Séria A

### Úloha č.1:

Rozsiahly program prekladáme v systéme s n procesormi. Nemôžeme zrejme predpokladať, že preklad sa zrýchli až n-krát. Ktoré faktory spôsobujú, že sa toto maximum nedosiahne?

Preklad programu sa n-krát nezrýchli, pretože časť výkonu procesorov zaberá ich réžia (prideľovanie úloh, resp. vlákien prekladu jednotlivým procesorom). Zabezpečovaním náväznosti prekladu jednotlivých modulov sa tiež stráca určitý čas.

### Úloha č. 2:

Sieťový systém s podporou zdieľania súborov rozlišuje zamykanie na zápis a zamykanie na čítanie. Čo sa stane, ak proces zamkol súbor na čítanie a chce to teraz zmeniť na zamknutie na zápis? Ako je možné zamknutie na zápis zmeniť na zamknutie na čítanie?

Len proces, ktorý zamkol súbor ho môže odomknúť. Takže proces, ktorý zamkol súbor na čítanie, môže zmeniť zámok na čítanie na zámok na zápis. A to buď tak, že najprv zruší zámok na čítanie, a potom súbor zamkne na zápis (pričom po zrušení prvého zámku sa môže stať, že ho nebude môcť hneď zamknúť na zápis, lebo ho má na zápis zamknutý iný proces), alebo tak, že zmení typ zámku (to sa stane len vtedy, ak iný proces nemá tento súbor zamknutý na zápis, inak procesu zostane zámok na čítanie).

### Úloha č. 3:

Súborový server používa tri identické disky. Ako je možné zabezpečiť systém pred chybou, ktorá by sa mohla stať na jednom z nich? Popíšte riešenie, ktoré využije kapacitu dvoch diskov a tretí použije na zálohu. Ako sa bude postupovať po zistení chyby?

Systém je možné zabezpečiť tak, že bity sa budú zapisovať striedavo na prvé dva disky a na tretí sa bude zapisovať paritný bit (1, ak počet jednotiek je párny; 0 inak). Napr.:



Ak na ktoromkoľvek disku nastane chyba, tak sa chybná časť nahradí paritnými bitmi vytvorenými podľa ostatných dvoch diskov (vytvorenie paritných bitov môžeme popísať binárnou funkciou ekvivalencie ). Napr.:

Pred chybou



Rekonštrukcia po chybe disku2



### Úloha č. 4:

DIR1/SUBDIR1/SUBDIR2/file.doc je cesta k súboru file.doc v systéme Novell Netware. [SRCEF] je filtrom pre adresár SUBDIR1, [SRF] filter súboru file.doc, [RWMF] moje používateľské práva k DIR1 a [RCEA] práva mojej skupiny k SUBDIR1. Aké sú moje efektívne práva pre súbor file.doc? Môžem kamarátovi sprístupniť súbory z adresára SUBDIR2? Zdôvodnite.

Moje efektívne práva pre súbor file.doc získam bitovým AND-om cez všetky prístupové práva (filter, zdedené práva od nadradených adresárov, moje používateľské práva a práva mojej skupiny pre súbor). Takže mám jediné prístupové právo [R] k file.doc. Kamarátovi tento súbor sprístupniť nemôžem, pretože nemám prístupové právo meniť atribúty súboru [A].

### Úloha č. 5:

Transakcia je postupnosť viacerých úkonov, ktoré musíme buď vykonať všetky, alebo ani jeden. Uvažujme server, ktorý vybavuje typické jednoduché transakcie - odpočítanie z jedného bankového konta a pripočítanie na druhé konto. Popíšte problémy, ktoré môžu nastať pri rôznych chybách v prenose a chybách vo vlastnom systéme.

Pri prenose sa môže poškodiť alebo úplne stratiť niektorý potrebný údaj – číslo účtu odosielateľa, prijímateľa, suma. Takže server nebude vedieť komu má odpočítať z konta, komu má pripočítať, o akú sumu sa jedná. Pri výpadku komunikačnej siete, elektrického prúdu alebo vlastného systému počas vykonávania transakcie, keď už nejaké operácie boli vykonané, sa môže stať, že ostatné operácie nebudú vykonané. Teda sa môže stať, že z jedného konta bude odpočítaná čiastka, ale nebude pripočítaná na druhé konto, čo napr. môže využiť útočník, ktorý prenikne do systému a pripíše si odpočítanú sumu na svoje konto.

### Úloha č. 6:

Problémy pri spracovaní transakcií je možné čiastočne riešiť žurnálom, do ktorého zapíšeme zmeny, ktoré sme urobili. Prečo je potrebné zapisovať do žurnálu staré i nové hodnoty zmenených premenných? Ktoré problémy nevyriešime ani žurnálom?

Ak sa zapisujú staré aj nové hodnoty, je vidieť, či prebehla nejaká zmena. Ak sa staré hodnoty rovnajú novým, pravdepodobne ide o chybu transakcie (záleží na transakcii). V tomto prípade sa dá podľa žurnálu nielen skontrolovať priebeh transakcie, ale aj nájsť chybná či chýbajúca časť transakcie. Žurnál nechráni systém pred útočníkom, teda neoprávneným vniknutím do systému.

### Úloha č. 7:

Popíšte hlavné rozdiely medzi bezspojovou a spojovo orientovanou komunikáciou. Porovnajte možnosti využitia. Aké sú problémy pri simulovaní spojovo-orientovaného protokolu pomocou bezspojového? V ktorých prípadoch je bezspojová komunikácia lepšia?

Bezspojová komunikácia je jednosmerná komunikácia a prebieha bez nadviazania spojenia s druhou stranou. Klient posiela otázky a čaká na odpoveď servera určitý čas (do timeoutu). Spojovo orientovaná komunikácia je obojsmerná komunikácia a prebieha po nadviazaní spojenia klienta so serverom. Spojenie sa udržuje počas celej komunikácie. Takáto komunikácia povoľuje napr. možnosť autentifikácie klienta (prihlásenie používateľa do systému). Bezspojová komunikácia je rýchlejšia a menej zaťažuje server pri veľkom počte klientov, s ktorými server nepotrebuje mať spojenie (www servery). Pri simulovaní spojovo orientovaného protokolu bezspojovým môže byť problém s udržaním simulovaného spojenia (keďže nejde o skutočné spojenie) pri veľkom počte klientov a následnom preťažení servera, ktorý nestíha obsluhovať požiadavky do timeoutu.

### Úloha č. 8:

Zoraďte jednotlivé fázy komunikácie pri využití rozhrania typu BSD sockets na strane servera {accept, close, listen, bind, write} podľa náväznosti a stručne jednou vetou každú fázu charakterizujte. Je možné použiť funkciu bind aj na strane klienta? Aký by to malo význam?

1. bind – metóda na priradenie voľného socketu reprezentovaného jeho číslom;

2. listen – načúvanie, teda čakanie na nadviazanie spojenia na danom sockete;

3. accept – prijatie správy od klienta, ktorá žiada o komunikáciu (ešte neobsahuje dáta);

4. write – posielanie správ klientovi;

5. close – ukončenie komunikácie a uvoľnenie socketu.

Funkciu bind je možné použiť aj na strane klienta.

### Úloha č. 9:

Procedúra Incr(var x,y:integer) { inc(x); inc(y) } zväčší o 1 obsah premenných x a y. Počiatočná hodnota v premennej i je 0. Ako sa zmení po volaní procedúry Incr( i, i ) lokálne a ako po jej volaní ako vzdialená procedúra (RPC) ? Vysvetlite.

Pri lokálnom volaní procedúry sú parametre volané odkazom a pri vzdialenom parametre volané odkazom musia byť zmenené na hodnoty. Takže lokálne volaná procedúra Incr(i,i) inkrementuje premennú i dvakrát (i=2), pretože dvakrát pracuje s tým istým odkazom. Ale keďže vzdialená procedúra pracuje s hodnotami, dvakrát inkrementuje hodnotu 0 ({inc(0); inc(0)}), takže výsledkom je i=1.

### Úloha č. 10:

Zvážte problémy, ktoré môžu vzniknúť pri prenášaní 32-bitových argumentov pri vzdialenom volaní procedúr. Procesory SPARC ukladajú slová systémom big endian (od najvyšších bytov), procesory Intel systémom little endian. Akú hodnotu by mal parameter 2 po prenesení zo systému SPARC (počítač SUN) do systému Intel? Ako je možné tieto problémy riešiť?

Parameter 2 (2 = 000 000 000 000 000 000 000 000 000 000 102) by po prenesení zo systému SPARC do Intelu mal hodnotu 1073741824 (1073741824 = 010 000 000 000 000 000 000 000 000 000 002). Tento problém sa dá riešiť buď konvertovaním údajov pred formátu, aký používa prijímateľ (big/little endian), alebo používaním abstraktného zápisu (pred údajmi sa pošle správa o tom, v akom formáte sú zapísané).

Séria B

### Úloha č.1:

Charakterizujte rozdiely medzi iteratívnym a konkurentným serverom a popíšte možnosti ich realizácie pomocou BSD socketov.



Iteratívny server so spojením dokáže komunikovať naraz len s jedným klientom. Komunikácia s ďalším klientom je možná až po ukončení spojenia. Konkurentný server so spojením reaguje na výzvu k nadviazaniu komunikácie vytvorením svojej kópie ako dcérskeho procesu. Táto jeho kópia potom komunikuje s klientom pomocou iného socketu (na obr. Socket2) a server môže zatiaľ rovnakým spôsobom obslúžiť ďalších klientov.

### Úloha č.2:

Popíšte typy parametrov volania vzdialenej procedúry (RPC) a spôsoby ich odovzdávania pri jej volaní.

Vzdialenej procedúra môže mať vstupné (IN) alebo vstupno-výstupné (INOUT) parametre. Vstupné sú reprezentované buď hodnotou alebo odkazom a nie sú menené počas behu procedúry (resp. ak neboli globálnymi premennými, môžu byť zmenené len lokálne, teda len počas behu procedúry a po jej skončení majú opäť pôvodnú hodnotu). Vstupno-výstupné parametre sú reprezentované odkazom a počas behu procedúry sú zmenené. Parametre reprezentované odkazom musia byť pri volaní vzdialenej procedúry nahradené hodnotami, čo zabezpečuje stub.

### Úloha č.3:

Aké sú hlavné rozdiely medzi distribuovaným a sieťovým operačným systémom? Uveďte príklady, kedy je výhodnejší jeden a kedy druhý.

Distribuovaný operačný systém poskytuje jednotný pohľad pre všetkých používateľov vyrovnaním rozdielov medzi uzlami (počítačmi). Každý používateľ má rovnaký operačný systém a pristupuje ku všetkým zdrojom rovnako. Adresárová štruktúra je pre používateľa strom, ktorý je nezávislý od fyzického umiestnenia súborov na jednotlivých uzloch. Distribuovaný operačný systém umožňuje nie len zdieľanie súborov a periférií ako aj sieťový operačný systém, ale aj zdieľanie výkonu uzlov.

