. Způsoby, jakými je toho možné docílit, jsou popsány v kapitole 4.2.2. Skript aktuálně obsahuje příklad příkazu pro vypnutí prováděného skrze ssh. Při jeho použití je však nutné opět počítat s ssh klíči bez hesel.

4.4 Rezervační aplikace

Zaručení stejných podmínek pro všechny zúčastněné týmy je předpokladem spravedlivého posouzení dosažených výsledků. Z toho důvodu je ve hře Bandwidth challenge využito technologie Vmware a přístup k vyhrazené lince je řízen rezervacemi konkrétních časů V jednu chvíli tak může provádět testy pouze jedna skupina. Výsledky testování proto nemohou být nikým ovlivněny. Správa těchto rezervací je zajištěna pomocí rezervační aplikace.

Aplikace byla napsána v jazyce JSP, který slouží k vývoji webových aplikací. Tento programovací jazyk umožňuje vložení kódu v jazyce JAVA, který se stará o dynamický obsah, do statických webových stránek. Pro jeho provedení je potřeba web server ovládající tuto technologii (v této bakalářské práci byl zvolen Apache Tomcat popsaný v sekci 4.2.5) a tzv. Java Development Kit (JDK), což je balík základních nástrojů pro vývoj aplikací v jazyce Java.

- **Vstupy a výstupy**

  Rezervační aplikace pracuje s textovými soubory, ve kterých jsou uloženy jednotlivé rezervace a přihlašovací údaje hráčů. Rezervace jsou uloženy ve dvou různých souborech z důvodu provázanosti aplikace s řídícím skriptem.

  Soubor s informacemi nutnými k přihlášení se dá považovat za vstupní soubor a administrátor ho vytváří ručně. Při registraci týmů účastnících se této hry bude od jejich členů vyžadováno přihlašovací jméno a hash hesla. Hašovací funkce přetváří vstupní posloupnost bytů na posloupnost pevné délky, přičemž z výsledného řetězce není možné zjistit ten původní. Jelikož jsou hesla uchovávána v textovém souboru, je z bezpečnostních důvodů nutné jejich ukládání v pozměněném tvaru, aby při případném odcizení souboru nemohla být použita. Pro vytvoření hashe hesla slouží následující příkaz

  ```
  echo -n heslo | sha256sum.
  ```

  Aby algoritmus pro porovnání přihlašovacích údajů pracoval správně, musí být reprezentace hesla vytvořena právě tímto způsobem. Především parametr n funkce echo je velmi důležitý. Bez něj je za řetězec hesla přidán znak konce řádku a vypočtená hash pak nesouhlasí.

  V souboru je rovněž uvedena informace o skupině, do které konkrétní hráč patří. Tento údaj se využívá při tvorbě rezervací a i při samotném přenosu virtuálních stojů. Každý hráč vždy vytváří rezervaci pro celou svou skupinu a všichni její členové mají přístup ke každé z nich.

  Ostatní soubory obsahující informace o rezervacích jsou vytvářeny samotnou aplikací. Jeden z nich obsahuje pouze údaje o začátku budoucích rezervací a označení skupiny, která ji vytvořila. Skript pro řízení přenosu virtuálních strojů totiž více informací nepotřebuje. Druhý naopak obsahuje kompletní seznam všech vytvořených rezervací.

- **Struktura aplikace**

  Aplikace se skládá ze dvou částí. První je dostupná všem návštěvníkům webových stránek a obsahuje kalendář s vyobrazením všech rezervací Dny, pro které byla vytvořena alespoň jedna rezervace, jsou barevně odlišeny od těch ostatních. Po kliknutí na políčko s příslušným datem se zobrazí výpis všech rezervací pro tento den a časová lišta, na které jsou opět barevně znázorněny. Je možné prohlížet rezervace i zpětně, ale pouze pokud byly vytvořeny v aktuálním roce.

  Druhá část je dostupná pouze registrovaným hráčům. Po přihlášení se uživateli zobrazí stránka s výpisem rezervací pro aktuální den a malé menu s tlačítky. Tato tlačítka nabízí tři funkce: Vytvoření nové rezervace, Odstranění rezervace a Restart počítače.

  Poslední ze jmenovaných je dostupné pouze v průběhu rezervace. Jeho funkcionalita zatím nebyla implementována, protože závisí na principu, jakým se budou restartovat fyzické servery (sekce 4.2.2).

