

# Otázky k lekci č. 11: Aplikační vrstva

## 1. Jak se vyvíjela představa o zařazení aplikací do aplikační vrstvy u RM ISO/OSI a TCP/IP?

- představa, že v aplikační vrstvě jsou provozovány celé aplikace není správná!!! (bylo by nutno standardizovat celé aplikace včetně UI atd...)
- místo toho v aplikační vrstvě je pouze základ aplikace, co spolupracuje s aplikacemi na jiných uzlech, jenž musí být standardizován
- zbytek aplikace nad aplikační vrstvou, zejména UI a již nemusí být standardizováno
- platí pro ISO/OSI i TCP/IP

### RM ISO/OSI

- snaha vytvářet bohaté a dokonalé aplikační protokoly
  - X.400 (elektronická pošta)
  - X.500 (adresářové služby)
  - FTAM (práce se soubory)
  - VT (vzdálené přihlašování)
  - CMIP (správa, management)
- většina z nich se neujala
- některé zůstaly
  - X.400 (do r. 200 MS Exchange na něm fungoval)
  - X.500 (odlehčením vznikl používaný protokol LDAP)

### TCP/IP

- vývoj od jednoduchého ke složitějšímu. Aplikace vznikly jako jednoduché a postupně se obohacovaly
- počáteční množina aplikací:
  - Telnet, rlogin (vzdálené přihlašování)
  - FTP
  - SMTP, RFC 822 (elektronická pošta)
- pak přidávány:
  - NFS (sdílení souborů)
  - NNTP (sdílení informací)
  - Gopher, WWW (zpřístupnění informací)
  - Archie, Wains, Veronica (vyhledávání informací)
- dochází ke vzniku aplikačních platform
  - el. pošta, WWW nejsou již jen služby/aplikace, ale platformy, kde lze tvořit nové služby

## 2. Jaký je rozdíl mezi sdílením a přenosem souborů? Které aplikační protokoly se k tomu používají, v rámci ISO/OSI a TCP/IP?

### přenos souborů

- služba (realizovaná aplikací)
- netransparentní (rozdělují se místní a vzdálené soubory) - je třeba znát umístění vzdálených s. a pracuje se s nimi jinak, než s místními
- pro přesun souborů potřeba podnikat explicitní akce (příkazy **GET, PUT**)
- TCP/IP - nejpoužívanějším protokolem pro přenos je FTP, dále TFTP (trivial FTP)
- ISO/OSI - protokol FTAM (file transfer access and management), realizuje přenos i sdílení

### sdílení souborů

- je transparentní (nerozlišují se vzdálené a místní soubory)
- pro přesun z místního na vzdálené a naopak netřeba podnikat explicitní akce (služba/aplikace to zajišťuje sama)
- TCP/IP - NFS (network FS), dále AFS (Athena FS), nově CIFS (common internet FS)
- ISO/OSI - protokol FTAM
- Microsoft, MS Windows - protokol SMB (service message blocks)

## 3. Jak funguje protokol FTP? Jaký je význam řídicího a datového spojení? Jaký charakter mají požadavky klienta a odpovědi serveru?

- FTP chápe implicitně soubor jako dále nestrukturovaný (bez vnitřní struktury), nepotřebuje tedy doprovodnou konvenci o formátu přenášených dat
- implicitně obsah souboru přenášen jako spojitá proud dat (spolehlivý, spojovaný pomocí protokolu TCP)
- implementace vychází z modelu klient/server
  - klient typicky aplikační program
  - server obvykle systémový proces (démon, rezidentní program...)
- zajištění potřebných funkcí v rámci FTP se dělí na:

- **interpret protokolu (PI, Protocol Interpreter)**
- **přenosový proces (DTP, data transfer process)**
- PI existuje trvale, DTP vzniká až na základě konkrétního požadavku
- používají se 2 různá spojení
  - řídicí (pro přenos příkazů)
  - datové (přenáší soubory)

#### **požadavky klienta a odpověď serveru**

- každý příkaz vyvolá alespoň jednu odpověď
- odpovědi číselné trojmístné (1.celkový charakter, 2.upřesňuje odpověď, 3.blíže specifikuje)
- hierarchický charakter odpovědi vychází vstříc různě inteligentním vyhodnocovacím procesům
  - hloupý klient / server se může spokojit jen s první číslicí
  - chytrý může využít všech tří
- **1xx** - předběžně kladná odpověď (akce byla zahájena, budou ještě další odpovědi)
- **2xx** - kladná definitivní odpověď
- **3xx** - kladná prozatimní (jsou nutné další příkazy)
- **4xx** - dočasná záporná (nepodařilo se, ale je vhodné opakovat)
- **5xx** - trvalá záporná (nepodařilo se, je nesmysl opakovat)