Sieťový operačný systém je otvorenejší, má menšie nároky na jednotlivé uzly. Každý uzol môže mať iný operačný systém. Zdroje sú špecifikované ako lokálne a vzdialené. Ku vzdialeným sa pristupuje po prihlásení sa do vzdialeného systému.

Výhodou distribuovaných operačných systémov je zdieľanie výkonu uzlov, jednotné používateľské rozhranie. Sieťový operačný systém je výhodnejší, keď nie je žiadané (potrebné) užšie zviazanie uzlov jednotným rozhraním a prístupom k zdrojom.

### Úloha č.4:

Charakterizujte aspoň tri funkcie relačnej vrstvy a tri funkcie prezentačnej vrstvy referenčného modelu OSI. Pre každú z funkcií popíšte aspoň jeden prípad použitia.

Relačná vrstva zabezpečuje identifikáciu relácií medzi procesmi, zriaďovanie relácií, ich udržiavanie, rušenie a správu kontrolných bodov, na základe ktorých sa relácie obnovujú od ich prerušenia, nie od začiatku.

Prezentačná vrstva zabezpečuje reprezentáciu údajov, šifrovanie a komprimáciu dát. Taktiež zabezpečuje konverziu formátu údajov medzi little endian, big endian, sieťovým formátom a abstraktnou syntaxou.

### Úloha č.5:

ISO OSI referenčný model je príkladom využitia viacvrstvovej architektúry pri popise zložitých štruktúr. Uveďte aspoň dva ďalšie príklady využitia viacvrstvového modelu pre popis štruktúr z iných oblastí vedy resp. praktického života.

Na prednáške bol uvedený príklad komunikácie afrického a ázijského biológa prostredníctvom ich tlmočníkov a telefónov. Podobne môžeme uvažovať komunikáciu riaditeľov dvoch spoločností, ktorí chcú urobiť fúziu. Riaditelia sa radia so svojimi právnikmi. Právnici prezentujú požiadavky svojich riaditeľov formulované „právnickou rečou“ a navzájom komunikujú napr. prostredníctvom telefónu.

Ďalším príkladom je situácia, keď sa dvaja ľudia chcú na seba pozerať. Popíšem to na jednom z nich smerom od najnižšej vrstvy k najvyšším. Oko vníma vlnové dĺžky. Takže oči môžeme považovať za fyzickú vrstvu podľa ISO OSI referenčného modelu. Oko rozkladá informáciu, ktorú prijalo v podobe vlnenia, na elementárne zložky. Tie sa prostredníctvom nervových vlákien dostávajú do senzorických oblastí mozgu, kde sa spracujú na nižšej, inštinktívnej úrovni. Ďalšie spracovanie nastáva v asociatívnej oblasti, kde sa elementárne zložky obrazu spájajú, a kde dochádza k samotnému zážitku vedomia.

### Úloha č.6:

Vypočítajte približne rýchlosť prenosu údajov v ľudskom oku za predpokladu, že rozozná vizuálne pole s cca 1000000 elementami v troch základných farebných zložkách (v každej s 64 rôznymi intenzitami) za 0,1 s. Odhady zdôvodnite.

Prenosovú rýchlosť získame vynásobením počtu elementov počtom základných farebných zložiek, rôznymi intenzitami vyjadrenými počtom bitov a počtom opakovaní za sekundu:

1 000 000 \* 3 \* 6 \* 10 = 180 Mbps .

Teda približná rýchlosť prenosu údajov v ľudskom oku je 180 Mbps.

### Úloha č.7:

Televízny signál používa šírku pásma 6 MHz. Koľko bitov za sekundu ním možno preniesť ak použijeme kódovanie pomocou štyroch úrovní (a neuvažujeme šum) ?

Podľa Nyquistovej vety:

(max. prenosová rýchlosť) ≤ 2 \* (šírka prenosového pásma) \* log2(veľkosť vstupnej abecedy) .

Dosadením dostaneme nerovnosť (max. prenosová rýchlosť) ≤ 2 \* 6 MHz \* log2 4 .

A teda televíznym signálom s použitím daného kódovania je možné preniesť maximálne 24 Mbps.

### Úloha č.8:

Odhadnite, koľko CD-ROM diskov potrebujeme na zakódovanie knižnice obsahujúcej 1000 kníh do kódu ASCII. Koľko bude trvať jej prenos modemom po linke s prenosovou rýchlosťou 64 kb/s? Do akej vzdialenosti by bolo výhodnejšie posielať tieto údaje autom na CD-ROM diskoch (predpokladajme, že za modemové spojenie platíme iba poplatok za miestny telefónny hovor)? Odhady zdôvodnite!

Označme priemerný počet znakov knihy α. Jeden znak sa v ASCII kóduje pomocou 7 bitov. Jednu knihu zakódujeme do 7\*α bitov, 1000 kníh do 7\*α kb. Počet 700MB CD-ROM diskov potrebných na uloženie jednej knihy je (7\*α)/(700\*1024\*8). Prenos knihy modemom po linke s prenosovou rýchlosťou 64kb/s trvá (7\*α)/64 s.

### Úloha č.9:

Teoreticky bezšumový kanál šírky 4 kHz snímkujeme každú milisekundu. Akú maximálnu prenosovú rýchlosť môžeme dosiahnuť? Zdôvodnite.

### Úloha č.10:

Digitálne údaje posielame kanálom so šírkou pásma 3 kHz (telefón) s odstupom signálu od šumu 20 dB. Akú najvyššiu prenosovú rýchlosť môžeme dosiahnuť? Za akých podmienok (kvality signálu) by sme mohli týmto kanálom posielať údaje rýchlosťou 1 Mb/s?

Séria C

### Úloha č.1:

Čím sa líši výpočtový model klient/server od modelu agent/manager? Uveďte aspoň dva príklady na každý z nich.

Aplikácia vo výpočtovom modeli klient/server je rozdelená na 2 časti, jedna časť beží na klientovi, druhá na serveri. Tieto časti môžu mať rôzne rozdelené 3 základné funkcie: prezentačnú (používateľské rozhranie), aplikačnú a správu dát. Jedným z možných riešení je Distributed Presentation, kde klient má na starosti len prezentačnú funkciu, server zas aplikačnú funkciu a správu dát. Príklady využitia: www server a www prehliadač (na strane klienta), databázový server a databázová aplikácia (na strane klienta).

Výpočtový model agent/manager je založený na asymetrickom vzťahu jeho prvkov: agentov a managera. V monitorovacích systémoch je agent program, ktorý poskytuje dáta tomu, kto si ich vyžiada (inému agentovi alebo managerovi), môže robiť nejaké výpočty. Manager je program, ktorý vyhodnotí zozbierané dáta a na základe výsledkov vykonáva ďalšie operácie. Príkladom využitia je monitorovanie stavu a správa siete na základe protokolu SNMP. Agenti môžu byť aj samostatnejší, môžu sa premiestňovať v sieti, robiť výpočty na rôznych miestach, čo im umožňuje ich nezávislosť na platforme. Takýto agenti sú dynamické programy pracujúce nad virtuálnym strojom (napr. JVM = Java Virtual Machine). Manager je potom statický program, ktorý generuje agentov a riadi ich činnosť.

### Úloha č.2:

Zdôvodnite, v ktorom z nasledujúcich prípadov je nutné použiť stavový a v ktorom stačí bezstavový server: a) posielanie html stránok b) prístup do databázy c) prenos a správa súborov d) správa presného času e) obsluha vzdialených terminálov f) prihlasovanie do siete g) diaľkové ovládanie prepojenia sietí h) monitorovanie vlastností siete.

1. bezstavový server – klient si vyžiada stránku, server si nemusí pamätať žiadny stav komunikácie;
2. bezstavový server – podobne ako v prípade a) , klient sa pýta, server odpovedá a jeho odpovede nie sú závislé od predošlých otázok a odpovedí;
3. stavový server – server si musí pamätať stav prenosu súborov, aby sa nestala napr. situácia pri prenose súboru, keď súbor bol zo serveru odoslaný klientovi a následne vymazaný, ale ku klientovi neprišiel úplný, alebo prišiel poškodený;
4. bezstavový server – server odpovedá klientovi na otázku, aký je presný čas, čo nevyžaduje pamätanie stavov komunikácie;
5. stavový server – je potrebný kvôli obsluhe terminálov, pri ktorej si server musí pamätať jej stav;
6. stavový server – server sa musí prepínať aspoň medzi dvoma stavmi: prihlásený/neprihlásený používateľ;
7. stavový server – podobne ako pri e) , podmienený obsluhou;
8. bezstavový server – monitorovanie si nevyžaduje zmeny stavu servera.

### Úloha č.3:

Objasnite význam pojmu Protokol vo viacvrstvovom komunikačnom modeli.

Protokol je súhrn pravidiel komunikácie dvoch rovnocenných vrstiev. Spôsob komunikácie v rámci jednej vrstvy je daný aspoň jedným protokolom. Bez neho by nebola normovaná komunikácia medzi rovnocennými vrstvami, a teda by sa mohlo stať, že by si vrstvy „nerozumeli“. Viac protokolov môže byť použitých z toho dôvodu, že jedna službu môže poskytovať viacero prostriedkov.

### Úloha č.4:

Chrbticové spojenie dĺžky 100 km pracuje prenosovou rýchlosťou 1.544 Mb/s. Rýchlosť šírenia signálu je 2/3 rýchlosti svetla. Koľko bitov sa súčasne zmestí do kábla?

Rýchlosť svetla je približne 300 000 km/s, čo je 300 000 000 ms-1. Čas, ktorý trvá prenos jedného bitu z jednej strany daného spojenia na druhú si vyjadríme takto:



a ten, keď prenásobíme prenosovou rýchlosťou spojenia, získame počet bitov, ktorý sa súčasne zmestí do kábla:

1.544 Mb/s \* 0,0005 s = 1 544 000 b s-1 \* 0,0005 s = 772b.

### Úloha č.5:

Na prepojenie medzi PC a modemom využijeme sériové rozhranie s kódovaním systémom Start-Stop so zabezpečením parity (jedným bitom), dvoma Stop bitmi a ôsmimi údajovými bitmi modulačnou rýchlosťou 9600 Bd. Akú najvyššiu rýchlosť prenosu údajov môžeme dosiahnuť ?

Na prenos 8 údajových bitov je dokopy potrebných 12 bitov (1 štartovací, 1 paritný, 2 Stop a 8 údajových bitov), teda údajové bity tvoria 2/3 prenášaných bitov. Keďže vstupná abeceda má veľkosť 2 (používa sa binárne kódovanie), najvyššiu rýchlosť prenosu údajov vyjadríme:

9 600 Bd \* (2/3) = 6 400 b/s.

