Pište prosím čitelně.

Písemná práce ke zkoušce z PSI

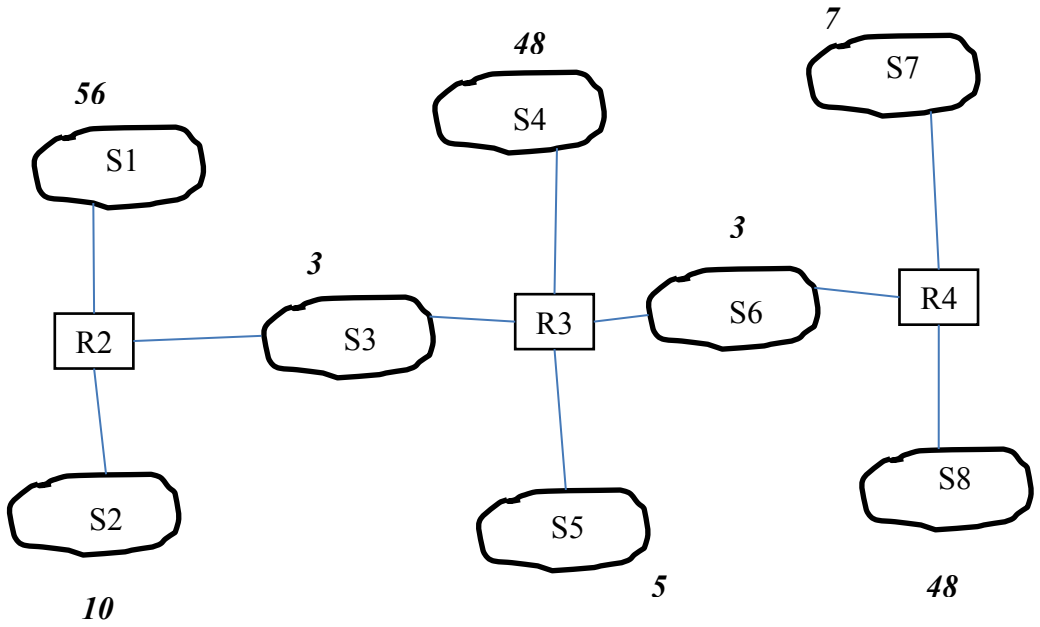
12.6.2013 (4)

1	2	3	4	5	6	7	8	9	10

Jméno:
Počet bodů:

1. Je dáno zapojení 8 IP sítí, přiřaďte IP adresy směrovačů. IP adresy subsítí přidělte podle zadaných rozsahů subsítí pomocí subnettingu z jednoho souvislého bloku adres (např. 20.0.0.0/16). Umožněte následný, nekomplikovaný supernetting.

   Norm8



Řešení:

Normálně bychom mohli rozdělit adresní prostor na 8 (2^3) stejných dílů a každé podsíti přidělit adresní prostor 2^{13} adres (13 bitů). Abych nemusel tak moc počítat, budu předpokládat, že 13 bitů je moc, že by stačilo pro každou subsíť pouze 8 bitů. V subnettingu tedy využiji 8 bitů pro subsíť a 8 bitů pro adresování uvnitř subsítě. Takže např. S1 bude mít adresu 20.0.0/24, S2 bude mít adresu 20.0.1/24, S3 adresu 20.0.2/24 atd. Subsíť bude mít 56 stanic a jedno rozhraní směrovače. Např. R2 20.0.0.1 a hostitelské systémy pak 20.0.0.2 až 20.0.0.57. Směrovač bude mít adresy rozhraní 20.0.0.1, 20.0.1.1, 20.0.2.1. Směrovací tabulka v R2 bude

S1: 20.0.0/24 rozhraní eth0
S2: 20.0.1/24 rozhraní eth1
S3: 20.0.2/24 rozhraní eth2
S4: ..... rozhraní eth2
S5: ..... rozhraní eth2
S6: ..... rozhraní eth2
S7: ..... rozhraní eth2
S8: ..... rozhraní eth2

A abyste udělali „nekomplikovaný“, potřebujete adresy v této tabulce agregovat, pokud možno sloučit všechny adresy směřující na jedno rozhraní do jedné adresy. To v tomto případě lze, pokud všechny subsítě na rozhraní eth2 dokážeme vyjádřit jako jednu adresu s posunutou maskou. Binárně je adresa S3 něco jako A.00000010.xxxxxxxxb. Pokud se nám podaří schovat S4 až S8 do části xxxxxxxx, tak máme vyhráno. Z tohoto pohledu musíme rozdělit adresní prostor mezi subsítě. Každá subsíť bude mít jinou délku síťové části adresy. Sousední adresy můžeme „zkrátit“ o příslušný počet bitů. Např. v routeru R4 zavedu adresy

S7: 20.0.1.01000000/28 – max 15 stanic a
S8: 20.0.1.00000000/26 – max 63 stanic a
S6: 20.0.1.00000000/29 – max 7 stanic

Mohu agregovaně adresovat jako 20.0.1.00000000/25, protože mají prvních 25 bitů stejných. V systému mi pak zbývá 20.0.1.10000000/25, které musím obdobně rozdělit mezi sítě S1 až S4. Pak ve směrovacích tabulkách bude co rozhraní, to jedna položka. A lépe už to nepůjde.

2. Použitím supernettingu navrhněte statické směrování tak, aby se minimalizoval počet položek ve směrovacích tabulkách aniž by bylo použito implicitní směrování (síť 0.0.0.0/0, příp. ::/0) Jako příklad uveďte obsah směrovacích tabulek R2 a R3.

Řešení viz výše. Implicitní adresa není třeba.

3. Připojte uvedené sítě k Internetu přes 1 rozhraní směrovače R4 a uveďte, jak se případně změní konfigurace routerů.

Abychom mohli připojit síť do Internetu, musíme v každém směrovači zadat adresy, které jsou vně naší sítě. Protože je patrně neznáme, použijeme implicitní adresu 0.0.0.0 s maskou 0.0.0.0 na posledním řádku směrovací tabulky. Směrovací tabulka musí být uspořádána podle délky masky, a pak také podle zvětšující se adresy (pro pořádek). Síť s nejdelší maskou (32 jedničkových bitů) bude v tabulce jako první. Síť s nejkratší maskou (samé nuly) bude poslední. Pronásobíme-li jakoukoliv adresu nulovou maskou, dostaneme nulovou adresu. Takže na tuto položku „zabere“ každá adresa, proto musí být poslední. My toho využijeme pro určení směru, který vede k hraničnímu směrovači.

Na poslední řádek tabulky uvedeme síť 0.0.0.0, maska 0.0.0.0 použití routeru ano, rozhraní vedoucí k hraničnímu směrovači.

4. Vysvětlete, co znamenají pojmy SlowStart, Contention Avoidance, Fast Retransmit, Fast Recovery a kde se uvedené algoritmy používají.

Tyto pojmy se týkají řešení problematiky zahlcení sítě. Velikost okénka v TCP stačí při přímém propojení dvou počítačů k tomu, aby rychlejší vysílač nezahltil pomalejší přijímač. Kapacita vedení omezuje rychlost přenosu, ale jestliže kapacitu nedokážeme využít, tak musí brzdit příjemce.

Jestliže ale připojíte tyto počítače do sítě, kde jsou směrovače s konečnými délkami vstupních front, pak nastane problém co se záplavou paketů, které se do front nevejdou. Čili jak to

zařídit, aby počítače generovali jen tolik paketů, kolik jich směrovač stačí zpracovat. Škrcení paketů (ICMP zpráva Source Quench) je účinná jen pro směrovač sousedíci s hostem. Existují i explicitní metody řízení toku dat, kdy routery, plazící jazyk po zemi, přidávají do záhlaví paketu značky, aby zdroj ubral na rychlosti. My jsme se zabývali implicitními metodami. V prvé řadě jak host pozná, že to s rychlostí přehnal. Směrovač mu paket zahodí a host nedostane po timeoutu ACK. Druhou možností je, že směrovač zahodí náhodně paket, ale další v sérii pošle. Pak host dostane duplicitní potvrzení, protože zahozený paket pořád chybí a další příchozí se do série zatím nehodí. Rychlost přenosu je možné spočítat jako podíl množství dat, odeslaných bez potvrzení a doby odezvy (RTT – Round Trip Time). Čili čím víc se nám podaří odeslat paketů bez potvrzení (sliding window), tím je přenos rychlejší. No a teď je třeba stanovit, jakou rychlost si můžeme dovolit. Slow start postupuje tak, že se pokouší exponenciálně navyšovat počet paketů poslaných bez potvzení. V okamžiku, když se nějaký ztratí (timeout), nastaví jakýsi práh na polovinu dosažené hodnoty. Sníží počet nepotvrzených paketů na jeden (někdy dva) a zkouší to exponenciálně navyšovat znovu, ale jen do hodnoty prahu. Pak se uklidní a počet nepotvrzených zvyšuje jen lineárně – to je fáze Contention Avoidance. Fast Retransmit funguje tak, že detekuje duplicitní ACK a nečeká na timeout. Fast Recovery funguje tak, že pokud se zjistí duplicitní ACK, stanoví se opět nový práh (polovina dosažené rychlosti) a pokračuje se fází Contention Avoidance. Pokud nastane timeout, pak je to nějaký velký průšvih (zahodilo se moc paketů) a v tomto případě se zase začne fází Slow Start.

5. Vysvětlete, kdy se požívá a jak funguje ICMP redirect.

Představte si, že máte k jednomu přepínači nebo HUBu připojeny kromě hostitelských systémů také 2 směrovače. Při konfiguraci hostu pochopitelně neznáte rozdělení adres v síti (tj. na které se dostanete přes první směrovač a na které přes druhý) a proto nastavíte implicitní směrování na jeden z nich. Teď se ale může stát, že pošlete data do prvního a ten je pošle do druhého směrovače, protože adresujete síť nebo host za druhým směrovačem. V tu chvíli to první směrovač pozná (směruje do směrovače na tomtéž přepínači) a pošle hostu ICMP redirect s informací, aby danou síť směroval přímo na druhý směrovač. Host přidá do směrovací tabulky další položku a jede dál.

6. Uveďte, jak se provádí vzájemné ověření pomocí symetrického šifrování a **bez pomoci** ověřovacího serveru.

Předpokladem pochopitelně je, že oba uzly, A i B, znají společný tajný klíč. Jinak to v symetrické kryptografii ani nejde. A se chce ověřit jako první (třeba klient A k serveru B). A pošle serveru náhodné číslo X, server ho zašifruje, přidá náhodné číslo Y a obojí pošle do A. A zkontroluje (dešifruje X), zašifruje Y a pošle zpět do B. Pokud by se někdo vydával za A (třeba T), pak neumí správně zašifrovat náhodné číslo Y. Pomůže si tak, že to Y pošle do B. Dobrák B mu je zašifruje pošle do T a T mu je vrátí. Což není dobře.

Takže A požádá B o ověření. B pošle náhodné číslo Y do A. A je zašifruje a přidá náhodné X. B dešifruje zašifrované Y a pokud je v pořádku, pak pošle zašifrované X. Takže pánové pozor, pokud od Vás někdo chce soukromá data a vydává se za banku, mobilního operátora apod., pak většinou chce ověřit, že se dovolal tam, kam chtěl. Vy musíte nejprve ověřit, že se

bavíte s pravým subjektem a teprve pak se nechat ověřit. Protože nesdílíte tajné heslo, musíte vymyslet dotaz tak, abyste si ověřili, že Vaši odpověď vzdálený subjekt opravdu zná, že z Vás tu informaci nemámí a sám neví nic.

7. Uveďte, jak se provádí zpracování analogového signálu (např. signál z mikrofonu) a jeho přenos počítačovou sítí. Co je třeba zahrnout do výpočtu celkového zpoždění?

Nejprve se signál navzorkuje (analogový vzorek v čase), pak se A/D převodníkem převede na číselnou hodnotu (kvantifikace), která se od analogové hodnoty pochopitelně liší. Pak se hodnota může nějak transformovat (např. nelineárně 16 bitů na 8). Osmibitové hodnoty se naskládají do paketu (paketizace - protože nemůžeme sítí posílat byte po byte). Celé pakety přeneseme a na straně příjemce opět rozložíme na slabiky a synchronně interpretujeme. Takže zpoždění je dáno rychlostí vzorkování (převod musí být rychleší – jinak co by s daty převodník A/D dělal, že?), počtem vzorků v paketu, dobou přenosu (včetně režie vysílání a příjmu). Na straně přijímače je třeba počítat s rozptylem doby doručení (jitter). Použijí se vyrovnávací paměti a data se interpretují s umělým zpožděním. Pokud to paket nestihne včas, musí se zahodit.

8. Co je to Leacky Bucket a Token Bucket. Kde se tyto algoritmy používají? Nakreslete zjednodušené grafy (časové závislosti), jak se projevují při přenosu dat.

Leacky Bucket znamená, že dále se propouší data maximálně rychlostí odkapávání – omezení rychlosti přenosu za routerem. Nakreslete závislost na čase sami.

Token Bucket znamená, že máte opět přísun tokenů, ale navíc vědro, ve kterém je zachycujete. Pokud přijdou data, prochází vstupní rychlostí tak dlouho, dokud se vědro nevyprázdní. Pak musejí čekat na přicházející kapky. Přenášíte-li webovou stránky, pak třeba projde celá rychle, ale dlouhý přenos videa poběží rychle jen do vyčerpání vědra. Pak se rychlost sníží na rychlost přicházejících tokenů.

9. Co jsou to distribuované hashovací tabulky v P2P sítích. Jak se mapují do tabulky soubory, jak se rozdělí tabulka mezi uzly P2P sítě? Jaké základní operace se používají pro manipulaci s tabulkou?

Haschovací tabulka má obecně na daném řádku informaci, ze které haschováním vznikne index tohoto řádku. V tomto případě P2P sítí je to tak, že je v tabulce uveden údaj, na kterém stroji se soubor najde. Čili ze jména souboru vypočtu hasch a soubor uložím na stroj, jehož adresu najdu v haschovací tabulce v řádku s indexem hasche. Teď ale jde o to, kam ty soubory uložit. Jak vhodně zvolit stroj pro uložení souboru. Tady bychom opět rádi viděli pokud možno rovnoměrné vytížení strojů. Takže např. v Chordu se haschuje i adresa stroje do stejné množiny jako soubory. Přidávám-li stroj, vypoču hasch a adresu stroje uložím na příslušný řádek haschovací tabulky. Jak ukládám soubor, viz výše. No a nyní jde o to, kam tabulku uložit. První možnost je mít jednu tabulku a přes ní se dostávat k souborům (může být kdekoliv). Po prohlédnutí N/2 položek najdu to, co hledám. Nebo použiji binární prohledávání. Druhou možností je, že tabulku rozhodím do jednotlivých uzlů tak, že vezmu

řádek ukazující na následující uzel, pak na druhý za ním, čtvrtý za ním, osmý za ním, atd. Když pak hledám, prohlédnu vlastní redukovanou tabulku, naleznu uzel s nejbližší vyšší hodnotou hasche (sekvenčně) a přejdu na něj. Buď tu soubor je, nebo opět prohlédnu sekvenčně jeho tabulku a najdu opět uzel s nejbližší vyšší hodnotou hasche a postup opakuji. To že struktura není lineární, ale kruhová souvisí s tím, že pokud se dostanete na konec, pokračujete od začátku.

Prováděné operace jsou lookup (vyhledávání) vrací kde soubor je a put, uložení souboru na správný uzel. Podstatné je, že soubory neukádáte nahodile, ale na vypočtená místa.

10. Vysvětlete princip Content Delivery Networks (CDN). Jak se udržuje konzistentnost nabízených dat v případě statických stránek a v případě dynamických stránek?

Problém je v tom, že potřebujeme vymyslit systém, který by při distribuci souborů nepřetěžoval ani komunikační linky, ani servery. Řešením jsou vyrovnávací paměti a replikace serverů. Historicky se nejprve používali vyrovnávací paměti v klientech a serverech (odlehčení disků), pak se vytvořily proxy a cache – umístění dat někde v síti, aby se nemusela pořád přenášet, přitom se nerozlišovalo, o jaká data jde (transparentní cache, lokální cache). No a poslední výkřik je přenést data co nejblíže ke klientovi a navíc jen ta, která bude potřebovat (geograficky, podle významu a zájmu, ...). Pokud budeme přenášet R/O soubory, pak je to bez problémů. R/W soubory (dynamické webové stránky) se musí generovat v těch listových serverech nebo vůbec v nich neumisťovat. Jenže generování stránek bývá závislé na databázích a máme tu problém opět. Distribuovat databáze nebo ne? Systém ale má další vlastnosti, jako třeba přidávání informací podle umístění klienta (reklamy). Někdy se do stránek přidává kód značkovacího jazyka, který vede listové servery k tomu, co mají přidat.

Přístup od originálního serveru k těmto náhražkám se zajistí směrováním (síťová úroveň), transformací přes DNS (podstrčení adresy replikačního serveru) nebo redirekcí v HTTP protokolu.