  Odstranění rezervací je možné pouze pokud se jedná o budoucí nebo o právě běžící rezervaci. Zrušení aktuální rezervace je výhodné například v případě, že tým dokončil testování dříve, než uběhl vyměřený čas. Stroje jsou tak poskytnuty dalším skupinám, a dochází tak k jejich efektivnímu využití.

  Nová rezervace může mít minimální dobu trvání 10 minut, protože při kratším čase by se neprojevily rozdíly v testovaných verzích protokolu TCP. Aplikace je samozřejmě ošetřena tak, aby nebylo možné vytvořit rezervaci do minulosti nebo aby se dvě rezervace nepřekrývaly. Stránka pro rezervování opět obsahuje barevnou časovou osu znázorňující již obsazené časové úseky. Potvrzením nastavené rezervace se upraví již zmíněné textové soubory, a změna je tak okamžitě aplikována. Po zobrazení potvrzení rezervace má uživatel možnost vytvořit další nebo se vrátit do menu s výběrem operací. Každá ze stran samozřejmě obsahuje tlačítko pro odhlášení.

Rezervační aplikace je součástí webových stránek vytvořených z důvodu propagace hry Bandwidth challenge. Jelikož je přímo provázaná s řídícím skriptem popsaným v kapitole 4.3, je rovněž připravena na přiloženém DVD i s potřebným web serverem.

4.5 Zprovoznění hry

V této sekci je podrobně popsán postup pro zprovoznění hry Bandwidth challenge. Potřebné softwarové nástroje a obraz disku „řídícího serveru“ jsou připraveny na přiloženém DVD.

Tato hra vyžaduje celkem 4 servery. Dva z nich slouží jako servery pro testování na vyhrazené lince. Musí být tedy připojeny k ní a zároveň k počítačové síti obsahující zbylé servery. Měly by obsahovat grafické karty schopné pracovat s technologií.

CUDA (*Compute Unified Device Architecture*), která je vysvětlena v sekci 4.2.6. Oba stroje musí být schopny startu s využitím počítačové sítě. Logicky je nutné, aby tato funkcionalita byla povolena. Při testování infrastruktury v rámci přípravy této bakalářské práce byl pro ni na dvou využitých strojích nalezen termín „NFS boot“. Další příprava těchto serverů závisí na způsobu jejich restartu. Možnosti jsou popsány v sekci 4.2.2.

Jeden ze zbylých serverů by měl obsahovat technologii VMware kvůli virtualizaci testovacích strojů (sekce 4.2.1). Při instalaci ESXi pro hru Bandwidth challenge je nezbytně nutné povolit ssh a vytvořit ssh klíče, které nebudou požadovat heslo při přihlášení z PXE serveru (sekce 4.2.3). Skript řídící přenos virtuálních strojů na fyzické totiž komunikuje přímo s ESXi pomocí ssh a také rsync[9] a není žádoucí, aby musel administrátor při každém spuštění skriptu zadávat příslušná hesla. Pro kopírování byl zvolen program rsync kvůli úspoře času. Tato funkce bohužel není součástí ESXi, proto je nutné získat její binární reprezentaci[10] kompilovanou právě pro ESXi.

Každý tým potřebuje dva virtuální stroje s operačním systémem, knihovnou vytvořenou pro účely testování a hardwarovou konfigurací nejlépe totožnou s fyzickými servery. Názvy těchto strojů je vhodné odvodit od označení jednotlivých skupin a upravit podle nich část řídícího skriptu (sekce 4.3) určenou k vyhledávání potřebného identifikačního čísla virtuálních strojů. Označení virtuálních serverů používá i rezervační aplikace pro výběr zařízení, které má být v průběhu rezervace restartováno.

Poslední stroj je určen k řízení běhu celé hry a poskytnutí služeb web serveru. Má funkci PXE serveru poskytujícího fyzickým počítačům obrazy operačních systémů, zároveň s nimi sdílí data z virtuálních strojů a umožňuje rezervaci testovací linky, protože zprostředkovává přístup k webovým stránkám s rezervační aplikací.

Na přiloženém DVD jsou všechny potřebné programy již připraveny a jejich funkčnost byla otestována. Před spuštěním hry je nutné upravit jejich konfigurační soubory podle použité síťové infrastruktury.

