

Elektronická pošta

Elektronická pošta v internetu (též označovaná jako SMTP pošta podle protokolu, který zajišťuje přenos) není proprietární (tj. není vlastněna žádnou firmou, vychází z plně otevřených standardů).

Elektronická pošta je

- rychlá (čas doručení se měří v minutách a sekundách)
- levná (i když záleží na konkrétním způsobu připojení)
- pohodlná (mnoho úkonů lze zautomatizovat – např. třídění došlých zpráv)
- efektivní (lze ji propojit s jinými aplikacemi, umožňuje snadné hromadné rozesílání)
- funguje „off-line“ způsobem (nevyžaduje současné připojení odesílatele a příjemce v konkrétní čase)

Historický základ

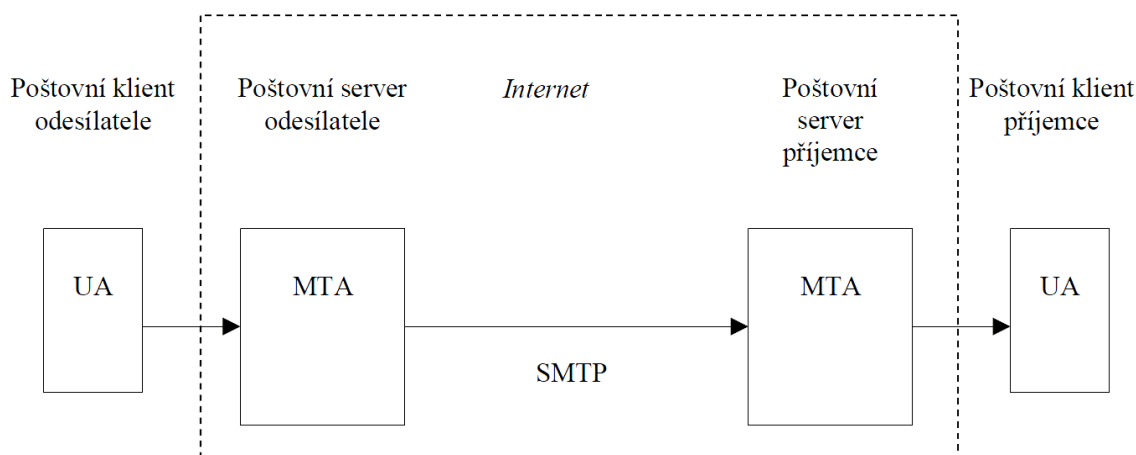
Zadání, které dostali autoři původní koncepce elektronické pošty po Internet, znělo asi takto: "můžete počítat s tím, že přenášet se budou krátké, čistě textové zprávy (chápej: tvořené základními ASCII znaky). Vymyslete vše tak, aby to bylo co možná nejefektivnější". Výsledkem bylo řešení, které skutečně naplnilo literu tohoto zadání: soustřeďuje se opravdu jen na co možná nejefektivnější přenos krátkých textových zpráv, tvořených základními ASCII znaky, a nezabývá se takovými "nadbytečnostmi", jakými jsou například přidávání netextových příloh k jednotlivým zprávám, možnost používání jiných znakových sad než "čistého ASCII", podpora hodně velkých zpráv (např. velikosti megabytů) apod. Faktem je, že dnes jsou požadavky tohoto typu (tj. přílohy, jiné znakové sady atd.) na elektronickou poštu kladeny, ale ona jim vstřícně nevyhází. Bylo by ale chybou svalovat za to vinu na autory původní koncepce internetové elektronické pošty – oni takovéto zadání nedostali.



Koncepce elektronické pošty, dodnes používané v Internetu, je založena na dvou základních standardech: dokumentech RFC821 a RFC822. První z nich, dokument RFC821, definuje přenosový protokol jménem SMTP (Simple Mail Transfer Protocol). Podle tohoto protokolu spolu komunikují jednotlivé poštovní úřady (poštovní servery, tj. jednotky MTA - Message Transfer Agents), když si mezi sebou předávají jednotlivé zprávy. Spojení probíhá na smluveném portu 25.

