9. Relačná a transportná vrstva

Čo by mal študent vedieť:

- ✓ funkcie relačnej vrstvy (vytvorenie relácie, riadenie dialógu, synchronizácia)
- ✓ služby relačnej vrstvy (polo duplex, plný duplex)
- ✓ protokoly relačnej vrstvy (RPC, SIP, WSP)
- ✓ funkcie transportnej vrstvy (multiplexácia dát aplikačných protokolov cez čísla portov (port numbers), riadenie toku, zabezpečenie, adresovanie)
- ✓ transportné služby (zabezpečený prenos, nezabezpečený prenos)
- ✓ protokoly transportnej vrstvy (UDP, TCP, RTP)
- ✓ využitie transportných protokolov
- ✓ spoluprácu transportnej vrstvy so sieťovou vrstvou (príprava dát na prenos po sieti)

Relačná vrstva (session layer)

Relačná vrstva je piatou vrstva modelu vrstvové sieťovej architektúry (OSI). Poskytuje služby prezentačnej vrstve tým, že poskytuje mechanizmus správy dialógu medzi aplikačnými procesmi koncového používateľa. Zároveň využíva služby transportnej vrstvy pri nadväzovaní a ukončovaní relácie *(session).*
Relačná vrstva je v ISO/OSI vrstvou, ktorá má toho má relatívne „najmenej na práci“. Pôvodná predstava autorov bola taká, že v tejto vrstve budú sústredené funkcie, ktoré uľahčujú a podporujú vzájomnú interakciu komunikujúcich strán označovanú relácie (*sessions).* Bolo to hlavne kvôli zaisteniu bezpečnosti interakcie, ako aj v zmysle podpory transakčného spracovania, kde je dôležité, aby žiadna transakcia nebola vykonaná len z časti, ale vždy úplne celá, alebo sa vôbec nerealizovala. Príkladom môže byť nejaká finančná transakcia, spočívajúca v odpísaní určitej čiastky z jedného účtu, a pripísanie tejto čiastky na účet iný. Určite by nebolo správne keby prebehla len jedna časť transakcie - odpísanie čiastky z jedného účtu a ostatné časti, napríklad pripísanie na iný účet, by sa už nevykonali. Na tomto príklade je zrejmá hlavná myšlienka potreby relačnej vrstvy, ale vzniká aj otázka, či je potrebné riešiť relačnú vrstvu pre všetky aplikácie rovnako alebo je vhodnejšie, aby si ich aplikácie riešili samé, podľa svojich vlastných predstáv. Je zrejmé, že napríklad bankové aplikácie budú mať podstatne prísnejšie požiadavky na "kvalitu" zaistenia transakcií, ako jednoduchšie databázové aplikácie, kde nehrozia také veľké následky pri nekorektnom prevedení transakcie. Preto je relačná vrstva riešená v rôznych technológiách rôzne. V technológii TCP/IP nie je táto vrstva vôbec uvažovaná, ale jej funkcie sú špecifikované v aplikáciách.

9.1 Relácia

Pojem **relácia** *(session)* je používaný v rozhlase alebo v televízii ako ucelená programová jednotka. Iný význam tohto pojmu je súvislosť niečoho s niečím alebo spojenia niečoho s niečím.

Význam tohto slova možno nájsť na http://sk.wikipedia.org/wiki/Rel%C3%A1cia. Podľa tohto zdroja je jedno z vysvetlení nasledovné: „v interaktívnych a transakčných systémoch jedna elementárna akcia používateľa systému spočívajúca v zadaní požiadavky a prevzatí výsledku, po anglicky session“. Táto špecifikácia je najpríbuznejšia komunikačným systémom, kde relácia na úrovni relačnej vrstvy je vytvorenie spojenia medzi dvomi koncovými používateľmi, ktorý si vymieňajú informácie. **Relácia** je tak každé spojenie, ktoré je zaisťované prostredníctvom jedného spojenia na transportnej vrstve. Treba tu však poznamenať, že jedno transportné spojenie môže zabezpečovať aj viac po sebe idúcich relácií.

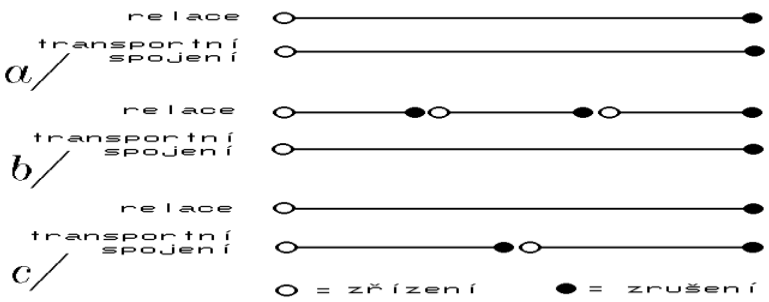
9.3. Funkcie relačnej vrstvy

Aby mohla byť splnená úloha relačnej vrstvy, ktorou je vytvárať, organizovať a synchronizovať dialóg medzi spolupracujúcimi relačnými vrstvami obidvoch systémov a riadiť výmenu dát medzi nimi, je potrebné zabezpečiť určité technické funkcie, ktoré budú túto úlohu zabezpečovať. Základné **funkcie relačnej vrstvy** sú:

- **nadviazanie, udržovanie a rušenie relácií**
- **riadenie dialógu** – zaistenie pravidelného striedania komunikačných systémov pri vysielaní
- **synchronizácia** – zaistenie súladu medzi dátami vysielanými odosielateľom a prijímanými príjemcom

9.3.1. Nadviazanie, udržovanie a rušenie relácií

Relačná vrstva vytvára používateľom na rôznych komunikačných systémoch spojenie. Spojenie umožňuje prenos dát, ako aj transportná vrstva, ale poskytuje aj rozšírené služby potrebné pre niektoré aplikácie. Vytvorenie spojenia je analogické vytvoreniu spojenia v telefónnej službe. Voľba telefónneho čísla je analogická transportnému spojeniu a vedenie rozhovoru je relácia medzi dvomi účastníkmi. Relácia je tak spojenie na úrovni vyššej, než je transportná vrstva. Spravidla je každé takéto **relačné spojenie** /relácia zaisťovaná jedným transportným spojením, ktoré je zriaďované a rušené pri zriaďovaní a rušení relácie, obr. 9.1.a. Sú však možné aj iné prípady. V obr. 9.1.b. jedno transportné spojenie zaisťuje dve alebo viac relácií. Tu je možné tiež použiť analógiu s telefónnym hovorom. Táto situácia odpovedá tomu, že dvaja účastníci telefónneho hovoru dokončia svoj rozhovor a namiesto zrušenia spojenia, odovzdajú telefón inej dvojici, ktorá môže začať nový rozhovor/reláciu.
Tretím prípadom relačného spojenia je prípad, keď dôjde k výpadku transportného spojenia a relačná vrstva zaistí pokračovanie relácie prostredníctvom nového transportného spojenia, obr. 9.1.c. V telefónnej analógii to odpovedá situácii, keď je prerušený hovor a účastníci musia znovu obnoviť spojenie, aby dokončili rozhovor.
Rozdiel od transportného spojenia je aj spôsob ukončenia relácie. V prípade transportného spojenia sú poskytované prostriedky pre jednostranné direktívne ukončenie spojenia, ktorému druhá strana nemôže zabrániť. Na úrovni relačnej vrstvy sa však predpokladá ukončenie na základe vzájomnej dohody. Jedna strana dá návrh na zrušenie spojenia a druhá strana má možnosť odmietnuť a zaistiť tak pokračovanie relácie.



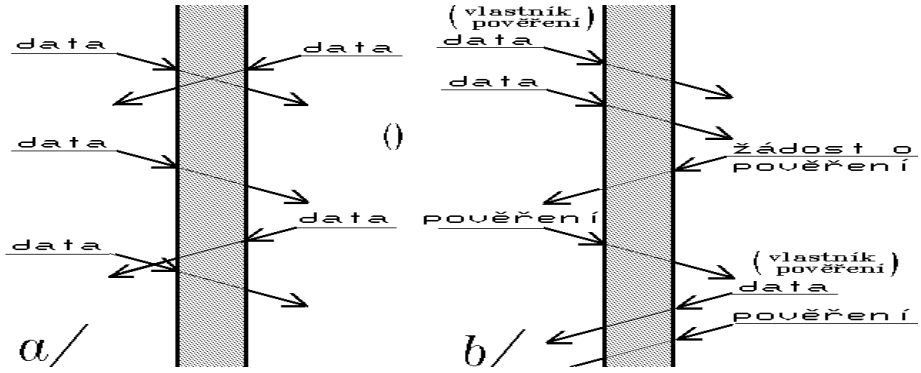
Obr. 9.1 Vzťah relácie a transportného spojenia

Na relačnej vrstve sú oznamované aj výnimočné stavy, ktoré riešia výnimočné situácie. Takúto situáciu rieši napríklad pri web službe hláška „Chcete obnoviť reláciu“, ak bola relácia z nejakých príčin zrušená.

9.3.2. Riadenie dialógu – zaistenie pravidelného striedania uzlov pri vysielaní

Ďalšou funkciou relačnej vrstvy je riadenie dialógov medzi oboma koncovými účastníkmi. Tak ako pri telefónnom rozhovore nie je možné, aby obaja účastníci hovorili súčasne, existujú aj v počítačových sieťach také aplikácie, ktoré vyžadujú koordinované striedanie oboch zúčastnených pri vysielaní. Sú dôležité pre aplikácie kde komunikujúce strany nepristupujú k rovnakej operácii v rovnakom čase a pri rôznych aplikáciách transakčného charakteru.
Riadenie dialógu umožňuje prevádzku oboch smeroch v rovnakom čase, alebo len jedným smerom v rovnakom čase. Ak je prevádzka iba jedným smerom v čase, tak relačná vrstva udržuje prenosovú cestu po ktorej pôjde prenos späť.
Relačná vrstva túto požiadavku zaisťuje pomocou mechanizmu/metódy odovzdávania **poverenia** k prenosu dát označovaného ***(data token).*** Na spravovanie týchto aktivít relačná vrstva poskytuje tokeny, ktoré môžu byť vymieňané. Len strana, ktorá má token môže vykonávať potrebné operácie. Vysielať dáta môže vždy len ten, kto vlastní poverenie/ token, obr. 9.2. Relačná vrstva pritom ponúka prostriedky, pomocou ktorých sa dá poverenie odovzdať, alebo si ho vyžiadať.