### Úloha č.6:

Načrtnite priebeh signálu, kódujúceho sekvenciu 0001110101 v kódovaní Manchester a Diferenciálny Manchester.



### Úloha č.7:

Navrhnite spôsob modulácie, ktorou by bolo možné preniesť kanálom šírky 4 kHz údaje rýchlosťou 40 kb/s. Aké budú minimálne požiadavky na kvalitu signálu ? (vyjadrite odhadom odstupom signálu od šumu v dB).

Na vyjadrenie minimálnych požiadaviek na kvalitu signálu potrebujeme vyjadriť podiel výkonu signálu a šumu:

40 kb/s <= 4kHz \* log2(1 + (Ps/Pn))

10 <= log2(1 + (Ps/Pn))

210 <= 1 + (Ps/Pn)

Ps/Pn = 1 023

Odstup signálu od šumu je potom 10 \* log 1 023, čo je približne 30dB.

### Úloha č.8:

Aký odstup signálu od šumu potrebujeme dosiahnuť na vedení, aby sa dal vytvoriť prenosový kanál T1 (1,544 Mb/s) v pásme 50 kHz ?

Na výpočet odstupu signálu od šumu musíme vypočítať podiel výkonu signálu a šumu:

1,544 Mb/s = 1 544 kb/s <= 50 kHz \* log2(1 + (Ps/Pn))

30,88 <= log2(1 + (Ps/Pn))

230,80 <= 1 + (Ps/Pn)

Ps/Pn >= 230,80 – 1 .

Toto dosadíme do vzorca na výpočet odstupu signálu od šumu 10 \* log(230,80 – 1), čo sa približne rovná 92,96 dB.

### Úloha č.9:

Koaxiálny kábel s prenosom signálu v kvalite 30 dB v rozsahu od 50 MHz po 350 MHz využijeme frekvenčným multiplexom po pásmach šírky 20 MHz. V každom pásme vynecháme ochranné pásmo po 1 MHz na okrajoch kvôli presluchom. Vypočítajte maximálnu celkovú prenosovú kapacitu kábla.

Do káblu so šírkou 300 MHz sa zmestí 13 pásiem (300 MHz delíme 22 MHz, čo je šírka pásma spolu s ochrannými pásmami po jeho okrajoch). Zo vzťahu pre odstup šumu od signálu vypočítame podiel výkonu signálu a šumu:

30dB = 10 \* log(Ps/Pn)

3 = log(Ps/Pn)

1 000 = Ps/Pn .

Maximálnu prenosovú rýchlosť vypočítam:

13 \* 20MHz \* 10 = 2,6 Gb/s .

### Úloha č.10:

Telefónne vedenie je uložené hviezdicovo od každého účastníka k telefónnej ústredni, káblová televízia používa jeden kábel, ktorý vedie ku každému účastníkovi. Predstavte si, že káblové rozvody sú z optického vlákna a prenesú 10 Gb/s. Ako by sa táto sieť dala využiť na telefónny rozvod ? Koľko účastníkov by mohlo komunikovať po jednom takom vlákne ?

Táto sieť by sa dala použiť na telefónny rozvod pomocou pulznej kódovej modulácie (PCM), ktorá na kódovanie spojenia používa rýchlosť 64 kb/s. Keďže prenosová rýchlosť optického vlákna je 10 Gb/s, počet účastníkov na jednom vlákne je (10 Gb/s) / (64 kb/s) = 156 250.

Séria D

### Úloha č.1:

Vysvetlite princíp presmerovania (redirection) v implementácii sieťových operačných systémov.

V sieťovom operačnom systéme je umožnené zdieľanie I/O zariadení (diskov, tlačiarní, atď.) a súborov v adresároch. Tieto zariadenia a adresáre si vzdialený počítač pripojí (Windows diskové zariadenia a adresáre ako logické disky, Linux príkazom mount namountuje do príslušného adresára). Užívateľ k nim potom pristupuje ako k lokálnym, resp. sa mu javia ako lokálne zariadenia a adresáre. Redirection je presmerovanie, ktoré vykonáva operačný systém a je to presmerovanie z pripojených zariadení a adresárov na ich fyzické umiestnenie.

Príklad: V Linuxe mám v adresári AL namountovaný adresár AV obsahujúci súbory f1, f2. V adresári AL mi zobrazí len súbory f1, f2, takže neviem, že tieto sú uložené na vzdialenom počítači v adresári AV. Mne sa to javí, akoby boli uložené niekde na mojom disku. Avšak operačný systém sa postará pomocou presmerovania (redirection) o prístup k ich fyzickému uloženiu na tom vzdialenom počítači.

### Úloha č.2:

Prenosový kanál pracuje s odstupom signálu od šumu 30 dB. Spočítajte, akú minimálnu šírku pásma potrebujeme, aby sme mohli dosiahnuť prenosovú rýchlosť a) 9600 b/s b) 64 kb/s c) 1.544 Mb/s d) 155.5 Mb/s e) 2 Gb/s.

Zo vzorca na výpočet odstupu signálu od šumu

 odstup = 10 \* log10 (Ps/Pn)

vyjadríme pomer intenzity signálu k intenzite šumu:

 30 = 10 \* log10 (Ps/Pn)

 103 = Ps/Pn.

Na výpočet minimálnej potrebnej šírky pásma využijeme Shannonovu vetu

 R <= H \* log2 (1 + (Ps/Pn)).

Dosadíme do tohto vzorca údaje a upravíme:

 R / log2 1001 <= H, resp. (R \* ln 2) / ln 1001 <= H.

a) R = 9 600 b/s; minimálna šírka pásma sa približne rovná 963,16 Hz.

b) R = 64 kb/s; minimálna šírka pásma sa približne rovná 6,422 kHz.

c) R = 1,544 Mb/s; minimálna šírka pásma sa približne rovná 154,908 kHz.

d) R = 155,5 Mb/s; minimálna šírka pásma sa približne rovná 15,602 MHz.

e) R = 2 Gb/s; minimálna šírka pásma sa približne rovná 200,658 MHz.

### Úloha č.3:

Digitálny signál modulovaný kvadratúrnou amplitúdovou moduláciou (so štyrmi stavmi fáze a dvoma úrovňami aplitúdy nosnej) posielame telefónnym kanálom (so šírkou pásma 3 kHz) modulačnou rýchlosťou 1200 Bd. Akú prenosovú rýchlosť môžeme dosiahnuť? Akú minimálnu kvalitu telefónnej linky pri tom potrebujeme?

Prenosovú rýchlosť (si označíme ako Rp) vypočítame podľa vzťahu:

 Rp = M \* log2 v .

Teda:

 Rp = 1 200 \* log2 (4 \* 2) = 3,6 kb/s .

Zvýšením modulačnej rýchlosti môžeme zvýšiť prenosovú rýchlosť. Obmedzenie zvýšenia prenosovej rýchlosti je určené vzťahom:

 R <= 2 \* H \* log2 v .

Po dosadení:

 R <= 2 \* 3000 \* log2 8 = 18 kb/s.

Na určenie kvality telefónnej linky opäť potrebujeme Shannonovu vetu. Po dosadení:

 18 000 <= 3 000 \* log2 (1 + (Ps/Pn))

Dostávame

 Ps/Pn >= 26 – 1 = 63,

čo použijeme na výpočet odstupu signálu od šumu:

 odstup = 10 \* log10 63,

a to sa približne rovná 17,993 dB.

### Úloha č.4:

Aký je rozdiel medzi demodulátorom v modeme a kóderom v kodeku ? (obidva konvertujú analógový signál na digitálny)

Kodek sa používa v telefónnych ústredniach. Kodér v ňom prevádza analógový signál na digitálny. Modem sa používa na pripojenie počítača do počítačovej siete prostredníctvom verejnej telekomunikačnej siete. Modem sa pripája na sériové rozhranie, a preto demodulátor v modeme naviac kóduje digitálny signál Start-Stop kódovaním, ktoré toto rozhranie používa.

### Úloha č.5:

Cez optické vlákno chceme prenášať obrazový signál v rozlíšení 1024 x 768 bodov, pričom farba každého bodu je kódovaná 24 bitmi a za jednu sekundu potrebujem preniesť 60 snímok. Akú potrebujem minimálnu šírku pásma ? Odhadnite, koľko takýchto kanálov (s potrebnou šírkou pásma) sa teoreticky vmestí do viditeľného spektra (vlnovej dĺžky približne 0,4 – 0,7 mikrometra) ?

Jedna snímka má 1024 x 768 bodov, čo je 786 432 bodov. Jeden bod je kódovaný 24 bitmi. Jednu snímku teda popisuje 18 874 368 bitov. Za sekundu potrebujeme preniesť 60 snímok, čo je 1 132 462 080 bitov. Tým máme vyjadrenú rýchlosť prenosu (1,132 Gb/s). Zo vzťahu

 R <= 2 \* H \* log2 v

vypočítame minimálnu potrebnú šírku pásma H (použijeme abecedu {0,1}, preto v = 2).

 1 132 462 080 <= 2 \* H \* log2 2

 H = 566 231 040 Hz = 566,231 MHz .

Na odhad počtu kanálov použijeme vzťah pre frekvenciu f a vlnovú dĺžku λ a rýchlosť svetla c

 f = c / λ ,

pomocou ktorého vypočítame minimálnu a maximálnu frekvenciu viditeľného spektra.

 f1 = c / λ1

 f1 = 300 000 / (0,7 / 1000 000) = 428571428571 kHz (zaokrúhlene)

 f2 = c / λ2

 f2 = 300 000 / (0,4 / 1000 000) = 750000000000 kHz .

Vypočítame šírku spektra

 f2 - f1= 321428571429 kHz (zaokrúhlene).

Túto vydelím šírkou kanála a dostávame, že do viditeľného spektra sa zmestí približne 567 573 kanálov so šírkou pásma 566,231 MHz.

### Úloha č.6:

Vstupný reťazec údajov obsahuje sekvenciu A, DLE, ETX, B, DLE, DLE, STX. Ako bude vyzerať odoslaná postupnosť ? Ako bude vyzerať odoslaná postupnosť ak použijeme metódu bit stuffing pri posielaní reťazca 0111111011111011110 ?

Odoslaná postupnosť prvého reťazca:

STX A DLE DLE DLE ETX B DLE DLE DLE DLE DLE STX STX,

kde podčiarknuté slová sú riadiacimi znakmi (STX = start of transmission, DLE = data link escape, ETX = end of transmission).

Odoslaná postupnosť druhého reťazca pri použití metódy bit stuffing:

01111110 011111 0 1011111 0 011110 01111110,

kde sekvencia 01111110 označuje začiatok a koniec, a kde po každej sekvencii piatich jednotiek je vložená 0 kvôli odlíšeniu sekvencie správy od jej konca.