V sekci 4.2.3 byly vyjmenovány všechny součásti PXE serveru a jejich úloha v této technologii. Testování PXE serveru probíhalo v počítačové síti společnosti ABRATICA s.r.o.[11], jejíž lokální sít obsahuje vlastní DHCP server umístěný na firewallu. Proto obraz připraveného stroje konfiguraci DHCP neobsahuje. Podrobný návod pro operační systém Debian, který popisuje instalaci i nastavení DHCP serveru, je dostupný na této adrese: http://www.debian-administration.org/articles/478. Cesty k souborům TFTP serveru na stroji připraveném na DVD se shodují s těmi v návodu.

Pro spuštění hry je dále nutné upravit konfigurační soubory PXE serveru. Vzorový

9 Program pro přenos souborů. Při přenosu nové verze již uloženého souboru stáhne pouze nová data. http://linux.die.net/man/1/rsync.

10 Návod na stažení: http://computerpr0n.com/2011/04/esxi-remote-management-part-2/ .

11 Společnost poskytující poradenství v oblasti počítačových sítí a zabívající se jejich správou. Bližší informace na http://abratica.cz/.

soubor je umístěn na DVD v adresáři TFTP serveru (/var/lib/tftp-boot/pxelinux.cfg/default). Podle něj musí být vytvořeny soubory pojmenované podle MAC adres fyzických serverů. Důvody jsou vysvětleny v sekci 4.2.3.

NFS server vyžaduje úpravu souboru /etc/exports. Tento soubor obsahuje cestu k adresáři, kde bude exportován obraz disku virtuálního stroje. Počítači, který si vyžádá soubory pomocí protokolu NFS, bude sdílen obsah právě tohoto adresáře. Zároveň je možné přístup omezit na základě IP adres. V případě hry Bandwidth challenge je toto omezení na místě, protože o soubory budou žádat dva různé počítače. Vzorové nastavení je součástí /etc/exports/, stačí pouze přidat odpovídající IP adresy fyzických serverů.

Soubor fstab, zmíněný v sekci 4.2, který obsahuje seznam všech připojovaných diskových oddílů, využívaný při startu systému, se nalézá ve stejném adresáři jako skript ovládající celý běh hry. Jelikož sdílení pomocí NFS vyžadují 2 servery, musí být vytvořeny i dva tyto soubory. Oba jsou již připraveny a není potřeba je nijak upravovat, pokud nedojde ke změně adresáře pro export obrazů disků virtuálních strojů.

Poslední změny už se týkají pouze připraveného skriptu. Z důvodu automatizace se velká část příkazů provádí na vzdálených strojích. Proto je nutné doplnit jejich IP adresy. Především se jedná o server s ESXi, ze kterého se kopírují obrazy disků virtuálních strojů, dále pak o adresy fyzických serverů. Musí však být zohledněn způsob jejich restartu.

Funkčnost skriptu především závisí na jeho opakovaném spouštění. K tomu slouží soubor crontab. Jelikož není nutné, aby se skript spouštěl, dokud není nakonfigurován, automatické provádění není zatím nastaveno.

Po úpravě zde zmíněných konfiguračních souborů a sestavení celé síťové infrastruktury je hra připravena k použití. Registrace hráčů však ještě vyžaduje vytvoření textového souboru, který využívá rezervační aplikace pro kontrolu přihlašovacích údajů. Vše je vysvětleno v sekci 4.3.1.

4.6 Informační materiály

Součástí zadání této bakalářské práce je tvorba informačních materiálů určených k propagaci hry Bandwidth challenge. Byly vytvořeny webové stránky, které spolu s informacemi o hře obsahují i již zmíněnou rezervační aplikaci. Dále vznikl propagační leták, který je součástí příloh k této práci.

Webové stránky jsou napsány stejně jako rezervační aplikace v jazyce JSP. Obsahují základní informace o hře a především zprostředkovávají přístup k rezervacím vyhrazené linky. Webový server, který je zobrazuje, umožňuje přístup pouze využitím protokolu HTTPS (*Hypertext Transfer Protocol Secure)* zajišťujícím šifrovaný přenos. Důvodem této implementace je přenos hesel v čisté podobě přes síť Internet. Dříve zmíněná hash se počítá až při zpracování dat aplikací. Aby nebylo možné hesla odposlechnout, je nutné jejich šifrování.

Obsah stránek je v českém jazyce, ale je možné později dodělat anglickou verzi. Aby nemusela být složitě upravována logika jednotlivý stránek, postačí vytvořit kopie všech stávajících souborů, přeložit obsah, kde bude potřeba a dále pak pomocí tzv. sezení (session) řídit načtení odpovídajících stránek.