#### **4. Naznačte průběh dialogu mezi klientem a serverem, v rámci protokolu FTP**

##### **příklad** (klient : server)

navázání spojení na uzel charon.isdn.cz na port 21 :

```

                220 charon.isdn.cz FTP server ready (definitivní kladná odpověď)
USER earchiv  :   331 password required for earchiv (prozatimní kladná, nutno zadat pswd)
PASS ***** :   230 User earchiv logged in (definitivní)
CWD /earchiv/ :   250 CWD command successful (definitivní)
RETR users.dat :  150 Opening Binary mode data connection for users.dat(1234 bytes)
                (prozatimní kladná, ještě budu dělat další akce)
(received 1234 bytes in 1 sec.....) : 226 Transfer complete (definitivní
ukončení spojení...

```

#### **5. Jaký je rozdíl mezi HTTP 1.0 a 1.1.? Co jsou metody a co hlavičky u HTTP?**

##### **HTTP** (pouze popis, co je HTTP - toho se otázka netýká)

- jednoduchý přenosový protokol přenášející v textovém tvaru
- používá transportních služeb TCP
- server pracuje na portu 80
- bezstavově
- navazuje samostatné spojení pro každý objekt na stránce (obrázek, ikonu apod...)
- komunikace tvaru **žádost-odpověď**
- odpovědi číselného charakteru (jako u FTP, nebo SMTP), ale součástí odpovědi je i WWW obsah!!!
- každá stránka může obsahovat řadu samostatných objektů (HTML kód, N obrázků, CSS externě...)
- každý objekt může být na jiném serveru

##### **HTTP 1.0**

- každý objekt na stránce získáván samostatně (je pro něj zřízeno samostatné transportní spojení s WWW serverem (na port 80), objekt vyžádán, přenesen a spojení ukončeno)

##### **HTTP 1.1**

- jsou-li objekty na stejném serveru, přenášeny společně (zřízeno jedno společné transportní spojení s WWW serverem, objekty postupně stahovány, teprve pak je transportní)

##### **metody**

- žádosti WWW klientů mají formu jednoduchých příkazů označovaných jako **metody**
- např metoda **GET, HEAD, POST** (dále pak třeba PUT, DELETE, LINK, UNLINK - nepoužívají se)
- žádosti klientů mohou být doplněny dalšími parametry - hlavičky
  - "If-Modified-Since <datum>" - stránka požadována, jen je-li novější
  - "Authorization" - pro zaslání id. údajů (jméno, heslo, ...)
  - ...
- server si nepamatuje historii komunikace s daným klientem --> komunikace je bezstavová --> požadavky různých klientů mohou být různě promíchány a serveru to nevadí!!!

## 6. Naznačte průběh dialogu mezi klientem a serverem, v rámci protokolu HTTP

### příklad

- požadavek klienta  
GET /index.html HTTP/1.0
- odpověď serveru (2xx)  
(hlavičková část - hlavičky protokolu HTTP)  
HTTP/1.0 200 OK  
Date: Mon, 22 May 2000 22:22:22 GMT  
Server: Czech-Net Apache  
Content-Length: 546  
Last-Modified: Thu, 08 Apr 1999 07:08:09 GMT  
Connection: close  
Content-Type: text/html; charset=windows-1250  
Expires: Thu, 08 Jan 1970 00:00:01 GMT  
<html><head>... (datová část)

## 7. Co a jak definují standardy SMTP, RFC 822 a MIME?

(budu užívat následující analogie: zpráva je list papíru, jež se vloží do obálky a teprve ta se přenáší)

### SMTP

- přenos zpráv mezi servery (simple mail transfer protokol)
- definuje obálku a způsob jejího přenosu (i to, co má být napsáno na této obálce)
  - některé položky hlavičky listu jsou kopírovány i na obálku (adresa příjemce, odesilatele)
- SMTP jako takové je přenosovým mechanismem pro přenos zpráv (využívá spolehlivých přenosových služeb TCP (ale může být implementován i nad jinými spolehlivými protokoly))
- chápe přenášená data jako text (jež užívá CRLF řádkování (windows) a 7-bitové ASCII znaky)

### RFC 822 (doporučení)

- formát zpráv a adres
- definuje a doporučuje, co má být napsáno na listu papíru

### POP3, IMAP

- stahování zpráv ze schránky na poštovním serveru

### MIME

- rozšíření (národní abecedy, přílohy, formátování ...)

## 8. Jak se doručuje el. pošta podle MX záznamů? Naznačte.

- e-mailové adresy mají nejčastěji tvar alias@doména -> např. sileny@blazen.cz
- odesílatel pozná, na který SMTP server má zprávu doručit tak, že pošle dotaz do DNS (domain name server), kde je definován pro každou doménu MX (mail exchanger) záznam, který definuje jeden, nebo více SMTP serverů, jež přijímají poštu pro danou doménu. Vráti adresy těchto serverů odesílateli a ten pak odešle zprávu na správný stroj.
- pozor!!! když jsme mluvili o odesílateli, nemyslíme stroj, obecně nemyslíme stoj, na kterém je e-mail psán, ale poštovní server, který odesílatel zná (má ho uveden v konfiguraci) a tomu zprávu předá na odeslání (vhodí do poštovní schránky) -> zpráva se předává pomocí protokolu SMTP

## 9. Naznačte průběh dialogu mezi odesílatelem a příjemcem, v rámci protokolu SMTP

### průběh - příklad

navázání transportního spojení na portu 25	220 xxxx.xxx.cz SMTP service ready
HELLO smtp.post.cz	250 xxxx.xxx.cz hello smtp.post.cz
MAIL FROM: <odesílatel@domena>	250 sender ok
RCPT TO: <prijemce1@domana> (zadam-li vice mailu, posle vice lidem)	250 recipient ok
RCPT TO: <prijemcex@domena> (skryta kopie)	250 recipient OK
DATA	354 Enter mail, end with "." on a line by itself
{hlavička zprávy dle RFC822}	
{tělo zprávy dle RFC822}	
{tečka, jako zakončující znak}	
	250 mail accepted
QUIT	221 xxxx.xxx.cz

## 10. V čem je problém netextových přenosů u SMTP pošty a jak jej řeší standard MIME?

- původně SMTP pošta určena jen pro přenos krátkých zpráv v čistém ASCII
- problém, když se někdo pokusí přenést jiné, než 7bitové znaky
  - může to dopadnout dobře
  - nemusí (nejvyšší bity ořezány)
- problém s přílohamy
  - pokud by k txt. zpr. byl přiložen dat. soubor, nemusel by projít (je obecně tvořen 8-bitovými znaky)
- problém s národními abecedami
- problém s formátováním (formátovací znaky taky 8bitové)
- jednoduchý princip řešení
  - všechno 8bitové se převede na 7-bitové, přeneso a pak vrátí
    - lze učinit mnoha způsoby - důležité je, dohodnout se na společném postupu

### **nesystematická řešení**

- týkají se pouze přibalování příloh
- UUENCODE - varianta přibalování příloh ze světa unixu
- BinHex - varianta od Macintoshe

### **systematické řešení: standard MIME**

- multipurpose internet multimedia extensions
- podporován většinou novějších klientů (1 zpráva i více příloh, příloha cokoliv, co lze zabalit do podoby souboru)
- jde psát česky
- jde provázat poštovní klient s aplikacemi (stačí kliknout na tlačítko s přílohou a otevře se WORD)
- definuje kódování (2 způsoby převodu 8b na 7b - Quoted Printable a Base64)
- definuje typování dat (text/html .....
- definuje rozšíření formátu zprávy (zavádí rozšíření dle RFC822)
- MIME standard je typickým příkladem vývoje aplikací v rámci TCP/IP

## 11. K čemu sloužily služby Gopher, WAIS a Archie, a čím (a proč) byly nahrazeny?

### **Gopher**

- prohrál v souboji s WWW
- vyvinut na University of Minnesota
- služba pro zpřístupnění informací
- uživatelé poskytují nabídku ve formě menu
  - jednotlivé položky uspořádány lineárně, položky textové (i celé menu)
  - položka může představovat (soubor, odkaz na jiné menu, přechod (bránu) do jiné služby, či aplikace)
- dnes funguje jen velmi málo Gopher serverů

### **Wais a Archie**

- původně samostatné služby elektronické pošty
- uživatelé museli používat specifické aplikace a specifický styl práce (znát příkazy)
- trend vedl k minimalizaci klientů
- důsledkem se rozpadal široký repertoár služeb TCP/IP a zůstaly dvě základní aplikace
  - WWW (browser)
  - pošta (klient poštovní)
- WWW a pošta se staly zákl. platformami a na nich dnes stavěny další služby, jež původně byly samostatné
  - el. pošta -> zprostředkovává též diskuse, el. konference, nástěnky apod...
  - WWW -> různé formy vyhledávání, transakce (nákupy), hry, e-learning, vzdálené přihlašování
- **přesto dnes vznikají samostatní klienti** (např. IM klienti ICQ, Skype ...)