### Úloha č.7:

Reťazec údajov 1001011 chcem zabezpečiť CRC kontrolou pomocou polynómu x5+x3+1. Ako bude vyzerať výsledná správa ?

Generátorom polynómu je reťazec 101001. Správu predĺžime o 5 núl a vydelíme týmto reťazcom pomocou operácie XOR:

100101100000

101001

00110010

 101001

 0110110

 101001

 0111110

 101001

 0101110

 101001

 0001110

Zvyškom po delení je reťazec 1110. Ten doplníme na dĺžku piatich bitov predpísaním jednej nuly, dostávame reťazec 01110. Výsledná správa je: 100101101110.

Overené na stránke <http://www.macs.hw.ac.uk/~pjbk/nets/crc/> .

### Úloha č.8:

Aké vlastnosti musí mať kontrolný polynóm, ktorý pri CRC kontrole odhalí a) jednobitovú chybu b) dve izolované chyby, vzdialené najviac o k bitov c) nepárny počet chýb

### Úloha č.9:

Lokálna sieť prenáša údaje rýchlosťou 10 Mb/s. Signál potrebuje na prechod medzi stanicami 0,02 ms. Komunikácia prebieha s pozitívnym potvrdzovaním rámcov. Pre akú veľkosť rámca bude sieť vyťažená aspoň na 50 percent?

Keďže sieť má byť vyťažená aspoň na 50%, čas ktorý čaká stanica na potvrdenie prijatia rámca je najviac rovnaký ako čas, ktorý trvá posielanie rámca. Prechod signálu je 0,02 ms. Za túto dobu sa prenesie 200 bitov, ktoré postačujú na poslanie potvrdenia. Teda poslanie potvrdenia trvá 0,02 ms. Z toho vyplýva, že rámec musí mať veľkosť aspoň 200 bitov.

### Úloha č.10:

Používame satelitný kanál s bezchybným prenosom 64 kb/s. Prijímame 512 bitové bloky, ktoré potvrdzujeme veľmi krátkymi správami metódou posuvného okna (sliding window). Aká je priepustnosť kanála (priemerná rýchlosť komunikácie s potvrdením) v prípade, že veľkosť okna je 1 resp. 8 resp. 16 resp. 128 blokov ? (Oneskorenie na satelit je cca 0,3 s).

Priemernú rýchlosť komunikácie si vyjadríme ako podiel počtu prenesených bitov (počet blokov β vynásobený počtom bitov jedného bloku) a času prenosu tohto počtu bitov (počet prenesených bitov vydelený prenosovou rýchlosťou zväčšený o oneskorenie na satelit dané posielaním okna a jeho potvrdením):

 

Veľkosť okna: Približná priepustnosť kanála:

1 blok 842,105 b/s

8 blokov 6,169 kb/s

16 blokov 11,253 kb/s

128 blokov 40,355 kb/s

Séria E

### Úloha č.1:

Správa, zabezpečená CRC kontrolou s polynómom x4+x3+x+1 prišla v tvare 1101011101001. Zistite, či nedošlo k chybe pri prenose.

To, či došlo k chybe pri prenose, zistíme tak, že vydelíme prijatú správu kontrolným polynómom. Ak zvyšok po delení je nenulový, došlo k chybe.

1101011101001

11011

0000111101001

 11011

 001011001

 11011

 0110101

 11011

 000011

Keďže sme po delení získali zvyšok 11, došlo k chybe pri prenose správy.

### Úloha č.2:

Sieť, využívajúca viacnásobný prístup s pripočúvaním s prenosovou rýchlosťou 10 Mb/s je 1 km dlhá. Rámce majú dĺžku 256 bitov (vrátane 32 bitovej hlavičky) a po odoslaní každého rámca chcem, aby prijímajúca stanica poslala potvrdenie dĺžky 32 bitov. Predpokladajme, že v sieti nevznikajú kolízie a signál sa šíri rýchlosťou 200 000 km/s. Akú najvyššiu efektívnu prenosovú rýchlosť (koľko bitov údajov za sekundu) možno dosiahnuť ?

Prenosovú rýchlosť siete si vyjadríme takto:

 10 Mb/s = 10 b/μs .

Signál prejde sieť za

 1 km / 200 000 km/s = 0,000 005 s = 5 μs .

Prenos jedného rámca trvá

 256 b / 10 b/μs = 25,6 μs,

k čomu musíme pripočítať dobu, za ktorú signál prejde cez sieť. Teda prenos jedného rámca od odosielateľa k prijímateľovi trvá 30,6 μs. Rovnako vypočítame dobu prenosu potvrdenia od prijímateľa správy k jej odosielateľovi:

 32 b / 10 b/μs + 5 μs = 8,2 μs .

Dokopy prenos správy a jej potvrdenia cez sieť trvá 38,8 μs. Za túto dobu sa prenesie 256 + 32 b, pričom len 224 b údajov. Z toho vyplýva, že sa prenesie 224 b za 38,8 μs. Úpravou dostávam najvyššiu efektívnu rýchlosť (najvyššiu, pretože sme počítali s minimálnym časom potrebným na prenos údajov):

 224 b / (1 000 000 / 38,8) s ≅ 5 773 195 b/s = 5,773 Mb/s .

### Úloha č.3:

Aký je význam Preambuly v rámci štandardu CSMA/CD ?

Preambula je určená fyzickej vrstve a je upozornením, že stanica vysiela. Za ňou nasleduje začiatok vysielaného rámca (sekvencia 11) a samotný rámec. Preambula taktiež zabezpečuje synchronizáciu času dát, kedy boli vysielané a času, kedy sú prijaté.

### Úloha č.4:

Zriaďujeme CSMA/CD sieť (neštandardnú), pracujúcu rýchlosťou 1 Gb/s na 1 km káblovom prepojení bez opakovačov. Signál sa šíri rýchlosťou 200 000 km/s. Aká musí byť minimálna dĺžka rámca ? Zdôvodnite. Čo by sa mohlo stať, ak by bol rámec menší ?

Keďže detekciu kolízií robia len vysielajúce stanice, je potrebné, aby vysielajúca stanica vysielala jeden rámec aspoň dvakrát tak dlho, ako trvá prenos signálu medzi najvzdialenejšími stanicami (najprv musí prejsť signál, ktorý vysiela jedna stanica k druhej, ktorá sa tiež pokúša vysielať, potom je potrebný čas na prenos jam signálu od tejto stanice ku všetkým ostatným, ktorý signalizuje kolíziu a následné súperenie; preto dvojnásobok slot time-u). Tento prenos signálu trvá

 1 km / 200 000 km/s = 0,000 005 s = 5 μs .

Z toho vyplýva, že prenos rámca trvá aspoň 10 μs. Prenosová rýchlosť siete je 1 Gb/s = 1 kb/μs, a teda minimálna veľkosť rámca je 10 kb.

Ak by bola veľkosť rámca menšia, mohlo by sa stať, že žiadna zo staníc nebude schopná detekovať chybu, resp. jedna zo staníc zúčastnených kolízie by mohla považovať svoj rámec za odvysielaný, pritom tento rámec sa dostal do kolízie a následne by sa nezúčastnila súperenia o médium, teda by sa rámec, ktorý odvysielala stratil.

### Úloha č.5:

V sieti s prístupom CSMA/CD sú pripojené dve stanice, ktoré majú stále pripravené údaje na vysielanie. Pre súperenie stanice využívajú štandardný exponenciálny odpočítavací algortimus. Aký je priemerný počet kolízií, ktoré vzniknú medzi dvoma úspešnými prenosmi ?

### Úloha č.6:

Čo je to kolízna doména a aký význam má tieto domény oddeľovať. Prečo keď chceme zvýšiť prenosovú rýchlosť, musíme zmenšiť veľkosť kolíznej domény ?

Kolízna doména je oblasť počítačov a všetkých zariadení, kde sa šíri kolízia. Oddeliť ich znamená rozdeliť jednu kolíznu doménu na dve menšie kolízne domény. Kolízia sa potom šíri v menšej oblasti, teda sa týka menšieho počtu počítačov. To uľahčuje odstraňovanie kolízií, zrýchľuje súperenie, pretože menej počítačov súperí o médium, a teda menej kôl binárneho odpočítavania je potrebných. Čím je počet kôl odpočítavania pri súperení menší, tým rýchlejšie sa odstráni kolízia. S počtom počítačov v kolíznej doméne sa zväčšuje pravdepodobnosť, že na odstránenie kolízie bude potrebných viac ako10 kôl odpočítavania, pričom 10 kôl je hranica, ktorej prekročenie znamená ukončenie súperenia a chybové hlásenie neúspešného vysielania. Zvýšenie efektívnej prenosovej rýchlosti dosiahneme tak, že skrátime čas, ktorý je potrebný na súperenie, skrátime slot time, ktorý je potrebné čakať pri odstraňovaní kolízií a na konci vysielania, kedy sa čaká dvakrát slot time kvôli detekcii prípadnej kolízie. Toto všetko dosiahneme zmenšením veľkosti kolíznej domény.

### Úloha č.7:

Prenosový kanál šírky 2 kHz s odstupom signálu od šumu približne 10 dB snímkujeme každú milisekundu. Akú maximálnu prenosovú rýchlosť môžeme dosiahnuť? Navrhnite spôsob modulácie, ktorá pri tomto snímkovaní umožní maximálnu rýchlosť dosiahnuť.

Vyjadríme si podiel intenzity signálu a šumu:

 10 = 10 \* log10 (Ps/Pn)

 10 = Ps/Pn .

Podľa Shannonovej vety vyjadríme maximálnu prenosovú rýchlosť (označenú ako R):

 R = 2 \* log2 (1 + 10)

 R = 6,919 kb/s .

### Úloha č.8:

Vysvetlite algoritmus súperenia, podľa ktorého stanice v sieti s prístupom CSMA/CD rozhodnú, ktorá z nich bude vysielať

Súperenie v sieti CSMA/CD sa realizuje formou binárneho odpočítavania. Toto prebieha v niekoľkých kolách. V každom kole generátor náhodných čísel vyberie pre každú stanicu náhodné číslo. V i-tom kole sú tieto čísla v intervale 0 až (2i-1) (v prvom kole sú to čísla 1 alebo 0, v druhom čísla 0 = 002, 1 = 012,
2 = 102, 3 = 112, atď.). Stanica, ktorá má najnižšie číslo, môže vysielať. Stanica s j-tym najnižším číslom čaká
(j-1)-krát slot time (stanica s druhým najnižším číslom čaká jeden slot time, s tretím najnižším číslom čaká dvakrát slot time, atď.). Po uplynutí tejto doby a ak je médiu voľné, môže stanica, ktorá je na rade začať vysielať. Ak opäť nastane kolízia, pokračuje sa ďalším kolom odpočítavania. Inak súperenie o médium skončilo. Tento algoritmus je nedeterministický, pretože nie je možné určiť čas, do ktorého bude môcť začať nejaká stanica vysielať.

### Úloha č.9:

V sieti s prístupom CSMA/CD využívajú stanice pre súperenie štandardný exponenciálny odpočítavací algortimus. V treťom kole odpočítavania si stanica vygenerovala náhodné číslo 3. Ako sa má zachovať? Popíšte jej činnosť vo všetkých možných prípadoch až do obnovenia komunikácie.

Prípady, ktoré môžu nastať si rozdelíme na dva typy: také, ktoré v tomto kole vedú k úspešnému ukončeniu súperenia a také, ktoré vedú k ďalšej kolízii a následne k ďalšiemu kolu odpočítavania. V oboch prípadoch môže byť takéto rozdelenie nižších čísel (vyššie nás zatiaľ nezaujímajú):

* táto stanica má najnižšie číslo
* je aspoň jedna stanica, ktorá má menšie číslo.

Uvažujme, že žiadne číslo menšie alebo rovné 3 nie je priradené viac ako jednej stanici. Potom sledovaná stanica čaká 3 slot time-y. Po uplynutí tejto doby začne vysielať, ak médium je voľné. Tento čas čaká v každom prípade, aj vtedy, ak má najmenšie číslo. Ak po odvysielaní nenastane v tomto kole už žiadna kolízia, skončí sa súperenie úspešne a pokračuje sa v komunikácii. Ak kolízia nastane, táto stanica sa už súperenia nezúčastňuje.

Ak jedno z čísel 0 až 3 je pridelené viacerým staniciam, nastane kolízia, ktorá sa rieši ďalším kolom odpočítavania. Nami sledovaná stanica sa nedostane v tomto kole k vysielaniu, pokiaľ nastane kolízia staníc s nižšími číslami. V prípade, že má viac staníc číslo 3 (aspoň dve, vrátane sledovanej stanice), stanica sa pri vysielaní rámca dostane do kolízie. Ak túto detekuje, vysiela jam signál, inak ho dostane.

### Úloha č.10:

V sieti s prenosovou rýchlosťou 10Mb/s používam prístupovú metódu CSMA/CD, ktorá umožňuje maximálnu vzdialenosť staníc 500 m (nie je štandardná). Chcem zvýšiť prenosovú rýchlosť na 100 Mb/s, ale ponechať štruktúru rámcov takú, aká bola v pôvodnej sieti. Aká bude môcť byť v novej sieti maximálna vzdialenosť staníc ? Zdôvodnite.

Veľkosť rámca (označíme ako r) dostaneme vynásobením prenosovej rýchlosti a času potrebného na prenos jedného rámca (označíme ako t). Vyjadríme si preto veľkosť rámca pri zadanej prenosovej rýchlosti a pri želanej rýchlosti:

 r1 = 10 b/μs \* t1

 r2 = 100 b/μs \* t2 .

Keďže veľkosť rámca chceme ponechať, platí:

 r1 = r2

 10 b/μs \* t1 = 100 b/μs \* t2

 t1 = 10 \* t2 .

Rýchlosť šírenia signálu si označíme ako v. Tá sa nemení. Maximálnu vzdialenosť staníc vyjadríme ako násobok času potrebného na prenos signálu a rýchlosti šírenia signálu (ako x označíme hľadanú maximálnu vzdialenosť staníc v novej sieti):

 500 m = t1 \* v

 x = t2 \* v = t1 / 10 \* v .

Po úprave dostávam hľadanú maximálnu vzdialenosť staníc v novej sieti:

 t1 = 500 m / v

 t1 = x / v \* 10

 500 / v = x / v \* 10

 x = 50 m.

Séria F

### Úloha č.1:

Charakterizujte význam a funkcie podvrstvy logického riadenia spoja (LLC) v komunikačnom modeli.

Referenčný model ISO/OSI bol navrhnutý pre dvojbodové spojenia (point-to-point). Viacbodové spojenie (multipoint link), ktoré využívajú LAN siete, si vyžaduje riadenie prístupu k spoločnému médiu. Tento prístup zabezpečuje podvrstva riadenia prístupu k médiu (MAC = Medium Access Control). Podvrstva logického riadenia spoja LLC v komunikačnom modeli zabezpečuje to, čo pôvodne mala zabezpečovať v referenčnom modeli ISO/OSI spojová vrstva (Data Link Layer), teda: rozdelenie dát na rámce a detekciu ich začiatku a konca (Framing), potvrdzovanie rámcov a riadenie údajov (pomocou protokolov Stop-And-Wait Protocol a Sliding Window Protocol), okrem detekcie a odstraňovania chýb v rámcoch (pomocou Error-Detecting Codes alebo Error-Correcting Codes), ktoré zabezpečuje podvrstva MAC. Podvrstva LLC tiež zabezpečuje identifikáciu sieťového protokolu.

### Úloha č.2:

Čo by sa stalo, keby stanica v sieti 10BASE-T s prístupom CSMA/CD poslala rámec menší ako 64 oktetov ? Popíšte najhoršie scenáre, vysvetlite a zdôvodnite problémy.

Na to, aby vysielajúca stanica detekovala kolíziu, musí vysielať tak dlho, ako trvá šírenie medzi dvoma najvzdialenejšími stanicami v sieti oboma smermi (2 \* slot time). Z definície 10 Mb/s Ethernetu vyplýva, že rámec musí mať aspoň 512b (= 64 oktetov). V prípade, ak by poslala stanica rámec menší ako 64 oktetov, tak nie je zaručené, že detekuje kolíziu, resp. zachytí collision jam. Teda si bude „myslieť“, že rámec bol v poriadku odoslaný, ale v skutočnosti sa stal súčasťou kolízie a nebol doručený.

### Úloha č.3:

V sieti CSMA/CD 10BASE-T prestal fungovať Switch (most). Dočasne ho chceme nahradiť Hubom (opakovačom) s príslušným počtom portov. Za akých podmienok je to možné ? Zdôvodnite. Za akých podmienok je naopak možné nahradiť Hub Switchom ?

Pri nahradzovaní switcha hubom je potrebné dodržať:

* Maximálnu vzdialenosť 5 000 metrov medzi najvzdialenejšími uzlami novovzniknutej kolíznej domény. Túto vzdialenosť je potrebné dodržať, aby bolo možné detekovať vzniknuté kolízie.
* Medzi každými dvoma stanicami môžu byť najviac 2 huby, prípadne 4 huby, pričom aspoň 2 z piatich segmentov, ktoré vytvárajú, nemajú pripojené žiadne uzly.
* Keďže hub nemá žiadnu pamäť, každý k nemu pripojený segment musí používať rovnakú prenosovú rýchlosť.

Nahradenie hubu switchom je vždy možné, dokonca prináša zrýchlenie, keďže switch oproti hubu prenosovú rýchlosť medzi stanicami nezdieľa, ale každej poskytuje rovnakú (v našom prípade 10 Mb/s).

### Úloha č.4:

Na prepojenie sietí S1 a S2 je použitý transparentný most A a na prepojenie sietí S2 a S3 transparentný most B. Stanice 1 a 2 sú zapojené v sieti S1, stanica 3 v sieti S2 a stanica 4 v sieti S3. Mosty majú (po resetovaní) prázdne prepájacie tabuľky. Postupne pošle stanica 1 rámec k stanici 2, 3 k 1, 4 k 3, 2 k 1 a stanica 3 k stanici 2. Popíšte, ako sa budú zapĺňať prepájacie tabuľky mostov a ktoré z uvedených piatich rámcov si bude môcť prečítať stanica číslo 4.

Transparentné mosty neprenášajú kolízie a zbytočnú komunikáciu medzi segmentmi. Rámce prenášajú len do toho segmentu, kde sa cieľová stanica rámca nachádza. Umiestnenie staníc v segmentoch si ukladajú do prepojovacej tabuľky. Po resetovaní je táto tabuľka prázdna. Jej napĺňanie prebieha tak, že ak most dostane rámec, ktorého odosielateľskú stanicu nemá zapísanú v tabuľke, zapíše si jej MAC adresu a segment, v ktorom sa stanica nachádza. Posielanie rámcov potom prebieha na základe záznamov v tejto tabuľke. Ak však most v nej nenájde záznam o cieľovej stanici rámca, pošle ho do každého segmentu.

Náš príklad popíšeme nasledujúcou tabuľkou (rámce sú posielané v poradí, v akom sú indexované). Prvý stĺpec je indexom posielaného rámca. Druhý stĺpec označuje odosielajúcu stanicu a tretí stĺpec cieľovú stanicu rámca. Štvrtý stĺpec obsahuje zoznam staníc, ktoré dostanú daný rámec. Piaty stĺpec obsahuje zoznam staníc zapísaných v prepojovacej tabuľke mostu A, šiesty stĺpec mostu B.



Stanica 4 dostane rámce označené indexom 1 a 5.

### Úloha č.5:

Čo sa stane, ak v sieti, prepojenej transparentnými mostmi, premiestnim stanicu z jednej kolíznej domény do inej ?

Keďže záznamy v prepojovacej tabuľke transparentného mostu (Learning Bridge) majú svoju životnosť, po (zvyčajne) 10-15 minútach sa záznam o umiestnení stanice v kolíznej doméne vymaže. Počas trvania tohto intervalu nebude možná komunikácia s premiestnenou stanicou, pretože most si ešte pamätá jej predošlé umiestnenie. Po uplynutí tohto intervalu je záznam vymazaný, preto je umožnené tejto stanici opäť komunikovať s ostatnými. Ak stanica pošle rámec, most sa týmto naučí jej nové umiestnenie v kolíznej doméne.

### Úloha č.6:

Väčšia firma má 20 pobočiek a v každej najviac 10 oddelení. Má pridelenú sieťovú IP adresu triedy B. Ako zorganizovať prideľovanie IP adries staniciam tak, aby každá pobočka i každé oddelenie malo samostatnú podsieť ? Najviac koľko staníc bude možné adresovať v oddelení a ako bude vyzerať sieťová maska ?

Maska siete triedy B bez podsietí je 255.255.0.0 . Na rozlíšenie 20 pobočiek je potrebných 5 bitov, na rozlíšenie 10 oddelení v pobočke sú potrebné 4 bity, takže na identifikáciu podsiete je potrebných 9 bitov. Maska siete potom vyzerá takto: 11111111 11111111 11111111 100000002 => 255.255.255.128, pričom na identifikáciu počítačov v oddeleniach zostalo 7 bitov. Preto v jednom oddelení je možné adresovať najviac 27 = 128 staníc.