Druhý dokument, RFC822, pak definuje formát těchto zpráv - říká, že se skládají z hlavičky a těla, definuje typ a přesný tvar (syntaxi i sémantiku) jednotlivých položek hlavičky, a říká také, co smí a co nesmí být v těle zprávy. Mimo jiné také definuje i přesný formát adres, které lze používat pro potřeby elektronické pošty - proto se také internetovým poštovním adresám říká "adresy á la RFC822".

Samotné elektronické poště, vycházející ze standardů RFC821 a 822, se pak říká "SMTP pošta", podle charakteristického protokolu SMTP.



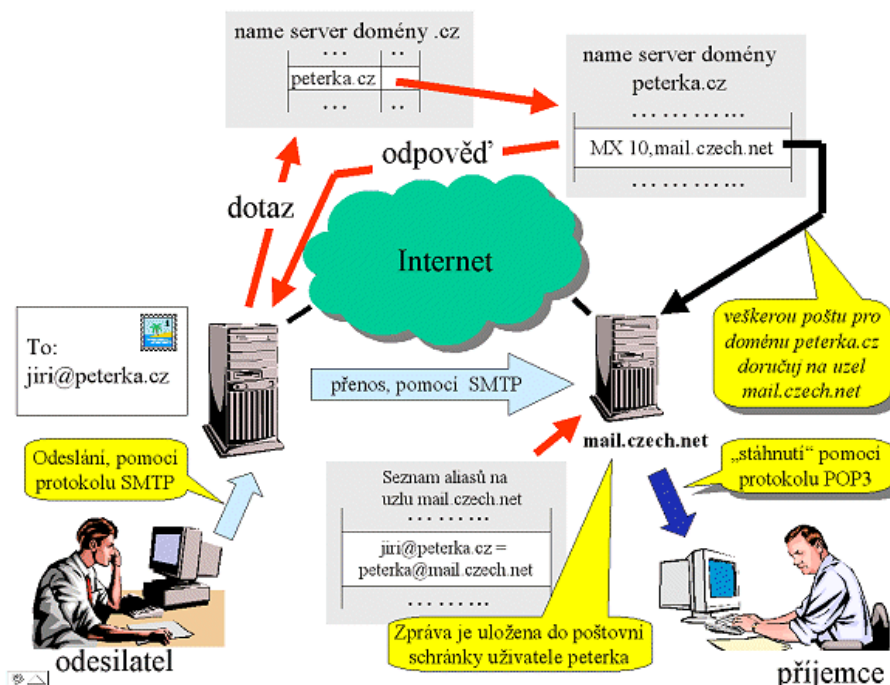
Obr. 1 – Přepavní systém

Jak funguje přenos elektronické pošty

Původní koncepce SMTP pošty počítala s tím, že adresy jednotlivých zpráv budou typu @. To umožňovalo, aby odesílající poštovní server, podle pravidel protokolu SMTP, navázal přímé spojení s cílovým uzlem (podle jeho adresy), zeptal se jej zda je ochoten přijmout poštu pro příslušného uživatele (uvedeného v adrese), a pokud ano, zprávu mu předal (načež přijímající poštovní server ji uložil do poštovní schránky příslušného uživatele).