.



Obr. 9.2 Vzájomné komunikácia v rámci relácie

Na úrovni relačnej vrstvy rozlišujú tri spôsoby vedenia dialógu:

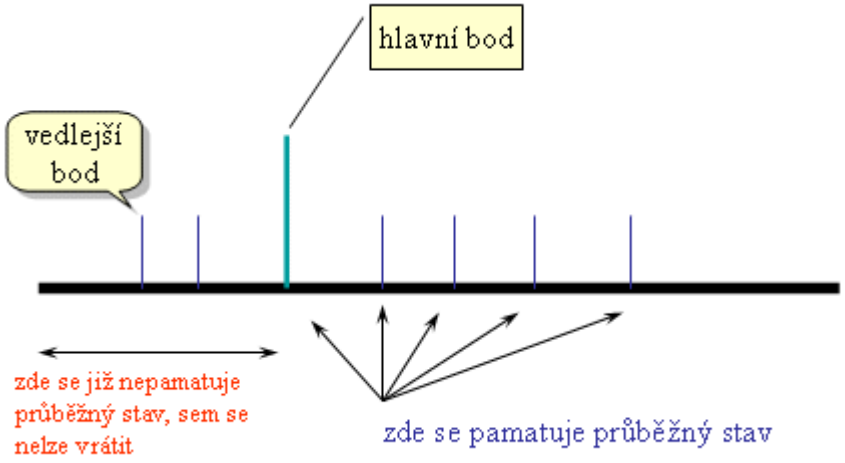
- plne duplexné *(full duplex, Two-Way-Simultaneous, TWS)*
- polo duplexné (*half duplex, Two-Way-Alternate TWA)*
- simplexné *(simplex, One-Way).*

Mechanizmus odovzdávania poverenia sa pritom týka len polo duplexného dialógu.Treba poznamenať, **že spôsob vedenia dialógu nemá nič spoločného s duplexným, polo duplexným alebo simplexným charakterom prenosového kanála**. Všetky spojenia v referenčnom modelu ISO/OSI sú zvyčajne plne duplexné a umožňujú súčasný prenos dát oboma smermi. Rôzne aplikácie však z rôznych dôvodov nemusia túto možnosť využívať a po plne duplexnom spojení môžu vytvoriť len polo duplexný dialóg.

9.3.3. **Synchronizácia** *(synchronization)*

Ďalšia služba relačnej vrstvy je synchronizácia. Jej význam si možno predstaviť v situácii prijímania dát na počítač , kde sú dáta zapisované, alebo sú hneď tlačené na tlačiareň. Ak dôjde k poruche zápisu na disk alebo nie je v tlačiarni papier, môže príjemca prísť o časť prenášaných dát, ktoré boli transportnou vrstvou prenesené. Aby sa vyhlo problému poskytuje relačná vrstva spôsob opakovania dát len od určitého bodu.

Mechanizmus použitý pri synchronizácii je nazývaný checkpointing a predstavuje vkladanie kontrolných bodov (check-point) do prenášaných dát. Tým umožňuje relačná vrstva možnosť: vrátiť sa ku určitému "kontrolnému bodu" a pokračovať od neho a nie od začiatku. Takýchto kontrolných bodov môže existovať viac a v prerušenej relácii je možné pokračovať od ktoréhokoľvek z nich. Jednotlivé kontrolné body sú akési body zotavenia, v rámci ktorých je uschovaný celý stav práve prebiehajúcej relácie, vrátane doposiaľ prenesených dát. Situáciu naznačuje nasledujúci obrázok 9.3.



Obr. 9.3 Zobrazenie hlavného a vedľajších kontrolných bodov

Prenášané dáta si pamätá odosielateľ. Aby si ich však nemusel pamätať všetky, čo predstavuje veľkú réžiu, rozlišuje relačná vrstva dva druhy kontrolných bodov hlavný *(major)* a vedľajší *(minor).* Rozdiel medzi nimi je vtom, že cez vedľajší sa dá vracať k dátam a cez hlavný nie. Pre vysielajúceho to znamená, že si musí pamätať prenášané dáta od hlavného kontrolného bodu. Staršie kontrolné body sa už tak môžu "uvoľňovať" a nie je možné sa k nim vracať.

Synchronizácia v relačnej vrstve je v úplne inom zmysle, ako je synchronizácia na úrovni fyzickej vrstvy.

9.4. Protokoly relačnej vrstvy

V TCP/IP architektúre nie je relačná vrstva špecifikovaná. Funkcie si zabezpečuje každá aplikácia sama, podľa svojich nárokov a požiadaviek. V tejto súvislosti sú často tvorcovia TCP/IP architektúry chválení za zmenšenie počtu vrstiev, čo však nie je vždy pravda. Mnoho protokolov hlavne pre podporu tradičných služieb cez TCP/IP technológie takúto vrstvu postrádajú a nahrádzajú ju vlastnými podvrstvami v konkrétnych protokoloch.
Tým, že mnohé aplikačné protokoly, ktoré vykonávajú funkcie relačnej vrstvy sú zaraďované do aplikačnej vrstvy vzniká nejednotnosť zaraďovania. To však nebráni tomu, že sa funkcie niektorých protokolov budú vysvetľovať pri rôznych vrstvách OSI modelu.

9.4.1. SIP – Session Initiation Protocol

Do relačnej vrstvy patria také služby „aplikačnej úrovne TCP/IP“, ktoré zaisťuje pre VoIP – Voice over Internet Protocol protokol SIP. Ten má podporu relácií dokonca vo svojom názve, pretože doslova znamená: podpora nadväzovania relácií. Je z oblasti internetovej telefónie a službám na báze VoIP slúži na nadväzovanie telefonických hovorov. Rieši také činnosti, ako je vyhľadanie volaného podľa telefónneho čísla tak, aby ku nemu mohlo byť nadviazané transportné spojenie a následne vedený hlasový hovor.
SIP je protokol pre komunikáciu v reálnom čase a podporuje relácie v rámci najrôznejších aplikácií internetu, ako sú telefónia, konferencie, multimédiá. Podporuje tieto služby:

- lokalizácie používateľa – určenie koncového systému pre danú komunikáciu,
- nadviazanie spojenia – stanovenie parametrov pre volajúceho a aj volanú stranu,
- dostupnosť používateľa – zistenie dostupnosti volanej strany a sledovanie prítomnosti,
- používateľské možnosti – určenie média a jeho parametrov.

SIP neumožňuje:

- manažment interaktívnych relácií po ich nadviazaní
- zaistiť kvalitu službu (QoS), pretože nevie uprednostňovať určitú prevádzku a rezervovať sieťové prostriedky, ale môže spolupracovať s protokolmi, ktoré zaisťujú QoS, napríklad RSVP - Resource ReSerVation Protocol, čo je protokol transportnej vrstvy a umožňuje rezervovať zdroje siete pre **integrované služby *(inregrated service)*** internetu. Integrované služby v internete – **IntServ,** je architektúra, ktorá špecifikuje prvky pre zabezpečenie kvality siete. Viac informácií na http://en.wikipedia.org/wiki/Integrated_services
- nie je protokol určený k prenosu veľkého objemu dát, ako je HTTP, miesto toho prenáša len malý objem dát potrebných pre nadviazanie interaktívnych relácií, okrem toho je ešte schopný prenášať krátke textové správy.

SIP pracuje na princípe klient –server. Medzi koncové body SIP používateľských agentov (obsahujúce klienta UAC, UA Client, a server UAS, UA Server), patria používateľské zariadenia ako SIP telefóny a PC s klientskym softvérom a brány do iných sietí (zvlášť brány pre IP telefóniu). Servery mohou fungovat jako zástupci (proxy), kdy zastupují klienty při předávání požadavků SIP na další server. Servery mohou podporovat přesměrování (redirect), kdy klienta informují o dalším skoku v síti, kam se má zpráva poslat, a klient nebo proxy následně kontaktuje doporučené zařízení sám. Registraci momentálního umístění klientů zpracovávají registrátoři (registrar), kteří informace o uživatelských agentech aktualizují v serveru umístění (location) nebo databázi.

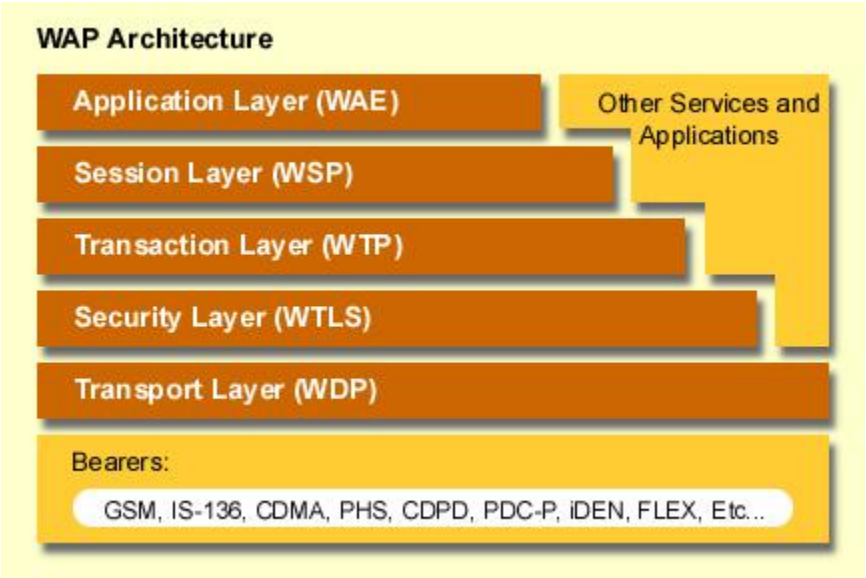
SIP URL mají různé formy (obecně `sip: user@domain`) a mohou obsahovat telefonní čísla.