### Úloha č.7:

IP adresu v hexadecimálnej reprezentácii A22F1582 prepíšte do bodkovanej reprezentácie. V sieti ktorej triedy sa nachádza táto stanica a aká je jej sieťová adresa za predpokladu, že jej sieťová maska je 255.255.240.0 ?

A22F158216 = 10100010 00101111 00010101 100000102 => 162.47.21.130 .

IP adresa v bodkovanej reprezentácii je 162.47.21.130 .

Táto IP adresa patrí do siete triedy B (IP adresy 128.0.0.0 – 191.255.255.255). Táto sieť je ďalej rozdelená podsiete (keby tak nebolo, maska siete by bola 255.255.0.0).

Sieťovú adresu získame binárnym ANDom IP adresy a masky siete:

IP adresa (162.47.21.130): 10100010 00101111 00010101 10000010

Maska siete (255.255.240.0): 11111111 11111111 11110000 00000000

Sieťová IP adresa: 10100010 00101111 00010000 00000000 .

Sieťová IP adresa v bodkovanej reprezentácii je 162.47.16.0 .

### Úloha č.8:

Stanica s IP adresou 160.144.190.100 v sieti s maskou 255.255.224.0 chce poslať IP paket na adresu 160.144.174.120. Ako bude postupovať? Ako bude postupovať pri odosielaní paketu na adresu 160.144.158.180?

Na to, aby stanica mohla poslať svoj paket, musí vedieť hardwarovú adresu cieľovej stanice. Pri jej zisťovaní postupuje tak, že najprv určí, či cieľová stanica sa nachádza v tej istej sieti (sieti s tou istou sieťovou IP adresou), a to tak, že spraví bitový AND cieľovej IP adresy a masky siete. Ak je jeho výsledkom sieťová IP adresa posielajúcej stanice, cieľová stanica je v tej istej sieti. V našom prípade musíme najskôr zistiť sieťovú IP stanice, ktorá posiela IP paket. Urobíme to opäť bitovým ANDom IP adresy posielajúcej stanice a jej masky siete:

IP adresa posielajúcej stanice (160.144.190.100): 10100000 10010000 10111110 11000100

Maska podsiete posielajúcej stanice (255.255.224.0): 11111111 11111111 11111110 00000000

Sieťová IP adresa posielajúcej stanice: 10100000 10010000 10111110 00000000 .

Teraz zistíme, či uvedené cieľové stanice sa nachádzajú v tej istej sieti:

IP adresa cieľovej stanice (160.144.174.120): 10100000 10010000 10101110 11111000

Maska podsiete posielajúcej stanice (255.255.224.0): 11111111 11111111 11111110 00000000

160.144.174.120 & 255.255.224.0 10100000 10010000 10101110 00000000 .

Výsledok sa zhoduje so sieťovou IP adresou posielajúcej stanice, takže cieľová stanica leží v tej istej sieti. Hardwarové adresy a IP adresy počítačov v sieti sa nachádzajú v ARP tabuľke. Z nej posielajúca stanica zistí hardwarovú adresu cieľovej stanice. Ak sa táto nenachádza, zistí sa pomocou ARP protokolu. A to tak, že sa vyšle ARP-REQUEST, t.j. rámec obsahujúci typ žiadosti (=1), hardwarovú adresu odosielateľa, IP adresu odosielateľa a IP adresu cieľovej stanice. Na túto výzvu odpovedá len stanica s príslušnou IP adresou (označenou ako IP cieľovej stanice). Odpovedá ARP-RESPONSEom, t.j. rámcom obsahujúcim typ žiadosti (=2), hardwarovú adresu odosielateľa, IP adresu odosielateľa, hardwarovú adresu cieľovej stanice a IP adresu cieľovej stanice. Po zistení hardwarovej adresy cieľovej stanice môže stanica poslať IP paket.

V druhom prípade máme:

IP adresa cieľovej stanice (160.144.158.180): 10100000 10010100 10011110 10110100

Maska podsiete posielajúcej stanice (255.255.224.0): 11111111 11111111 11111110 00000000

160.144.158.180 & 255.255.224.0 10100000 10010100 10011110 00000000 .

Výsledok sa nezhoduje so sieťovou IP adresou, to znamená, že cieľová stanice nie je v tej istej sieti. Preto odosielajúca stanica posiela IP paket na ROUTER (smerovač), ktorý zabezpečí poslanie IP paketu do príslušnej siete.

### Úloha č.9:

Popíšte spôsob defragmentácie IP paketu v cieľovej stanici. Čo sa stane, keď príde oneskorený fragment paketu po vyčlenenom čase (timeoute) ?

K fragmentácii IP paketu dochádza vtedy, keď je potrebné z jednej siete poslať do druhej siete paket väčší ako je v tej sieti umožnené. Paket sa potom rozdelí do viacerých paketov príslušnej veľkosti. Táto fragmentácia je potom popísaná v ich hlavičkách. Súčasťou hlavičky IP paketu sú 4 FLAG bity, vyjadrujúce príznaky fragmentácie. Medzi nimi je 1 bit označený MF, ktorý vyjadruje, či za ním nasleduje ďalší fragment. Ďalej hlavička IP paketu obsahuje 12 bitový fragment offset, ktorý popisuje posunutie v rámci fragmentácie. Podľa tohto fragment ofsetu sa usporiadajú dáta paketov, teda jednotlivé fragmenty pôvodného IP paketu. Ak aspoň jeden fragment nepríde do timeoutu, tak sa zahodia všetky ostatné fragmenty a vyžiada sa opäť celý fragmentovaný IP paket . Oneskorený fragment paketu sa musí zahodiť.

### Úloha č.10:

V čase zavedenia IP verzie 6 predpokladajme, že sa za jednu pikosekundu pridelí milión adries. Na ako dlho zásoba adries vystačí ?

Za jednu pikosekundu, ktorá odpovedá 1\*10-12 sekundy sa pridelí 106 adries. Protokol IPv6 používa adresu dĺžky 16B = 128 b, teda je možné vygenerovať 2128 adries. Čas, ktorý táto zásoba adries vystačí, je vlastne počet sekúnd, koľko bude generovať všetky možné adresy. Ten vyjadríme:

2128 / 106 = t /10-12

2128 / 1018 = t

Čas, ktorý nám zásoba adries vystačí, je približne 10 790 283 070 806 rokov (vo výpočte nie sú zohľadnené priestupné roky).

Séria G

### Úloha č.1:

Prečo nie je možné na prepojenie sietí Ethernet (10BASE-T) a FastEthernet (100BASE-TX) použiť opakovač (hub) ? Vysvetlite čo najnázornejšie. Ako tieto siete prepojiť?

Keďže Ethernet (10BASE-T) a FastEthernet (100BASE-TX) používajú rôzne prenosové rýchlosti. Na ich prepojenie je potrebná nejaká vyrovnávacia pamäť, ktorá zabezpečí prechod medzi nimi. Problém nastáva najmä pri prechode z FastEthernetu na 10Mb Ethernet, pretože ten používa nižšiu prenosovú rýchlosť a nie je schopný preniesť rovnaké množstvo dát v porovnateľnom čase. Vyrovnávacia pamäť sa potom používa vlastne na oneskorenie dát

Opakovač (hub) nemá vlastnú pamäť, preto nie je možné ho použiť na prepojenie týchto dvoch sietí, resp. akýchkoľvek sietí s rôznymi prenosovými rýchlosťami. Na prepojenie sietí s rôznymi prenosovými rýchlosťami je možné použiť buď most (bridge) alebo switch, obe tieto zariadenia majú vlastnú vyrovnávaciu pamäť.

### Úloha č.2:

Na odstránenie nežiaducich prepojení sietí transparentnými mostami sa použije algoritmus hľadania minimálnej kostry. V akom grafe sa táto kostra hľadá a aké sú dôsledky - čo sa stane, keď algoritmus túto kostru nájde.

Podmienkou pre nájdenie minimálnej kostry grafu je, aby graf, v ktorom minimálnu kostru hľadáme, bol súvislý. Platí, že každý súvislý graf má kostru. Ak je splnená táto podmienka, zvolený algoritmus pre hľadanie minimálnej kostry ju vždy nájde. Ak ju nenájde, znamená to, že nebola splnená podmienka súvislosti grafu, teda bolo prerušené niektoré spojenie (v grafe reprezentované hranou) medzi smerovačmi (v grafe reprezentované vrcholom).

### Úloha č.3:

Predpokladajme, že všetky stanice i smerovače v sieti pracujú bezchybne. Je možné, aby paket prišiel do zlej cieľovej stanice ? Ukážte príklad.

Keďže stanice a smerovače pracujú bezchybne, na to, aby došlo k nesprávnemu doručeniu paketu, musí nastať chyba pri prenose dát, alebo musí dôjsť k narušeniu bezpečnosti siete. Doručenie paketu do zlej cieľovej stanice spôsobené poškodením paketu pri prenose je možné len vtedy, ak je zmenená cieľová IP adresa paketu uvedená v jeho hlavičke (inak by paket nebol doručený vôbec). Keďže však hlavička obsahuje kontrolný súčet celej hlavičky, chyba paketu musí byť taká, aby sedel tento kontrolný súčet. Narušenie bezpečnosti môže nastať na viacerých miestach a rôznymi spôsobmi, napr. odpočúvaním komunikácie.

### Úloha č.4:

ARP aj RARP priraďujú adresy z jedného adresovacieho priestoru do druhého. Funkciou sú teda veľmi podobné. Čím sa líšia v implementácii ? Obidva protokoly poskytujú služby sieťovej vrstve. Sú teda protokolmi spojovej vrstvy ? Odpovede zdôvodnite !

Pomocou ARP protokolu sa zisťuje hardwarová (ďalej len HW) adresa odpovedajúca nejakej IP adrese počítača. ARP request obsahuje typ žiadosti 3, HW adresu a IP adresu odosielateľa requestu a IP adresu cieľovej stanice. Stanica s hľadanou IP adresou odpovedá pomocou ARP response, ktorý obsahuje to, čo ARP request doplnený o HW adresu cieľovej stanice, pričom typ žiadosti je zmenený na 2. RARP protokol slúži bezdiskovým počítačom na zisťovanie ich IP adresy. Tu však neodpovedá stanica s nejakou hľadanou adresou, ale poverený server (resp. iný uzlový počítač, ktorý voči tomuto počítaču vystupuje ako server IP adries). Bezdiskový počítač po zavedení operačného systému pri štarte nepozná svoju IP adresu, pozná len svoju HW adresu. Preto si pri štarte systému zistí pomocou RARP svoju IP adresu. RARP request obsahuje typ žiadosti 3 a HW adresu bezdiskového počítača. Server odpovedá prostredníctvom RARP response, ktorý obsahuje typ žiadosti 4, HW adresu a IP adresu bezdiskového počítača, ktorý poslal RARP request, a HW adresu a IP adresu servera, ktorý odpovedá.

Protokoly ARP a RARP sú protokolmi sieťovej vrstvy. Nie sú protokolmi spojovej vrstvy, pretože IP adresa je definovaná až v sieťovej vrstve. Teda na jej získanie (resp. získanie hardwarovej adresy podľa IP adresy) je potrebné využiť nie len služby spojovej vrstvy ale aj sieťovej.

### Úloha č.5:

Popíšte položky smerovacej tabuľky smerovača v sieti Internet a postup, ako podľa tejto tabuľky smerovač rozhodne o ďalšom smerovaní IP paketu.

Smerovacia tabuľka smerovača obsahuje tieto položky: cieľová sieťová IP adresa, maska cieľovej siete, identifikácia rozhrania, adresa najbližšieho smerovača v príslušnom smere, metrika (resp. cena cesty, či počet preskokov). Tabuľka môže obsahovať aj ďalšie položky.

Smerovač rozhoduje o smerovaní IP paketu nasledujúcim spôsobom. IP adresu cieľovej stanice, ktorá je uvedená v hlavičke paketu, vynásobí po bitoch logickou operáciou AND s maskou cieľovej siete prvého záznamu smerovacej tabuľky. Výsledok porovná so sieťovou IP adresou prvého záznamu. Ak sa nezhodujú, opakuje postup s ďalším záznamom, až kým nenájde príslušnú sieťovú IP adresu. Ak prejde všetkými záznamami a nenájde odpovedajúci, pošle paket na defaultne nastavený smerovač. Ak nájde príslušnú cieľovú sieťovú IP adresu, podľa identifikácie k nej odpovedajúceho rozhrania pošle paket do daného rozhrania.

### Úloha č.6:

Siete A, B, C, D, E, F, prepojené smerovačmi, používajú na obnovovanie smerovacích tabuliek algoritmus smerovania podľa vektora vzdialeností. Počiatočná hodnota vektora v smerovači siete A je (0,2,1,3,15,15) (teda A je priamo spojená so sieťami B, C, D s oneskorením - vzdialenosťou - 2, 1, resp. 3, s E a F nemá priame spojenie). Smerovač v sieti A postupne dostáva vektor (8,0,6,9,4,2) od B, vektor (3,5,0,2,9,4) od C a vektor (6,9,8,0,6,9) od D. Ako sa mení jeho smerovacia tabuľka ?

Počiatočná vektora smerovača A je (0, 2, 1, 3, 15, 15). Keď smerovač dostane od iného smerovača vektor, jeho zložky zvýši o hodnotu spojenia medzi ním a smerovačom, ktorý vektor poslal. Keď smerovač A dostane vektor (8, 0, 6, 9, 4, 2) od smerovača B, zvýši jeho položky o 2, čo je hodnota spojenia medzi nimi (medzi smerovačom A a B). Dostane vektor (10, 2, 8, 11, 6, 4) a ten porovná so svojím vektorom. Keďže nenájde nové položky, nepridáva žiadne nové záznamy. Ďalej porovná jednotlivé položky (i-tu zložku jedného vektora s i-tou zložkou druhého vektora). Ak nájde hodnotu, ktorá je menšia ako v počiatočnom vektore (smerovača A), opraví záznam odpovedajúci danej hodnote, keďže našiel kratšiu cestu. Teda nový vektor smerovača A je (0, 2, 1, 3, 6, 4). Po porovnaní s vektorom smerovača C (3, 5, 0, 2, 9, 4) sa vektor smerovača A nemení, pretože žiadna zložka vektora (4, 6, 1, 3, 10, 5) nie je menšia ako zložka vektora smerovača A, rovnako po porovnaní s vektorom smerovača D. výsledný vektor smerovača A je (0, 2, 1, 3, 6, 4).

### Úloha č.7:

Smerovače A,B,C,D majú počiatočné údaje o vzdialenostiach spojení AB=11, AC=2, AD=7, BD=3, CD=1. Iné spojenia neexistujú. Ako sa bude postupne meniť tabuľka vzdialeností v A ? Akú hodnotu dosiahne ? Čo sa stane, ak sa po čase preruší spojenie medzi BD ? Čo sa stane a ako sa budú meniť tabuľky v A, keď sa po čase preruší aj spojenie AB ?

Šírenie vektorov vzdialeností medzi smerovačmi A, B, C, D a ich zmeny popíšeme nasledujúcou tabuľkou. V kroku 0 sú zapísané počiatočné hodnoty vektorov vzdialeností. Odosielateľ označuje smerovač, ktorý posiela svoj vektor smerovaču označenému ako prijímateľ. V stĺpcoch A-D sú uvádzané vektory vzdialeností smerovačov A-D po prijatí vektorov od odosielateľov. Spôsob výpočtu vektorov je naznačený v predošlom príklade. Tabuľka zachytáva šírenie vektorov vzdialeností do momentu, kedy sa už vektory vzdialeností prestanú meniť (ďalšie posielanie vektorov medzi smerovačmi neovplyvní ich smerovacie tabuľky). Zmeny vektora vzdialeností závisia od toho v akom poradí smerovač A dostane jednotlivé vektory. Výsledný vektor to však neovplyvní.



Po prerušení spojenia medzi smerovačmi B a D sú ich vektory vzdialeností (6, 0, 4, 15) a (3, 15, 1, 0). Po troch (resp. štyroch) cykloch sa nepotvrdí údaj o spojení medzi B a D, preto sa začnú meniť údaje o vzdialenostiach medzi smerovačmi. Výsledný vektor vzdialeností smerovača A je (0, 11, 2, 3). Po prerušení spojenia aj medzi A a B sa postupuje rovnako, výsledný vektor vzdialeností smerovača A je
(0, 15, 2, 3).

### Úloha č.8:

Prečo je potrebné členiť sieť Internet na Autonómne systémy ? Prečo nestačí rozčlenenie na podsiete (broadcastové domény), keď sa smeruje aj tak podľa adries podsietí ?

Autonómny systém je systémom broadcastových domén, ktoré používajú len jeden IGP protokol (Interior Gateway Protocol) pre smerovanie IP paketov smerovačmi. IGP protokolom je zvyčajne buď RIP alebo OSPF protokol. Takéto autonómne systémy sú potrebné pre ohraničenie podsietí, v rámci ktorých je používaný rovnaký IGP protokol. Ak by nebolo takéto ohraničenie, nastali by problémy na miestach (smerovačoch), kde by sa stretávali rôzne smerovacie protokoly. Autonómne systémy sú prepojené border routerami, ktoré používajú EGP protokol (Exterior Gateway Protocol). EGP protokol rieši problém smerovania IP paketov medzi autonómnymi systémami, ktoré môžu používať rôzne IGP protokoly. V EGP protokole môžu byť definované aj obmedzenia na rýchlosť a priepustnosť spojenia. Autonómne systémy zvyčajne vytvárajú providery (jeden provider môže mať aj viac autonómnych systémov), takže autonómny systém môžeme chápať aj ako ohraničenie podsiete jedného providera (resp. viacerých podsietí jedného providera).

### Úloha č.9:

Aký je význam protokolu UDP ? Nestačilo by v prípade nespojovanej služby rovno použiť IP paket ?

UDP protokol je protokol transportnej vrstvy zabezpečujúci bezspojovú komunikáciu.

V prípade nespojovej služby nestačí použiť len IP paket, pretože v jeho hlavičke nie je vyhradené miesto pre určenie čísel portov. Čísla portov sú potrebné na jednoznačné určenie komunikujúcich procesov.

### Úloha č.10:

Predpokladajme, že v štádiu nadväzovania spojenia použijeme len dvojfázové potriasanie rúk (two-way handshake) - teda po prvých dvoch SYN segmentoch považujeme spojenie za nadviazané. Uvážte možnosti uviaznutia (deadlocku) v tomto prípade. Uveďte príklad na uviaznutie alebo dokážte, že nenastane.

Séria H

### Úloha č.1:

Objasnite význam a použitie položky TOS (type of services) typu služby v záhlaví IP paketu.

Prvé 3 bity položky TOS označujú prioritu IP paketu (od 0 po 7), ostatných 13 bitov označuje D/T/R, teda oneskorenie (Delay), priepustnosť (Throughput) a spoľahlivosť (Reliability). Podľa D/T/R sa smerovač môže rozhodovať pri hľadaní cesty, avšak mnohé smerovače túto položku ignorujú.

### Úloha č.2:

Charakterizujte postup dynamického vytvárania smerovacej tabuľky smerovača, využívajúci protokol RIP (Routing Information Protocol).

RIP protokol využíva dynamické smerovanie pomocou vektora vzdialeností. Smerovače autonómneho systému si medzi sebou posielajú po uplynutí určitého časového intervalu svoje vektory vzdialeností. Keď smerovač (označme tento smerovač ako A) dostane vektor vzdialeností od iného smerovača (označme tento smerovač ako B), hľadá najprv údaje o prepojení, ktoré nemá vo svojej smerovacej tabuľke. Ak nájde takéto prepojenie, teda IP adresu siete, ktorú nemá vo svojej smerovacej tabuľke, zapíše si záznam o tomto prepojení: IP adresu siete, masku siete, rozhranie, v ktorom sa nachádza smerovač B (pretože na toto rozhranie potom posiela pakety), IP adresu smerovača B (pretože ten je aktuálne najbližším smerovačom v tomto smere) a ako údaj o metrike zapíše údaj o metrike z vektora vzdialeností B zvýšený o hodnotu metriky prepojenia medzi týmito dvoma smerovačmi A a B. Ďalej smerovač porovnáva hodnoty svojho vektora vzdialeností s vektorom vzdialeností smerovača B, ktorého jednotlivé položky zvýši o vzdialenosť medzi týmito dvoma smerovačmi A a B. Ak nájde údaj, ktorý je menší, opraví príslušný záznam vo svojej tabuľke – zapíše novú hodnotu metriky, a ak je potrebné, aktualizuje rozhranie a IP adresu najbližšieho smerovača v tomto rozhraní na rozhranie a IP adresu smerovača B. Takýmto spôsobom sa dynamicky pridávajú a menia záznamy v smerovacích tabuľkách smerovačov používajúcich RIP protokol.

### Úloha č.3:

V záhlaví TCP paketu sa vyskytuje položka potvrdzovacie číslo (acknowledgement number). Čo budeme do nej zapisovať a ako sa využije v cieľovej stanici ? Ako súvisí toto číslo s veľkosťou posuvného okna ?

Stream oktetov, prostredníctvom ktorého sa uskutočňuje komunikácia medzi dvoma procesmi (príp. dvoma stanicami) sa čísluje sekvenčným číslom, teda každý oktet z tohto streamu má svoje poradové číslo. Potvrdzovacie číslo označuje sekvenčné číslo prvého oktetu streamu, ktorý proces očakáva od procesu, s ktorým komunikuje. Týmto potvrdzovacím číslom taktiež proces potvrdzuje prijatie oktetov s nižšími sekvenčnými číslami. Cieľový proces (resp. cieľová stanica) toto číslo využije na nastavenie posuvného okna metódy sliding window a posiela potom oktety číslované od tohto potvrdzovacieho čísla.

Pri nadväzovaní spojenia potvrdzovacie číslo uvedené v druhom a treťom TCP segmente potvrdzuje len jeden oktet. Súvisí to s tým, že prvý a druhý TCP segment, ktoré potvrdzujú tieto čísla, ešte neobsahujú dáta. Sú to segmenty slúžiace len na nadviazanie spojenia a jeho synchronizáciu. Preto ostatné oktety hlavičky nie je pri nadväzovaní spojenia potrebné číslovať.

Potvrdzovacie číslo nesúvisí s veľkosťou posuvného okna, súvisí s jeho umiestnením.

### Úloha č.4:

Popíšte postup nadväzovania spojenia protokolom TCP, popíšte obsah a význam údajov v prvých dvoch segmentoch komunikácie.

Nadväzovanie spojenia protokolom TCP prebieha prostredníctvom trojfázového handshake-u. Majme dva procesy A a B. Proces A nadväzuje spojenie s procesom B. Hlavička prvého TCP segmentu, ktorý pošle proces A procesu B bude obsahovať:

* číslo portu procesu A, ktoré tento proces identifikuje (Source Port);
* číslo portu procesu B, ktoré tento proces identifikuje (Destination Port);
* sekvenčné číslo, od ktorého sa začína číslovať stream oktetov, ktoré posiela proces A procesu B;
* ľubovoľné potvrdzovacie číslo, pretože teraz nie je ani potrebné ani platné, keďže ACK=0;
* dĺžku TCP hlavičky;
* príznaky:
	+ URG=0, pretože tento segment ešte nemôže obsahovať žiadne dáta, ani dôležité;
	+ ACK=0, nepotvrdzujú sa žiadne prijaté oktety, keďže toto je prvý segment spojenia;
	+ PSH=0, keďže žiadne dáta nemôžu byť vo vyrovnávacej pamäti pre toto spojenie;
	+ RST=0, lebo toto je začiatok spojenia, spojenie ešte neexistuje, preto nie je možné ho už ukončiť;
	+ SYN=1, pretože sa jedná o synchronizáciu spojenia;
	+ FIN=0, podobne ako pri RST;
* Window Size a Urgent Pointer nie sú ešte platné položky;
* Checksum, teda kontrolný súčet segmentu.

Prvý a druhý TCP segment handshake-u neobsahuje žiadne dáta, sú to len synchronizačné segmenty, ktoré nastavujú parametre spojenia. Druhý segment, ktorý posiela proces B procesu A obsahuje:

* číslo portu procesu B (Source Port);
* číslo portu procesu A (Destination Port);
* sekvenčné číslo, od ktorého sa začína číslovať stream oktetov, ktoré posiela proces B procesu A;
* potvrdzovacie číslo rovné sekvenčnému číslu procesu A zvýšené o 1, čo znamená, že proces B potvrdzuje prijatie prvého synchronizačného TCP segmentu procesu A;
* dĺžku TCP hlavičky;
* príznaky:
	+ URG=0, pretože tento segment ešte nemôže obsahovať žiadne dáta, ani dôležité;
	+ ACK=1, nepotvrdzujú sa žiadne prijaté oktety, keďže toto je prvý segment spojenia;
	+ PSH=0, keďže žiadne dáta nemôžu byť vo vyrovnávacej pamäti pre toto spojenie;
	+ RST=0, pokiaľ proces B nechce ukončiť nadväzovanie spojenia s procesom A;
	+ SYN=1, pretože sa jedná o synchronizáciu spojenia;
	+ FIN=0, podobne ako pri RST;
* veľkosť posuvného okna (Window Size) metódy sliding window, ktorá označuje aký maximálny počet oktetov môže poslať proces A procesu B;
* Urgent Pointer nie je platná položka, kvôli príznaku URG;
* Checksum, teda kontrolný súčet segmentu.

Tretí TCP segment, ktorý posiela proces A procesu B, už nemá príznak SYN=1, preto už môže obsahovať aj dáta. Tento segment potvrdí predošlý synchronizačný segment potvrdzovacím číslom(rovné sekvenčnému číslu procesu B zvýšenému o 1) a nastaví veľkosť posuvného okna pre proces A.

### Úloha č.5:

Akým spôsobom sa realizujú služby relačnej vrstvy (session level) referenčného modelu OSI v sieti Internet ?

Služby relačnej vrstvy referenčného modelu OSI v sieti internet zabezpečuje transportný protokol TCP. TCP je protokol pre komunikáciu so spojením. A práve toto spojenie je vlastne relácia. Identifikácia relácií sa zabezpečuje prostredníctvom identifikácie procesov, teda číslami portov. Vytváranie relácií zodpovedá nadväzovaniu komunikácie, ktoré sa vykonáva trojfázovým handshake-om. Ukončovanie relácií sa realizuje buď pomocou príznaku RST v hlavičke TCP segmentu, alebo pomocou príznaku FIN a následnými potvrdeniami. Správa kontrolných bodov relácií je čiastočne realizovaná mechanizmom sliding window, ktorý vyžaduje potvrdzovanie streamu prijatých oktetov.

### Úloha č.6:

Čo je doménové meno a kde sa uchovávajú informácie o doménových menách? Môžem sa pripojiť do Internetu pokiaľ nemám doménové meno ? Ako možno doménové meno získať ? Ako prebehne zisťovanie IP adresy z doménového mena rekurzívnym spôsobom ?

Doménové meno je symbolická adresa používaná na aplikačnej úrovni. Informácie o doménových menách sa uchovávajú na DNS (Domain Name System) serveroch príslušných domén. Bez doménového mena je možné sa pripojiť do Internetu, pretože doménové mená sú len logickou adresáciou a nie sú nutné pre komunikáciu, na tú postačí IP adresa. Doménové meno je možné získať registráciou, ktorú zabezpečujú organizácie ako napr. IANA.

DNS server, ktorý dostane rekurzívnu žiadosť o priradenie IP adresy k doménovému menu, postupuje nasledovným spôsobom: Najprv zistí, či má vo svojej vyrovnávacej (cache) pamäti toto priradenie. Ak nemá, hľadá IP adresu podľa doménového mena. Ak je toto doménové meno v jeho doméne, má informáciu o IP adrese k nemu prislúchajúcej vo svojej tabuľke. Ak je hľadané doménové meno v jeho poddoméne, posiela žiadosť DNS serveru, ktorý je v tejto poddoméne. Ak nie je doménové meno ani v cache pamäti, ani v doméne daného DNS servera, ani v niektorej jeho poddoméne, posiela žiadosť nadradenému DNS serveru. Takto sa postupne posielajú žiadosti až pokým sa nenájde príslušné priradenie.

### Úloha č.7:

Čo je to reverzná doména a na čo slúži? Uveďte príklad adresovania pomocou reverzného doménového mena. Kde sa uchovávajú informácie o reverzných doménových menách? Musí mať každá stanica v Internete aj reverzné doménové meno?

Na reverzný preklad, teda zistenie doménového mena prislúchajúceho IP adrese slúži strom IP adries podobný stromu doménových mien. Top level doména tohto stromu sa nazýva arpa. Pod ňou je definovaná doména in-addr. Pod ňou sa potom nachádzajú domény pomenované prvým číslom zo štvorice čísel tvoriacich IP adresu. Takto je vytvorený strom domén nazývaných reverznými doménami.

Príkladom reverzného doménového mena je pre IP adresu 158.197.196.9 reverzné doménové meno 9.196.197.158.in-addr.arpa .

Reverzné doménové mená sú uložené na serveroch organizácií, ktoré prideľujú IP adresy. Teda organizácia, ktorá pridelila IP adresu, spravuje reverzné doménové meno k nej prislúchajúce.

Každá stanica v Internete nemusí mať reverzné doménové meno, ale potom jej nebudú poskytnuté niektoré služby alebo nebudú fungovať správne (napr. FTP).

### Úloha č.8:

Binárny súbor má veľkosť 2540 bytov. Posielame ho elektronickou poštou, pričom dĺžka jedného riadka nemôže presiahnuť 76 znakov. Aká bude veľkosť súboru po zakódovaní systémom Base64 ? Aká by bola jeho približná veľkosť po zakódovaní spôsobom Quoted-printable za predpokladu rovnomernej pravdepodobnosti výskytu hodnôt bytov v pôvodnom súbore ?

### Úloha č.9:

Aký je rozdiel medzi pasívnym a aktívnym FTP prenosom ? Prečo sú aktívne FTP prenosy z pohľadu bezpečnosti citlivejšie ako pasívne ? Akým spôsobom umožniť klientom vnútornej siete pomocou rozšíreného filtra komunikáciu s vonkajšími FTP servermi len cez pasívne FTP prenosy ? Akým spôsobom umožniť komunikáciu protokolom Telnet pre klientov vnútornej siete von a zakázať používať Telnet spojenie pre vonkajších klientov do vnútra siete ?

### Úloha č.10:

Čo sú zástupné (proxy) servery a aké je ich využitie vo vytváraní bezpečnostných brán (firewallov)?

Zástupné (proxy) servery sú umiestnené na rozhraní lokálnej siete (vnútornej siete) s Internetom (vonkajšou sieťou). Ich úlohou je sprostredkovať komunikáciu na aplikačnej úrovni medzi klientom vnútornej siete a vonkajšou sieťou. Klient nekomunikuje priamo s vonkajšou sieťou, ale s proxy serverom. Ten potom komunikuje s vonkajšou sieťou. Preto sú pre vonkajšou sieť klienti vnútornej siete ukrytí. Táto vlastnosť zástupných serverov sa využíva pri vytváraní firewallov a taktiež pri vlastnej adresácii počítačov vo vnútornej sieti. Na proxy serveroch môžu byť na spojovej úrovni filtre, ktoré prepúšťajú len pakety povolenej komunikácie a zvyšujú tak bezpečnosť komunikácie. Taktiež môžu zabezpečovať ďalšie obmedzenia, ako napr. zakázať sprostredkovanie prístupu klientov do niektorých lokalít (resp. na niektoré WWW stránky) a zakázať niektoré ďalšie služby. Zástupné servery majú vlastnú cache pamäť, do ktorej si na istý čas ukladajú prenášané dáta. Ak opätovne príde požiadavka na tieto dáta, nesťahujú ich z Internetu, ale ich poskytujú zo svojej cache. Čím urýchľujú komunikáciu a znižujú objem prenášaných dát.