Konkrétní postup

1. Uživatel si spustí klientský program a pomocí vestavěného editoru napíše zprávu. Zpráva je upravena tak, aby vyhovovala standardu rfc822 a pokud zpráva obsahuje nepovolené znaky (s diakritikou) či přílohy, je na ni aplikován standard MIME.
2. Když uživatel napsal všechny zprávy, zvolí ve svém klientském programu volbu odeslat. UA najde ve své konfiguraci MTA (tj. server odchozí pošty) a pomocí protokolu SMTP zprávu odešle na MTA. UA zde tedy vystupuje jako SMTP klient (v komunikaci s MTA jakožto SMTP serverem).
3. SMTP server uloží zprávu do fronty zpráv, které má zaslat na poštovní úřady příjemců. MTA odesílatele vybírá zprávy z fronty a pokouší se je zaslat na poštovní úřady příjemců. Nejprve se podívá na část adresy vpravo od zavináče (např. peterka.cz) a snaží se ptát systému DNS (systému doménových jmen v Internetu), kam má být doručována pošta pro doménu peterka.cz. Odpověď mu může poskytnout pouze name server této domény, na který se odesílající server se svým dotazem musí obrátit (pokud nezná jeho adresu, musí se zeptat nejprve serveru nadřazené domény .cz, viz. níže uvedený obrázek). V name serveru domény peterka.cz by měl být tzv. MX záznam (záznam typu Mail Exchange), který vyjadřuje požadovanou informaci - v tomto konkrétním případě fakt, že veškerá pošta pro uživatele z domény peterka.cz má být fakticky doručována na počítač se jménem mail.czech.net. Proto odesílající MTA naváže spojení s MTA příjemce (mail.czech.net), a zprávu mu pošle. Celá komunikace opět probíhá protokolem SMTP, takže MTA odesílatele zde vystupuje jako SMTP klient. Pokud se spojení s MTA příjemce nedaří, vrací MTA odesílatele zprávy zpět do fronty a zkouší je zaslat později. Pokud se odeslání nepodaří v předepsaném limitu, jsou zprávy vráceny odesílateli jako nedoručitelné.
4. MTA příjemce zprávy uloží do příslušné přihrádky podle adresy v části alias a to do složky nových zpráv.
5. Příjemce v okamžiku, kdy má čas, spustí svůj klientský program UA. Protože jeho nové zprávy jsou dosud na jeho MTA, musí je nejprve vyzvednout, což provádí pomocí protokolu POP3 nebo IMAP.
6. Vyzvednuté zprávy jsou uloženy do lokální složky nových zpráv v klientském programu příjemce. Uživatel si pak prohlíží a čte zprávy za pomoci vestavěného editoru ve svém klientském prostředí.



SMTP pošta je sedmibitová!

Velmi důležitou vlastností protokolu SMTP je způsob, jakým se dívá na přenášená data. Přesně v souladu s původním zadáním předpokládá, že jsou sedmibitová - příslušný standard (dokument RFC821) konkrétně říká, že zprávy mohou obsahovat kterýkoli ze 128 ASCII znaků, čímž je míněna základní sada ASCII znaků, z nichž každý je zobrazitelný právě v sedmi bitech (neboť pro rozlišení 128, neboli 2^7 , stačí právě 7 bitů)!

Standard definující protokol SMTP přitom počítá s možností, že přenosové cesty budou ve skutečnosti 8-bitové, neboli že budou přenášet najednou celé osmice bitů (tzv. oktety), a pro tento případ stanoví, že sedmibitové znaky mají být vkládány do osmibitových bytů na nejméně významné bitové pozice (tj. mají být zarovnávány doprava), zatímco nejméně významnější (nejlevější) bit má být nastaven na nulu.

Co je nesmírně podstatné a důležité, je skutečnost že definice protokolu SMTP neříká nic o 8-bitových znacích ve zprávách, neboli o tom, jak se má zachovat implementace protokolu SMTP v případě, kdy dostane k přenesení zprávu obsahující nějaké 8-bitové znaky. Některé konkrétní implementace protokolu SMTP mohou "honorovat" nejméně významnější bit a přenášet jej stejně věrně, jako ostatní bity. Platnému znění standardu však vyhovují i takové implementace SMTP, které se vůči nejméně významnějšímu bitu chovají jinak, například tak že jej zcela ignorují (a na výstupu jej nastaví pevně na 0).

V praxi je pak celkový efekt následující: bude-li se někdo snažit přenést přes určitou posloupnost poštovních serverů komunikujících protokolem SMTP zprávu obsahující osmibitové znaky, pak nemá zaručeno, že obsah zprávy bude doručen korektně. Může se stát, že ano (a je to docela pravděpodobné), ale není to zaručeno - jistota je pouze v případě, kdy zpráva obsahuje pouze znaky znázornitelné v sedmi bitech.