Webové stránky jsou připraveny ke spuštění na stroji, který obsahuje přiložené DVD.

Informační leták má motivovat hráče k navštívení webových stránek a účasti v této hře. Byl vytvořen ve freewarovém programu pro tvorbu vektorové grafiky Inkscape. Barevně odpovídá koncepci webových stránek a jeho část tvoří hlavní stranu tohoto webu. Obrázek použitý jako podklad vznikl pouze pro tuto bakalářskou práci kombinací fotografie koncovky síťového kabelu a grafu navrženého v programu Open Office Calc. Motiv prasklého skla byl dokreslen ručně.

Informační leták zatím neobsahuje konkrétní kontaktní údaje, ale je pro ně vyhrazena plocha ve spodní části. Před spuštěním hry zde bude doplněna adresa webových stránek a email pro registraci hráčů.

5 Závěr

Protokol TCP prošel od svého vzniku výrazným vývojem. Bylo navrženo mnoho variant, které více či méně úspěšně řešily problémy tohoto protokolu na linkách s velkou kapacitou a zpožděním. Tato bakalářská práce navrhuje výběr těch nejlepších variant nebo případnou tvorbu nových prostřednictvím hry.

Zadáním této práce bylo připravit síťovou a výpočetní infrastrukturu spolu s potřebnými softwarovými nástroji pro hru Bandwidth challenge. Byla navržena počítačová síť obsahující potřebné stroje pro uskutečnění této hry a testovací linku určenou k ověření vlastností jednotlivých variant v praxi. Všechny její součásti jsou popsány v této bakalářské práci a centrální server řídící chod celé hry je připraven na přiloženém DVD. Po drobných úpravách, kterým se věnuje sekce 4.5 a přípravě potřebné počítačové sítě je tedy možné tuto hru uskutečnit.

Dále byla vytvořena aplikace sloužící k rezervaci testovací linky a knihovna, která poskytuje hráčům potřebná data a zároveň vyhodnocuje úspěšnost jejich testování. Jelikož mají všechny skupiny stejné výchozí podmínky (virtuální stroje pro vývoj, rezervovaný přístup k testovací lince a data generovaná stejným způsobem), je zajištěno spravedlivé hodnocení jejich výsledků.

Součástí zadání byla také příprava informačních materiálů. Byly vytvořeny webové stránky poskytující informace o hře, ale jejich hlavní součástí je především rezervační aplikace. Příloha této práce obsahuje i návrh propagačního letáku k této hře.

Hra Bandwidth challenge nabízí netradiční možnost srovnání variant protokolu TCP a především umožňuje účastníkům této hry přístup k reálné síťové infrastruktuře.

Literatura:

[1] HLADKÁ Eva, *Transportní vrstva* [online]. 2010, [cit. 1. 1. 2012]. Dostupný na: <https://is.muni.cz/auth/el/1433/jaro2011/PB156/um/lecture5.pdf>.

[2] LOW Steven, PETERSON Larry, WANG Limin, *Understandig TCP Vegas: Theory and Practice* [online]. 2000, [cit. 1. 1. 2012]. Dostupný na <ftp://ftp.cs.princeton.edu/techreports/2000/616.pdf>.

[3] *FAST TCP: From Theory to Experiments* [online]. 2005, [cit. 1. 1. 2012]. Dostupný na: <http://authors.library.caltech.edu/8589/1/JINieeen05.pdf>.

[4] SKALA Martin, *Hybrid congestion control for high-speed networks* [online]. 2011, [cit. 1. 1. 2012]. Dostupný na: <https://is.muni.cz/auth/el/1433/podzim2011/PV177/um/3_Skala-HCC-TCP.pdf>.

[5] *Virtual Machine to Physical Machine Migration* [online]. 2000, upraveno 2010, [cit. 1. 1. 2011]. Dostupné na: <http://www.vmware.com/support/v2p/doc/V2P_TechNote.pdf>.

[6] HOLLIS, *Preboot Execution Environment (PXE) Specification* [online]. 1999, [cit. 1. 1. 2011]. Dostupné na:<http://download.intel.com/design/archives/wfm/downloads/pxespec.pdf>
.

Přílohy:

Na přiloženém DVD je obraz disku stroje potřebného pro řízení síťové infrastruktury.

Obsahuje:

PXE server

NFS server

Apache Tomcat a webové stránky

knihovna pro generování dat.

Následnující strana obsahuje návrh informačního letáku.