Příklady: `sip:abc@lupa.cz` - adresa počítače uživatele abc v doméně lupa.cz, `sip:+420-212345678@lupa.cz` – telefonní číslo uživatele dosažitelné prostřednictvím brány.

Podpora jak webové adresace, tak telefonních čísel umožňuje IP komunikaci bez větších problémů přecházet mezi telefonní sítí a Internetem. Uživatelé na kterékoli síti tak mohou komunikovat s kýmkoli na telefonní síti nebo na Internetu, aniž by museli vyměnit svá stávající zařízení. IP zařízení s podporou SIP (telefony, počítače) mohou komunikovat přímo, pokud znají URL druhé strany.

9.4.2. WAP Wireless Access Protocol

Je aplikačný protokol pre bezdrôtové zariadenia. Je to technológia, umožňujúca prezeranie webových stránok, najčastejšie z mobilných telefónov. WAP nie je iba protokol, ale má aj svoju vlastnú architektúru. WAP špecifikuje architektúru založenú na vrstvách, ktorá je veľmi podobná klasickému sieťovému OSI referenčnému modelu. Túto architektúru môžeme prezentovať modelom, ktorý je znázornený nižšie na obrázku a je zložený z piatich vrstiev (prenosová, bezpečnostná, transakčná, relačná a aplikačná). obr. 9.4.



Obr. 9.4 WAP model

Aplikačná vrstva reprezentuje bezdrôtové aplikačné prostredie (WAE – Wireless Application Environment), ktoré priamo podporuje vývoj WAP aplikácií prostredníctvom WML (Wireless Markup Language: Značkovací jazyk v štruktúre podobný HTML.

Relačná vrstva reprezentuje bezdrôtový relačný protokol **WSP (Wireless Session Protocol),** čo je v podstate ekvivalent HTTP pre WAP prehliadače. Podobne ako HTTP, i WAP vyžaduje prehliadač a server (ako napr.: web server), no v tomto prípade komunikácia pomocou HTTP by bola neefektívna, keďže je to protokol navrhnutý pre "drôtovú" komunikáciu.

Výhodou WSP je, že udržiava presnú šírku prenosového pásma na bezdrôtových linkách a predovšetkým pracuje s relatívne kompaktnými binárnymi údajmi, kým HTTP pracuje hlavne s textovými údajmi.

Bezdrôtový relačný protokol (WSP) poskytuje aplikáciám dve spojovacie služby. Prvá je spojovo-orientovaná služba, operujúca nad spoľahlivým transportným protokolom WTP a druhá, nespojovaná služba nad protokolom WDP s negarantovaným doručením.

V základoch je WSP založený na HTTP s nejakými dodatkami a modifikáciou k optimalizácii jeho použitia v bezdrôtových spojoch. Je to teda transakčne -orientovaný protokol, založený na koncepte požiadavka/odpoveď.

WSP tiež definuje tzv. Push operácie servera, ktorými môže server poslať nevyžiadaný obsah ku klientovi. Toto môže byť využívané pre vysielanie správ (messages) alebo pre služby posielajúce napr. prehľad správ (v titulkoch) alebo ceny akcií, ktoré môžu byť prispôsobené pre každé klientské zariadenie.

WSP poskytuje tieto hlavné služby:

- Zavedenie spoľahlivého spojenia od klienta k serveru
- Výmena obsahu medzi klientom a serverom s využitím kompaktného kódovania
- Pozastavenie a obnovenie spojenia
- Prenášanie Push obsahu od servera ku klientovi asynchronickou metódou

9.4.3. RPC (*Remote procedure call)*

RCP, tiež **vzdálené volání procedur** je technológia umožňujúca aplikácii vykonať procedúru, ktorá môže byť uložená na inom mieste ako volajúca aplikácia, na inom počítači v sieti.
Najprv sa identifikátor procedúry aj s parametrami zabalí do formy vhodnej pre prenos - *marshalling*. Následne sa balíček odošle. Po prijatí na vzdialenom mieste sa balíček rozbalí, identifikuje sa procedúra a po vykonaní sa výsledok opäť zabalí a odošle späť.
Existuje mnoho rôznych implementácií tohoto mechanizmu. V databázach Oracle sa bežne používa volanie procedúr z inej databáze prostredníctvom databázových liniek. Medzi známe RPC implementácie patria tiež produkty MSRPC od firmy Microsoft, ktoré boli využité i pri vývoji DCOM, .Net Remoting či Java RMI.

Nevýhodami je, že klient vždy čaká než mu server vráti odpoveď. Vlastné volanie cez sieť môže byť veľmi pomalé, či dokonca vzdialené rozhranie môže byť nedostupné. Vzdialené rozhranie tretej strany môže byť bez varovania zmenené.

9.4.4. Iné protokoly relačnej vrstvy

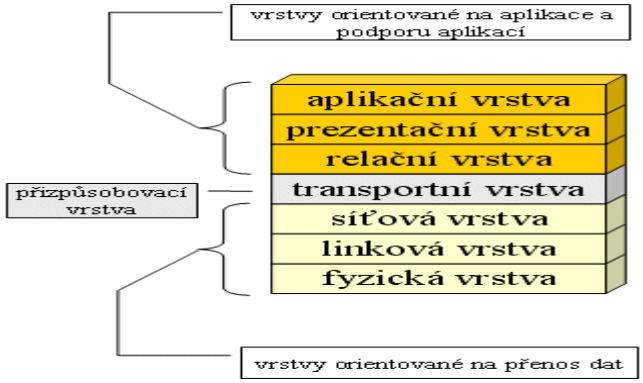
- NetWare Core Protocol (NCP)
- AppleTalk Session Protocol (ASP)
- NetBIOS

Transportná vrstva *(Transport Layer)*

Transportná vrstva je deliacou vrstvou medzi sieťou a koncovým zariadením. Je prvou vrstvou, ktorá nie je vo vnútorných uzloch siete/smerovačoch, ale je v koncových uzloch siete. Pretože jej hlavnou úlohou je zabezpečenie vzájomnej komunikácie koncových uzlov, označuje sa ako ***end-to-end*** komunikácia.
V referenčnom modeli je hlavne preto, aby vyšším vrstvám OSI modelu poskytovala kvalitnejšie prenosové služby, než dokáže poskytovať sieťová vrstva. Poskytuje služby pre relačnú vrstvu, resp. pre funkcie relačnej vrstvy obsiahnuté v aplikačných protokoloch.
Vyššie vrstvy sú tak izolované od špecifík používanej komunikačnej podsiete od jej technológie a od všetkých nedokonalostí siete. Zatiaľ čo aplikačná, prezentačná a relačná vrstva sa zaoberá problematikou aplikácii, nižšie štyri vrstvy majú na starosti záležitosti týkajúce sa prenosu dát.
Okrem základnej úlohy vytvorenia zabezpečenia komunikácie medzi koncovými uzlami siete sú jej úlohy aj zabezpečenie kvality služieb a spoľahlivosti prenosu dát. Možno povedať, že poskytuje službu prenosu dát a „chráni“ vyššie vrstvy pred detailmi realizácie prenosu. Je rozhraním medzi poskytovateľmi služieb prenosu *(bearer services)* a používateľmi všetkých ostatných poskytovaných IK služieb.
Je tiež prvou vrstvou, ktorá v rámci uzlu rozlišuje jednotlivé entity/procesy prostredníctvom portov. Pre nižšie vrstvy je však ďalej nedeliteľným celkom.
Má niektoré funkcie podobné ako sieťová vrstva a jej význam je hlavne v tom, že poskytuje spoľahlivú prenosovú službu.

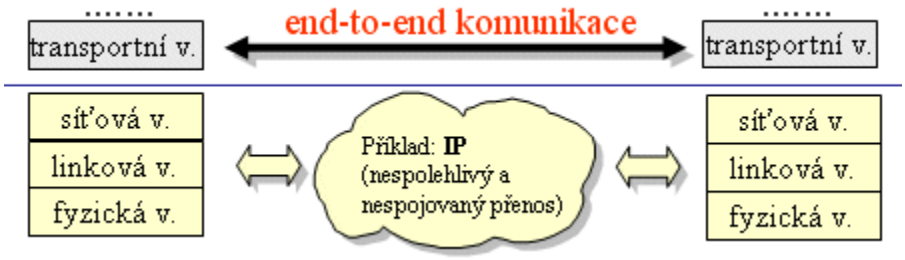
9.5. Umiestnenie transportnej vrstvy v OSI modeli

Na transportnú vrstvu sa môžeme pozerať ako na prispôsobovaciu vrstvu medzi tromi spodnými a tromi vrchnými vrstvami.



Obr. 9.5. Transportná vrstva v OSI modeli

Aj keď úlohou transportnej vrstvy je komunikácia end-to-end ako je znázornené na obrázku, 9.6 musí využívať služby sieťovej vrstvy, ktorá je bezprostredne pod ňou. Tá vyhľadáva vhodné prenosové cesty v sieti cez jednotlivé smerovače. Takouto službou poskytuje transportnej vrstve „ilúziu“, že koncové uzly sú prepojené každý s každým. Preto sa transportná vrstva môže sústrediť iba na komunikáciu koncových uzlov a výmeny dát medzi nimi.



Obr. 9. 6 End-to-end komunikácia

9.6. Funkcie transportnej vrstvy

9.6.1. Zriadenie, udržiavanie a uvoľnenie transportného spojenia

Na transportnej vrstve sa najčastejšie vytvára sa spojovo orientované *(conection oriented)* spojenie. To znamená, že sa najskôr vytvorí spojenie a potom sa začnú prenášať dáta. Vytvorenie spojenia prebieha výmenou správ medzi PC, ktoré chcú komunikovať.

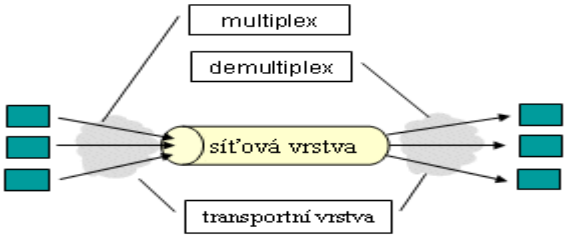
Najskôr sa snaží počítač odosielateľa vytvoriť spojenie tým, že sa pokúsi o synchronizáciu s druhým počítačom, tým že pošle segment s príznakom SYN. Následne sa obojstranne dohodne spojenie medzi počítačmi, ktoré chcú komunikovať. Po dohodnutí spojenia potvrdí počítač príjemcu synchronizáciu poslaním segmentu s príznakom SYN. Nakoniec odosielateľov počítač potvrdzuje vytvorenie spojenia pomocou príznaku ACK a môžu sa začať prenášať dáta.

Spojenie môže ukončiť ktorákoľvek strana a to tým, že odošle TCP segment s príznakom FIN. Toto ukončenie sa nazýva aktívne ukončenie spojenia *(active close).* Čo znamená, že strana, ktorá ho poslala už ďalej nebude odosielať dáta, ale môže stále prijímať.

Pokiaľ odosielateľ nepošle TCP segment s príznakom FIN, tak stále odosiela dáta. Toto spojenie sa nazýva polo uzavreté *(half close*). Keď pošle príznak FIN aj on tak potom sa ukončí spojenie úplne.
animácia

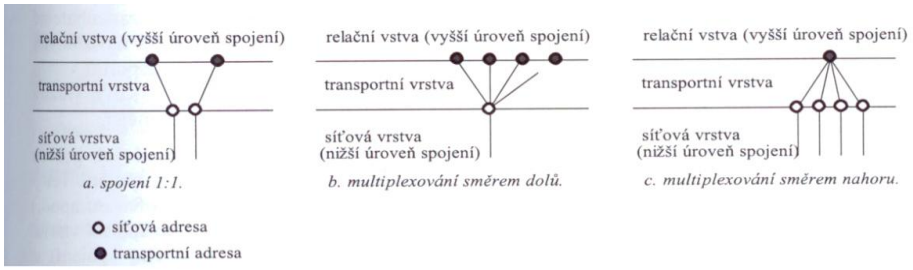
9.6.2. Multiplexovanie na transportnej vrstve

Pre priblíženie funkcií na transportnej vrstve je možné uviesť nasledovný príklad. Na jednom počítači môže byť spustených viac aplikácií, ktoré komunikujú s inými uzlami v sieti nezávisle na sebe. Používateľ môže mať spustený web prehliadač a elektronickú poštu. Prijíma dáta, ale nemusí byť zrejmé komu patria web prehliadaču alebo poštovému klientovi? Sieťová vrstva takýto problém nerieši. Pre ňu je vždy uzol len ako jeden celok. Takže ak prijme dáta pre daný uzol, odovzdá ich transportnej vrstve a tá musí rozlíšiť, komu v rámci daného počítača dáta patria. Podobne je to pri vysielaní, keď transportná vrstva dáta prijíma a rozlišuje od koho dáta dostáva a túto informáciu musí zachovať pre príjemcu dát. Sieťová vrstva tak vytvára prenosový kanál s tým, že transportná vrstva zaisťuje na strane odosielateľa zlúčenie dát od rôznych odosielateľov *(multiplex)* a na strane príjemcu zase potrebné rozdelenie *(demultiplex)* podľa toho, komu sú dáta určené. Príklad je na obr. 9. 10



Obr. 9.10 Multiplex a demultiplex na transportnej vrstve

Štandarde sa každé jednotlivé transportné spojenie realizuje pomocou jedného sieťového spojenia, obr. 9.11 a. Pokiaľ ale transportná vrstva požaduje spojenie vyššou rýchlosťou než je než je prenosová rýchlosť jedného sieťového spojenia, môže byť jedno transportné spojenie realizované niekoľkými sieťovými spojeniami, obr. 9.11.c. Môže nastať aj opačná situácia, obr. 9.11.b. Do jedného sieťového spojenia je multiplexovaných viac transportných spojení. Takáto požiadavka vznikne pri použití viacerých terminálov, ktoré vyžadujú samostatné transportné spojenia, sú fyzicky blízko a využívanie sieťovej vrstvy je len sporadické. Preto je neekonomické vytvárať samostatné sieťové spojenie.

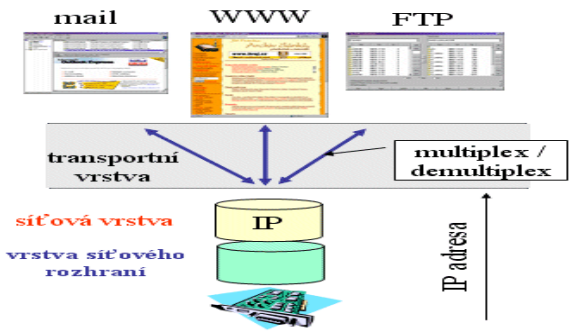


Obr. 9.11 Multiplexovanie na transportnej vrstve

Poznámka: Nemá nič spoločné s multiplexami na fyzickej vrstve

9.6.3. Adresovanie

Príklad využitia jedného sieťového spojenia pre viac transportných spojení znamená aj použitie jednej sieťovej adresy a viac transportných adries. Príklad je možné vidieť na obr. 9.12., kde je potrebné demultiplexovať tri transportné spojenia.

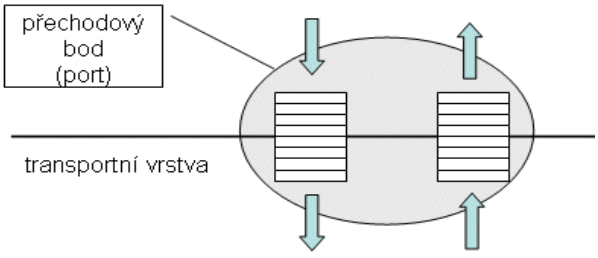


Obr. 9.12 Demultiplexovanie a multiplexovanie na transportnej vrstve

Z poznatkov sieťovej vrstvy je známe, že v sieťovej vrstve sa používajú sieťové adresy, v TCP/IP sú to IP adresy, ktoré identifikujú jednotlivé uzly. Tieto sieťové adresy nerozlišujú jednotlivých príjemcov a odosielateľov v rámci daného uzlu. Rozlišovanie jednotlivých príjemcov a odosielateľov robí až transportná vrstva. K tomu musí používať konkrétne adresy, označované ako transportné adresy. Tie majú relatívny charakter a rozlišujú príjemcov a odosielateľov iba v rámci daného uzla.

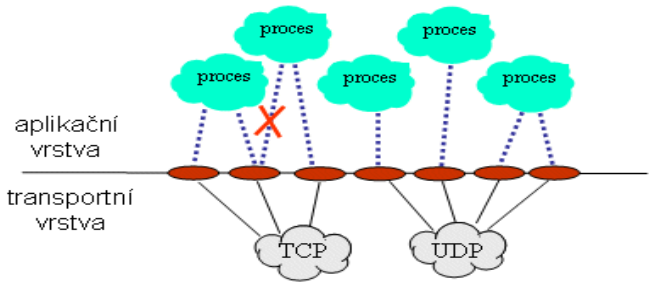
Problém je v tom, že na rôznych systémových platformách, MS Windows, Unix, Linux, a iné, môžu byť odosielatelia a príjemcovia rôzneho typu, môžu to byť rôzne procesy, prípadne systémové úlohy. Ich identifikácia v rámci príslušnej platformy sa môže veľmi líšiť. Okrem toho, príslušné entity (procesy, úlohy, atď,) vznikajú a zanikajú dynamicky. Takže nie je dopredu známe, kto (aký proces, úloha) bude existovať a mal by prijať dáta určené pre www alebo poštu.

Tu však nejde tak veľmi o to, kto nejaké dáta prijíma, ale že sú prijímané dáta nejakej služby. Rovnako pri vysielaní je klientovi jedno, aký proces je na strane servera, kde posiela svoje požiadavky. Dôležité je vedieť, že takýto server existuje a môže mu poslať požiadavky. Takže postačuje adresovanie, že príjemca je ten, kto poskytuje príslušnú službu, nie proces. Príslušné adresovanie používané na transportnej vrstve však nie je založené priamo na predstave služieb, ale je založená na predstave prechodových bodov SAP *(Service Access Poins)* medzi transportnou a bezprostredne vyššou vrstvou. Takáto predstava je znázornená na obr. 9.13.



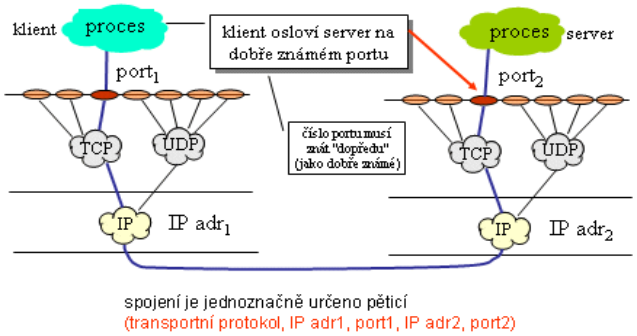
Obr. 9.13 Prechodový bod medzi vrstvami

Tieto prechodové body je možné prirovnať k vyrovnávacím pamätiam *(buffers)*, kde na jednej strane sa dáta vkladajú a na druhej vyberajú.
Prechodové body v TCP/IP sú porty počítača. Výhodou portov ako prechodových bodov je to, že môžu byť rovnaké na všetkých systémových platformách a majú rovnaké vlastnosti. Rovnaké môže byť aj označenie portov a tým ich adresovanie na transportnej úrovni. V TCP/IP sú porty adresované poradovými číslami, presnejšie celými nezápornými číslami (0,1, 2, 3, 4 atd.). V praxi sa hovorí o číslach portov. Takže transportnými číslami sú v TCP/IP práve tieto čísla.
Treba s uvedomiť, že port nie je ešte proces, alebo úloha, ktorá skutočne spracováva a generuje nejaké dáta. Port je iba prechodový bod s režimom fronty, za ktorým je schovaná úloha. Porty vždy existujú apriori, zatiaľ čo aplikačné entity (procesy, úlohy) vznikajú dynamicky, podľa momentálnych potrieb. V praxi to znamená, že ak je vytvorený nejaký proces, a ten má komunikovať v prostredí siete, musí byť pridružený k nejakému portu. Kto s takýmto procesom chce komunikovať, musí vedieť číslo portu a svoje dáta smeruje tomuto portu, presnejšie na cieľový uzol a v rámci neho na príslušný port.
Všeobecne platí, že jeden proces alebo úloha môžu byť asociované s viacerými portami, naopak jeden port môže byť združený len s jedným procesom. Dôvod je ten, že port by nevedel vyberať , komu jednotlivé procesy odovzdať. Priradenie portov a procesov je na obr. 9.14



Obr. 9.14 Priradenie portov a procesov

Príklad celej komunikácie je znázornený na obr. 9.15.

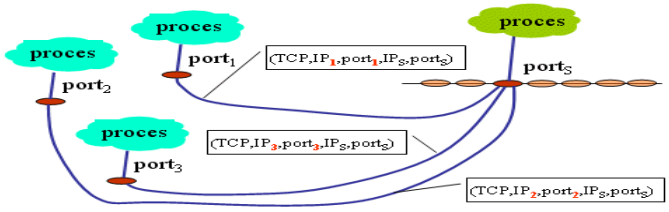


Obr. 9.15 Aplikačné spojenie

Treba pripomenúť, že jeden server môže poskytovať svoje služby cez jeden port viacerým klientom. Môže vzniknúť otázka, ako server rozpozná, komu má čo poslať. Odpoveď je jednoduchá, každá požiadavka musí mať nasledovné informácie:

- Označenie transportného protokolu
- IP adresu a port klienta
- IP adresu a port servera

Viac spojení k jednému serveru je ilustrované na obr. 9.16.



Obr. 9.16 Viac spojení k jednému serveru

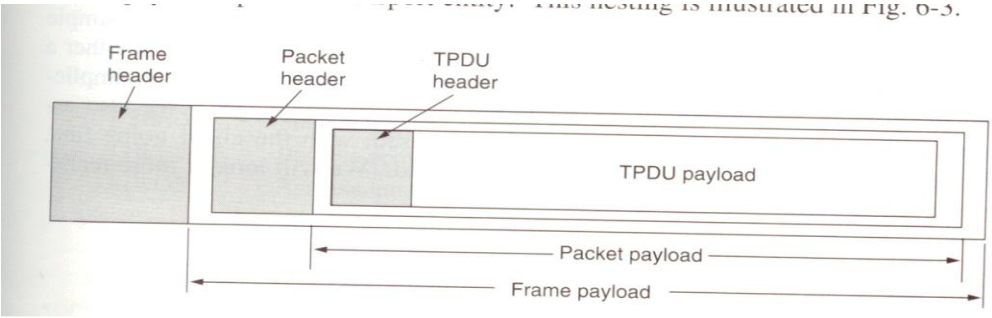
Rovnaká komunikácia je aj pri použití UDP protokolu.

Dôležitá poznámka je k určovaniu **čísla portov**. Ako môže napríklad poštový klient vedieť, že má odosielať poštu svojmu serveru na port 25? Je tu dohodnutá konvencia, a tá musí byť všetkým známa. Nie však koncovým používateľom služby ale tým, ktorý konfigurujú a prevádzkujú servery a ich klientov.
To, ktorý port je čomu priradený je dopredu stanovené, v TCP/IP sú to dobre známe porty *(well-known ports)* a ide o porty v rozsahu 0 – 1023. Príslušné konvencie spravuje IANA (Internet Assigned Numbers Autority). Sú publikované každých 6 mesiacov ako RFC dokumenty. Konkrétne sú zverejňované na http://www.iana.org/assignments/port-numbers . Mála časť je v tabuľke.

Port #	**Popis**
21	FTP
23	Telnet
25	SMTP
69	TFTP
70	Gopher
80	HTTP
88	Kerberos
110	POP3
119	NNTP
143	IMAP
161	SNMP

Štruktúra TPDU – Transport Protocol Data Unit

Transportná vrstva v systéme odosielateľa delí dáta z vyššich vrstiev na **segmenty** a v systéme príjemcu ich znova skladá do dátového toku. Príklad TPDU a jeho začlenenie do paketu a rámca je na obr. 9.17.



Obr. 9.17 TPDU

Konkrétna štruktúra TPDU je rôzna pre rôzne protokoly.

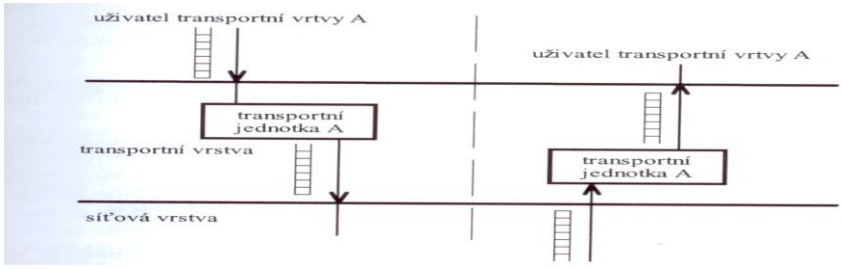
9.6.4. Riadenie toku dát

Riadenia toku dát je realizované nasledujúcimi spôsobmi:

- Potvrdzovaním (*acknowledgement*- ACK).
- Kontrolou toku *(flow control*) pomocou techniky okna (*sliding Windows)*

Potvrdzovanie

Ak chce používateľ jednej transportnej vrstvy poslať dáta používateľovi inej transportnej vrstvy, používateľ na vysielacej strane vytvára dáta a dáva ich do fronty k vysielaniu. V príslušných vrstvách dochádza k vytváraniu front a tiež k zdržaniu dát vo frontách. Existujú 4 druhy front, ako je znázornené na obrázku 9.9.



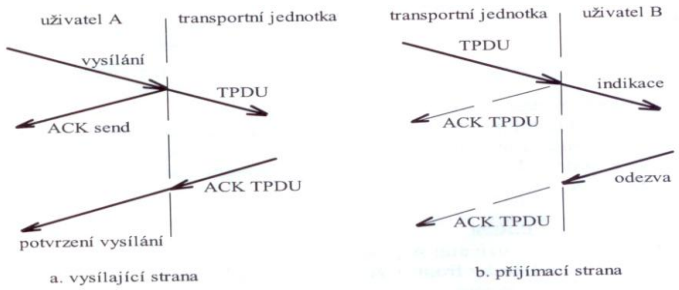
Obr. 9.9. Fronty pri prenose dát

Pri riadení toku dochádza k tomu, že dáta sú postupne odovzdávané vždy po ich povolení od zdroja k cieľu . Pritom sú odovzdávané vysielajúcej strane dva typy potvrdení:

- Potvrdenie od prijímajúcej stanice *(peer flow control)*
- Potvrdenie od vlastnej transportnej vrstvy *(interface flow control)*

Spôsob potvrdzovania dát je znázornený na obrázku 9.10.

Pri vysielaní použije používateľ funkciu vysielanie. Potom transportná vrstva generuje jednu alebo niekoľko TPDU *(Transport Protocol Data Unit).* A používateľovi potvrdí prevzatie dát „ACK send“ (ACKnowledge). Na prijímacej strane transportná jednotka TPDU zloží a a ich prijatie hlási príjemcovi ako „indikácia“. Obidve tieto potvrdenia sú transformované do dátovej jednotky „ACK TPDU“
Riadenie je potrebné preto, lebo prijímateľ nie je schopný dáta odoberať alebo to nie je schopná ani transportná vrstva. Preto musí mať transportná jednotka možnosť zastaviť alebo spomaliť rok dát, aby nedochádzalo k preplneniu vyrovnávacích jednotiek.



Obr. 9. ? Riadenie toku dát medzi používateľom a transportnou vrstvou

Animácia

Kontrola toku *(Flow control)*
Medzi počítačmi je vytvorený komunikačný kanál, prostredníctvom ktorého si vymieňajú počítače dáta (segmenty). Príjemca má na svojej strane buffer, ktorý ukazuje, ako je zaneprázdnený spracovaním posielaných segmentov. Posielajú sa dáta (segmenty) od odosielateľa ku príjemcovi, ktoré sa ukladajú do buffera príjemcu. Môže sa stať že sa buffer zaplní. Po zaplnení buffera posiela príjemca odosielateľovi správu o tom že nemôže momentálne prijímať ďalšie dáta (segmenty). Keď sa spracujú dáta, zaplnenie buffera sa zníži.
Preto teraz posiela príjemca odosielateľovi správu o tom, že už je zase je schopný prijímať ďalšie dáta (segmenty). Takáto kontrola toku má nasledujúce funkcie:

- Stará sa, aby sa nestratili dáta
- Zabezpečuje aby zdroj nezaplnil buffre cieľa, lebo potom by musel zrušiť segment
- Poskytuje komunikáciu medzi zdrojom a cieľom

Preťaženie môže vzniknúť z 2 dôvodov:

1. vysoko rýchlostný počítač môže generovať prenos rýchlejšie ako môže sieť prenášať
2. ak chce veľa počítačov naraz posielať do určitých cieľov

Technika okna
Technika okna vznikla preto, aby sa mohlo preniesť väčšie množstvo dát bez toho, že by sa musel prenos zakaždým potvrdiť. Okno je množstvo dát, ktoré je schopné prijať príjemca.
Ak odosielateľ pošle viac segmentov naraz potom ich prijímateľ príjme len toľko koľko sa mu zmestí do vyrovnávacej pamäte (buffra). Ak príjemca neprijme segment, tak jeho vyrovnávacia pamäť je plná a nie je schopná segment prijať. Ako náhle sa pamäť uvoľní môže zase príjemca spracovávať ďalšie segmenty. Pri posielaní segmentov sa zaplňuje buffer príjemcu, ktorý ich má

za úlohu spracovať. Môže sa stať, že sa buffer zaplní a príjemca už nebude schopný prijať ďalší segment. Optimálna veľkosť okna zaistí pomalý štart.
Animácia

9.1 Služby transportnej vrstvy

Účelom transportnej vrstvy je poskytovať transparentný prenos dát medzi koncovými používateľmi, čím odbremeňuje vyššie vrstvy od nutnosti poskytovať spoľahlivého a efektívneho dátového prenosu. Transportná vrstva má na starosti spoľahlivosť daného spojenia
Rozlišujú sa dva druhy služieb na transportnej úrovni:

- Spojovo orientované (Connection Oriented -CO), ktoré vyžadujú zriadenia, udržanie a ukončenie logického spojenia medzi používateľmi na transportnej vrstve. Umožňujú riadenie toku, dodržanie postupnosti odosielania segmentov a pod. Hovoríme o spoľahlivom prenose dát.
- Nespojovo orientované (Connectionless –CL) , ktoré vyžadujú kontrolu prenosu na vyšších vrstvách OSI. Pre niektoré druhy prenosov sú tieto služby výhodné. Jedná sa o aplikácie v reálnom čase, kedy straty nespôsobia výrazné zhoršenie prenosu.

9.7.Spolupráca transportnej a sieťovej vrstvy

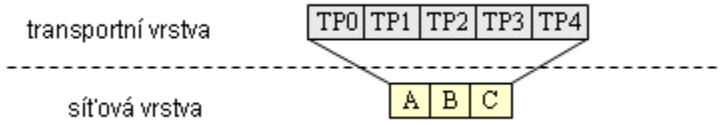
Význam transportnej vrstvy je závislý od toho, akú službu poskytuje transportnej vrstve sieťová vrstva. Podľa toho sú aj špecifikované kategórie transportných protokolov.
V OSI modeli sa predpokladá, že sieťová vrstva bude vytváraná v rôznych prostrediach , ktoré sa líšia počtom výpadkov spojení a straty dát. Napríklad LAN siete predpokladajú relatívne malý počet výpadkov a strát, naproti tomu rozľahlé siete môžu strácať dáta častejšie a častejšie môže dochádzať k výpadkom spojenia.
Preto sú pre OSI špecifikované tri rôzne varianty sieťových protokolov:

- kategória A: pre prostredie, kde dochádza k minimálnym (žiadnym) stratám paketov a minimálnym (žiadnym) výpadkom spojení
- kategória B: pre minimálne (žiadne) straty paketov, a občasné výpadky spojení
- kategória C: pre prostredie kde sú občasné straty paketov a občasné výpadky spojení

Od transportnej vrstvy sa tak očakáva, že bude zachovávať spojovaný spôsob komunikácie a že bude kompenzovať určité výpadky spojení a straty dát. Preto je špecifikovaných päť rôznych variant transportných protokolov označovaných ako TP0 - TP5:

- trieda TP0: je lenjednoduchou nadstavbou nad sieťovým protokolom kategórie A, nemení jeho vlastnosti
- trieda TP1: je nadstavbou nad sieťovým protokolom kategórie B, obmedzuje prípadné výpadky spojení
- trieda TP2: je nadstavbou nad A, dokáže využiť jedno sieťové spojenie pre viac transportných spojení
- trieda TP3: je nadstavbou nad B, obmedzuje prípadné výpadky spojení a dokáže využiť jedno sieťové spojení pre viac transportných spojení
- trieda TP4: je nadstavbou nad C, obmedzuje prípadné výpadky spojení a straty paketov.

Súvislosť medzi protokolmi sieťovej a transportnej vrstvy je na obr. 9.7.

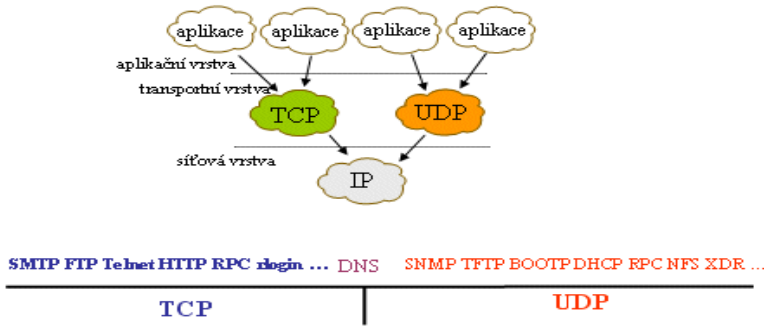


Obr. 9.7 Kategórie sieťových a transportných protokolov

V TCP/IP, kde je na sieťovej vrstve preferované poskytovanie nespojovanej nespoľahlivej služby, je otázka spoľahlivosti požadovaná od koncových uzlov, teda od transportnej vrstvy. Keďže nie je nadväzované spojenie, nie sú ani žiadne výpadky siete a tak ich nie je treba kompenzovať. Nie sú tak rozlišované žiadne protokoly sieťovej vrstvy a vystačí sa s protokolom IP. Tento protokol môže byť implementovaný v akomkoľvek prostredí, nad akoukoľvek linkovou vrstvou. Preto sa v TCP/IP technológii poskytujú na úrovni transportnej vrstvy dva protokoly:

- protokol UDP (*User Datagram Protocol)*- je len jednouchou nadstavbou nad protokolom IP a funguje rovnako ako on (tj. nespojovane a nespoľahlivo)
- protokol TCP *(Transmission Control Protocol)*- je už zložitejšou nadstavbou nad protokolom IP, a mení jeho spôsob na spojovaný a spoľahlivý.

Aplikácie si môžu samé vyberať, ktorý transportný protokol chcú používať. Tradičné počítačové aplikácie, prenos súborov, elektronická pošta preferujú spoľahlivosť a dávajú prednosť TCP. Naproti tomu multimediálne aplikácie dávajú prednosť UDP, pretože sa nezdržuje zaisťovaním spoľahlivosti a tak dokáže prenášať dáta rovnomernejšie a z menším oneskorením. Príklad použitia je na obr. 9.8.



Obr. 9.8 Voľba transportných protokolov v TCP/IP

9.8. Protokoly transportnej vrstvy

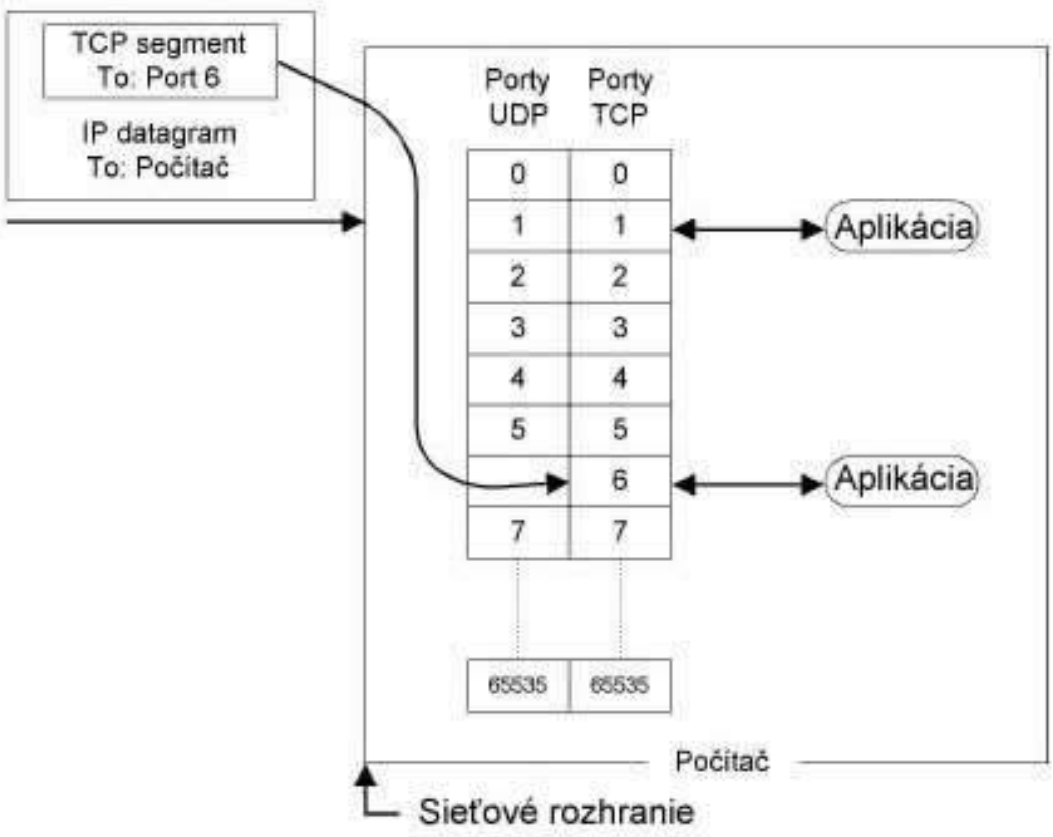
- **TCP** (**T**ransport **C**ontrol **P**rotocol),
- **UDP** (**U**ser **D**atagram **P**rotocol);
- **RTP** (**R**eal-time **T**ransport **P**rotocol)

9.8.1. TCP protokol

Protokol TCP je protokolom transportnej vrstvy. Zatiaľ čo protokol IP prenáša dáta medzi ľubovoľnými počítačmi zapojených v internete, tak protokol TCP prenáša dáta medzi dvomi konkrétnymi aplikáciami bežiacich na týchto počítačoch.

Protokol TCP poskytuje **spojovanú službu *(connection oriented)***, tj. služba ktorá naviaže spojenie – vytvorí na dobu spojenia virtuálny okruh. Okruh je plne duplexný (dáta sa prenášajú súčasne na sebe nezávisle obidvomi smermi). Prenášané bajty sú číslované. Stratené alebo poškodené dáta sú znovu vyžiadané. Integrita prenášaných dát je zabezpečená kontrolným súčtom. Ochranou prenášaných dát sa zaoberajú napr. protokoly SSL, S/MINE.

Konce spojení (odosielateľ a príjemca) sú určené tzv. číslom portu. Toto číslo je dvoj bajtové, takže môže nadobúdať hodnoty 0 až 65535. Pre protokol UDP je iná sada portov ako pre protokol TCP, tj. napr. port 53/tcp nemá nič spoločné s portom 53/udp. Cieľová aplikácia je v internete adresovaná (jednoznačne určená) IP-adresou, číslom portu a použitým protokolom (TCP alebo UDP). Protokol IP dopraví IP-paket na konkrétny počítač. Podľa čísla cieľového portu operačný systém pozná ktoré aplikácie má TCP-segment doručiť.
Základná jednotka prenosu v protokole TCP je TCP segment.



Obr. Porty TCP a UDP

9.8.2. TCP segment

TCP segment sa skladá z týchto častí:

- **Zdrojovýport (source port)** - je port odosielateľa TCP segmentov

- **Cieľový port (destination port)** – je port príjemcu TCP segmentu.
- Pätica sa skladá: zdrojový port, cieľový port, zdrojová IP-adresa, cieľová IP-adresa a protokol (TCP) jednoznačne identifikuje v danom okamihu spojenie.

- **Poradové číslo odosielaného bajtu** - je poradové číslo prvého bajtu TCP segmentu v toku dát od odosielateľa k príjemcovi(TCP segment nesie bajty od poradového čísla odosielaného bajtu až do dĺžky segmentu). Tok dát v opačnom smere má samostatné číslovanie svojich dát. Keďže poradové číslo odosielaného bajtu je 32 bitov dlhé, tak po dosiahnutí hodnoty 2^{32}-1 nadobudne cyklicky opäť hodnotu 0. Číslovanie obyčajne nezačína od nuly, ale číslovanie by malo začínať od náhodne zvoleného čísla. Vždy keď je nastavený "príznak" SYN, tak operačný systém odosielateľa začne znovu číslovať, tj. vygeneruje štartovacie poradové číslo odosielaného bajtu, tzv. ISN (initial sequence number).

- **Poradové číslo prijatého bajtu** - vyjadruje číslo nasledujúceho bajtu, ktorý je príjemca pripravený prijať, tj. príjemca potvrdzuje, že správne prijal všetko až do poradového čísla prijatého bajtu mínus jedna.

- **Dĺžka hlavičky** – vyjadruje dĺžku hlavičky TCP segmentu v násobkoch 32 bitov (4 bajty)

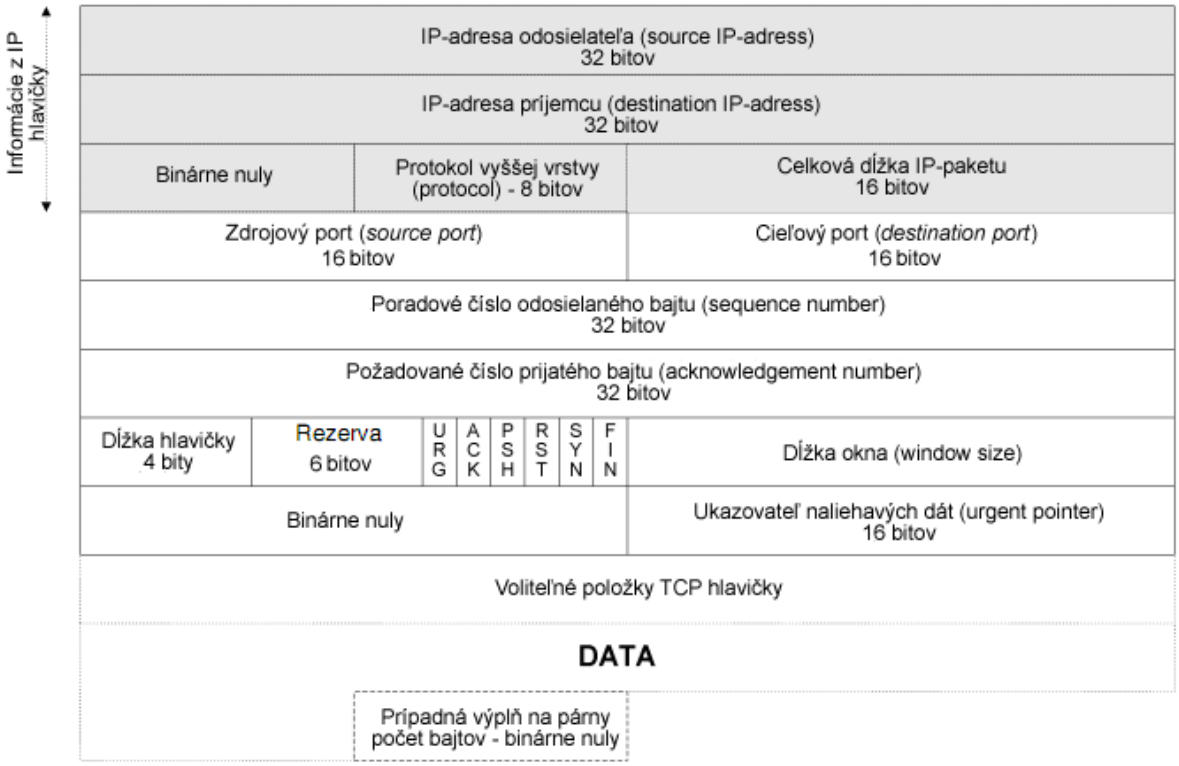
- **Dĺžka okna** - vyjadruje prírastok poradového čísla prijatého bajtu, ktorý bude príjemcom jasne akceptovaný.

- **Ukazovateľ naliehavých dát** - môže byť nastavený iba v prípade, že je nastavaný príznak URG. Ak sa pripočíta tento ukazovateľ k poradovému číslu odosielajúceho bajtu, tak ukazuje na koniec úseku naliehavých dát. Odosielateľ si praje, aby príjemca tieto naliehavé dáta prednostne spracoval. Tento mechanizmus používa napr. protokol TELNET. V TCP segmente nesúcim príkaz ABORT(signalizuje žiadosť o zrušenie procesu) sa nastaví príznak URG a vyplní sa ukazovateľ naliehavých dát ukazujúci na príkaz ABORT.

Animácia

Voliteľné položky hlavičky

Povinné položky hlavičky tvoria 20B. Za povinnými položkami nasledujú voliteľné položky. Voliteľná položka sa skladá z typu voliteľnej položky, dĺžky voliteľnej položky a hodnoty. Dĺžka TCP hlavičky musí byť deliteľný štyrmi. V prípade, že dĺžka hlavičky by nebola deliteľná štyrmi, tak sa záhlavie doplňuje jedným bajtom na konci. Keďže pole hlavičky je iba 4 bity dlhé, tak toto pole môže nadobúdať maximálne hodnoty $1111_2=15_{10}$. Dĺžka hlavičky sa udáva v násobkoch 4, teda záhlavie môže byť maximálne 15x4=60 bajtov. Povinné položky zaberú 20 bajtov, takže na voliteľné zostáva najviac 40 bajtov.



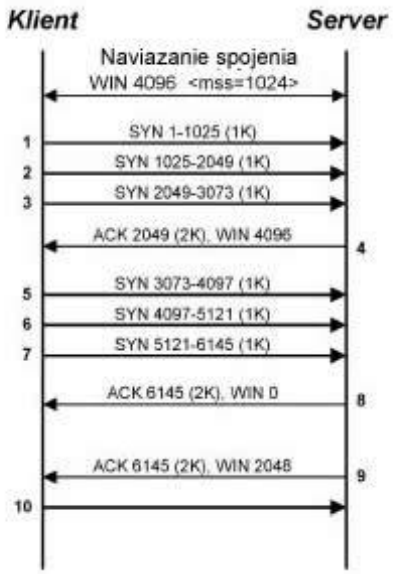
Obr. Polia z ktorých sa počíta kontrolný súčet TCP hlavičky.

9.1.1 Technika okna v TCP/IP

Okno je množstvo dát, ktoré je schopné prijať príjemca.

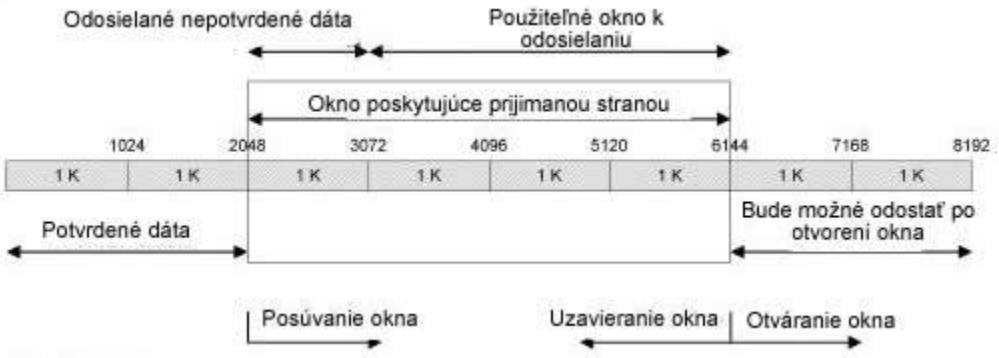
Teraz je naším problémom, keď chce klient odoslať veľké množstvo dát. Klient (resp. server) môže odosielanie dát druhej strane, bez toho by ich príjem mal potvrdený až do tzv. okna (Window – skratkou WIN) .

Predstavme si, že klient zo serverom naviazal spojenie a vzájomne sa dohodli na maximálnej veľkosti segmentu (MSS) o veľkosti 1 K (tj. 1024 B) a vzájomne veľkosť okna 4 K (tj. 4096 B).



Obr. 9 Technika okna

Klient začne s odosielaním dát, odošle segmenty 1,2 a 3, Potom dostane od servera potvrdenie 4, ktoré potvrdzuje segmenty 1a2. Klient v zapätí odosiela segmenty 5,6 a 7. Lenže server dáta medzi tým nedokázal spracovať a dáta mu zaplnili vyrovnávaciu pamäť, preto segmentom 8 síce potvrdí prijatie segmentu 3,5,6 a 7, ale zároveň klientovi uzavrie okno, tj. klient nemôže s odosielaním dát pokračovať. Potom čo server spracuje časť dát (2KB), tak umožní klientovi pokračovať v odosielaní, ale neotvorí mu segmentom 9 okno celé – iba 2 KB, preto že všetky dáta vo vyrovnávacej pamäti ešte nespracoval a pre viacej dát nie je miesto.



Obr. 1.4.5.2: Okno

Prvé 2 KB sú už potvrdené, okno je teda posunuté za bajt 2048. Tieto potvrdené dáta už klient nemusí udržovať v pamäti. Odoslané, ale nepotvrdené dáta (segment 3) tvoria 1 KB. Klient môže teda odoslať bez ďalšieho potvrdenia 3 KB dát.

9.2 UDP protokol

V balíku protokolov Internetu poskytuje UDP veľmi jednoduché rozhranie medzi sieťovou vrstvou pod a aplikačnou vrstvou nad. UDP neposkytuje žiadne záruky doručenia a

odosielateľova UDP vrstva si pri už raz odoslaných správach neudržuje žiadny stav. UDP pridáva iba kontrolné súčty a schopnosť roztrieďovať UDP pakety medzi viaceré aplikácie bežiace na jednom počítači.

Kvôli chýbajúcej spoľahlivosti sa UDP aplikácie musia zmieriť s nejakými stratami, chybami alebo duplikáciami. Niektoré aplikácie (ako napríklad TFTP) môžu podľa potreby pridávať jednoduchý mechanizmus spoľahlivosti do aplikačnej vrstvy. Aplikácie používajúce UDP našťastie najčastejšie opravný mechanizmus nepotrebujú a dokonca ním môžu byť zdržované. Pokiaľ aplikácia vyžaduje vysoký stupeň spoľahlivosti, môže sa namiesto nej použiť TCP alebo opravné kódy.

Keďže UDP nemá mechanizmus, ktorým by predchádzal a ovládal preťaženie siete, proti možnému kolapsu spôsobenému veľkou neovládanou UDP prevádzkou je potreba mechanizmu v samotnej sieti. Inými slovami: keďže odosielatelia UDP nemôžu detekovať upchanie siete, je často jediným nástrojom pre zvládnutie nadmernej UDP prevádzky zahadzovanie paketov na routeroch a iných sieťových prvkoch. Ako čiastočné riešenie tohto problému sa postupne navrhuje DCCP (Datagram Congestion Control Protocol).

Aj keď celkové množstvo UDP prevádzky na typickej sieti je rádovo len niekoľko percent, UDP používa celá rada kľúčových služieb, vrátane DNS, SNMP, DHCP a RIP.

Z predchádzajúceho obrázku je vidno, že hlavička UDP protokolu je veľmi jednoduchá. Obsahuje čísla zdrojového a cieľového portu čo je analogické protokolu TCP. Čísla portov protokolu UDP nesúvisia s číslami portov protokolu TCP. Protokol UDP má svoju nezávislú sadu čísel portov. Pole dĺžka dát obsahuje dĺžku UDP paketu (dĺžka hlavičky + dĺžka dát). Minimálna dĺžka je 8, tj. UDP paket obsahujúci len hlavičku a žiadne dáta. Pole kontrolný súčet nemusí byť povinne vyplnené. Výpočet kontrolného súčtu je tiež v protokolu UDP nepovinný. V minulosti bolo u niektorých počítačov zvykom výpočet kontrolného súčtu vypínať – jednalo sa o počítače s inštalovaným systémom NFS (Network File System). Dôvodom bolo zrýchlenie odozvy počítača.

9.3 RTP protokol

RTP je transportný protokol, ktorý zabezpečuje doručovanie interaktívnych dát (najčastejšie je to video a audio) v reálnom čase. Presnejšie zaisťuje doručenie segmentov v správnom poradí pomocou časových pečiatok *(timestamp),* sekvenčných čísel apod. Protokol poskytuje služby na identifikáciu obsahu *(payload),* sekvenčné číslovanie, časové pečiatky a monitorovanie doručenia. Typicky beží RTP na UDP protokole, môže byť ale použitý aj v iných sieťových a transportných protokoloch.

Treba poznamenať, že RTP sám o sebe neposkytuje žiadny mechanizmus na zaistenie včasného doručenia dát alebo poskytovania nejakého QoS (Quality of Service), ale spolieha sa na služby nižších úrovní. Sekvenčné číslo obsiahnuté v RTP umožňuje prijímateľovi

rekonštruovať postupnosť segmentov od odosielateľa, môže byť ale tiež použité na určenie správnej pozície segmentu, napr. pri dekódovaní videa bez nutnosti dekódovať predchádzajúce segmenty.

Pri prenose informácií v reálnom čase je možné zanedbať určitú stratu paketov, prípadne ich doručenie v nesprávnom poradí, ak takto dosiahneme menšie oneskorenie pri samotnom prenose dát.

Kľúčové slová

1. ***Relačná vrstva***
2. ***Relácia (session)***
3. ***Relačné spojenie***
4. ***Vytváranie relácií***
5. ***Riadenie dialógu v relácii***
6. ***Full duplex, Two-Way-Simultaneous, TWS***
7. ***Half duplex, Two-Way-Alternate, TWA***
8. ***Synchronizácia na relačnej vrstve***
9. ***Checkpointing***
10. ***SIP – Session Initiation Protocol***
11. ***Integrované služby internetu (IntServ)***
12. ***WSP (Wireless Session Protocol)***
13. ***RPC (Remote procedure call)***
14. ***Transportná vrstva***
15. ***End-to-end komunikácia***
16. ***Transportné spojenie***
17. ***Multiplexovanie na transportnej vrstve***
18. ***Transportné adresovanie***
19. ***SAP (Service Access Poins)***
20. ***Port***
21. ***Číslo portu***
22. ***TPDU – Transport Protocol Data Unit***
23. ***Segment***
24. ***Riadenie toku dát***
25. ***Kontrola toku (Flow Control)***
26. ***Technika okna***
27. ***Spojovo orientovaná služba (Connection Oriented Services)***
28. ***Nespojovo orientovaná služba (Connectionless services)***
29. ***TCP (Transport Control Protocol)***
30. ***UDP (User Datagram Protocol)***
31. ***RTP (Real-time Transport Protocol)***

Kontrolné otázky

1. Medzi ktorými vrstvami je špecifikovaná relačná vrstva?
2. Aká je hlavná úloha relačnej vrstvy?
3. Ktoré z uvedených významov platia pre pojem relácia?
4. Čo znamená pojem relácia v relačnej vrstve?
5. Ktoré sú základné funkcie relačnej vrstvy?
6. Kedy sa vytvára relácia?
7. Aké sú typy vzťahov relačných a transportných spojení?
8. V ktorých prípadoch môže byť použité jedno relačné spojenie a viac transportných spojení?
9. Odpovedá vždy jedno relačné spojenie aj transportnému spojeniu?
10. Čo zabezpečuje funkcia riadenie dialógu v relačnej vrstve?

11. Aký mechanizmus sa používa pre riadenie dialógu v relačnej vrstve?
12. V čom ja základná idea metódy odovzdávania poverenia?
13. V akých spojeniach je nevyhnutné využívať mechanizmus odovzdávania poverenia?
14. Aký je rozdiel medzi plne duplexným a polo duplexným dialógom v relačnej vrstve?
15. Aký je rozdiel medzi polo duplexom v relačnej a vo fyzickej vrstve?
16. K čomu je potrebná synchronizácia v relačnej vrstve?
17. Aký je rozdiel medzi synchronizáciou v relačnej a fyzickej vrstve?
18. Čo znamená pojem checkpointing?
19. K akému účelu je využívaný mechanizmus označovaný ako checkpointing.
20. Prečo je SIP radený medzi relačné protokoly?
21. Čo sú integrované služby internetu IntServ?
22. Aký je rozdiel medzi integrovanými službami v ISDN a integrovanými službami v internete?
23. Ktorý z uvedených protokolov je používaný v relačnej vrstve WAP?
24. Čo je hlavnou úlohou transportnej vrstvy?
25. V akých uzloch siete je implementovaná transportná vrstva?
26. Ktorý typ služieb je poskytovaný sieťovou vrstvou transportnej vrstve?
27. Prečo sa transportná vrstva označuje aj ako prispôsobovania vrstva?
28. Čo znamená pojem end-to-end komunikácia?
29. Ktoré z uvedených funkcií patria transportnej vrstve?
30. V ktorých typoch spojenia je potrebné vytvárať, udržiavať a uvoľňovať transportné spojenie?
31. Aký je rozdiel medzi transportným a relačným spojením?
32. Aký je rozdiel medzi multiplexovaním na transportnej a fyzickej vrstve?
33. Čo sa multiplexuje na transportnej vrstve?
34. Aké typy multiplexu sú na transportnej vrstve?
35. Aké typy multiplexov sú na fyzickej vrstve?
36. Kedy potrebné použiť multiplexovanie označované ako smerom hore?
37. Kedy je vhodné použiť multiplexovanie smerom dolu?
38. Čo v sieti identifikujú sieťové adresy?
39. Čo identifikujú transportné adresy?
40. Čím sú určené transportné adresy v TCP/IP architektúre?
41. Aké sú možnosti priradenia portov a procesov aplikácií?
42. Prečo nemôže byť eden port asociovaný s dvomi dva procesy?
43. Ako sú priraďované porty k aplikáciám?
44. Kto spravuje konvencie portov?
45. Ako sa označuje TPDU – Transport Protocol data Unit?
46. Aký je dôvod riadenia toku dát na transportnej vrstve?
47. Ktoré z vymenovaných spôsobov sú používané pre riadenie toku dát na transportnej vrstve?
48. Aký je význam techniky okna na transportnej vrstve?
49. Aké druhy služieb sú rozlišované na transportnej vrstve?
50. Ktoré z protokolov patria k protokolom transportnej vrstvy?
51. Aký je rozdiel medzi protokolmi TCP a UDP?
52. Pre aké aplikácie je vhodný RTP protokol?
53. Prečo je aplikácie v reálnom čase nepotrebujú spoľahlivú spojovanú službu?