Ponaučení z právě uvedených faktů by mělo být následující: pokud je předem známo, přes které poštovní servery bude zpráva procházet (resp. přes které konkrétní implementace protokolu SMTP), a všechny budou podporovat osmibitový přenos, pak je možné zařazovat do zpráv i osmibitové znaky. V opačném případě (který je v praxi bohužel nejčastější) je nutné se vyvarovat osmibitových znaků uvnitř zpráv, a to jak uvnitř jejich hlaviček, tak i uvnitř jejich těl.

V čem je skutečný problém?

V dnešní době již neplatí původní předpoklad o tom, že jednotlivé zprávy budou v praxi obsahovat pouze čisté ASCII znaky - dnes je velký tlak na to, aby bylo možné používat v hlavičce i v těle zpráv znaky různých národních abeced, a stejně tak je velmi žádoucí přibalovat ke zprávám všelijaké přílohy, které také nemají charakter čistých ASCII textů. Obecně se na tyto znaky národních abeced i netextové přílohy lze dívat jako na posloupnosti osmibitových znaků, které ale obecně neprojdou pouze sedmibitovým přenosovým mechanismem. Řešení však není nijak principiálně těžké: posloupnost osmibitových znaků se před odesláním přetransformuje na posloupnost sedmibitových znaků ("čistých ASCII znaků"), která přenosovým mechanismem na bázi protokolu SMTP zaručeně projde, a na druhé straně se přetransformuje zpět do původní podoby. Tak kde je vlastně nějaký problém?

Skutečný problém není až tak technického rázu, jako spíše rázu organizačního: způsobů, jak převést posloupnosti osmibitových znaků na posloupnost znaků sedmibitových, je mnoho. Problém je v tom, jak se sjednotit na jednom z nich, nebo alespoň jak zařídit věci tak, aby každý příjemce vždy bezpečně poznal, jaké dekódování má použít, aby přijatou posloupnost vrátil zpět do její původní podoby. Dalším problémem je to, jak poznat že v přenášené zprávě vůbec je nějaká takováto část, která má být určitým způsobem přetransformována (a kde přesně se nachází). A aby ještě nebylo všem problémům konec, dalším z nich je jak poznat, co je takováto část zač a jak ji dále zpracovat, až se jí podaří přetransformovat zpět do její původní podoby. Jde například o dokument editoru MS Word, který má být předán tomuto editoru k zobrazení a dalšímu zpracování? Nebo se jedná o spreadsheet, který má být "předhosen" konkrétnímu tabulkovému procesoru? Zde opět jde o problémy, ke kterým není nijak těžké najít vyhovující technické řešení - nejtěžší je, jak přimět ostatní zainteresované subjekty ke konsensu o vhodnosti takového řešení, a jak jej prosadit do praxe jako standardní přístup, který by všichni měli znát a dodržovat.

Aktuální standard MIME (Multipurpose Internet Mail Extensions)

Původní koncepce elektronické pošty vyhovovala pouze potřebám přenosu textových zpráv v anglickém jazyce (tj. bez diakritiky). Užívalo se sedmibitové kódování. Požadavek přenášet poštou i jiná data (zejména osmibitová) si vynutil modifikaci původní koncepce standardu SMTP a RCF 822.

Standard MIME tedy usiluje o to, aby na rozdíl od "klasické" SMTP pošty bylo možné:

- používat i jiné než "čisté ASCII znaky" v těle jednotlivých zpráv. V našich zeměpisných šířkách to například znamená možnost psát text zprávy s plnou českou diakritikou
- používat znaky národních abeced (ne-ASCII znaky) i v položkách hlaviček zpráv, tedy například možnost psát tzv. předmět zprávy (Subject) s diakritikou.
- přikládat k jednotlivým zprávám nejrůznější druhy příloh, obecně cokoli co lze "zabalit" do formy souboru. Tedy například formátovaný text, produkovaný nejrůznějšími textovými editory, spreadsheetové soubory, databázové soubory, obrázky, digitalizované zvukové nahrávky, videoklipy apod. Požadavkem přitom bylo to, aby příloh mohlo být i více než jen jedna (obecně libovolně mnoho)
- umožnit, aby tělo samotné zprávy mohlo být tvořeno také něčím jiným, než jen "čistým ASCII textem", bez jakékoli možnosti formátování apod. Příkladem může být situace, kdy tělo zprávy tvoří HTML kód (neboli zdrojový tvar WWW stránky).

Standard MIME je nadstavbou standardu SMTP, který se nemění a týká se poštovních klientů.

Možnosti používat v těle i v hlavičce zpráv jiné než čisté ASCII znaky, stejně tak jako možnosti přikládat ke zprávám nejrůznější přílohy převážně netextového charakteru, stojí v cestě pouze sedmibitový (garantovaný) charakter přenosového mechanismu, daný použitým protokolem SMTP.

Standard MIME proto musí definovat způsob, jakým osmibitové znaky (resp. osmibitové byty) převést před přenosem na znaky sedmibitové, a po přenosu je zase vrátit zpět do původní podoby. Jde vlastně o potřebu definovat vhodný způsob kódování (anglicky: encoding).

MIME definuje dokonce dva takovéto způsoby kódování, označované jako "quoted-printable" a "Base64", ale nepředepisuje striktně který má být kdy použit. Pouze doporučuje použití kódování "quoted printable" tam, kde se výchozí text příliš neodlišuje od čistého ASCII textu (tj. neobsahuje příliš mnoho "cizorodých" znaků), a alternativní kódování Base64 tam, kde je odlišnost od čistého ASCII textu větší. Logika, ze které toto doporučení vychází, je dána konkrétní povahou obou způsobů kódování.

Kódovací algoritmus Quoted-printable

Toto kódování je určeno pro data, která z větší části obsahují tisknutelné ascii znaky. Výsledkem kódování je ascii text, který je i bez dekódování z velké části pro člověka čitelný. Kódován může být i text, který obsahuje pouze ascii znaky a to např. z důvodu zajištění integrity dat, pokud data prochází přes gateway, která provádí náhradu znaků a/nebo zarovnání řádky.

Vrátí-li se nám náš mail jako nedoručitelný, pak se často setkáváme, že mailový server, který zprávu vrací, použije právě kódování quoted-printable.

Pravidla kódování

- Libovolná osmice bitů (kromě znaků konce řádky) je nahrazena znakem "=" následovaných hexadimální hodnotou osmice bitů.
- Osmice s desítkovou hodnotou od 33 do 60 a od 62 do 126 včetně jsou nahrazeny ascii znaky (od ! do < a od > do ~).
- Osmice s hodnotou 9 a 32 jsou nahrazeny TAB a SPACE znaky. Nesmí být na konci řádku.
- Konec řádky je vyjádřen CRLF.
- Zakódovaná řádka musí mít maximálně 76 znaků. Pokud je řádka delší použije se měkký konec řádky, tj. vloží se znak = na konec řádky.

Každý "osmibitový" znak je tak vlastně nahradí trojicí sedmibitových znaků, například naše "Č", které má ve znakové sadě ISO-8859-2 kód C8 (hexadecimálně), nahradí trojicí znaků "=C8" apod., a mezeru nahradí podtržítkem. Například slůvko "Článek" je pak v tomto kódování reprezentováno řetězcem "=C8l=E1nek". Při troše cviku (a malé četnosti diakritiky) je zakódovaný tvar pro člověka stále srozumitelný. Při větší četnosti jiných než čistých ASCII znaků však roste neefektivnost tohoto způsobu kódování.

Kódovací algoritmus Base64

Je určeno především pro obecná binární data, tj. data, kde nepřevažují standardní ASCII znaky. Kódovaná data jsou pouze o třetinu delší než data původní. Pro převod se používá tabulka Base64 o 64 znacích (a znak "="). Pro kódování 64 znaků je třeba 6 bitů ($2^6=64$). Znak "=" (65) se používá ke speciálnímu účelu.

Z hlediska kódování se na zprávu nehledí jako na proud osmic bitů (bajtů), ale jako na proud šestic bitů. Každá šestice se pak kóduje podle tabulky Base64. Na začátku kódování se kódovaný text rozdělí na sekvence 24 bitů a ty následně na 4 šestice bitů. Každá šestice reprezentuje jeden znak v abecedě Base64. Kóduje se zleva doprava. Každých 6 bitů je nahrazeno odpovídajícím znakem z tabulky znaků abecedy Base64.

Č	l	á	n	é
C : 8	6 : C	E : 1	6 : E	6
11001000	01101100	11100001	01101110	01101100
110010	000110	110011	100001	011011
50	6	51	33	27
y	G	z	h	b

Zdrojový řetězec "Článek" je v kódování Base64 nahrazen řetězcem "yGzhbmVrI", který snad dostatečně názorně dokumentuje skutečnost, že tento způsob kódování je pro člověka zcela nesrozumitelný.

Tabulka Base64

Value	Encoding	Value	Encoding	Value	Encoding	Value	Encoding
0	A	17	R	34	i	51	z
1	B	18	S	35	j	52	0
2	C	19	T	36	k	53	1
3	D	20	U	37	l	54	2
4	E	21	V	38	m	55	3
5	F	22	W	39	n	56	4
6	G	23	X	40	o	57	5
7	H	24	Y	41	p	58	6
8	I	25	Z	42	q	59	7
9	J	26	a	43	r	60	8
10	K	27	b	44	s	61	9
11	L	28	c	45	t	62	+
12	M	29	d	46	u	63	/
13	N	30	e	47	v		
14	O	31	f	48	w	(pad)	=
15	P	32	g	49	x		
16	Q	33	h	50	y		

MIME typ

Způsoby kódování, které standard MIME zavádí, umožňují "protlačit" skrz přenosový mechanismus v podstatě libovolná data, která mají být přenesena. S vlastními daty však musí být vždy přenesena i informace o tom, co jsou tato data zač - zda například jde o obrázek ve formátu GIF, dokument ve formátu MS Word, text ve formátu jazyka HTML apod. Právě podle této informace se totiž příjemce bude rozhodovat, co s příslušným datovým objektem učiní (například tehdy, když mu uživatel dá kliknutím najevo, že si jej přeje zobrazit). Je dobré si uvědomit, že tato informace o typu dat je nezávislá na způsobu jejich kódování - fakt, že určitá data jsou pro potřeby přenosu zakódována např. pomocí kódování Base64, ještě neříká nic o tom, co jsou tato data zač a jak má být s nimi dále naloženo.

Pro potřeby identifikace datového obsahu používá standard MIME dvouúrovňovou "klasifikaci", obecně zapisovanou jako dvojice "typ/podtyp" (anglicky: type/subtype). Na vyšší úrovni se rozeznává celkem sedm možných hlavních typů (které jsou dále upřesňovány konkrétními podtypy):

typ "text": `Content-Type: text/html; charset=ISO-8859`
Podle definice se jedná o jakýkoli text, který je apriorně srozumitelný a nevyžaduje použití specifických programů, které by teprve dokázaly "porozumět" jeho obsahu. V praxi se sem řadí například:

- "čistý ASCII text", bez jakéhokoli formátování. Pro něj se používá podtyp "plain" (tj. celý tzv. MIME typ je zde "text/plain")
- text v jazyku HTML (MIME typ "text/html").
- text ve formátu "richtext" či "enriched", který je definován v dokumentech RFC 1341, resp. 1896.

Mezi "textové" typy (tj. pod hlavní typ "text") se naopak nezařazují takové texty, které obsahují nějaké binární formátovací informace - což jsou například nativní formáty textových editorů typu Word apod.

Parametr CHARSET indikuje použitou znakovou sadu. Implicitní hodnota je us-ascii.

typ "image": `Content-Type: image/jpeg`
Pro obrázky

typ "audio": `Content-Type: audio/wav`
Pro zvuková data

typ "video": `Content-Type: video/mpeg`
Pro obrazová data

typ "application": `Content-Type: application/msword; name="soubor.doc"`
Obecně pro všechny ostatní druhy dat, které musí být pro potřeby zpracování "předhozeny" nějakému konkrétnímu programu (aplikaci, odsud označení hlavního typu), nebo mají samy "spustitelný" charakter, jako například skripty v jazyky JavaScript.

Na celém konceptu tzv. MIME typů je zajímavý ještě jeden fakt - jde o řešení, které se prosadilo i jinde, mimo rámec elektronické pošty, v situaci kdy také bylo třeba přesně vyspecifikovat, co jsou určitá data zač. Konkrétně se jedná v komunikaci mezi WWW serverem a jeho klientem, uskutečňovanou prostřednictvím protokolu HTTP (HyperText Transfer Protocol). Také zde je totiž nutné, aby server klientovi explicitně řekl, jaký je typ dat, která mu právě posílá. No a k tomu se využívá již jednou zavedená dvouúrovňová hierarchie MIME typů.

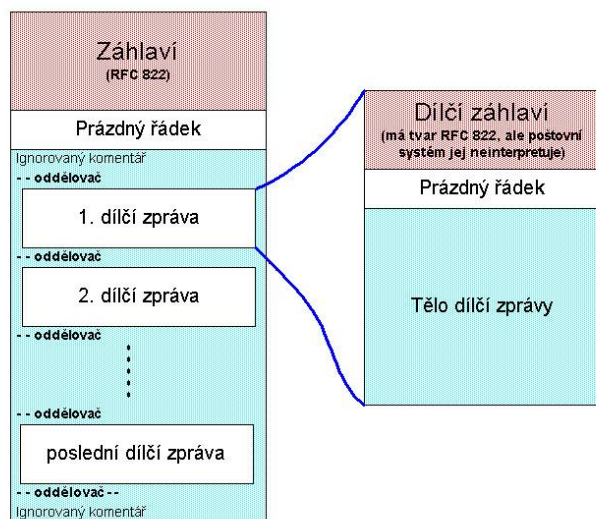
MIME typ multipart

Doposud jsme se zabývali pouze zprávami, které se skládaly pouze z jedné části.



Nyní se budeme zabývat zprávami složenými z více jednotlivých částí.

- Zpráva obsahuje přílohy
- Zpráva může být odeslána současně ve více tvarech (např. čistý text, HTML, audio)
- Dlouhá zpráva může být transportována jako několik kratších



Pokud tělo zprávy obsahuje několik různých částí (několik dílčích zpráv), tak každá část začíná za úvodním oddělovačem. Tento oddělovač je do zprávy zaveden jako parametr boundary v rámci definice hlavičky MIME typu multipart. Jedná se o speciální sekvenci znaků, která se nesmí vyskytnout nikde uvnitř jednotlivých částí.

Více nezávislých částí:

Content-Type: multipart/mixed;
boundary="gc0p4J:2408t"

Více částí se shodnými informacemi v odlišném tvaru:

Content-Type: multipart/alternative;
boundary="gc0p4J:2408t"

Dále pak následují hlavičky první dílčí části, prázdný řádek a vlastní tělo této zprávy. Poslední část je ukončena koncovým oddělovačem.

Jednotlivé dílčí zprávy nejsou interpretovány podle RFC822. Mohou, ale také nemusí obsahovat hlavičky (prázdný řádek za záhlavím však musí být vždy uveden). Pokud nejsou hlavičky u části uvedeny, uplatní se implicitní hlavičky ze záhlaví celé zprávy.

Oddělovací text definovaný parametrem boundary se používá také jako dílčí oddělovač jednotlivých částí zprávy. Z kraje oddělovacího řádku se uvedou dvě pomlčky "- -" následované uvedeným textem. Maximální délka oddělovače je 70 znaků.

Příklad zprávy, která bude zobrazena buď jako obyčejný text, zformátovaný text nebo přehrána hlasovým výstupem.

```
From: bilbo@pvt.net
To: gandalf@hrad.org
Subject: Pozvani
MIME-Version: 1.0
Content-Type: multipart/alternative; boundary="gc0p4J:2408t"

Pro pripad, ze Tvuj prohlizec neumi MIME,
tak Te proste zvu na vylet .

--gc0p4J:2408t
Content-Type: text/plain; charset=us-ascii

Zvu te i bez hacku a carek

--gc0p4J:2408t
Content-Type: text/html; charset=us/ascii

Pridej se <H1>k nam</H1>

--gc0p4J:2408t
Content-Type: audio/basic
Content-Transfer-Encoding: base64

UklGRkYtAABXQVZFZm10IBAAAAAABAEEAILYAACJWAAABAAGAZGF0YSItAACAgICAgICAgICA
gICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICA
gICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICA
...

--gc0p4J:2408t
```

Dalšími speciálně používanými MIME typy jsou Multipart/Signed, který specifikuje zprávu skládající se ze dvou částí (zpráva a elektronický podpis této zprávy) a Multipart/Encrypted pro zprávu zašifrovanou (zprávu v elektronické obálce). Není definován typ pro zprávy elektronicky podepsané a zároveň šifrované. Na takovýto požadavek je nutné uplatnit oba typy postupně za sebou.

Této problematice se budeme podobněji věnovat v samostatném textu.

Znaky v hlavičce, které nejsou ASCII

Znaky, které nejsou ASCII, by se v žádném případě neměly v záhlaví zprávy vyskytnout. Na otázku co se stane, když se tam takový znak vyskytne, není jednoznačná odpověď. Zpráva může dojít příjemci bez problému, zpráva může být nějakým serverem na cestě vrácena nebo se může i ztratit. RFC2047 řeší otázku jak do parametrů hlaviček dodat ne-ASCII znaky.

Problém se i zde skládá ze dvou částí. První je „Co takový znak vyjadřuje (obdoba Content-Type)?“ Např. hexadecimálně znak F8 může v jedné abecedě představovat ř a v jiné azbukové š. Druhá pak „Jak je kódován do ASCII (obdoba Content-Transfer-Encoding)?“.

Syntaxe parametru hlavičky je v tomto případě: =?charset?kódování?řetězec?=-

Parametr kódování je buď **b** pro **Base64** nebo **q** pro **quoted printable**.

Například, pokud chce odesílatel Václav Vopička uvést do zprávy své jméno v diakritice, tak musí zabezpečit správné kódování hlavičky: From: =?iso8859-2?q?V=E1clav Vopi=E8ka?=-

Nejčastější SMTP hlavičky

Hlavičky končí prvním prázdným řádkem a obsahují údaje ve formátu:

<klíč> : <parametry>

Klíč musí začínat na začátku řádku. Delší parametry mohou být rozděleny do více řádků, každý další řádek se uvozuje bílým místem. Hlavičky se dělí na strukturované a nestrukturované. Nestrukturované se počítačem nezpracovávají a jejich parametry mohou obsahovat libovolnou posloupnost znaků (př: Subject:). Strukturované jsou tvořeny tak, aby mohly být analyzované počítačem a jejich vytváření má určitá pravidla, například zápis adres jako <adresa>, komentářů (komentář), textu obsahujícího speciální znaky "text s mezerou", doménového literálu [ip.ad.re.sa]. Takto je např. tvořena hlavička Received:

From: adresy odesílatelů
To: adresáti
Cc: adresy na které se dopis odešle i když nejsou hlavními adresáty
Bcc: tajné kopie. Jako cc, ale adresy se vymažou z ostatních mailů
Subject: předmět
Return Path: zpáteční cesta k odesílateli
Received: záznamy přidané během zpracování
Replay to: adresa pr zaslání odpovědi
Sender: adresa skutečného odesílatele pokud je jiná než From: nebo je ve From: více adres.
Message ID: identifikace zprávy
Resent-klic: při přeposílání se klíče původních hlaviček uvodí Resent-
X-neco: doplňující hlavička (např. X-priority: 3)
Mime-version: požítá verze MIME. viz dále

Minimální hlavičky obsahují klíče From:, Date: a To: nebo Bcc: