Aplikační protokoly

- Klasické protokoly
  - Telnet – terminál
    - → ssh šifrovaný terminal
  - FTP – přenos souborů
  - SMTP – protokol elektronické pošty
    - → Nadstavba MIME
    - → POP, IMAP - přístup k schránce
- Moderní protokoly
  - WWW
    - → URL
    - → http v0,9 1,0, 1,1, http/2 a http/3
    - → html, xml
    - → relace

Michal Račák

Aplikační protokoly

- Routovací protokoly
  - RIP, (OSPF), BGP
- Speciální protokoly
  - NTP – čas
  - NNTP – newsy
  - SNMP – řízení sítě
  - LDAP – autentizace
- Okamžité zprávy
  - ICQ, IRCP, XMPP
- Proudové služby
  - RTSP, RTP
- Peer to peer
  - Gnutella, eMule, bitttorent
- Telefonování
  - SIP – navazování spojení
  - Skype
  - MGCP – řízení bran
- Anonymizační protokoly
  - TOR, I2P, Freenet
- Vzdálené disky
  - NFS, CIFS

Michal Řeřicha

Fyzické sítě

- Drátové
  - historie obsahuje řadu technologií (DECnet, Token ring, atd.)
  - přežil jen Ethernet
- Bezdrátové
  - IEEE 802.11 – Wi-Fi – různé rychlosti
  - Dvoubodové spojení – bezdrátový spoj mezi dvěma pevnými body
  - GSM (UMTS HSPDA) – mobilní sítě

Michal Řepka

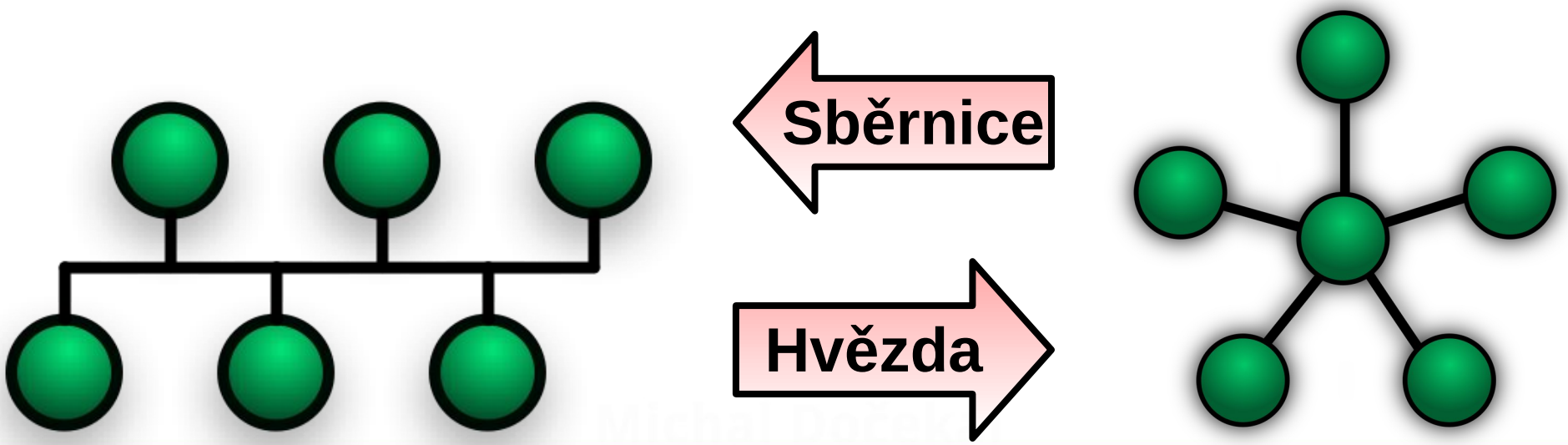
Ethernet: historie

- Původně projekt v laboratořích PARC Xeroxu
- Původně koaxiální kabel, později kroucená dvojlinka a nakonec i optika
- 1975: rychlost 3Mb/s
- 1979: 10Mbit a postupné zvyšování rychlosti
- sběrnicová technologie využívající technologii CSMA/CD pro přístup ke sdílenému médiu (koaxiálnímu kabelu) pro detekci a zabránění kolizí

Michal Račák

Ethernet: historie

- příchod kroucené dvojlinky
  - kroucená dvojlinka umožňuje vytváření pouze dvoubodových spojů
  - pro propojení více zařízení je potřeba hub nebo switch, ke kterému je počítač připojen samostatným kabelem



Ethernet: vlastnosti

- **48 bitová MAC adresa**
- **Rámce** s hlavičkou a patičkou
- Hlavička rámce obsahuje adresu síťového rozhraní **odesílatele** a **příjemce**
- Patička rámce obsahuje **kontrolní součet**

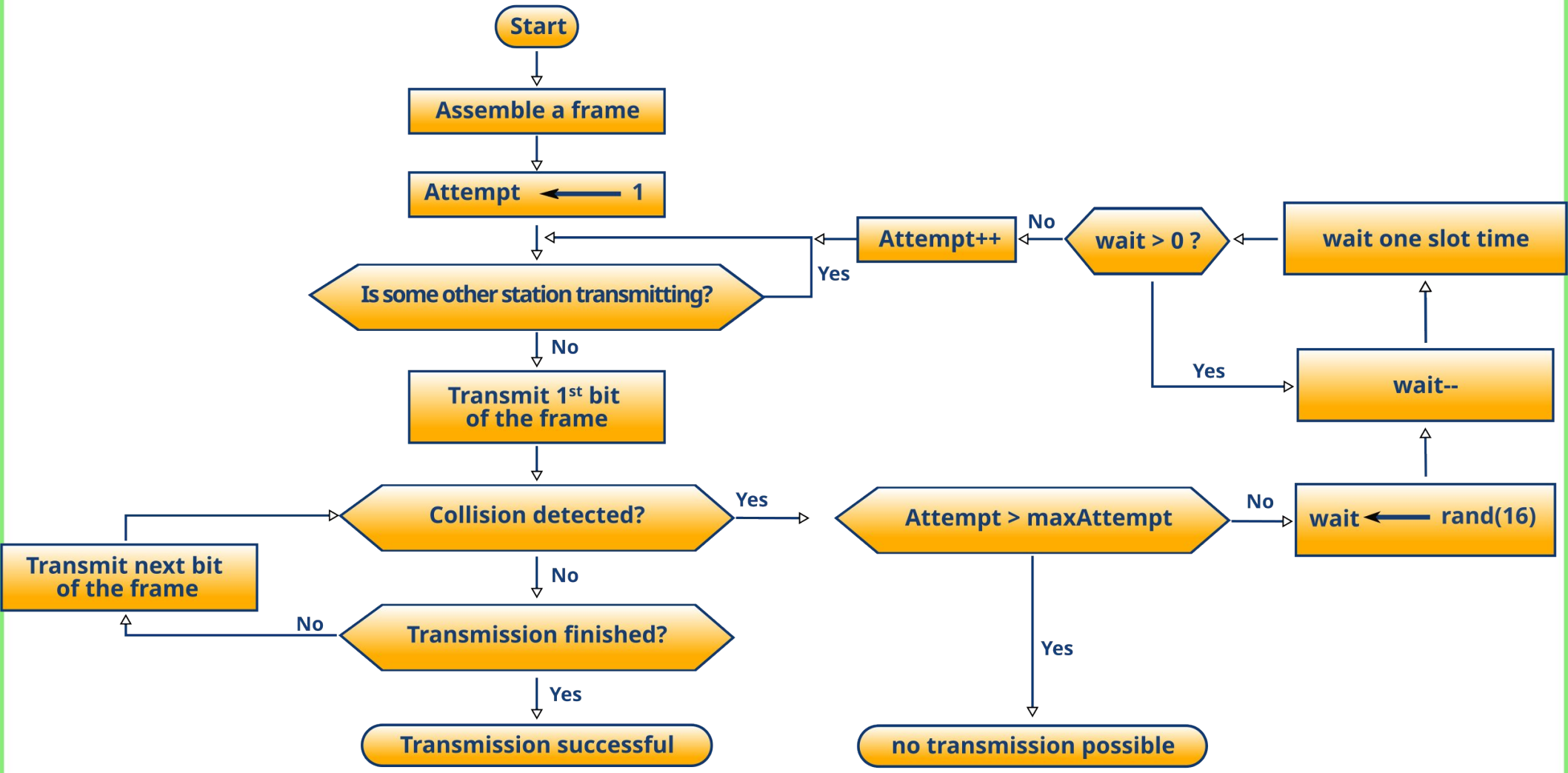
Michal Bečák

Přístupové metody ke sdílenému médiu

- **CSMA/CD**: **C**arrier **S**ense **M**ultiple **A**cces with **C**ollision **D**etection (ethernet)
- uzel, který chce vysílat, **zkontroluje**, jestli nevysílá někdo jiný
  - **ano**: počká, až bude klid
  - **ne**: začne vysílat a současně hlídá, jestli nevysílá někdo jiný
- v případě kolize uzel odvysílá signál indikující kolizi, odmlčí se na náhodně stanovenou dobu a pak se pokusí o nové vysílání

Michal Račák

CSMA/CD algoritmus (pro zajímavost)



Ethernetový rámec

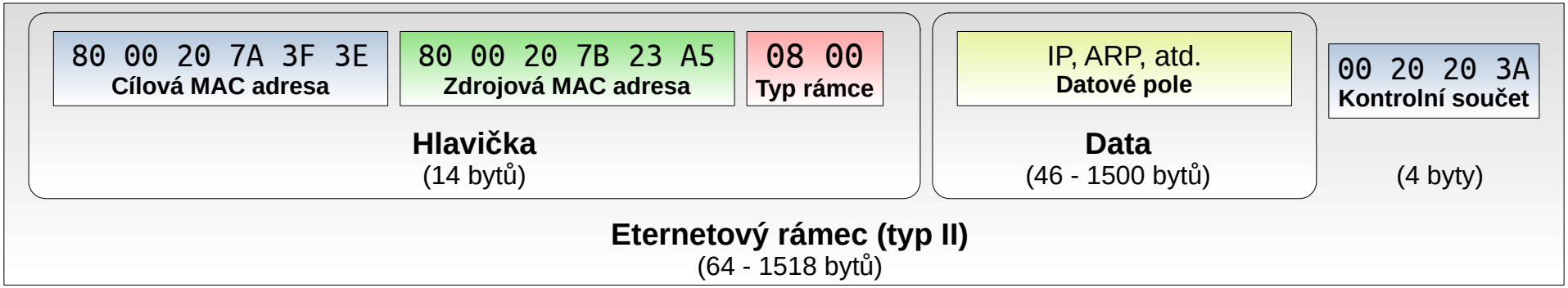
- **minimální** velikost rámce: **64 bytů** kvůli schopnosti spolehlivě detekovat kolizi
  - odesílatel potřebuje být schopen detekovat **kolizi** ještě **před** odvysíláním **celého rámce**
  - a datům nějakou dobu trvá, než obsadí celý kabel, a umožní tak účastníkům sběrnice detekovat cizí vysílání
  - bez tohoto minima by bylo možné aby dvě strany odvysílaly krátký rámec **současně**, aniž by detekovali cizí vysílání, a došlo by tak k **nezjištěné kolizi** a zkomolení (narušení integrity) dat

Ethernet: duplex

- **poloviční duplex** (half duplex)
  - „Tady Orel, přepínám...“
  - může vysílat vždy jen jedna strana
  - může docházet ke kolizím (tj. je třeba CSMA/CD)
- **plný duplex** (full duplex)
  - obě strany vysílají současně přes nezávislé kanály
  - nedochází ke kolizím

Michal Řepka

Struktura ethernetového rámce



- 48 bitová fyzická (nebo také MAC) adresa

  `80:00:20:7B:23:A5`

- Rámce mají hlavičku, data a patičkou
  - hlavička obsahuje fyzickou (MAC) adresu odesílatele a příjemce
  - patička obsahuje kontrolní součet

Michal Pokorný

Fyzické adresy

- ethernetová (fyzická či MAC) adresa je jedinečný identifikátor **síťového rozhraní** (network interface: síťová karta, wifi karta, apod.)
  - je přiřazována danému zařízení při výrobě (odtud *fyzická adresa*)
  - u moderních karet jde změnit
  - 48 bitů, zapisuje se jako šestice dvouciferných hexadecimálních čísel oddělených oddělovačem (dvojtečka, pomlčka, atd.)

Fyzické adresy

- první polovina adresy (24 bitů) je přidělena jednotlivým výrobcům, zbytek si definují výrobci sami

MAC-48: původní standard, přejmenováno na:

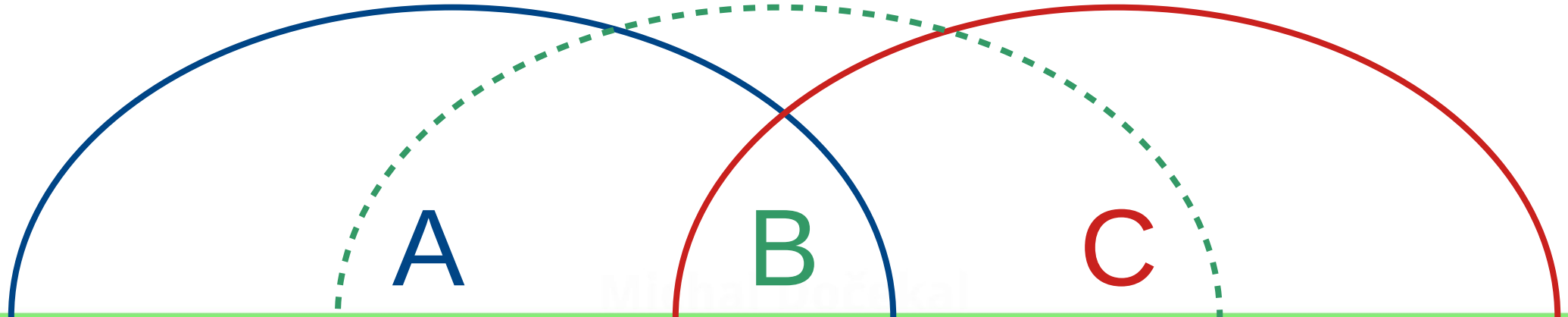
EUI-48: nový standard, rozšíření použití i na nehardwarová síťová rozhraní (např. Bluetooth)

EUI-64: rozšířený adresní prostor, použití např. IEEE 1394 (FireWire) a IPv6 – modifik. EUI-64

Michal Pořízek

Přístupové metody (WiFi)

- CSMA/CA: Carrier Sense Multiple Acces with Collision Avoidance
  - **Carrier sense**: před vysíláním posloucháme, jestli už někdo jiný nevysílá
    - pokud ne, nemusí to ovšem znamenat, že někdo jiný opravdu nevysílá (viz ***problém skrytého uzlu:***)

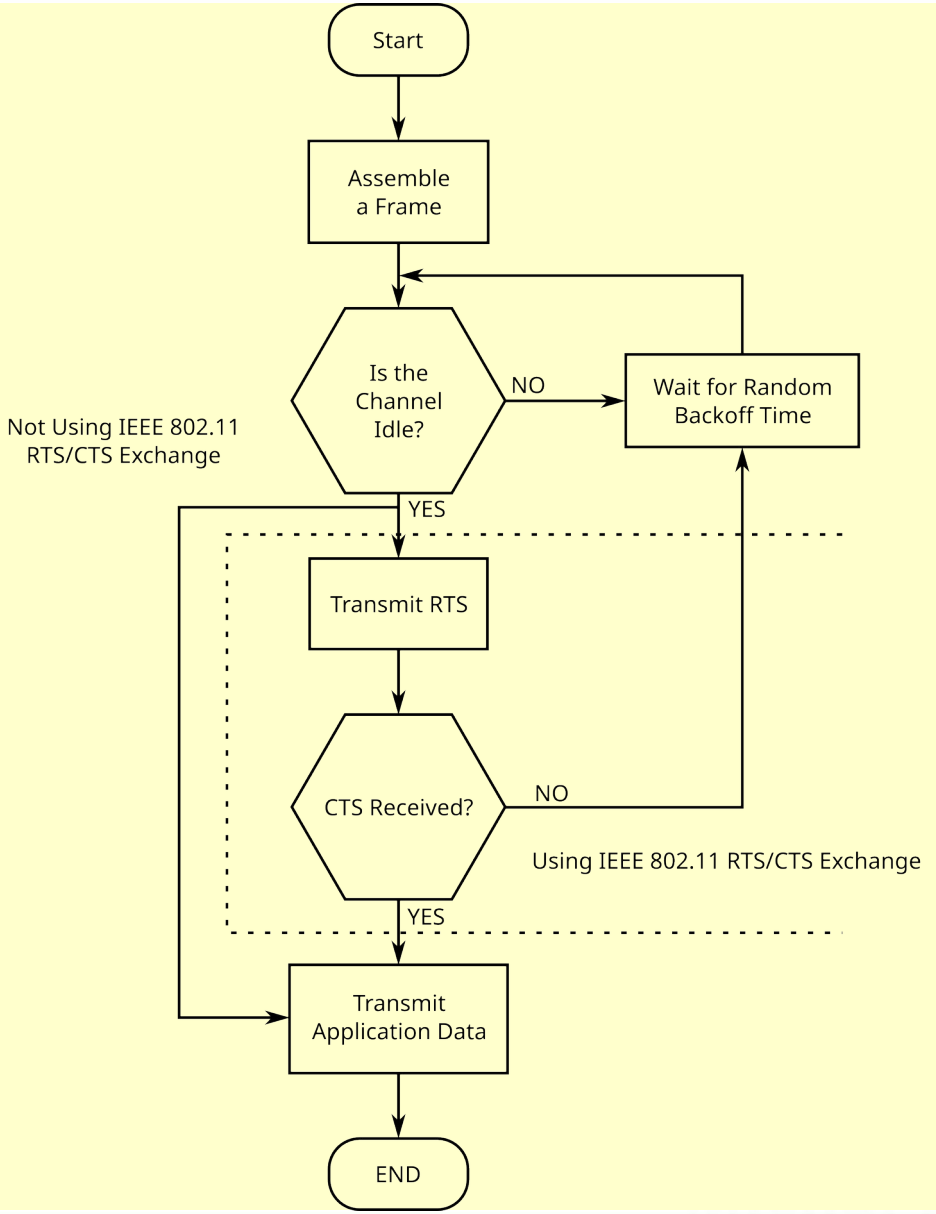


Přístupové metody (WiFi)

- CSMA/CA: Carrier Sense Multiple Acces with Collision Avoidance
  - **Carrier sense**: před vysíláním posloucháme, jestli už někdo jiný nevysílá
    - pokud ne, nemusí to ovšem znamenat, že někdo jiný opravdu nevysílá (viz *problém skrytého uzlu*)
  - **Collision avoidance**: když někdo vysílal, čekáme náhodně zvolenou dobu až dotyčný vysílat přestane, a teprve pak znovu posloucháme, jestli někdo nevysílá
  - **IEEE 802.11 RTS/CTS** – částečné řešení problému skrytého uzlu – odesílatel se nejdřív ptá, jestli může vysílat, a vysílá, až dostane signál, že smí

Michal Pecelka

CSMA/CA algoritmus (pro zajímavost)



- RTS: request to send
- CTS: clear to send

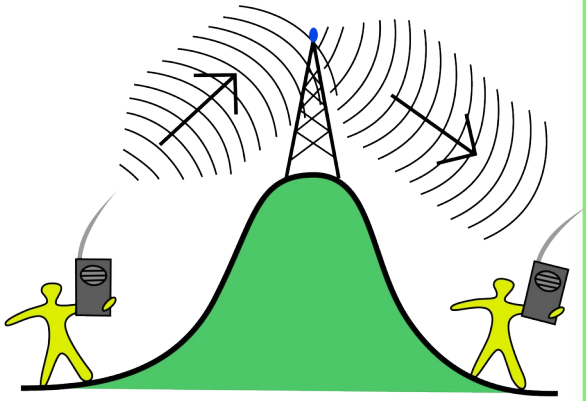
Technologická zařízení

- **Repeater** and **hub**
  - Na elektrické rovině
  - Pouze obnovují signál
- **Bridge** and **switch**
  - Oddělení provozu
  - Zasílání pouze příslušným MAC adresám
- **Router**
  - Pracuje na úrovni IP protokolu (síťová vrstva)

Michal Račák

Opakovač (repeater)

- pracují na **fyzické** vrstvě (L1)
- **rozšiřují** segment sítě
  - zesiluje či regeneruje (opakuje) signál, a prodlužuje tak jeho dosah, čímž umožňuje propojit více zařízení v jednom segmentu sítě
  - na opakovač dorazí slabší signál, opakovač jej odvysílá plnou silou dál
- v sítích jsou **transparentní** (nejsou vidět)
- v moderních sítích: WiFi, optika, atd.

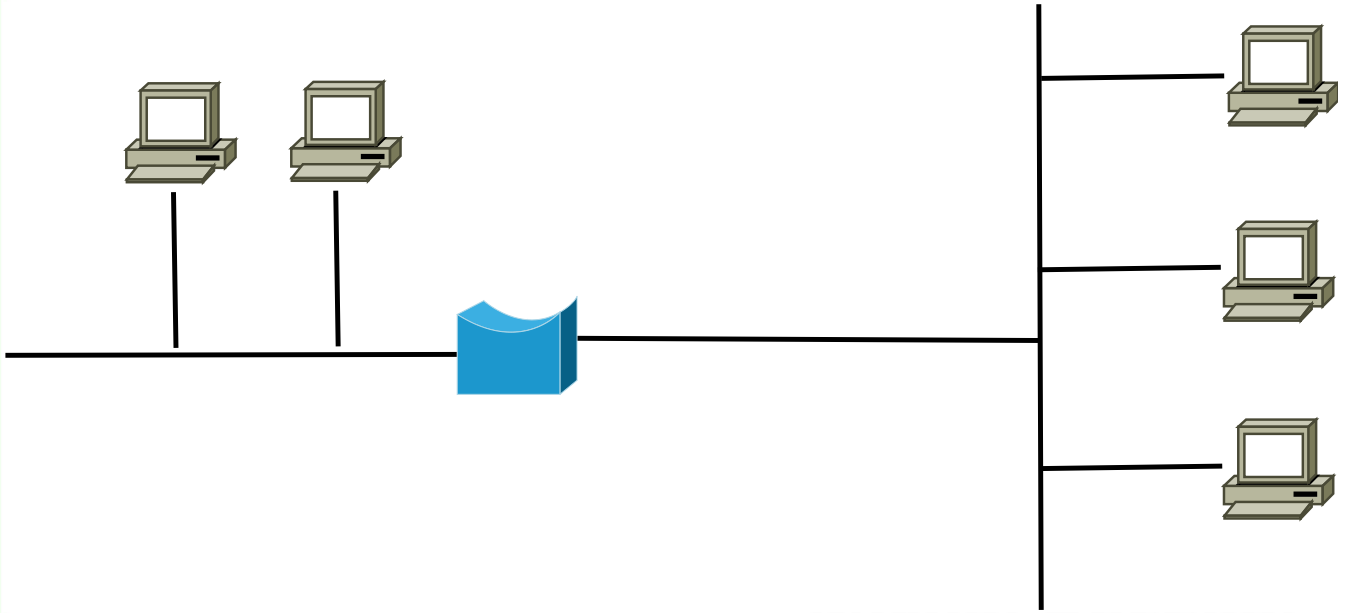


Rozbočovač (hub)

- **rozbočovače** (huby) - předchůdci switchů
  - fungují jako víceportové **opakovače** (repeater)
  - **opakují signál**
- pracují na **fyzické** vrstvě (L1)
- **rozšiřují segment** sítě, v síti jsou **transparentní**
- **nejsou selektivní**, přeposílají vše všem

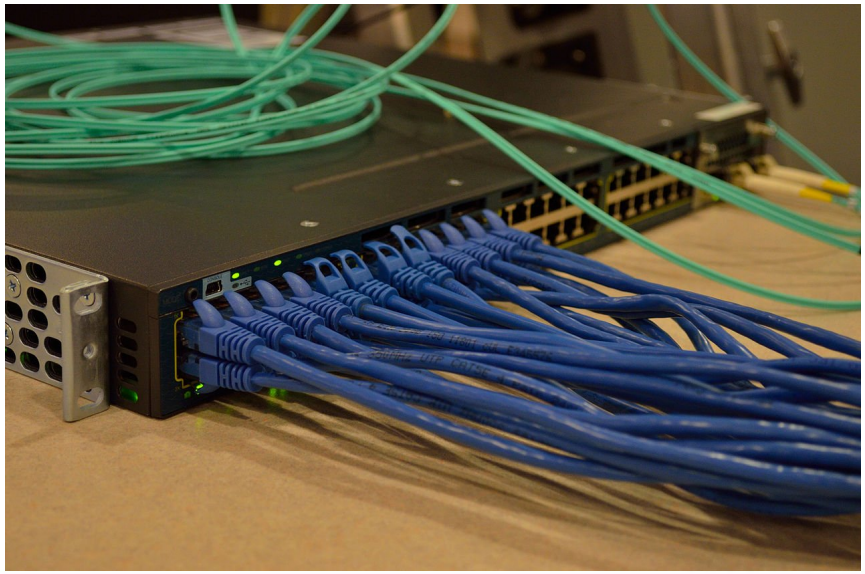
Mosty (bridge)

- pracují na **linkové vrstvě** (L2)
- **rozpoznávají** MAC adresy v přijatých datových rámcích a **selektivně** je posílají dál (oddělení provozu)
- **propojují** dva a více **segmentů** sítě do **jedné sítě**



Přepínače (switch)

- pracují na **linkové vrstvě** (L2)
- nejrozšířenější aktivní prvek lokálních sítí
- ve své podstatě je switch vylepšený víceportový most
- **inteligentně** (selektivně) předává ethernetové rámce mezi porty **v rámci jedné fyz. sítě**



Přepínače (switch)

- SOHO switch (na doma, do malé kanceláře: méně portů, automatizované, nelze konfigurovat)

Přepínače (switch)

- chytrý switch (více portů, konfigurovatelné přes nějaké rozhraní: web, příkazová řádka, ...)

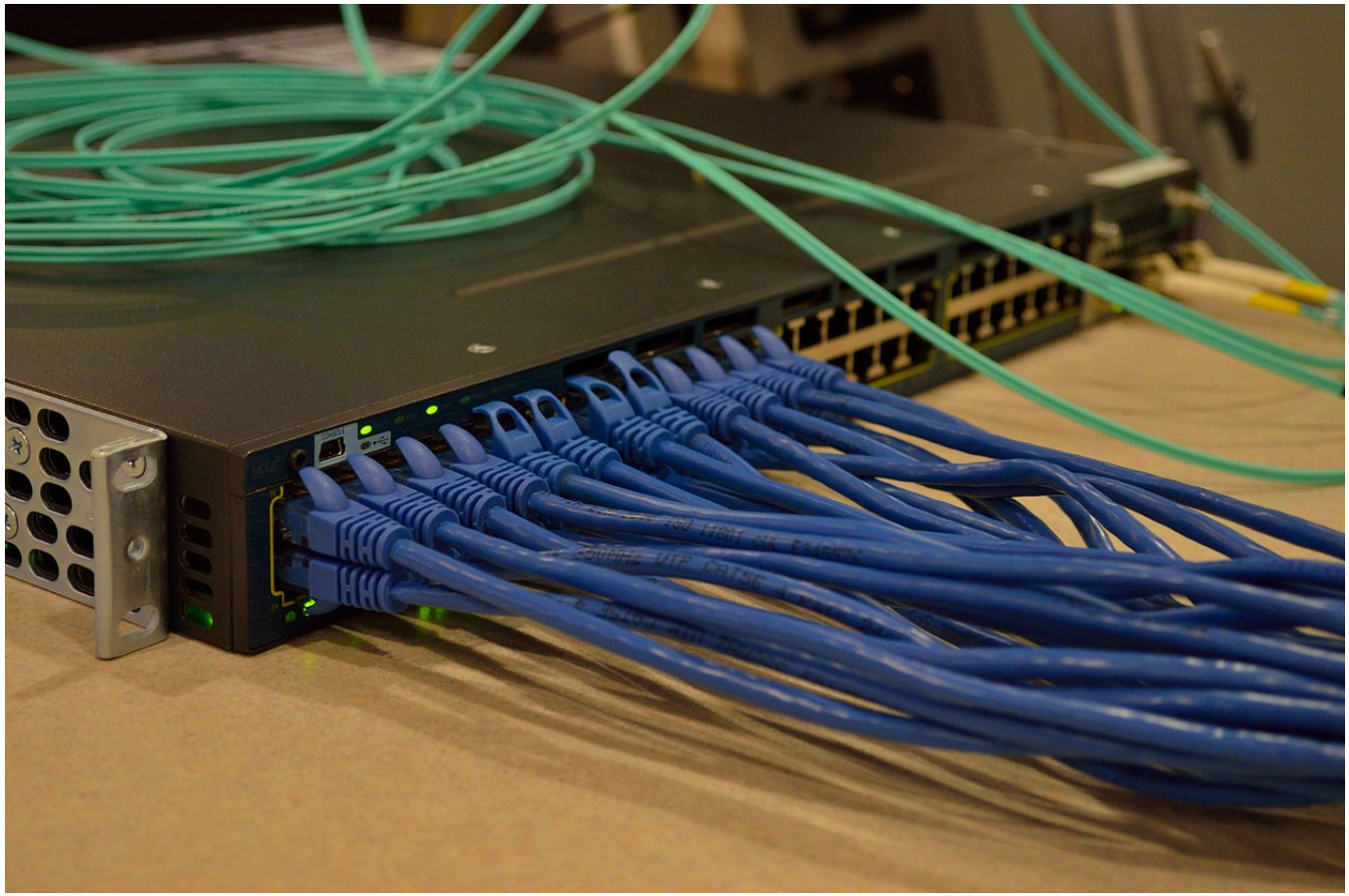
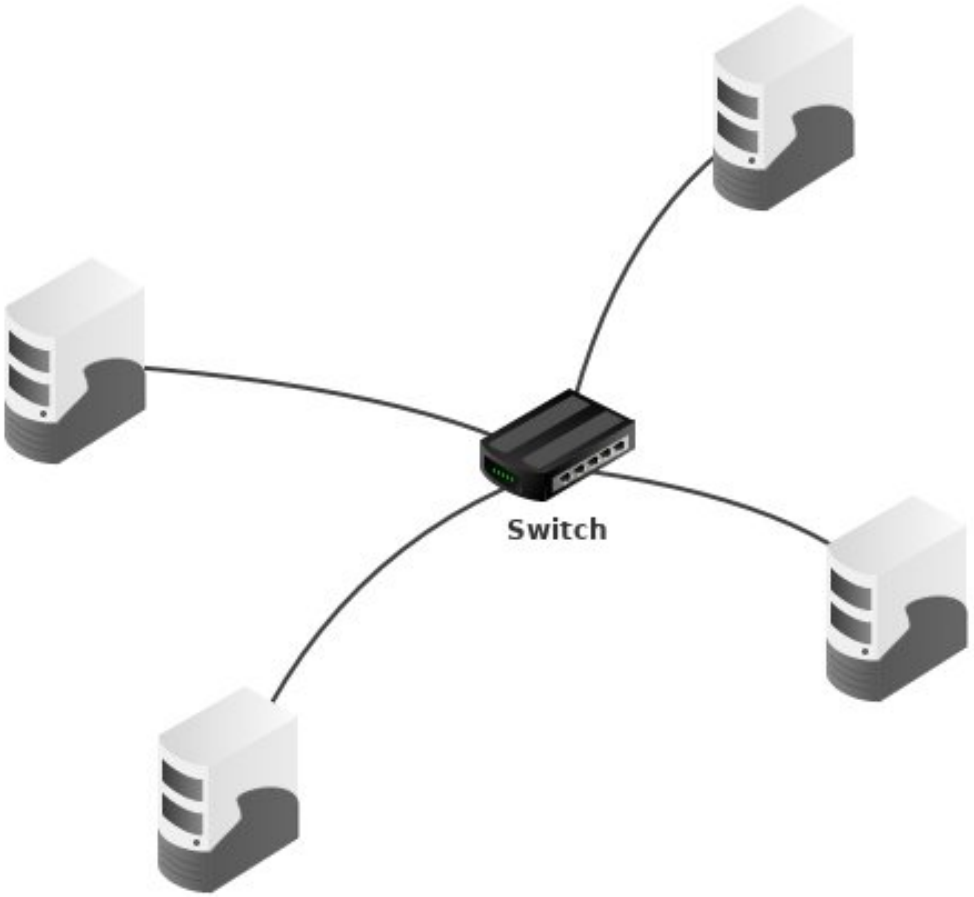


Schéma sítě, přepínač (switch)

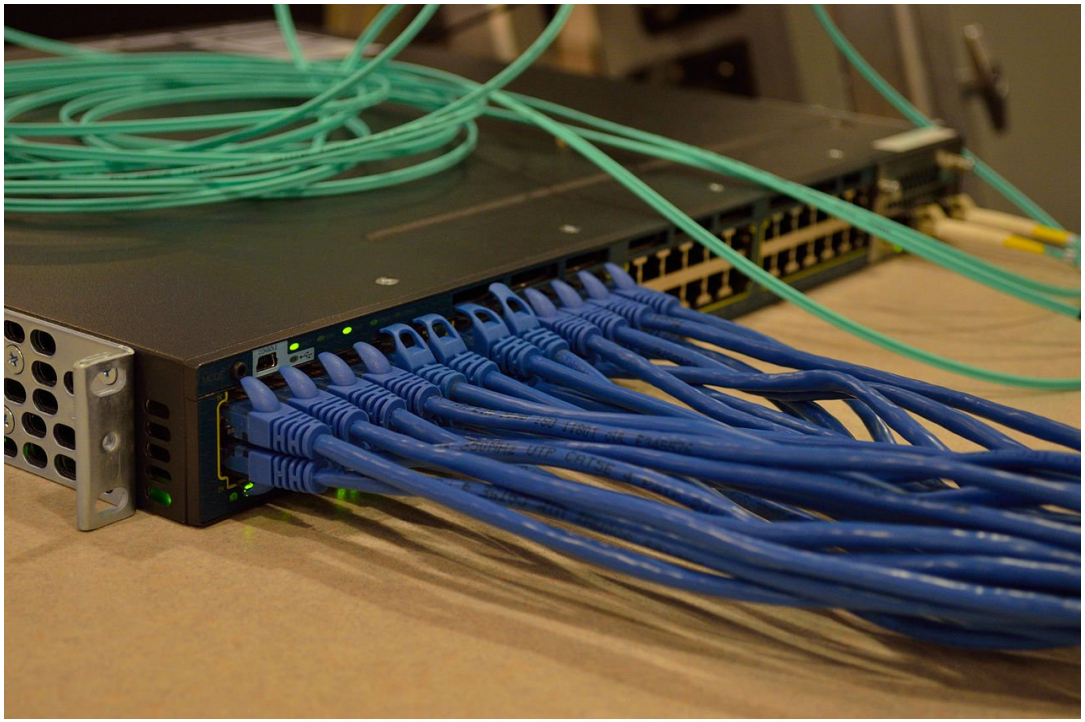


Směrovače (router)

- nejinteligentnější aktivní prvek
- pracují na síťové vrstvě, rozesílají pakety dle směrovacích tabulek
- propojují **více sítí**

L3 switch = switch + router

- jako L3 switch se označují zařízení, jež zajišťují jak funkci **switche**, tak funkci **routeru** (a často i další funkce)



Přehled aktivních síťových prvků

- **síťové karty** (network card)
  - zapojení počítače do fyzické sítě
  - rámce
- **přepínače** (switch)
  - propojení více počítačů ve stejné fyzické síti
  - rámce
- **směrovače** (router)
  - stojí na rozhraní dvou a více fyzických sítí a přenáší data mezi nimi
  - IP datagramy

Michal Račátek

Nízkoúrovňové protokoly

- **ARP**: Address Resolution Protocol
- protokoly pro **automatickou konfiguraci** síťového rozhraní (IP adresa a další náležitosti)
  - RARP: Reverse Address Resolution Protocol
  - BOOTP: Bootstrap Protocol
  - DHCP: Dynamic Host Control Protocol
- další protokoly
  - IARP: Inverse Address Resolution Protocol

Michal Rešetka

Nízkoúrovňové protokoly

- **ARP**: zjištění MAC adresy k IP adrese
- protokoly pro **automatickou konfiguraci** síťového rozhraní (IP adresa a další náležitosti)
  - RARP – mapování MAC → vlastní IP a nic víc
  - BOOTP – kompletní konfigurace, provoz bezdisk. stanic
  - DHCP – jako BOOTP + zápůjčky IP adres
- další protokoly
  - IARP – mapování MAC → cizí IP (ATM, Frame Relay)

Michal Rešetka

Chceme poslat IP datagram Ivovi

IP datagram

Honza
IP adresa: 192.168.0.1
Maska: 255.255.255.0
MAC: 12:34:56:78:9A:BC

Ivo
IP adresa: 192.168.0.2

eth. rámec

Hlavička IP	DST: 192.168.0.2 SRC: 192.168.0.1	DATA

Hlavička (ethernet)	DST MAC: ??? SRC MAC: 12:34:56:78:9A:BC	Patička

Michal Pokorný

ARP

- ARP = Address Resolution Protocol
- umožňuje se **v rámci místní sítě** zeptat, jakou **MAC adresu** má uzel s konkrétní **IP adresou**
- za tímto účelem se posílá ARP žádost

Hlavička (ethernet)	DST MAC: FF:FF:FF:FF:FF:FF SRC MAC: 12:34:56:78:9A:BC	Jakou MAC adresu má uzel 192.168.0.2? Tady 192.168.0.1 (12:34:56:78:9A:BC). Dík.	Patička

- ethernetový rámec se odesílá na **všesměrové vysílání** v místní síti (MAC FF:FF:FF:FF:FF:FF)
- dotčený uzel pak pošle ARP odpověď

Hlavička (ethernet)	DST MAC: 12:34:56:78:9A:BC SRC MAC: CB:A9:87:65:43:21	Uzel 192.168.0.2 má MAC: CB:A9:87:65:43:21. Nemáš zač.	Patička

Protokoly pro autokonfiguraci síťového rozhraní

- **RARP** (Reverse ARP)
  - funguje podobně jako ARP, jen v rámci fyz. sítě
  - mapuje MAC jen na IP adresu **a nic víc**
- **BOOTP** (Bootstrap Protocol, náhrada RARP)
  - využívá UDP, může poskytovat službu i v jiných sítích
  - 1985: poskytuje IP adresu, gateway, a informace ke stažení souboru s obrazem OS (umožňuje provoz bezdisk. stanic), časem rozšířen pro předávání dalších informací
  - statické mapování MAC na poskytnuté údaje

Michal Řeřicha

Protokoly pro autokonfiguraci síťového rozhraní

- **DHCP** (Dynamic Host Control Protocol)
  - založeno na BOOTP, umožňuje vše jako BOOTP
  - poskytuje IP adresu, masku, gateway, DNS servery, atd.
  - umožňuje stanovit rozsah IP adres pro **zápůjčky adres**
    - klient požádá, dostane IP na omezenou dobu, pak buď znovu požádá nebo se adresa vrací zpět do seznamu volných adres k zapůjčení (a klient ji musí přestat používat)

Princip funkce DHCP

- Klient používá port **68** → server poslouchá na portu **67**
- Klient odešle paket **DHCPDISCOVER** na lokální broadcast
  - cílová IP adresa: `255.255.255.255`, cílová MAC: `FF:FF:FF:FF:FF:FF`
- Server odpoví paketem **DHCPOFFER** s nabídkou IP adresy
  - Klient může obdržet více nabídek, ale smí si vybrat pouze jednu
  - Klient by si měl ověřit, zda-li nabídnutou IP adresu již někdo nepoužívá
  - Klient pošle ARP dotaz a neobdrží-li odpověď, považuje IP adresu za volnou
- Následně o IP adresu požádá server paketem **DHCPREQUEST**
- Server zápůjčku potvrdí paketem **DHCPACK** a klient si ji nastaví
- Klient musí před uplynutím doby zapůjčení obnovit
- Uplyne-li lhůta zapůjčení bez obnovení zápůjčky, musí klient IP adresu přestat používat a DHCP server ji může znovu nabídnout jinému klientovi

Michal Pecháč

Zdroje: Wikipédie, RFC

Elektronická pošta

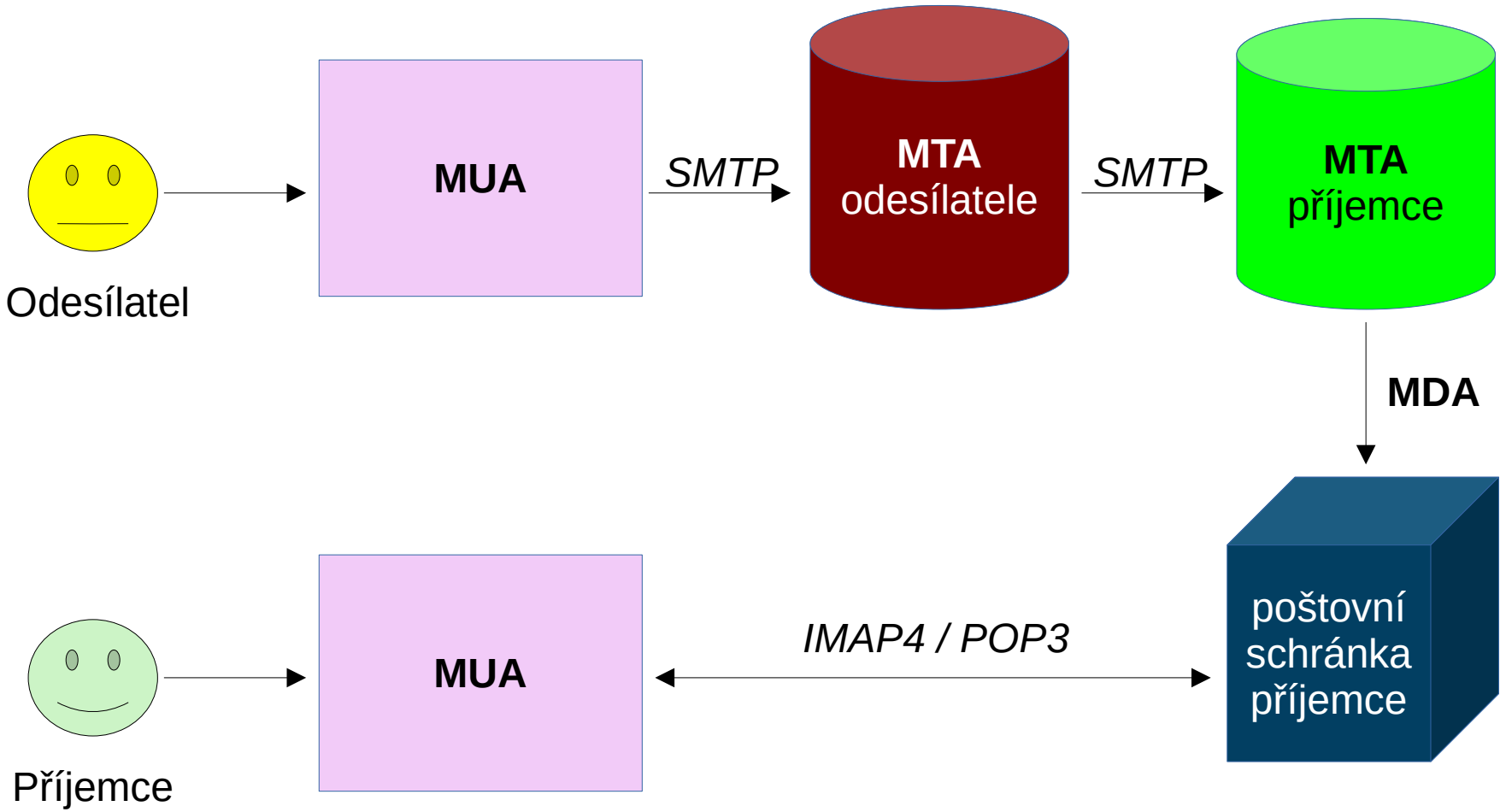
SMTP

POP

IMAP

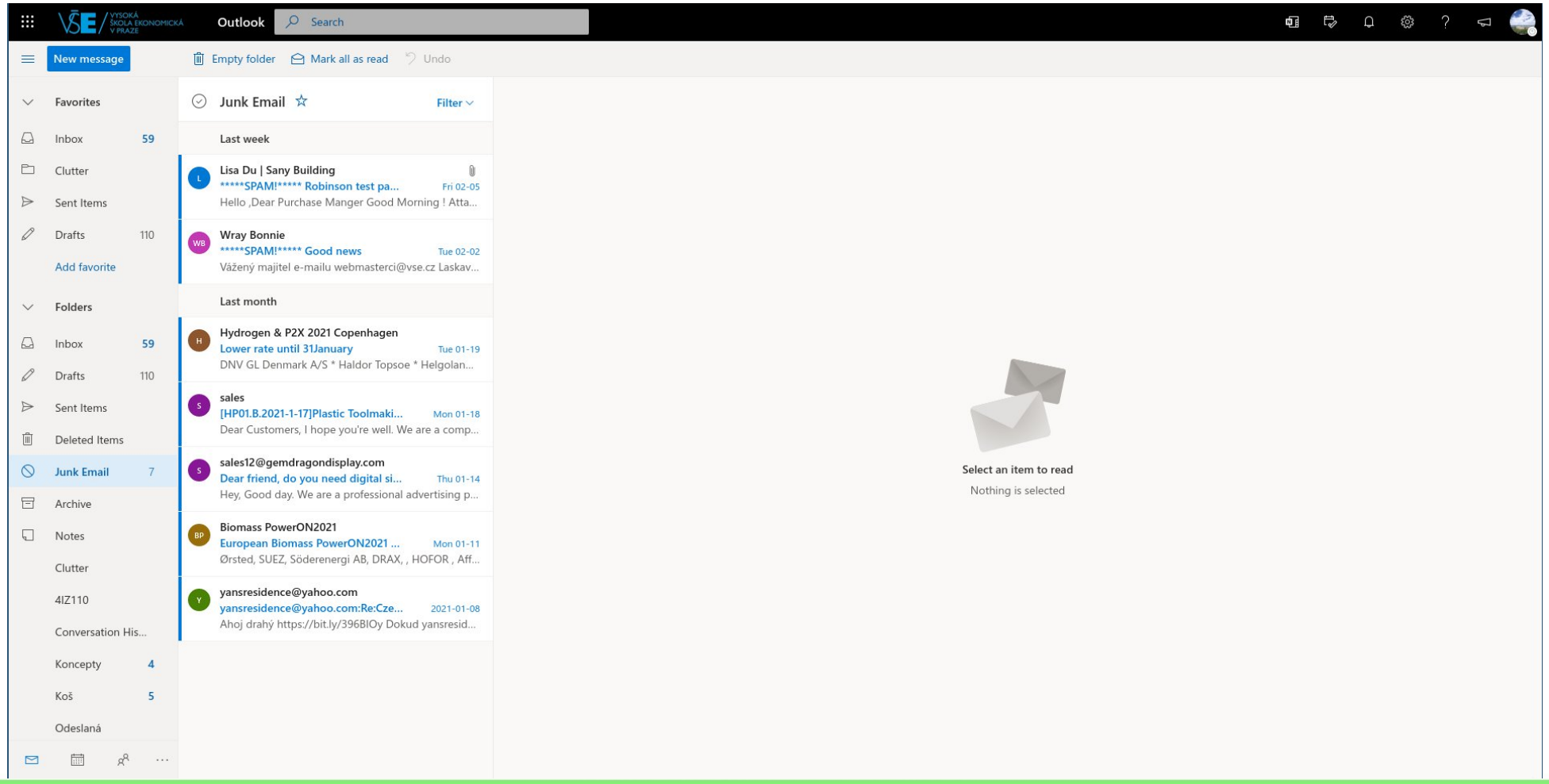
Michal Pořízek

Životní cyklus e-mailu



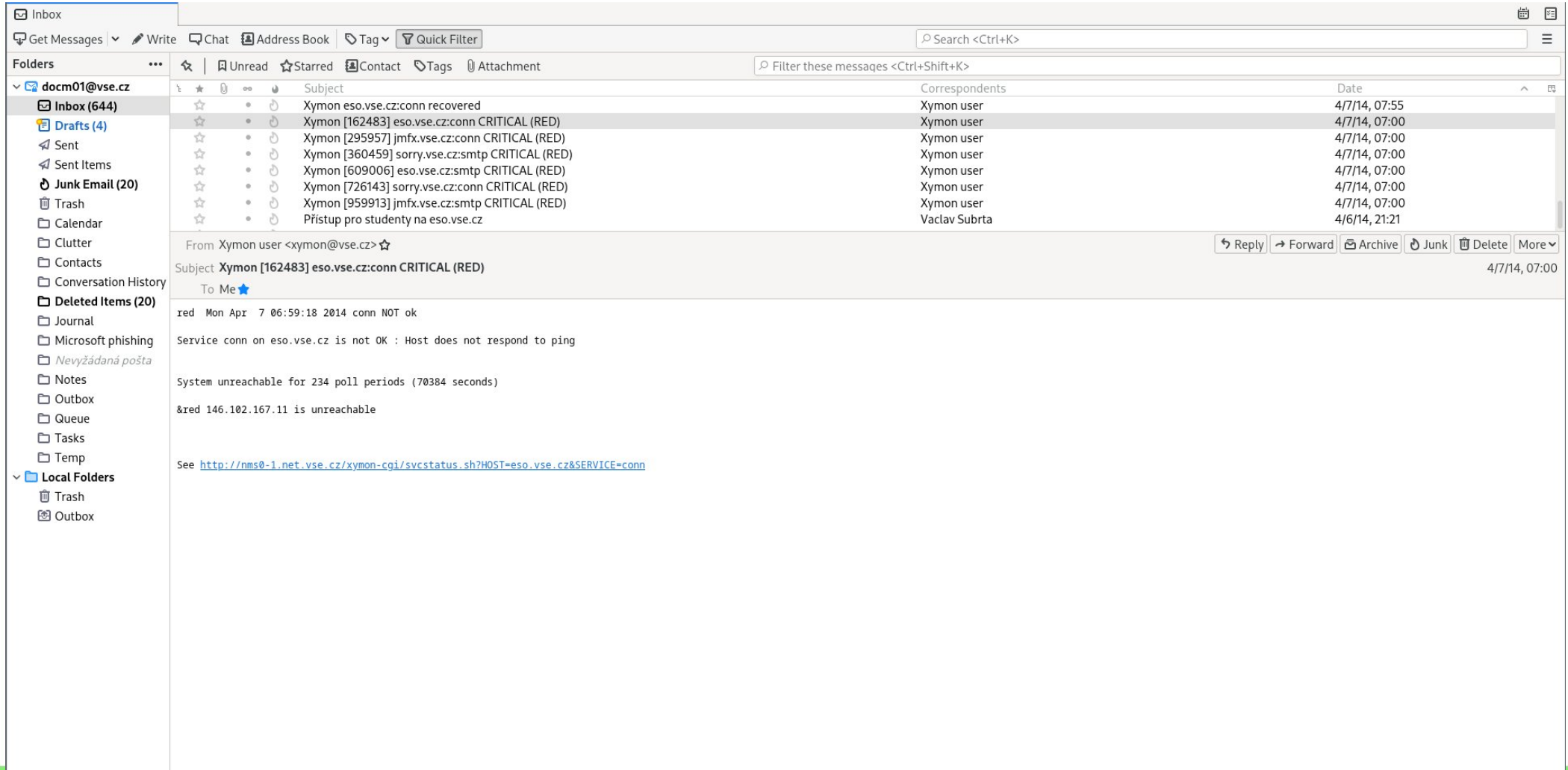
MUA: Mail User Agent

- klientský software pro práci s poštou (zde webové rozhraní)



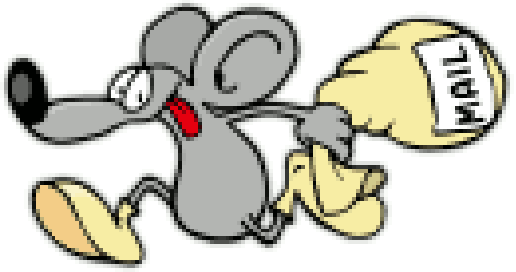
MUA: Mail User Agent

- klientský software pro práci s poštou (zde e-mailový klient)



MTA: Mail Transfer Agent

- software pro přenos/doručování pošty
- poštovní server
- **protokol SMTP**



POSTFIX

exim

www.exim.org

sendmail.org

MDA, MSA

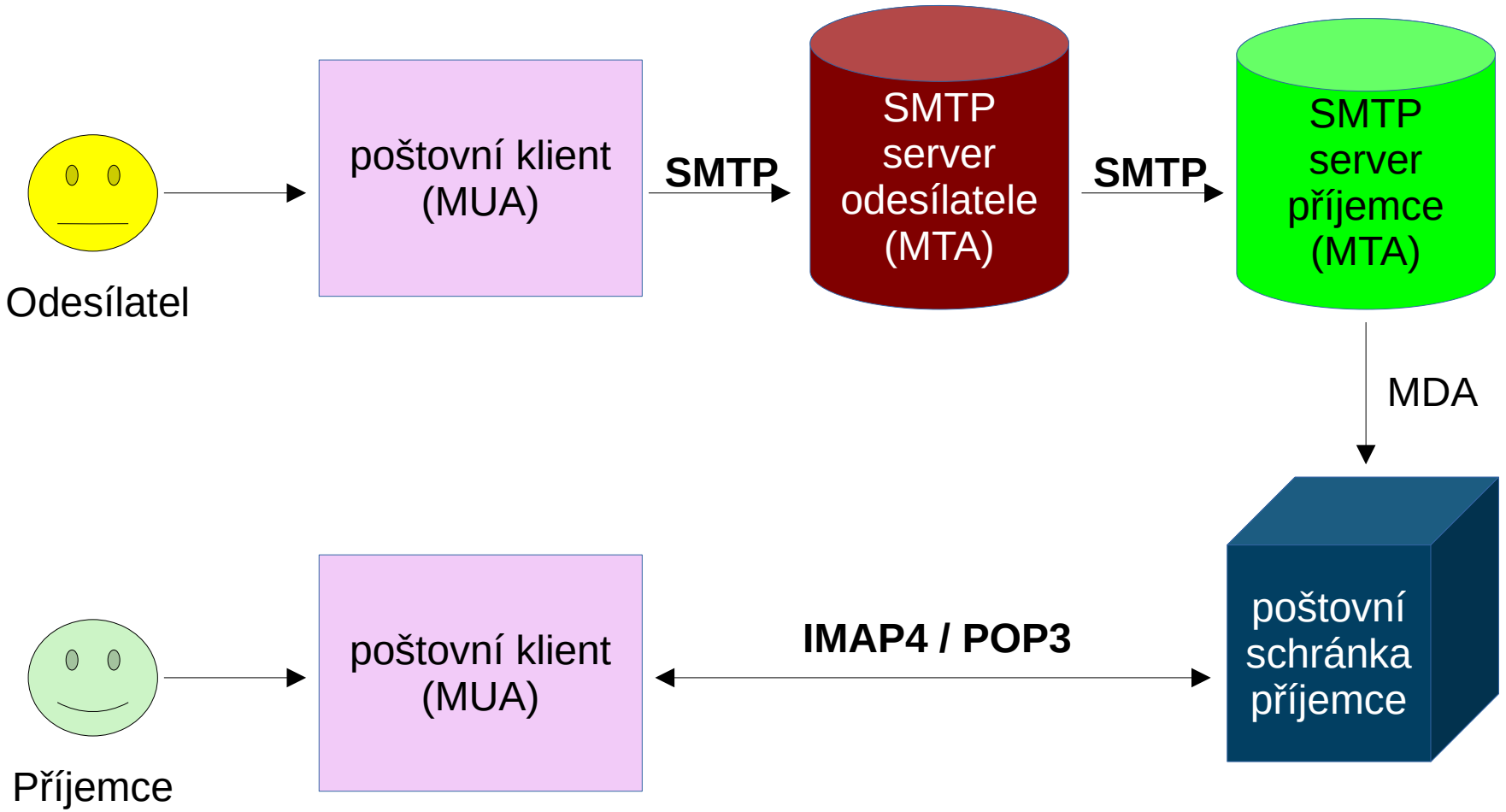
- **MDA**: Mail Delivery Agent – software doručující e-mail do schránky příjemce
  - zpravidla přidruženo k SMTP serveru, např. *maildrop* či *procmail*
- **MSA**: Mail Submission Agent – software přebírající od klienta poštu k doručení
  - také SMTP server
  - zpravidla omezená funkčnost oproti **MTA**
  - roli **MSA** ale může zastávat i „plnohodnotný“ **MTA** (tj. SMTP server)

Aplikační protokoly elektronické pošty

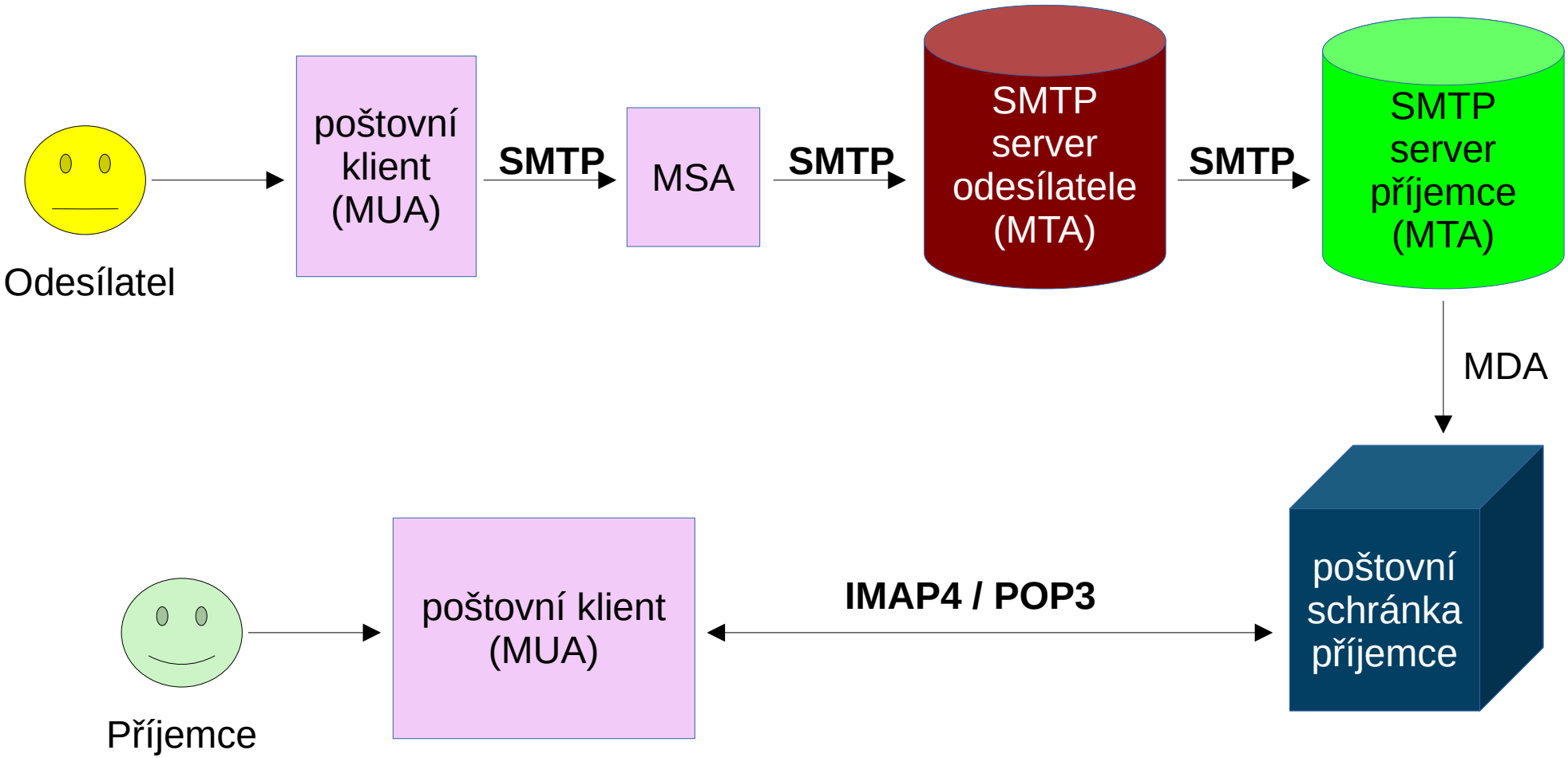
- **SMTP**: simple mail transfer protocol
  - přenos e-mailů mezi poštovními servery
  - využívá **DNS**
    - poštovní servery pro konkrétní doménu (např. vse.cz) jsou specifikovány pomocí MX záznamů
    - MX záznamy mají prioritu
- **IMAP4, POP3**: přístup uživatele k poštovní schránce

Michal Počtek

Životní cyklus e-mailu



Životní cyklus e-mailu



Protokol SMTP

Simple
Mail
Transfer
Protocol

Michal Pořízek

Protokol SMTP

- vznikl v roce 1981, RFC 788
- **textový protokol**
- TCP port 25
- využívá DNS
- předpokládá se stálá dostupnost poštovního serveru (s krátkodobým výpadkem si poradí)
- protokol je typu "push" – kdo má data, vyvolá spojení a data "tlačí" dále (doručuje e-mail)

Michal Račinský

Protokol SMTP

- omezená množina použitelných znaků: 7bitové ASCII
  - → nepodporuje přenos binárních dat
  - → ani znaky národních abeced

Michal Bečák

SMTP: textový protokol

```
220 krkavec.net ESMTP Postfix (Debian/GNU)
HELO vse.vse.cz
250 krkavec.net
MAIL FROM: docm01@vse.cz
250 2.1.0 Ok
RCPT TO: root@krkavec.net
250 2.1.5 Ok
DATA
354 End data with <CR><LF>.<CR><LF>
Subject: Ahoj

Ahoj svete.
.
250 2.0.0 Ok: queued as 81DF31BD732
```

SMTP: textový protokol

```
server:  220 krkavec.net ESMTP Postfix (Debian/GNU)
klient:  HELO vse.vse.cz
server:  250 krkavec.net
klient:  MAIL FROM: docm01@vse.cz
server:  250 2.1.0 Ok
klient:  RCPT TO: root@krkavec.net
server:  250 2.1.5 Ok
klient:  DATA
server:  354 End data with <CR><LF>.<CR><LF>
klient:  Subject: Ahoj

         Ahoj svete.
         .
server:  250 2.0.0 Ok: queued as 81DF31BD732
```

Příkazy SMTP relace

HELO `fqdn`	ohlášení klienta, původní SMTP *fqdn*: plně kvalifikované doménové jméno
MAIL FROM: `mail`	e-mail odesílatele, uvozuje doručování nové zprávy např. `MAIL FROM:<docm01@vse.cz>`
RCPT TO: `mail`	e-mail příjemce (může se opakovat) např. `RCPT TO:<karel@krkavec.net>`
DATA	přechod do režimu posílání zprávy, ukončuje se sekvencí: `<ENTER> . <ENTER>`
RSET	zruší aktuální transakci
VRFY `mail`	ověří, zda-li existuje daná schránka (bývá vypnuto)
NOOP	neprovede nic, udržuje spojení aktivní
QUIT	požádá server o ukončení spojení

E-mailová adresa a DNS

- přesná specifikace viz RFC 2821, sekce 5
- v tomto případě SMTP server vyšle DNS dotaz na MX záznamy pro doménu „krkavec.net“

Michal Rafajal

E-mailová adresa a DNS

- dostane odpověď:

```
krkavec.net.    IN    MX    20 spathiphyllum.cz.
krkavec.net.    IN    MX    10 krkavec.net.
```

- vybere server s nejvyšší prioritou

Michal Ročkai

E-mailová adresa a DNS

- dostane odpověď:

```
krkavec.net.    IN    MX    20 spathiphyllum.cz.
krkavec.net.    IN    MX    10 krkavec.net.
```

- vybere server s nejvyšší prioritou

Michal Ďačka

E-mailová adresa a DNS

- dostane odpověď:

```
krkavec.net.    IN    MX    20 spathiphyllum.cz.
krkavec.net.    IN    MX    10 krkavec.net.
```

- vybere server s nejvyšší prioritou

Michal Ďačka

E-mailová adresa a DNS

- a tomuto serveru zkusí doručit e-mail
- pokud to nejde, zkusí server s nižší prioritou
- když vystřídá všechny, tak chvíli počká a zkusí to znovu

Michal Ročárek

E-mailová adresa a DNS

- jak se zachová v této situaci?

```
krkavec.net.    IN    MX      10 cervicek.cz.
krkavec.net.    IN    MX      10 smirak.eu.
```

Michal Ďačka

E-mailová adresa a DNS

- jak se zachová v této situaci?

náhodný výběr

```
krkavec.net.    IN    MX    10 cervicek.cz.
krkavec.net.    IN    MX    10 smirak.eu.
```

Michal Řeřicha

E-mailová adresa a DNS

- jak se zachová, pokud nebudou k dispozici žádné MX záznamy pro danou doménu?

Michal Ročkai

E-mailová adresa a DNS

- jak se zachová, pokud nebudou k dispozici žádné MX záznamy pro danou doménu?

využije se **A** záznam, ale jen pokud MX záznam chybí

Michal Řeřicha

Omezená množina použitelných znaků

- 1 byte = 8 bitů
- binární data (programy, audiovizuálie, dokumenty, atd.) využívají všech 8 bitů na byte
- textová data *mohou* využívat pouze prvních 7 bitů
- a to je i případ e-mailu
- v e-mailu tedy nejde použít nic kromě 7-bitového ASCII
  - Hodnoty **`0-127`**, binárně **`0000 0000`** – **`0111 1111`**

Michal Račák

ASCII TABLE

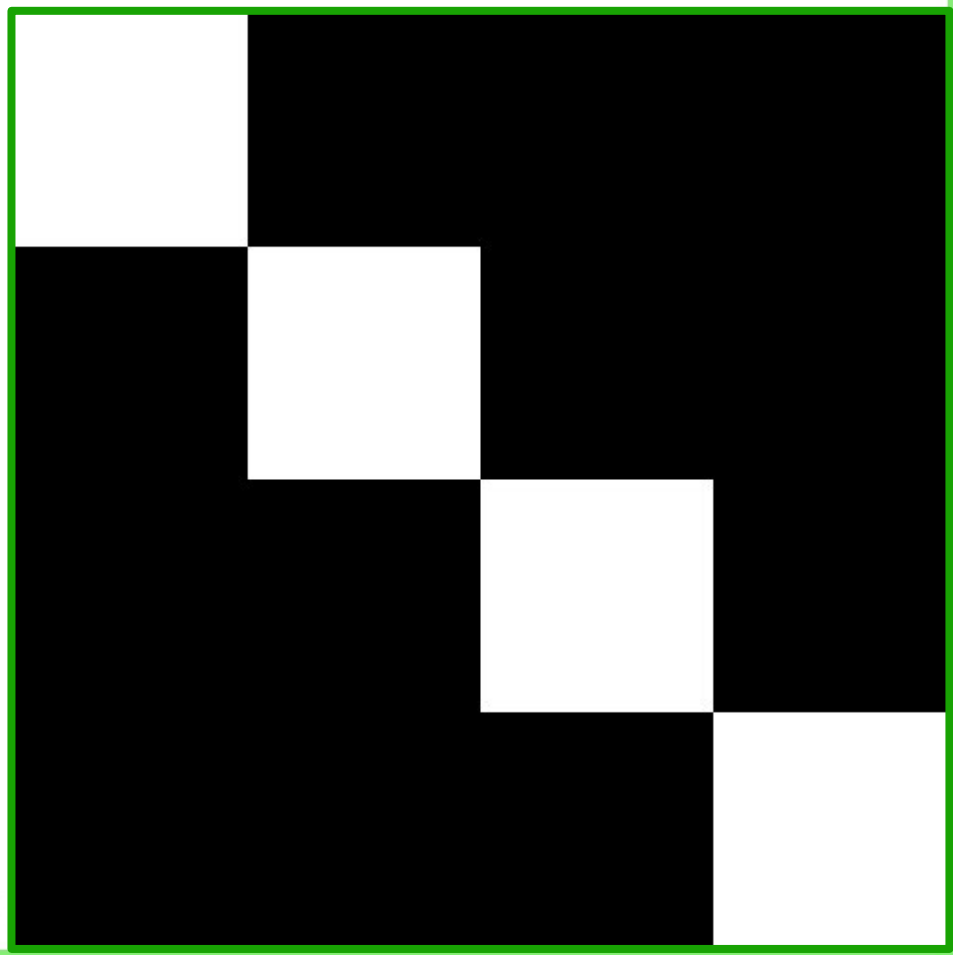
Decimal	Hexadecimal	Binary	Octal	Char
0	0	0	0	*[NULL]*
1	1	1	1	*[START OF HEADING]*
2	2	10	2	*[START OF TEXT]*
3	3	11	3	*[END OF TEXT]*
4	4	100	4	*[END OF TRANSMISSION]*
5	5	101	5	*[ENQUIRY]*
6	6	110	6	*[ACKNOWLEDGE]*
7	7	111	7	*[BELL]*
8	8	1000	10	*[BACKSPACE]*
9	9	1001	11	*[HORIZONTAL TAB]*
10	A	1010	12	*[LINE FEED]*
11	B	1011	13	*[VERTICAL TAB]*
12	C	1100	14	*[FORM FEED]*
13	D	1101	15	*[CARRIAGE RETURN]*
14	E	1110	16	*[SHIFT OUT]*
15	F	1111	17	*[SHIFT IN]*
16	10	10000	20	*[DATA LINK ESCAPE]*
17	11	10001	21	*[DEVICE CONTROL 1]*
18	12	10010	22	*[DEVICE CONTROL 2]*
19	13	10011	23	*[DEVICE CONTROL 3]*
20	14	10100	24	*[DEVICE CONTROL 4]*
21	15	10101	25	*[NEGATIVE ACKNOWLEDGE]*
22	16	10110	26	*[SYNCHRONOUS IDLE]*
23	17	10111	27	*[ENG OF TRANS. BLOCK]*
24	18	11000	30	*[CANCEL]*
25	19	11001	31	*[END OF MEDIUM]*
26	1A	11010	32	*[SUBSTITUTE]*
27	1B	11011	33	*[ESCAPE]*
28	1C	11100	34	*[FILE SEPARATOR]*
29	1D	11101	35	*[GROUP SEPARATOR]*
30	1E	11110	36	*[RECORD SEPARATOR]*
31	1F	11111	37	*[UNIT SEPARATOR]*
32	20	100000	40	*[SPACE]*
33	21	100001	41	**!**
34	22	100010	42	**"**
35	23	100011	43	**#**
36	24	100100	44	**$**
37	25	100101	45	**%**
38	26	100110	46	**&**
39	27	100111	47	**'**
40	28	101000	50	**(**
41	29	101001	51	**)**
42	2A	101010	52	*****
43	2B	101011	53	**+**
44	2C	101100	54	**,**
45	2D	101101	55	**-**
46	2E	101110	56	**.**
47	2F	101111	57	**/**
48	30	110000	60	**0**
49	31	110001	61	**1**
50	32	110010	62	**2**
51	33	110011	63	**3**
52	34	110100	64	**4**
53	35	110101	65	**5**
54	36	110110	66	**6**
55	37	110111	67	**7**
56	38	111000	70	**8**
57	39	111001	71	**9**
58	3A	111010	72	**:**
59	3B	111011	73	**;**
60	3C	111100	74	**<**
61	3D	111101	75	**=**
62	3E	111110	76	**>**
63	3F	111111	77	**?**
64	40	1000000	100	**@**
65	41	1000001	101	**A**
66	42	1000010	102	**B**
67	43	1000011	103	**C**
68	44	1000100	104	**D**
69	45	1000101	105	**E**
70	46	1000110	106	**F**
71	47	1000111	107	**G**
72	48	1001000	110	**H**
73	49	1001001	111	**I**
74	4A	1001010	112	**J**
75	4B	1001011	113	**K**
76	4C	1001100	114	**L**
77	4D	1001101	115	**M**
78	4E	1001110	116	**N**
79	4F	1001111	117	**O**
80	50	1010000	120	**P**
81	51	1010001	121	**Q**
82	52	1010010	122	**R**
83	53	1010011	123	**S**
84	54	1010100	124	**T**
85	55	1010101	125	**U**
86	56	1010110	126	**V**
87	57	1010111	127	**W**
88	58	1011000	130	**X**
89	59	1011001	131	**Y**
90	5A	1011010	132	**Z**
91	5B	1011011	133	**[**
92	5C	1011100	134	****
93	5D	1011101	135	**]**
94	5E	1011110	136	**^**
95	5F	1011111	137	**_**
96	60	1100000	140	**`**
97	61	1100001	141	**a**
98	62	1100010	142	**b**
99	63	1100011	143	**c**
100	64	1100100	144	**d**
101	65	1100101	145	**e**
102	66	1100110	146	**f**
103	67	1100111	147	**g**
104	68	1101000	150	**h**
105	69	1101001	151	**i**
106	6A	1101010	152	**j**
107	6B	1101011	153	**k**
108	6C	1101100	154	**l**
109	6D	1101101	155	**m**
110	6E	1101110	156	**n**
111	6F	1101111	157	**o**
112	70	1110000	160	**p**
113	71	1110001	161	**q**
114	72	1110010	162	**r**
115	73	1110011	163	**s**
116	74	1110100	164	**t**
117	75	1110101	165	**u**
118	76	1110110	166	**v**
119	77	1110111	167	**w**
120	78	1111000	170	**x**
121	79	1111001	171	**y**
122	7A	1111010	172	**z**
123	7B	1111011	173	**{**
124	7C	1111100	174	**|**
125	7D	1111101	175	**}**
126	7E	1111110	176	**~**
127	7F	1111111	177	*[DEL]*

Binární data (8bitů na bajt), obrázek

```
01000111 01001001 01000110 00111000 00110111 01100001 00000100 00000000
00000100 00000000 10000000 00000001 00000000 00000000 00000000 00000000
11111111 11111111 11111111 00101100 00000000 00000000 00000000 00000000
00000100 00000000 00000100 00000000 00000000 00000010 00000101 00001100
00001110 10000110 01111010 01010001 00000000 00111011
```

```
47 49 46 38 37 61 04 00
04 00 80 01 00 00 00 00
ff ff ff 2c 00 00 00 00
04 00 04 00 00 02 05 0c
0e 86 7a 51 00 3b
```

```
GIF87a..
........
...,....
........
..zQ.;
```

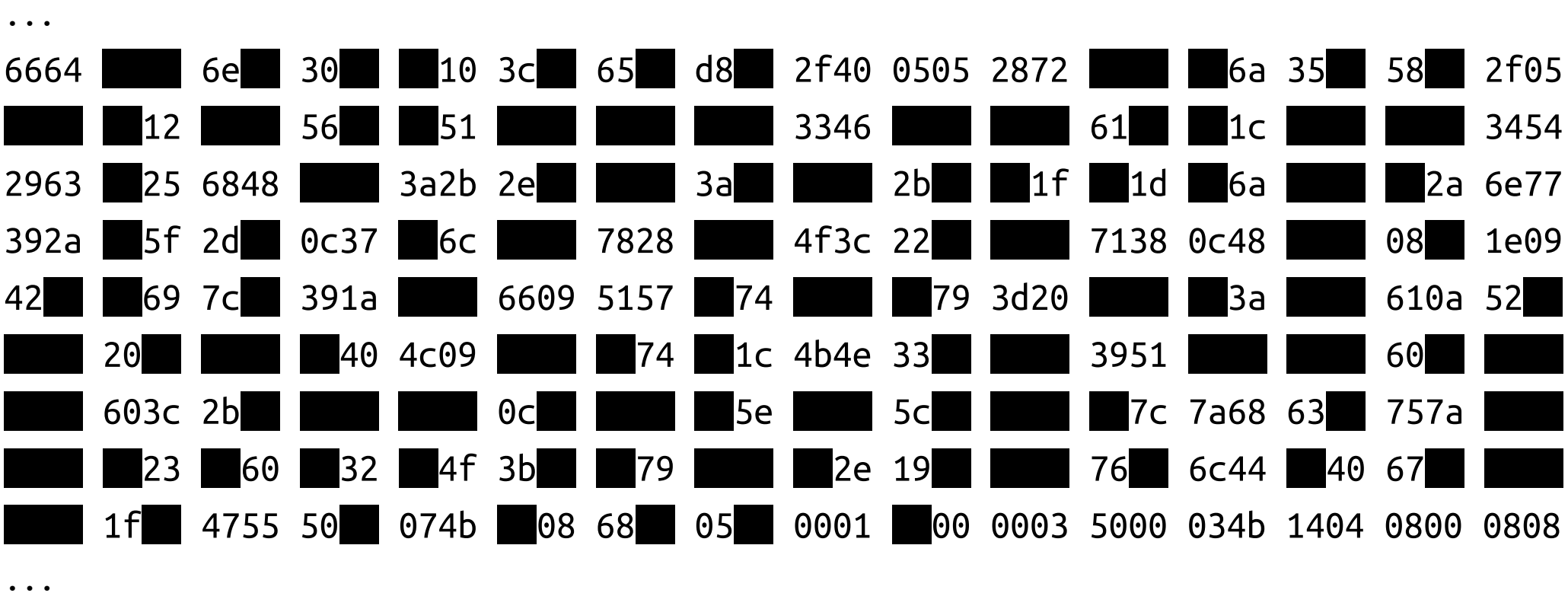


Prázdný dokument, formát ODF, binární data

```
...
6664 93cd 6ecd 3083 8410 3cef 6585 d8ce 2f40 0505 2872 b9ca 9f6a 35c0 5886 2f05
9af2 de12 8ebe 5693 a451 eaaa d49f aee3 3346 8edf 9bb4 61ed c81c b28b c1a8 3454
2963 ca25 6848 e9b5 3a2b 2ebb a5b9 3adb d8da 2bb6 9a1f f11d 836a 9fa5 ba2a 6e77
392a 965f 2d85 0c37 cf6c a2b3 7828 f39a 4f3c 22bc d5c1 7138 0c48 b4c6 088e 1e09
428d f569 7ce4 391a e2ce 6609 5157 eb74 90a0 f779 3d20 dd8b a93a b2a0 610a 52b6
87bd 20b6 8191 8d40 4c09 84ca 8374 ba1c 4b4e 33c5 f396 3951 a1c1 e2ed 60c7 d3bd
a283 603c 2bf0 cf8e ded8 0c88 e5a6 e05e b0bd 5cee df83 957c 7a68 63ee 757a c5d3
dff1 9423 9c60 ee32 ea4f 3bbb ff79 fee7 cf2e 19ff afe2 76b4 6c44 ce40 67a3 90fa
b4e7 1ff2 4755 50af 074b b408 68f7 05d2 0001 8300 0003 5000 034b 1404 0800 0808
...
```

Michal Pořízek

Prázdný dokument, formát ODF, binární data



SMTP

- řešení omezené množiny použitelných znaků?

  kódování

! nezaměňujte **kódování** (převod znaků/dat do jiné reprezentace, např. pomocí ASCII z písmen na binární číslo) a **šifrování** (převod nezabezpečených dat na zašifrovaná data, tedy data čitelná pouze se speciální znalostí – dešifrovací klíč)

Rozšíření ESMTP

Extended
Simple
Mail
Transfer
Protocol

ESMTP: Extended SMTP

- *zpětně kompatibilní* **rozšíření protokolu SMTP**
- rok 1995, RFC 1869
- klient používající ESMTP by měl použít uvozující příkaz **EHLO** (místo HELO)
- 7-bitové kódování pro přenos 8-bitových dat:
  - kódování `quoted printable`: RFC 2045, sekce 6.7
  - kódování `base64`: RFC 2045, sekce 6.8
- a řada dalších rozšíření:

Michal Račinský

Rozšíření (extensions)

- **PIPELINING**: možnost odesílat více SMTP příkazů za sebou bez nutnosti čekat na odezvu serveru
- **SMTP-AUTH**: autentizace uživatele (v dnešní době naprosto nezbytná)
- **STARTTLS**: šifrování na transportní vrstvě
- **SIZE**: `RFC 1870`
  - server: maximální velikost e-mailu, který je server ochoten přijmout
  - klient: velikost aktuálně doručovaného e-mailu

Rozšíření (extensions)

- **8BITMIME**: přenos 8-bitového materiálu v rámci SMTP
  - RFC 6152 (rok 2011), původ RFC 1426 (rok 1993)
  - maximálně 998 oketů na řádce s tím, že byty s hodnotou 13 a 10 (CR a LF) smí být pouze na konci řádku
- **BINARYMIME**: kompletní binární přenos obsahu
  - RFC 3030 (rok 2000), původ RFC 1830 (rok 1995, experimentální)
  - řada poštovních serverů však **dodnes** nepodporuje (Postfix, Sendmail, atd.)

Michal Račinský

Rozšíření (extensions)

- **CHUNKING**: příkaz **`BDAT`** (alternativa příkazu **`DATA`**) kterým je možné rozdělit odesílání dat na kousky (chunks)
  - **`BDAT velikost_kousku poslední_kousek?`**
  - vazba na **BINARYMIME**, ale lze použít i bez **BINARYMIME**
  - vhodné pro velké maily nebo přenos binárních dat přes **BINARYMIME**

Michal Bořek

Nové příkazy ESMTP

`EHLO` `fqdn`	**ohlášení klienta**, který umí **ESMTP** *fqdn*: plně kvalifikované doménové jméno (**F**ully **Q**ualified **D**omain **N**ame)
`AUTH` `metoda`	spustí **autentizaci** pomocí zadané **autentizační metody** např. `PLAIN`, `LOGIN`, `CRAM-MD5`, `DIGEST-MD5`, atd. `LOGIN` kóduje komunikaci do BASE64, leč nešifruje
`STARTTLS`	spustí **šifrování na transportní vrstvě** **TLS**: **T**ransport **L**ayer **S**ecurity (nástupce SSL)
`SIZE`	klient může specifikovat velikost e-mailu pro doručení: např. `MAIL FROM:<docm01@vse.cz> SIZE=500000`

(E)SMTP komunikace

```
220 krkavec.net ESMTP Postfix (Debian/GNU)
EHLO localhost
250-krkavec.net
250-PIPELINING
250-SIZE 40960000
250-ETRN
250-STARTTLS
250-AUTH PLAIN LOGIN
250-AUTH=PLAIN LOGIN
250-ENHANCEDSTATUSCODES
250-8BITMIME
250 DSN
MAIL FROM: docm01@vse.cz
250 2.1.0 Ok
```

Michal Dočekal

(E)SMTP komunikace

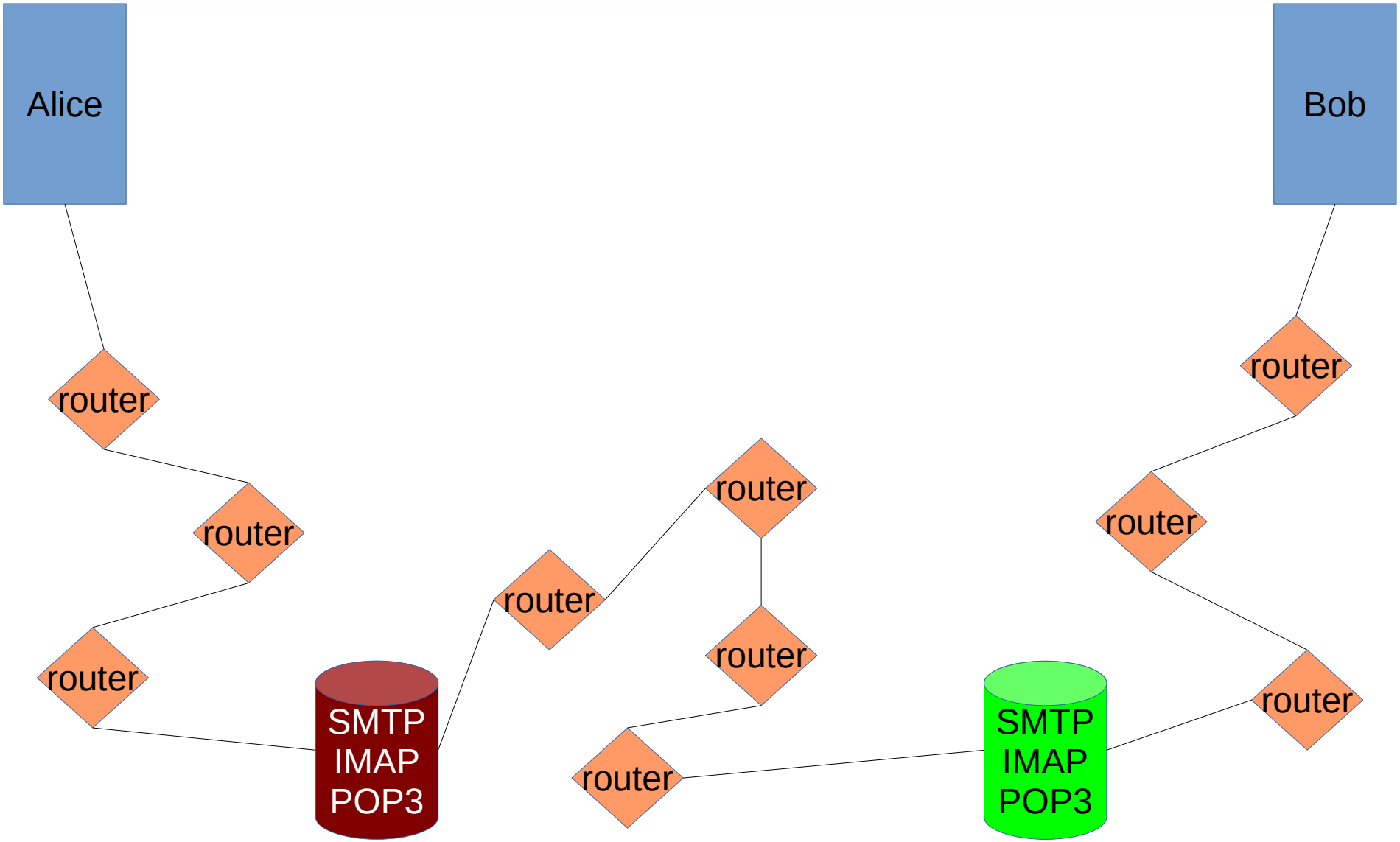
```
RCPT TO: root@krkavec.net
250 2.1.5 Ok
DATA
354 End data with <CR><LF>.<CR><LF>
From: docm01@vse.cz
To: root@krkavec.net
Date: Wed, 23 Mar 2022 15:09:48 +0100
Subject: Hi

Hi,

this is a test e-mail.

Regards,
Michal


.
250 2.0.0 Ok: queued as E77845481ACD
```

Vaše komentáře

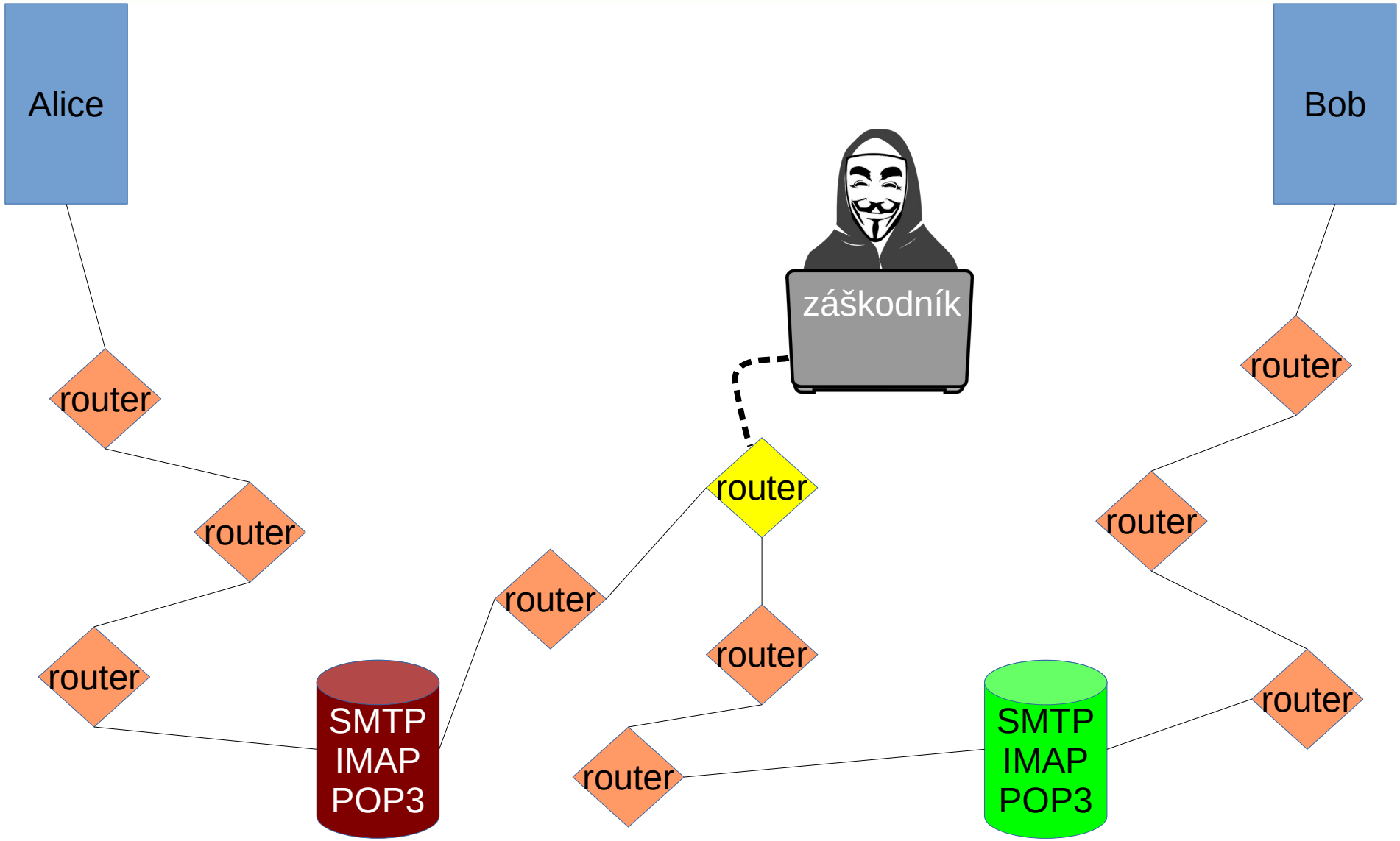
- Máte nějaké poznámky k ukázkám SMTP komunikace?

Michal Ročkal

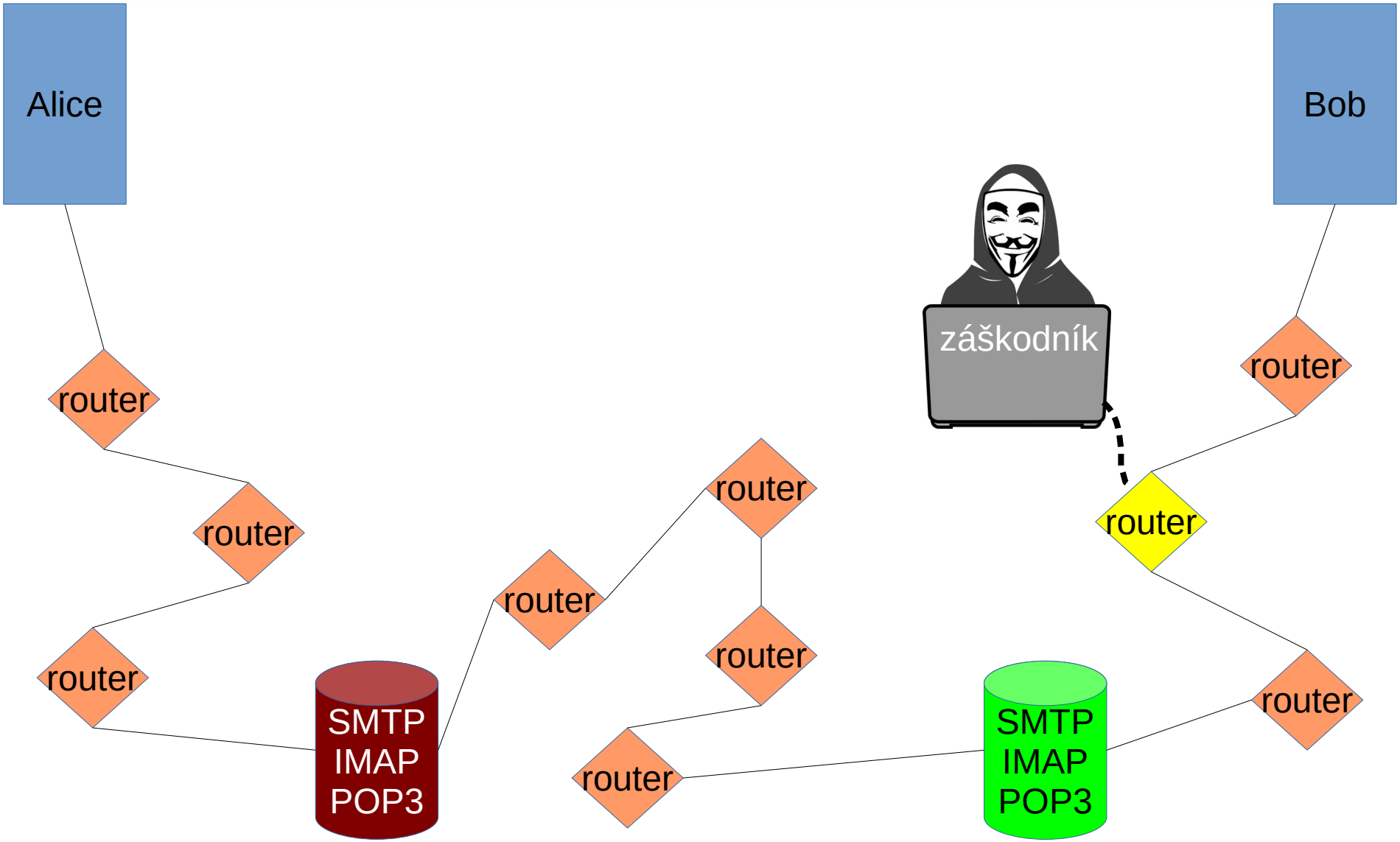
Praktický pohled



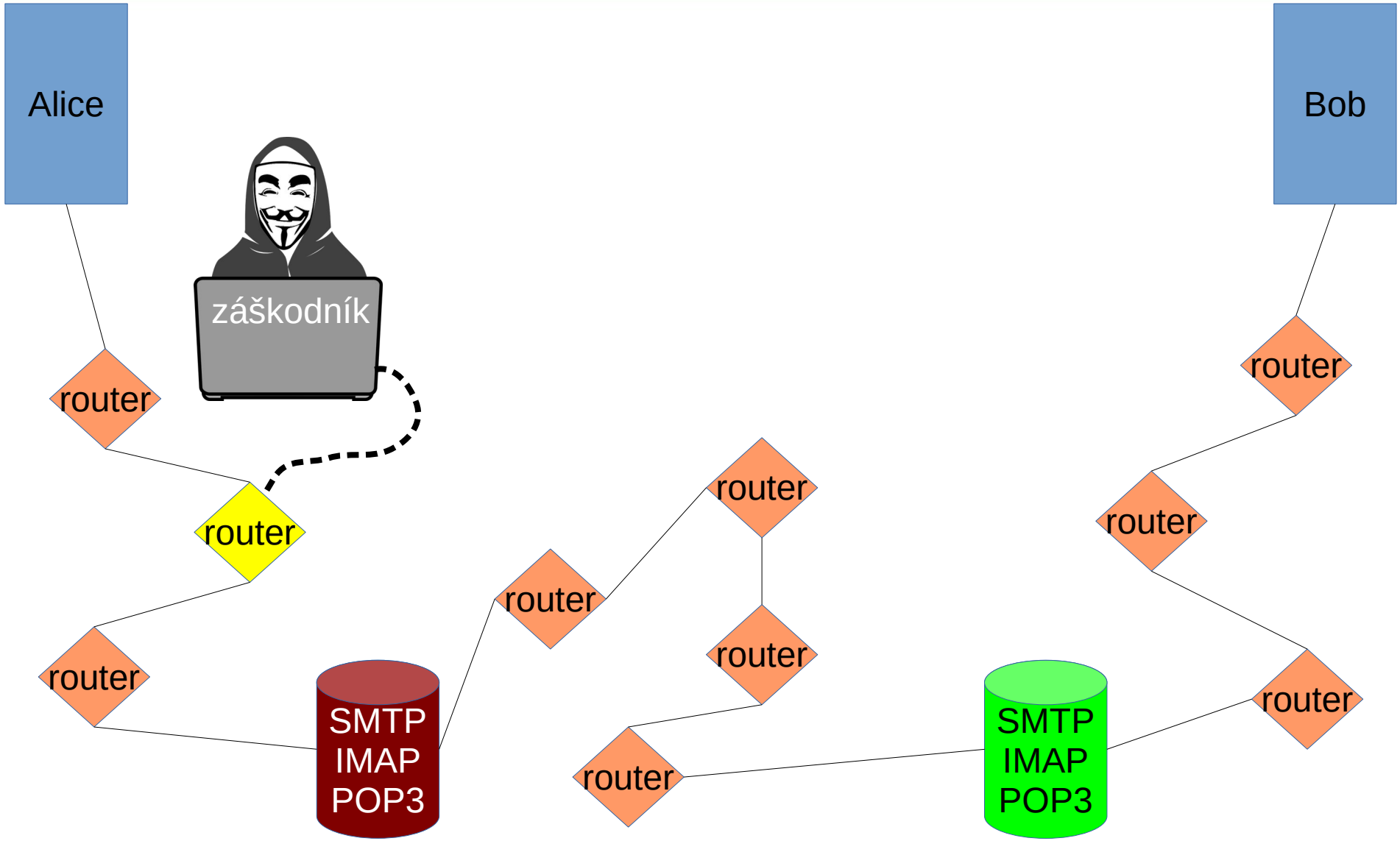
Praktický pohled



Praktický pohled



Praktický pohled



Návrh SMTP a bezpečnost

- SMTP nebyl navržen s ohledem na bezpečnost
- původní návrh SMTP neobsahoval šifrování
- předpoklad, že si poštovní servery budou vzájemně věřit
- přišel SPAM a útočníci
- řešilo a řeší se různými cestami, něco z toho je nevyřešeno dodnes
- aktuální rozumně nastavené poštovní servery jiným nevěří a jsou paranoidní

Šifrování přenosu e-mailu

- máme k dispozici zhruba tři cesty přes veřejnou síť, kudy musí jeden e-mail putovat (MUA_1 → SMTP, $SMTP_1$ → $SMTP_2$, MUA_2 → IMAP/POP3/web)
- jako odesílatel máte kontrolu nad šifrováním jedné z těchto cest – zbytek jde mimo vás
- šifrování je *zpravidla* vynucováno mezi klientem a poštovním serverem (ale je to jen moderní zvyklost, povinnost to není)
- šifrování mezi SMTP servery vynucováno není, použije se pouze pokud je nastavené a oba MTA se na něm dohodnou

Michal Račák

Šifrování a bezpečnost

- SMTP, IMAP4 i POP3 mají šifrované varianty, které se mohou, ale také nemusí použít
- v nešifrované podobě je e-mail na cestě plně čitelný (čistý text), takže jej lze zachytit, popřípadě i pozměnit (man in the middle útok)
- i když se komunikace šifruje, poskytovatel mailu má stále přístup k obsahu vaší pošty
- žádná bezpečnost/soukromí e-mailů prakticky neexistuje, pokud se o to člověk nepostará sám: **OpenPGP**, **S/MIME**

Michal Račinský

STRIPTLS

- **STRIPTLS** je útok, jehož cílem je obejít šifrování tak, že pomocí **MITM** pozmění komunikaci mezi SMTP serverem a klientem, aby se nešifrovalo
  - v SMTP je šifrování dobrovolné (Opportunistic TLS)
  - vyžadování šifrování lze nastavit, ale nedělá se to, neboť ne každý SMTP server šifrování podporuje
  - po zadání **EHLO** se z podporovaných funkcí serveru odstraní **STARTTLS**, takže klient si myslí, že server TLS neumí, a proto doručí mail bez šifrování
  - MITM útočník tak získá přístup k textu mailu (a může jej i pozměnit)

Michal Poslušný

STRIPTLS útok: ukázka

Nepozměněná komunikace

```
220 smtp.gmail.com ESMTP
ehlo nicnetusici.klient.th
250-smtp.gmail.com
250-SIZE 35882577
250-8BITMIME
250-STARTTLS
250-ENHANCEDSTATUSCODES
250-PIPELINING
250 SMTPUTF8
```

Pozměněná komunikace

```
220 smtp.gmail.com ESMTP
ehlo nicnetusici.klient.th
250-smtp.gmail.com
250-SIZE 35882577
250-8BITMIME
250-ENHANCEDSTATUSCODES
250-PIPELINING
250 SMTPUTF8
```

Další čtení: PDF

Pretty Good Privacy

- dnes otevřený standard OpenPGP, RFC 4880
  - Phil Zimmermann
- šifrování a podepisování obsahu e-mailů využívající asymetrickou kryptografii
- GNU Privacy Guard: FOSS implementace
- pluginy do poštovních klientů, problém: je třeba používat poštovního klienta
- alternativa k Public Key Infrastructure (PKI) v podobě Web of Trust - síť důvěry
  - Alice získala Bobův veřejný klíč, ale nemůže se Bobovi dovolat, aby si ověřila fingerprint klíče; Bobův klíč je však podepsán Karlem, kterého Alice zná a věří mu, tudíž věří i Bobovu klíči

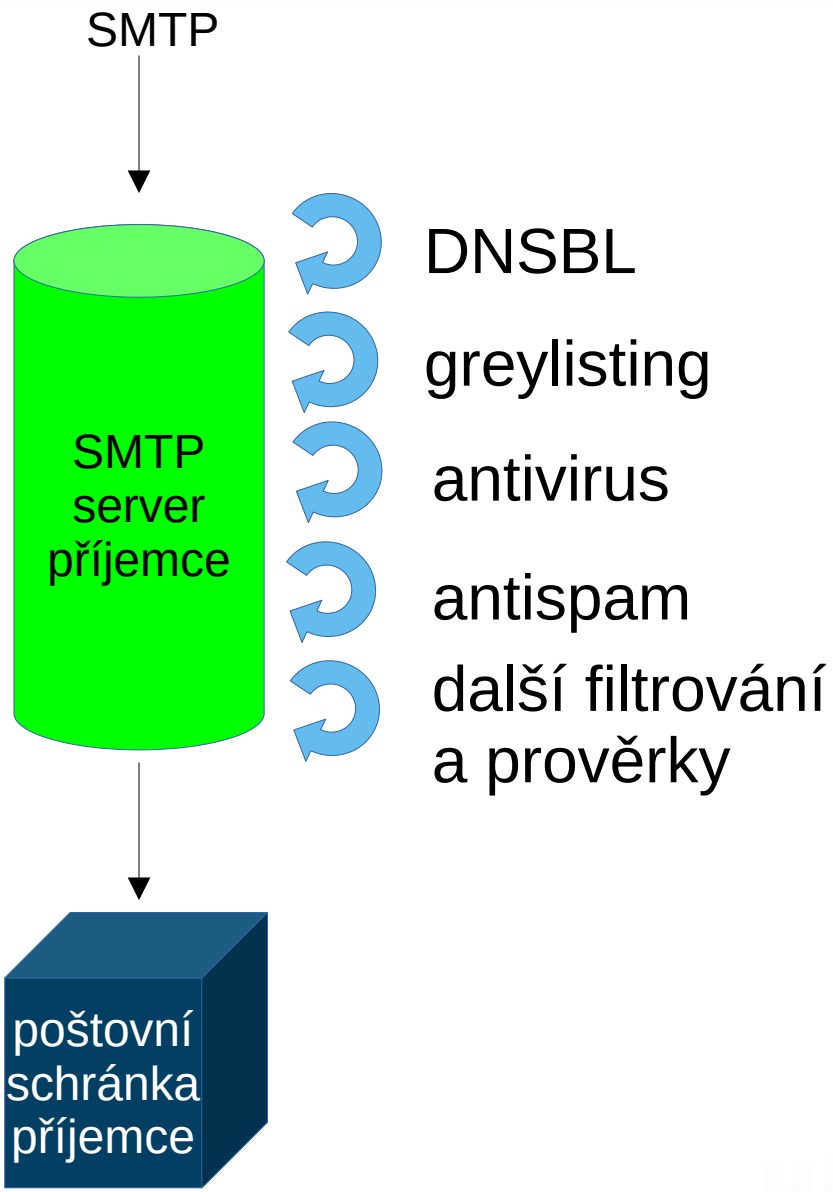
Michal Račák

Zpracování pošty

Spamassassin
Greylisting
DNSBL
Antivirus

Michal Pořízek

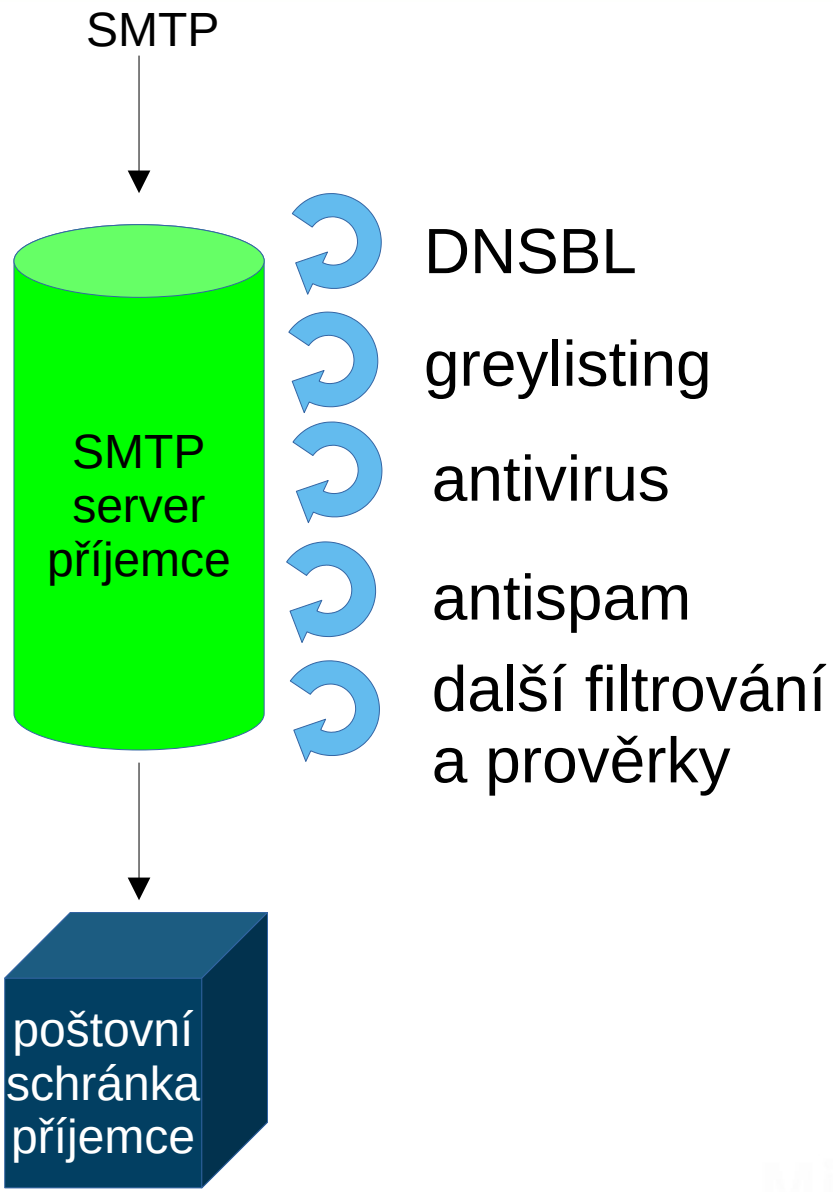
Zpracování pošty



- pořadí zpracování a filtrů záleží na správci / nastavení poštovního serveru
- některé filtrování je součástí poštovního serveru, jiné existuje v podobě samostatných komponent běžících buď na samotném SMTP serveru nebo i jinde
- pro zpracování externími službami je mail vyřazen z fronty pro doručení, předán externímu nástroji a poté vrácen zpět do fronty – jeho obsah se mění (minimálně o hlavičky)

Michal Račák

Zpracování pošty



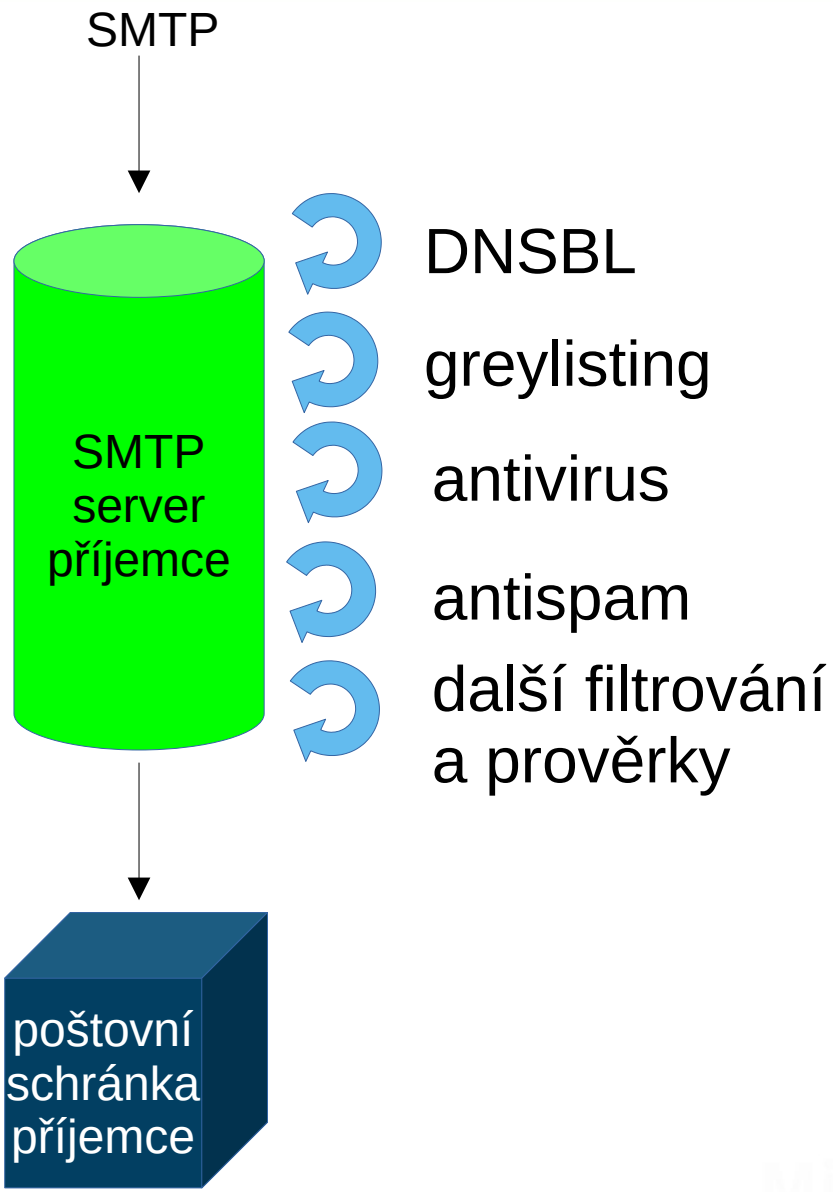
- ochrana proti spamu je obvykle komplexní a obsahuje více kroků / filtrů / opatření
  - **DNS blacklist** – externí služba, která sonduje a zaznamenává do databáze, které IP adresy nebo rozsahy odesílaly spam, a umožňuje poštovním serverům prověřovat, zda-li IP adresa odesílatele mailu není náhodou na blacklistu; služba využívá DNS
    - např. spammer odesílá z IP `1.2.3.4`
    - odešle se DNS dotaz na **A** záznam `4.3.2.1.sbl-xbl.spamhaus.org`
    - existuje-li záznam, klient je odmítnut

Michal Račák

```
220 krkavec.net ESMTP Postfix (Debian/GNU)
EHLO loge.gunlock.buzz
250-krkavec.net
250-PIPELINING
250-SIZE 40960000
250-ETRN
250-STARTTLS
250-ENHANCEDSTATUSCODES
250-8BITMIME
250-DSN
250 CHUNKING
MAIL FROM:<24492-44778-290-6442-karel=krkavec.net@mail.gunlock.buzz>
BODY=8BITMIME
RCPT TO:<karel@krkavec.net>
250 2.1.0 Ok
554 5.7.1 Service unavailable; Client host [134.73.146.41] blocked
using sbl-xbl.spamhaus.org; https://www.spamhaus.org/sbl/query/SBLCSS
QUIT
221 2.0.0 Bye
```

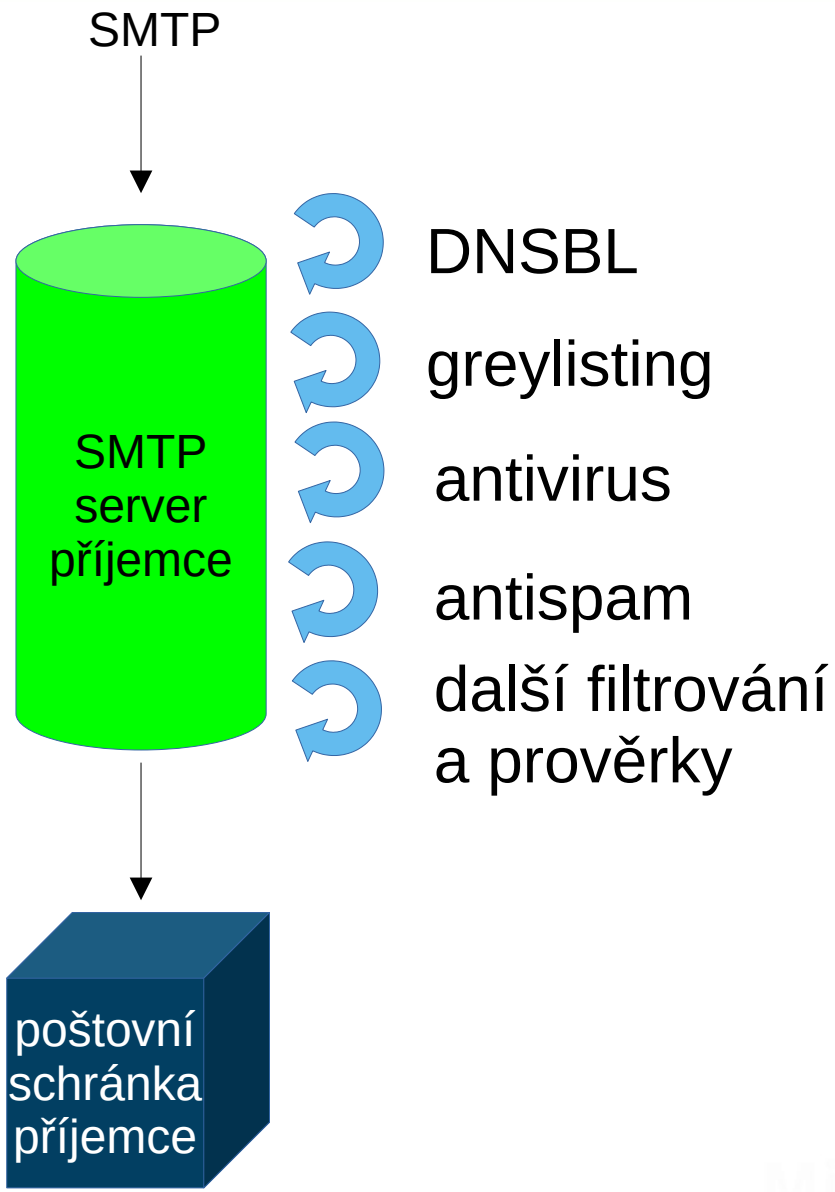
```
220 krkavec.net ESMTP Postfix (Debian/GNU)
EHLO loge.gunlock.buzz
250-krkavec.net
250-PIPELINING
250-SIZE 40960000
250-ETRN
250-STARTTLS
250-ENHANCEDSTATUSCODES
250-8BITMIME
250-DSN
250 CHUNKING
MAIL FROM:<24492-44778-290-6442-karel=krkavec.net@mail.gunlock.buzz>
BODY=8BITMIME
RCPT TO:<karel@krkavec.net>
250 2.1.0 Ok
554 5.7.1 Service unavailable; Client host [134.73.146.41] blocked
using sbl-xbl.spamhaus.org; https://www.spamhaus.org/sbl/query/SBLCSS
QUIT
221 2.0.0 Bye
```

Zpracování pošty



- ochrana proti spamu je obvykle komplexní a obsahuje více kroků / filtrů / opatření
  - **greylisting** – nástroj, který na základě analýzy 3 faktorů (IP odesílatele, mailu odesílatele a mailu příjemce) zjistí, jestli odesílatele v dané už konfiguraci viděl
    - pokud ano, mail je propuštěn dál
    - pokud ne, mail je dočasně odmítnut
    - klasické poštovní servery opakují doručení, spammeři často nikoliv
    - *může za zpoždění vaší pošty…*

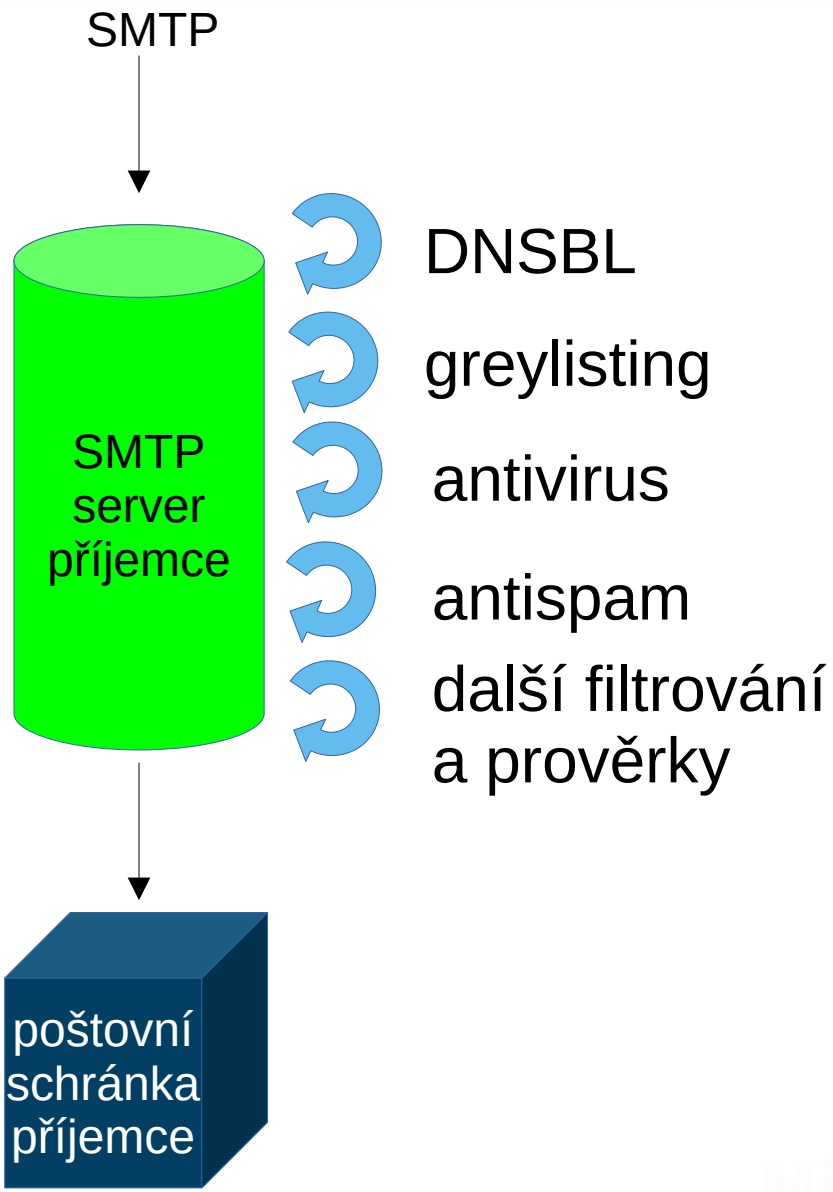
Zpracování pošty



- ochrana proti spamu je obvykle komplexní a obsahuje více kroků / filtrů / opatření
  - **antispam** – zpravidla pokročilý nástroj (např. spamassassin), který analyzuje e-mail pomocí mnoha různých testů (náročné na zpracování)
  - vyhoví-li mail konkrétnímu testu, jsou mu dle testu přičteny nebo odečteny body
  - má-li mail dostatek bodů, je označen za spam
  - často se používá v kombinaci s pravděpodobnostním filtrováním na základě Bayesova vzorce (**Bayesův filtr**)

Michal Špaček

Zpracování pošty



- samozřejmostí bývá antivirová kontrola (na linuxových poštovních serverech často zajišťuje clamav)

Michal Račák

Anatomie e-mailu

Hlavička

Tělo

Michal Ročárek

Anatomie e-mailu

- Formát je popsaný v `RFC 5322`
- **Hlavička**
  - je jen **jedna** a sestává se z řady **polí** (jméno: hodnota)
  - každý řádek s tisknutelným znakem na začátku uvozuje jméno pole, pokračuje dvojtečkou (oddělovač), za kterou následuje hodnota, např.:

    ```
    Subject: Vyhráli jste milion dolarů
    ```

  - hodnota pole může pokračovat na dalším řádku, je-li prvním znakem na řádku mezera nebo tabulátor
  - standardně podporuje pouze **`US-ASCII`** (v nedávné době **`RFC 6531`** a **`6532`** umožňují používat **`UTF-8`** znaky, ale servery to musí podporovat)
- **Tělo zprávy**
  - samotný obsah zprávy
  - od hlavičky jej odděluje **prázdný řádek**

Michal Račák

Pole v hlavičce

- Povinná pole (dle RFC 822)

  `From:`

  `Date:`

  `To:` nebo **`Bcc:`**
- Doporučená pole

  `Message-ID:`

  `In-Reply-to:`
- Další

  `Subject:`

  `Received:`

Michal Rešetka

Cesta e-mailu a hlavička

- putování e-mailu přes poštovní servery se zaznamenává do polí hlavičky
- stejně jako řada dalších zajímavých informací
- zkuste si zobrazit hlavičku k některému ze svých e-mailů
- (příklad následuje)

Michal Ročkai

E-mail (příklad)

Received: from VI1PR06MB5407.eurprd06.prod.outlook.com (2603:10a6:803:c1::17) 1
by DB7PR06MB5915.eurprd06.prod.outlook.com with HTTPS; Tue, 2 Feb 2021
06:42:19 +0000
Received: from MR2P264CA0102.FRAP264.PROD.OUTLOOK.COM (2603:10a6:500:33::18)
by VI1PR06MB5407.eurprd06.prod.outlook.com (2603:10a6:803:c1::17) with
Microsoft SMTP Server (version=TLS1_2,
cipher=TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384) id 15.20.3805.19; Tue, 2 Feb
2021 06:42:16 +0000
Received: from VE1EUR01FT056.eop-EUR01.prod.protection.outlook.com 2
(2603:10a6:500:33:cafe::7a) by MR2P264CA0102.outlook.office365.com
(2603:10a6:500:33::18) with Microsoft SMTP Server (version=TLS1_2,
cipher=TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384) id 15.20.3805.18 via Frontend
Transport; Tue, 2 Feb 2021 06:42:16 +0000
Authentication-Results: spf=fail (sender IP is 146.102.16.2)
smtp.mailfrom=suzano.sp.gov.br; o365.vse.cz; dkim=fail (no key for signature)
header.d=suzano.sp.gov.br;o365.vse.cz; dmarc=none action=none
header.from=suzano.sp.gov.br;
Received-SPF: Fail (protection.outlook.com: domain of suzano.sp.gov.br does
not designate 146.102.16.2 as permitted sender)
receiver=protection.outlook.com; client-ip=146.102.16.2; helo=vse.vse.cz;

E-mail (příklad)

```
Received: from vse.vse.cz (146.102.16.2) by                                3
VE1EUR01FT056.mail.protection.outlook.com (10.152.3.115) with Microsoft SMTP
Server (version=TLS1_2, cipher=TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384) id
15.20.3784.11 via Frontend Transport; Tue, 2 Feb 2021 06:42:15 +0000
Received: from localhost (localhost [127.0.0.1])                           4
by vse.vse.cz (Postfix) with ESMTP id 6F42B1A009F;
Tue, 2 Feb 2021 07:42:15 +0100 (CET)
X-Virus-Scanned: Debian amavisd-new at vse.cz
X-Spam-Flag: YES
X-Spam-Score: 10.877
X-Spam-Level: **********
X-Spam-Status: Yes, score=10.877 tagged_above=0 required=7
tests=[BAYES_50=0.8, DKIM_INVALID=0.1, DKIM_SIGNED=0.1,
FORGED_SPF_HELO=1, HTML_MESSAGE=0.001, LOCAL_MS_ANTISPAM=3,
LOTS_OF_MONEY=0.9, RCVD_IN_DNSBL_JUSTSPAM=1.5,
RCVD_IN_DNSWL_NONE=-0.0001, RCVD_IN_SBL=0.141, RCVD_IN_SBL_CSS=3.335,
SPF_FAIL=0.001, SPF_HELO_PASS=-0.001] autolearn=no autolearn_force=no
```

Michal Pavelka

E-mail (příklad)

Received: from vse.vse.cz ([127.0.0.1]) 5

by localhost (vse.vse.cz [127.0.0.1]) (amavisd-new, port 10024)

with ESMTP id FdNS47Z1Da9r; Tue, 2 Feb 2021 07:42:14 +0100 (CET)

X-Greylist: whitelisted by SQLgrey-1.8.0

Received: from EUR01-DB5-obe.outbound.protection.outlook.com (mail-db5eur01lp0204.outbound.protection.outlook.com [IPv6:2a01:111:f400:7e02::204]) 6

by vse.vse.cz (Postfix) with ESMTPS id 662771A009C

for <webmasterci@vse.cz>; Tue, 2 Feb 2021 07:42:14 +0100 (CET)

Received: from DU2PR04CA0142.eurprd04.prod.outlook.com (2603:10a6:10:231::27) 7

by AM0PR0602MB3505.eurprd06.prod.outlook.com (2603:10a6:208:19::19) with

Microsoft SMTP Server (version=TLS1_2,

cipher=TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384) id 15.20.3805.24; Tue, 2 Feb

2021 06:42:12 +0000

Michal Řeřicha

E-mail (příklad)

Received: from DB5EUR01FT063.eop-EUR01.prod.protection.outlook.com
(2603:10a6:10:231:cafe::3b) by DU2PR04CA0142.outlook.office365.com
(2603:10a6:10:231::27) with Microsoft SMTP Server (version=TLS1_2,
cipher=TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384) id 15.20.3805.17 via
Frontend
Transport; Tue, 2 Feb 2021 06:42:12 +0000
Authentication-Results-Original: spf=pass (sender IP is
177.222.18.34)
smtp.mailfrom=suzano.sp.gov.br; vse.cz; dkim=fail (no key for
signature)
header.d=suzano.sp.gov.br;vse.cz; dmarc=bestguesspass action=none
header.from=suzano.sp.gov.br;compauth=pass reason=109
Received-SPF: Pass (protection.outlook.com: domain of
suzano.sp.gov.br
designates 177.222.18.34 as permitted sender)
receiver=protection.outlook.com; client-ip=177.222.18.34;
helo=mail.suzano.sp.gov.br;

E-mail (příklad)

Received: from mail.suzano.sp.gov.br (177.222.18.34) by 9

DB5EUR01FT063.mail.protection.outlook.com (10.152.5.139) with Microsoft SMTP

Server (version=TLS1_2, cipher=TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384) id

15.20.3784.11 via Frontend Transport; Tue, 2 Feb 2021 06:42:11 +0000

Received: from localhost (localhost [127.0.0.1]) 10

by mail.suzano.sp.gov.br (Postfix) with ESMTP id EA3A515C6292

for <webmasterci@vse.cz>; Tue, 2 Feb 2021 03:20:00 -0300 (-03)

11

Received: from mail.suzano.sp.gov.br ([127.0.0.1])

by localhost (zimbra01.suzano.sp.gov.br [127.0.0.1]) (amavisd-new, port 10032)

with ESMTP id c7DEoFb_SKQT for <webmasterci@vse.cz>;

Tue, 2 Feb 2021 03:20:00 -0300 (-03)

E-mail (příklad)

```
Received: from localhost (localhost [127.0.0.1])                    12
by mail.suzano.sp.gov.br (Postfix) with ESMTP id 434131583574
for <webmasterci@vse.cz>; Tue, 2 Feb 2021 02:07:06 -0300 (-03)
DKIM-Filter: OpenDKIM Filter v2.10.3 mail.suzano.sp.gov.br 434131583574
DKIM-Signature: v=1; a=rsa-sha256; c=relaxed/relaxed; d=suzano.sp.gov.br;
s=5C35487C-155C-11EB-8F76-B2BAA081A366; t=1612242426;
bh=FmnDWenRENjoh+EtaEIHkdhQqmgEIsnteQeRydKV5xY=;
h=MIME-Version:To:From:Date:Message-Id;
b=rVGTiDBFfvW5jVraiO8/GLsY/v9B/TmPNmGSPDdrE4AbiKP0vkNX38NmyU+MAZZvI
Ea2Dzz3HGmIiYDLLozbvd+eQoCdfZTRnguaeRzgXbNO6R/GwyoGhtHg5KZ1mskzoz6
xvwNcTgb/aNaRWWHzNtU7fZG6ewWYytnhwkD2chiO9MXUBy+/tanY/EO9QRg0JWIDI
cavGXelojJ8I2blinS25t8C7Hyzj1ek/UV0eQJiMgzDE6H0IXKll/lQ2BAK362lZzt
D/529NW7krimWQCSvHU58oGZ+aQeHAEyepfdQ+aU/id7Ab6yGA6dgjQAdMLP5B2Mlh
8SfCuTxFctBoQ==
X-Virus-Scanned: amavisd-new at suzano.sp.gov.br
Received: from mail.suzano.sp.gov.br ([127.0.0.1])                   13
by localhost (zimbra01.suzano.sp.gov.br [127.0.0.1]) (amavisd-new, port 10026)
with ESMTP id EsIIrWoOduhJ for <webmasterci@vse.cz>;
Tue, 2 Feb 2021 02:07:06 -0300 (-03)
```

E-mail (příklad)


```
Received: from [192.168.43.147] (unknown [129.205.124.57])
by mail.suzano.sp.gov.br (Postfix) with ESMTPSA id 5E8231587E4C
for <webmasterci@vse.cz>; Tue, 2 Feb 2021 00:22:28 -0300 (-03)
Content-Type: multipart/alternative;
boundary="==============0398553340=="
MIME-Version: 1.0
Subject: *****SPAM!***** Good news
To: webmasterci@vse.cz
From: "Wray Bonnie" <emtherezinha.pereira@suzano.sp.gov.br>
Date: Mon, 01 Feb 2021 19:22:24 -0800
Reply-To: therrien@gracemanonfounda.com
Message-Id: <20210202032230.5E8231587E4C@mail.suzano.sp.gov.br>


Vážený majitel e-mailu webmasterci@vse.cz Laskave potvrdte
vlastnictví svého e-mailu webmasterci@vse.cz. Bylo náhodne
vybráno po losování elektronického pocítace Spinball, abychom
obdrželi dar ve výši 850 000,00 USD od Bonnie Wray. Další
podrobnosti odešlete potvrzovacím e-mailem na adresu:
therrien@gracemanonfounda.com
```

důležité hlavičky

tělo mailu

Pole `Received:`

- každý SMTP server, který e-mail přijme, přidá na začátek hlavičky e-mailu pole `Received:`
- v případě, že je mail předán externí službě např. pro antivirovou či antispamovou kontrolu, vrací se po zpracování zpět do fronty SMTP serveru, a to vyvolá přidání dalšího `Received:`
- jelikož se pole `Received:` přidávají na začátek hlaviček, chcete-li vypátrat původ e-mailu, musíte se podívat na ***poslední*** pole `Received:`

Michal Ďačok

Antispam

Spamassassin

E-mail (příklad)

Received: from vse.vse.cz (146.102.16.2) by 3
VE1EUR01FT056.mail.protection.outlook.com (10.152.3.115) with Microsoft SMTP
Server (version=TLS1_2, cipher=TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384) id
15.20.3784.11 via Frontend Transport; Tue, 2 Feb 2021 06:42:15 +0000
Received: from localhost (localhost [127.0.0.1]) 4
by vse.vse.cz (Postfix) with ESMTP id 6F42B1A009F;
Tue, 2 Feb 2021 07:42:15 +0100 (CET)
X-Virus-Scanned: Debian amavisd-new at vse.cz
X-Spam-Flag: YES
X-Spam-Score: 10.877
X-Spam-Level: **********
X-Spam-Status: Yes, score=10.877 tagged_above=0 required=7
tests=[BAYES_50=0.8, DKIM_INVALID=0.1, DKIM_SIGNED=0.1,
FORGED_SPF_HELO=1, HTML_MESSAGE=0.001, LOCAL_MS_ANTISPAM=3,
LOTS_OF_MONEY=0.9, RCVD_IN_DNSBL_JUSTSPAM=1.5,
RCVD_IN_DNSWL_NONE=-0.0001, RCVD_IN_SBL=0.141, RCVD_IN_SBL_CSS=3.335,
SPF_FAIL=0.001, SPF_HELO_PASS=-0.001] autolearn=no autolearn_force=no

E-mail (příklad)

```
Received: from vse.vse.cz (146.102.16.2) by                                    3
VE1EUR01FT056.mail.protection.outlook.com (10.152.3.115) with Microsoft SMTP
Server (version=TLS1_2, cipher=TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384) id
15.20.3784.11 via Frontend Transport; Tue, 2 Feb 2021 06:42:15 +0000
Received: from localhost (localhost [127.0.0.1])                               4
by vse.vse.cz (Postfix) with ESMTP id 6F42B1A009F;
Tue, 2 Feb 2021 07:42:15 +0100 (CET)
X-Virus-Scanned: Debian amavisd-new at vse.cz
X-Spam-Flag: YES
X-Spam-Score: 10.877
X-Spam-Level: **********
X-Spam-Status: Yes, score=10.877 tagged_above=0 required=7
tests=[BAYES_50=0.8, DKIM_INVALID=0.1, DKIM_SIGNED=0.1,
FORGED_SPF_HELO=1, HTML_MESSAGE=0.001, LOCAL_MS_ANTISPAM=3,
LOTS_OF_MONEY=0.9, RCVD_IN_DNSBL_JUSTSPAM=1.5,
RCVD_IN_DNSWL_NONE=-0.0001, RCVD_IN_SBL=0.141, RCVD_IN_SBL_CSS=3.335,
SPF_FAIL=0.001, SPF_HELO_PASS=-0.001] autolearn=no autolearn_force=no
```

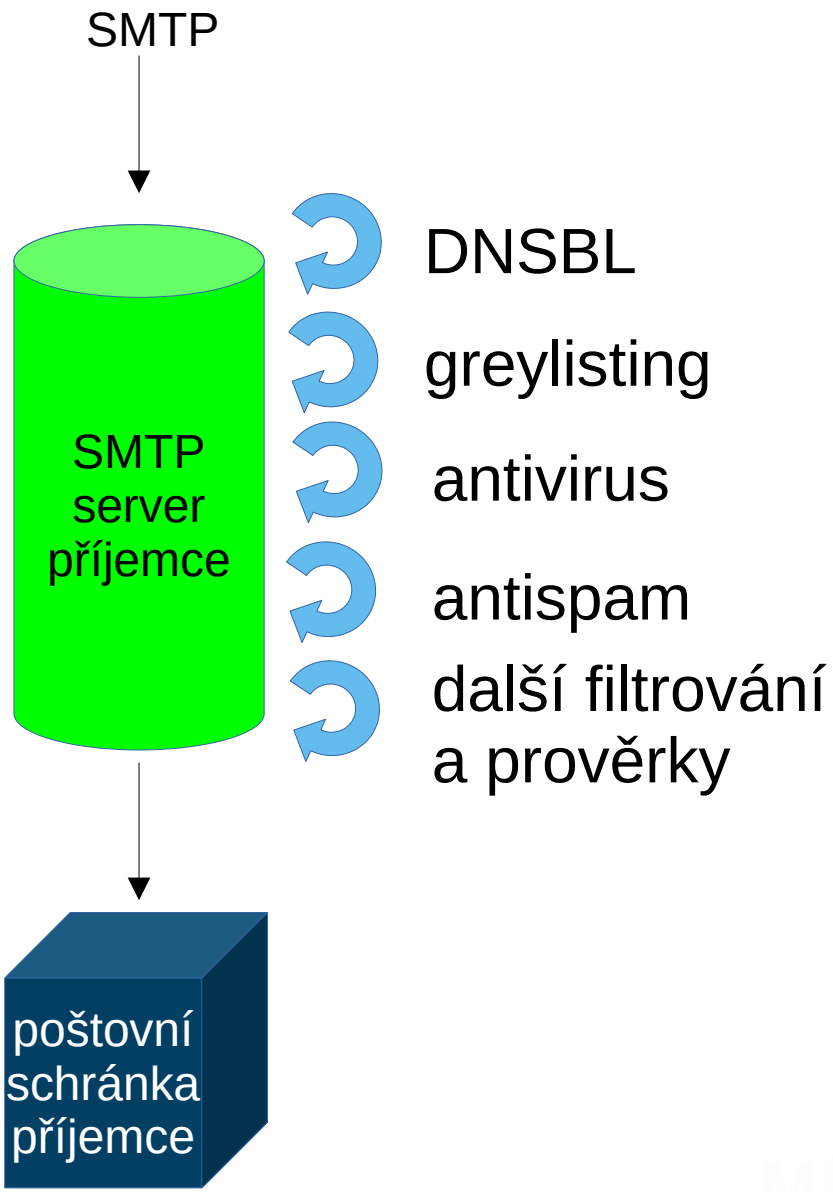
antivirová kontrola

E-mail (příklad)

```
Received: from vse.vse.cz (146.102.16.2) by                                  3
VE1EUR01FT056.mail.protection.outlook.com (10.152.3.115) with Microsoft SMTP
Server (version=TLS1_2, cipher=TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384) id
15.20.3784.11 via Frontend Transport; Tue, 2 Feb 2021 06:42:15 +0000
Received: from localhost (localhost [127.0.0.1])                             4
by vse.vse.cz (Postfix) with ESMTP id 6F42B1A009F;
Tue, 2 Feb 2021 07:42:15 +0100 (CET)
X-Virus-Scanned: Debian amavisd-new at vse.cz
X-Spam-Flag: YES
X-Spam-Score: 10.877
X-Spam-Level: **********
X-Spam-Status: Yes, score=10.877 tagged_above=0 required=7
tests=[BAYES_50=0.8, DKIM_INVALID=0.1, DKIM_SIGNED=0.1,
FORGED_SPF_HELO=1, HTML_MESSAGE=0.001, LOCAL_MS_ANTISPAM=3,
LOTS_OF_MONEY=0.9, RCVD_IN_DNSBL_JUSTSPAM=1.5,
RCVD_IN_DNSWL_NONE=-0.0001, RCVD_IN_SBL=0.141, RCVD_IN_SBL_CSS=3.335,
SPF_FAIL=0.001, SPF_HELO_PASS=-0.001] autolearn=no autolearn_force=no
```

antispam
(spamassassin)

Zpracování pošty



- ochrana proti spamu je obvykle komplexní a obsahuje více kroků / filtrů / opatření
  - **antispam** – zpravidla pokročilý nástroj (např. spamassassin), který analyzuje e-mail pomocí mnoha různých testů (náročné na zpracování)
  - vyhoví-li mail konkrétnímu testu, jsou mu dle testu přičteny nebo odečteny body
  - má-li mail dostatek bodů, je označen za spam
  - často se používá v kombinaci s pravděpodobnostním filtrováním na základě Bayesova vzorce (**Bayesův filtr**)

E-mail (příklad)

antispam (spamassassin)

```
X-Spam-Flag: YES
X-Spam-Score: 10.877
X-Spam-Level: **********
X-Spam-Status: Yes, score=10.877 tagged_above=0
required=7
tests=[BAYES_50=0.8, DKIM_INVALID=0.1, DKIM_SIGNED=0.1,
FORGED_SPF_HELO=1, HTML_MESSAGE=0.001,
LOCAL_MS_ANTISPAM=3, LOTS_OF_MONEY=0.9,
RCVD_IN_DNSBL_JUSTSPAM=1.5, RCVD_IN_DNSWL_NONE=-0.0001,
RCVD_IN_SBL=0.141, RCVD_IN_SBL_CSS=3.335, SPF_FAIL=0.001,
SPF_HELO_PASS=-0.001] autolearn=no autolearn_force=no
```

Další opatření proti SPAMu

- **Sender Policy Framework** (SPF)
  - pomocí SPF záznamů v DNS (typ **TXT**) lze specifikovat, které poštovní servery jsou oprávněny odesílat poštu z dané domény
- **DomainKeys Identified Mail** (DKIM)
  - SMTP server podepisuje odchozí poštu soukromým klíčem, veřejný klíč je uložen v DNS (typ **TXT**) a příjemce si může ověřit, zda-li byl e-mail skutečně odeslán příslušným SMTP serverem a ne někým jiným
- **Domain-based Message Authentication, Reporting and Conformance** (DMARC)
  - založeno na SPF a DKIM, vlastní DNS záznam (typ **TXT**), větší kontrola nad tím, co se má s mailem stát, pokud validace selže

Čtení pošty

POP3
IMAP4

Michal Počekal

POP3

- **P**ost **O**ffice **P**rotocol
- textový protokol (jako SMTP)
- TCP port `110` (šifrovaná varianta: `995`)
- aktuálně používaná je verze 3 (viz `RFC 1939`)
- princip: připojím se k serveru, postahuji nové maily, odpojím se a čtu poštu offline
- jen **jeden** klient v jednom čase
- méně komplexní než IMAP

Michal Bořátka

POP3 – příkazy

USER username	přihlašovací jméno k přihlášení
PASS password	heslo k přihlášení
STAT	počet zpráv a velikost schránky
LIST	vrátí seznam zpráv, ke každé přiřadí **číslo**
RETR číslo	stáhne zprávu danou **číslem**
DELE číslo	určí zprávu danou **číslem** k výmazu
NOOP	nevykoná žádnou akci, drží spojení živé
RSET	zruší všechny příznaky k výmazu v dané relaci
TOP číslo počet	vrátí daný *počet* řádek ze zprávy dané číslem
QUIT	ukončí spojení a zprávy určené k výmazu vymaže

Michal Pačes

POP3 komunikace

```
S: <čekání na spojení na TCP portu 110>
  C: <otevření spojení>
  S:    +OK POP3 server ready
<1896.697170952@dbc.mtview.ca.us>
  C:    APOP mrose c4c9334bac560ecc979e58001b3e22fb
  S:    +OK mrose's maildrop has 2 messages (320 octets)
  C:    STAT
  S:    +OK 2 320
  C:    LIST
  S:    +OK 2 messages (320 octets)
  S:    1 120
  S:    2 200
  S:    .
```

Michal Račák

POP3 komunikace

```
C:    RETR 1
S:    +OK 120 octets
S:    <POP3 server posílá zprávu 1>
S:    .
C:    DELE 1
S:    +OK message 1 deleted
C:    RETR 2
S:    +OK 200 octets
S:    <POP3 server posílá zprávu 2>
S:    .
C:    DELE 2
S:    +OK message 2 deleted
C:    QUIT
S:    +OK dewey POP3 server signing off (maildrop empty)
C:  <uzavírá spojení>
S:  <čeká na další spojení>
```

Michal Dočekal

IMAP4

- **Internet Mail Access Protocol**
- textový protokol (jako SMTP a POP3)
- TCP port 143 (šifrovaná varianta 993)
- aktuálně používaná je verze 4 (viz RFC 1939)
- **online** i **offline** práce: maily lze postahovat a číst off-line, ale také zůstat on-line a stahovat jen hlavičky a těla zpráv stahovat dle potřeby
- připojeno může být **více klientů** najednou
- podpora **více** poštovních **složek**
- **vyhledávání** zpráv na serveru (hledá server)

Michal Račák

Příklad IMAP4 komunikace

```
C: <otevření spojení>
S:    * OK IMAP4rev1 Service Ready
C:    a001 login mrc secret
S:    a001 OK LOGIN completed
C:    a002 select inbox
S:    * 18 EXISTS
S:    * FLAGS (\Answered \Flagged \Deleted \Seen \Draft)
S:    * 2 RECENT
S:    * OK [UNSEEN 17] Message 17 is the first unseen
message
S:    * OK [UIDVALIDITY 3857529045] UIDs valid
S:    a002 OK [READ-WRITE] SELECT completed
```

Příklad IMAP4 komunikace

```
C:   a003 fetch 12 full
S:   * 12 FETCH (FLAGS (\Seen) INTERNALDATE "17-Jul-1996 02:44:25 -0700"
      RFC822.SIZE 4286 ENVELOPE ("Wed, 17 Jul 1996 02:23:25 -0700 (PDT)"
      "IMAP4rev1 WG mtg summary and minutes"
      (("Terry Gray" NIL "gray" "cac.washington.edu"))
      (("Terry Gray" NIL "gray" "cac.washington.edu"))
      (("Terry Gray" NIL "gray" "cac.washington.edu"))
      ((NIL NIL "imap" "cac.washington.edu"))
      ((NIL NIL "minutes" "CNRI.Reston.VA.US")
      ("John Klensin" NIL "KLENSIN" "MIT.EDU")) NIL NIL
      "<B27397-0100000@cac.washington.edu>")
      BODY ("TEXT" "PLAIN" ("CHARSET" "US-ASCII") NIL NIL "7BIT" 3028
      92))
S:   a003 OK FETCH completed
```

Michal Ďačka!

Příklad IMAP4 komunikace

```
C:   a004 fetch 12 body[header]
S:   * 12 FETCH (BODY[HEADER] {342}
S:   Date: Wed, 17 Jul 1996 02:23:25 -0700 (PDT)
S:   From: Terry Gray <gray@cac.washington.edu>
S:   Subject: IMAP4rev1 WG mtg summary and minutes
S:   To: imap@cac.washington.edu
S:   cc: minutes@CNRI.Reston.VA.US, John Klensin <KLENSIN@MIT.EDU>
S:   Message-Id: <B27397-0100000@cac.washington.edu>
S:   MIME-Version: 1.0
S:   Content-Type: TEXT/PLAIN; CHARSET=US-ASCII
S:
S:   )
S:   a004 OK FETCH completed
C    a005 store 12 +flags \deleted
S:   * 12 FETCH (FLAGS (\Seen \Deleted))
S:   a005 OK +FLAGS completed
C:   a006 logout
S:   * BYE IMAP4rev1 server terminating connection
S:   a006 OK LOGOUT completed
```

Michal Dočekal

MIME

Multipurpose
Internet
Mail
Extensions

Michal Dobiáš

MIME

- `Multipurpose Internet Mail Extensions`
  - podpora jiných kódování než **`US ASCII`** → podpora národních abeced
  - přenos **binárních dat**
  - **přílohy**
  - i pole v hlavičce (jako např. pole `Subject`) mohou využívat jiná kódování
  - navržen pro SMTP, ale některé aspekty jsou použity i u HTTP

Michal Bořek

MIME hlavičky

- Hlavička udávající použití a verzi MIME

```
MIME-Version: 1.0
```

- Hlavička udávající typ daného obsahu

```
Content-Type: text/plain; charset=UTF-8
```

- Hlavička udávající typ kódování obsahu

```
Content-Transfer-Encoding: quoted-printable
```

Michal Bořecký

Mail může mít více MIME částí

- zpráva může mít několik různých **MIME částí** (part), každá pak má specifikovaný svůj vlastní **`Content-Type`** a **`Content-Transfer-Encoding`**
- **hlavní MIME hlavička** má `Content-Type: multipart/mixed` s parametrem `boundary`, který obsahuje identifikátor, který uvozuje a končí každou samostatnou **MIME část**
- viz příklad:

Michal Račák

Příklad užití mime

```
Subject: =?UTF-8?B?UMWZw63FoWVybsSbIMW+bHXFpW91xI1rw70ga8WvxYggw7pwxJts?=
 =?UTF-8?B?IMSPw6FiZWxza8OpIMOzZHk=?=
Message-ID: <20171222082206.45ed4be3@resistance.local>
X-Mailer: Claws Mail
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="MP_/P+H47B1B18LXNp970evBS+t"

--MP_/P+H47B1B18LXNp970evBS+t
Content-Type: text/plain; charset=UTF-8
Content-Transfer-Encoding: quoted-printable
Content-Disposition: inline

Dobr=C3=BD den,

p=C5=99=C3=AD=C5=A1ern=C4=9B =C5=BElu=C5=A5ou=C4=8Dk=C3=BD k=C5=AF=C5=88 =
=C3=BAp=C4=9Bl =C4=8F=C3=A1belsk=C3=A9 =C3=B3dy.

S pozdravem a p=C5=99=C3=A1n=C3=ADm hezk=C3=A9ho dne
```

Příklad užití mime

Michal Do=C4=8Dekal

--MP_/P+H47B1B18LXNp970evBS+t

Content-Type: image/png

Content-Transfer-Encoding: base64

Content-Disposition: attachment; filename=obrazek.png

iVBORw0KGgoAAAANSUhEUgAAABEAAAATCAIAAAD5x3GmAAAAA3NCSVQICAjb4U/gAAACGUlEQVQo
z72Tv2vqUBTHj1orOCgWXLJEohB/IAhNhmziIEJRpzppsZPQwcXi2r+gWym20EHpoi5ODgoaGgdB
MkUKikSFCoKpQ1VsRHvfEF/re0J5Xd5nuvfy/cA959yrQgjBD1HDz/lfzgHLsl6v93O/Xq/L5XKx
WOx2u5PJBMMwi8Xi8/mi0ejh4eE2VK/X0W8Gg0EgEKBpOpfL8Tw/nU57vV6lUolEIhRFPT09KbEv
ZzweHx8fX1xcLBYLtMft7S1BEKIo/uFcXl5SFDWfzxFCtVrt5OSEJEm/388wDMdxHx8fwWAwmUx+
ObIskySZWYRQs1mE8fxQqEgy/JgMHA4HBzHIYTu7+/dbjdC6ECpajgczmYzl8sFADc3N+Fw+PT0
FABwHNfpdErGarW+vr6uVqttr8fjsXIKAM/PzzRN77fYaDRqtVqtVrt19Ho9ALy/vwPAcrnUaDT7
jiRJGIapVCq18t4IggCAfr+vrNvt9m5alUAEARBubxaEITNZmMymY6OjlqtFgDE4/F8Pl8qlUaj
0cPDgyRJjUZDFMVsNptKpQAAnE7n1dUVQuj6+trpdL69vSGE7u7uGIZxuVzpdJplWYZhPB7P4+Pj
dqafU1ssFh6P5/z8fDaboW9R7f4fnufPzs5MJlMikXC73WazudPpYBhmt9t3y1P99eckScpkMtVq
9eXlxWAw2Gy2WCwWCoW+c/6FX9LjWhT47iegAAAAAElFTkSuQmCC

--MP_/P+H47B1B18LXNp970evBS+t--

Michal Dořekal

MIME hlavičky

- **`MIME-Version:`**
  - stále k dispozici pouze verze 1.0
- **`Content-Type:`**
  - zprávy obsahující jen prostý text: `text/plain`, výchozí hodnota
  - textové zprávy s přílohami: `multipart/mixed` s rozdělením na dvě části, jedna `text/plain`, druhá má typ dle typu přenášeného souboru a hlavičku `Content-Disposition: attachment`, často s parametrem `filename=jmeno_souboru.ext`
  - odpověď s uloženou původní zprávou: opět `multipart/mixed` s `text/plain` částí v podobě odpovědi a původní zprávou uloženou jako `message/rfc822` část
  - alternativní obsah – zpráva může obsahovat více variant zprávy, typicky `text/plain` varianta a `text/html` varianta
  - atd.

Michal Račák

MIME hlavičky

- Přehled možných **`Content-Type`** viz zde:
  - application
  - audio
  - font
  - example
  - image
  - message
  - model
  - multipart
  - text
  - video

Michal Bořek

MIME hlavičky

- **`Content-Transfer-Encoding:`**
  - specifické pro `SMTP`, v `HTTP` tato hlavička není
  - kódování obsahu konkrétní MIME části
  - **7bit**: implicitní režim, až 998 octetů v jednom řádku, povolené znaky 1..127, znaky CR a LF mohou být jen na konci řádku
  - **8bit**: klient i server musí podporovat rozšíření **`8BITMIME`** (RFC 6152), povoleno je až 998 octetů v jednom řádku, povolené znaky 1..255, ale CR a LF mohou být stále jen na konci řádku
  - **binary**: klient i server musí podporovat rozšíření **`BINARYMIME`** (RFC 3030), povolena je libovolná sekvence oktetů, řádky mohou být libovolně dlouhé – jediné rozšíření, které podporuje přenos binárních dat, poštovní servery jej ale mnohdy nepodporují, včetně aktuálního Postfixu

Michal Račák

MIME hlavičky

- **`Content-Transfer-Encoding:`**
  - možnosti kódování obsahu, který nevyhovuje omezení **7-bitů** na byte:
  - **`quoted-printable`**: pro národní znaky abeced v textu, escapování nevyhovujících znaků
  - **`base64`**: pro binární data, kódování celého obsahu

Michal Bořek

Kódování metodou `quoted printable`

- *Escapování* znaků, které nepatří do 7-bitového **`US-ASCII`** kódování
- kódují se **jen** nevyhovující znaky, ostatní zůstávají nezměněné (tj. znaky **`33-126`**, ale s výjimkou znaku **`61`** – rovnítko)
- znaky **`9`** (tabulátor) a **`32`** (mezera) netřeba kódovat, pokud však nejsou na konci řádku, pak se kódují
- znaky **`13`** (CR) a **`10`** (LF), pokud tvoří konec řádku, zůstávají jako CRLF sekvence konce řádku – v jiném významu se kódují jako **`=0D`** a **`=0A`**
- každý nevyhovující **jeden** znak se nahradí **právě** třemi znaky

Michal Bořek

ASCII TABLE

Decimal	Hexadecimal	Binary	Octal	Char
0	0	0	0	*[NULL]*
1	1	1	1	*[START OF HEADING]*
2	2	10	2	*[START OF TEXT]*
3	3	11	3	*[END OF TEXT]*
4	4	100	4	*[END OF TRANSMISSION]*
5	5	101	5	*[ENQUIRY]*
6	6	110	6	*[ACKNOWLEDGE]*
7	7	111	7	*[BELL]*
8	8	1000	10	*[BACKSPACE]*
9	9	1001	11	*[HORIZONTAL TAB]*
10	A	1010	12	*[LINE FEED]*
11	B	1011	13	*[VERTICAL TAB]*
12	C	1100	14	*[FORM FEED]*
13	D	1101	15	*[CARRIAGE RETURN]*
14	E	1110	16	*[SHIFT OUT]*
15	F	1111	17	*[SHIFT IN]*
16	10	10000	20	*[DATA LINK ESCAPE]*
17	11	10001	21	*[DEVICE CONTROL 1]*
18	12	10010	22	*[DEVICE CONTROL 2]*
19	13	10011	23	*[DEVICE CONTROL 3]*
20	14	10100	24	*[DEVICE CONTROL 4]*
21	15	10101	25	*[NEGATIVE ACKNOWLEDGE]*
22	16	10110	26	*[SYNCHRONOUS IDLE]*
23	17	10111	27	*[ENG OF TRANS. BLOCK]*
24	18	11000	30	*[CANCEL]*
25	19	11001	31	*[END OF MEDIUM]*
26	1A	11010	32	*[SUBSTITUTE]*
27	1B	11011	33	*[ESCAPE]*
28	1C	11100	34	*[FILE SEPARATOR]*
29	1D	11101	35	*[GROUP SEPARATOR]*
30	1E	11110	36	*[RECORD SEPARATOR]*
31	1F	11111	37	*[UNIT SEPARATOR]*
32	20	100000	40	*[SPACE]*
33	21	100001	41	**!**
34	22	100010	42	**"**
35	23	100011	43	**#**
36	24	100100	44	**$**
37	25	100101	45	**%**
38	26	100110	46	**&**
39	27	100111	47	**'**
40	28	101000	50	**(**
41	29	101001	51	**)**
42	2A	101010	52	*****
43	2B	101011	53	**+**
44	2C	101100	54	**,**
45	2D	101101	55	**-**
46	2E	101110	56	**.**
47	2F	101111	57	**/**
48	30	110000	60	**0**
49	31	110001	61	**1**
50	32	110010	62	**2**
51	33	110011	63	**3**
52	34	110100	64	**4**
53	35	110101	65	**5**
54	36	110110	66	**6**
55	37	110111	67	**7**
56	38	111000	70	**8**
57	39	111001	71	**9**
58	3A	111010	72	**:**
59	3B	111011	73	**;**
60	3C	111100	74	**<**
61	3D	111101	75	**=**
62	3E	111110	76	**>**
63	3F	111111	77	**?**
64	40	1000000	100	**@**
65	41	1000001	101	**A**
66	42	1000010	102	**B**
67	43	1000011	103	**C**
68	44	1000100	104	**D**
69	45	1000101	105	**E**
70	46	1000110	106	**F**
71	47	1000111	107	**G**
72	48	1001000	110	**H**
73	49	1001001	111	**I**
74	4A	1001010	112	**J**
75	4B	1001011	113	**K**
76	4C	1001100	114	**L**
77	4D	1001101	115	**M**
78	4E	1001110	116	**N**
79	4F	1001111	117	**O**
80	50	1010000	120	**P**
81	51	1010001	121	**Q**
82	52	1010010	122	**R**
83	53	1010011	123	**S**
84	54	1010100	124	**T**
85	55	1010101	125	**U**
86	56	1010110	126	**V**
87	57	1010111	127	**W**
88	58	1011000	130	**X**
89	59	1011001	131	**Y**
90	5A	1011010	132	**Z**
91	5B	1011011	133	**[**
92	5C	1011100	134	****
93	5D	1011101	135	**]**
94	5E	1011110	136	**^**
95	5F	1011111	137	**_**
96	60	1100000	140	**`**
97	61	1100001	141	**a**
98	62	1100010	142	**b**
99	63	1100011	143	**c**
100	64	1100100	144	**d**
101	65	1100101	145	**e**
102	66	1100110	146	**f**
103	67	1100111	147	**g**
104	68	1101000	150	**h**
105	69	1101001	151	**i**
106	6A	1101010	152	**j**
107	6B	1101011	153	**k**
108	6C	1101100	154	**l**
109	6D	1101101	155	**m**
110	6E	1101110	156	**n**
111	6F	1101111	157	**o**
112	70	1110000	160	**p**
113	71	1110001	161	**q**
114	72	1110010	162	**r**
115	73	1110011	163	**s**
116	74	1110100	164	**t**
117	75	1110101	165	**u**
118	76	1110110	166	**v**
119	77	1110111	167	**w**
120	78	1111000	170	**x**
121	79	1111001	171	**y**
122	7A	1111010	172	**z**
123	7B	1111011	173	**{**
124	7C	1111100	174	**|**
125	7D	1111101	175	**}**
126	7E	1111110	176	**~**
127	7F	1111111	177	*[DEL]*

Kódování metodou `quoted printable`

- princip kódování: nevyhovující znak se nahradí **rovnítkem + hexadecimální hodnotou ASCII znaku**
  - např. **=C3**
- Řádky nesmí být delší než **76** znaků – delší řádky se rozdělí pomocí speciálního znaku **rovnítko** (soft line break)

Michal Bořák

Kódování metodou quoted printable

```
. . .
Subject: =?iso-8859-2?Q?Dom=E1c=ED_=FAkol_z_IZI216/2?=
Date: Sun, 20 May 2001 22:09:55 +0200
MIME-Version: 1.0
boundary="----=_NextPart_000_0021_01C0E179.9A067220"
. . .
Content-Type: text/plain; charset="iso-8859-2"
Content-Transfer-Encoding: quoted-printable
Dobr=FD den.
Pos=EDl=E1m V=E1m odkaz na HTML str=E1nku (dom=E1c=ED =FAkol), kter=E1 =
umo=BE=F2uje p=F8=EDstup na jinou str=E1nku p=F8es protokoly FTP a HTTP.
=
. . .
nen=ED implementov=E1no p=F8=EDmo do str=E1nek, jak tomu b=FDv=E1.
D=ECkuji za pochopen=ED.
. . .
```

Michal Bořátal

Kódování metodou quoted printable

```
. . .
Subject: =?iso-8859-2?Q?Dom=E1c=ED_=FAkol_z_IZI216/2?=
Date: Sun, 20 May 2001 22:09:55 +0200
MIME-Version: 1.0
boundary="----=_NextPart_000_0021_01C0E179.9A067220"
. . .
Content-Type: text/plain; charset="iso-8859-2"
Content-Transfer-Encoding: quoted-printable
Dobr=FD den.
Pos=EDl=E1m V=E1m odkaz na HTML str=E1nku (dom=E1c=ED =FAkol), kter=E1 =
umo=BE=F2uje p=F8=EDstup na jinou str=E1nku p=F8es protokoly FTP a HTTP. =
. . .
nen=ED implementov=E1no p=F8=EDmo do str=E1nek, jak tomu b=FDv=E1.
D=ECkuji za pochopen=ED.
. . .
```

Michal Bořátal

Kódování metodou base64

- nenahrazuje jedno zdrojové písmeno jiným cílovým písmenem.
- zdrojový text se rozdělí na **šestibitové skupiny** a ty se nahrazují písmeny z cílové abecedy, která obsahuje právě 64 znaků
- Znaky cílové abecedy leží v první polovině ASCII tabulky
- Tři písmena na vstupu (čtyři šestice bitů) jsou zakódována čtyřmi písmeny na výstupu

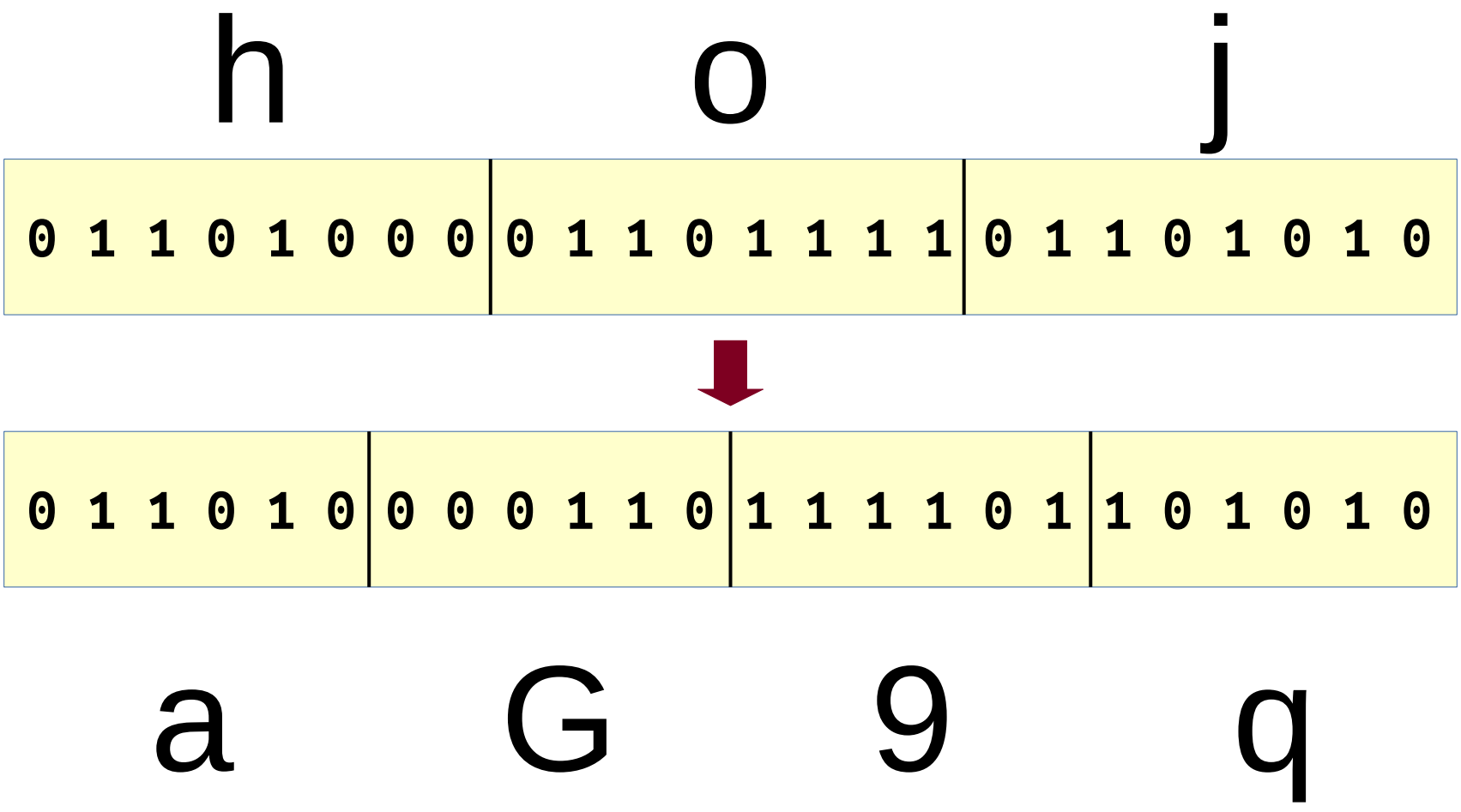
Michal Bačák

Kódovací tabulka base64

Dek.	Bin.	Znak	Dek.	Bin.	Znak	Dek.	Bin.	Znak	Dek.	Bin.	Znak
0	000000	A	16	010000	Q	32	100000	g	48	110000	w
1	000001	B	17	010001	R	33	100001	h	49	110001	x
2	000010	C	18	010010	S	34	100010	i	50	110010	y
3	000011	D	19	010011	T	35	100011	j	51	110011	z
4	000100	E	20	010100	U	36	100100	k	52	110100	0
5	000101	F	21	010101	V	37	100101	l	53	110101	1
6	000110	G	22	010110	W	38	100110	m	54	110110	2
7	000111	H	23	010111	X	39	100111	n	55	110111	3
8	001000	I	24	011000	Y	40	101000	o	56	111000	4
9	001001	J	25	011001	Z	41	101001	p	57	111001	5
10	001010	K	26	011010	a	42	101010	q	58	111010	6
11	001011	L	27	011011	b	43	101011	r	59	111011	7
12	001100	M	28	011100	c	44	101100	s	60	111100	8
13	001101	N	29	011101	d	45	101101	t	61	111101	9
14	001110	O	30	011110	e	46	101110	u	62	111110	+
15	001111	P	31	011111	f	47	101111	v	63	111111	/

Vyplňování hluchých míst (padding): =

Princip zakódování do base64

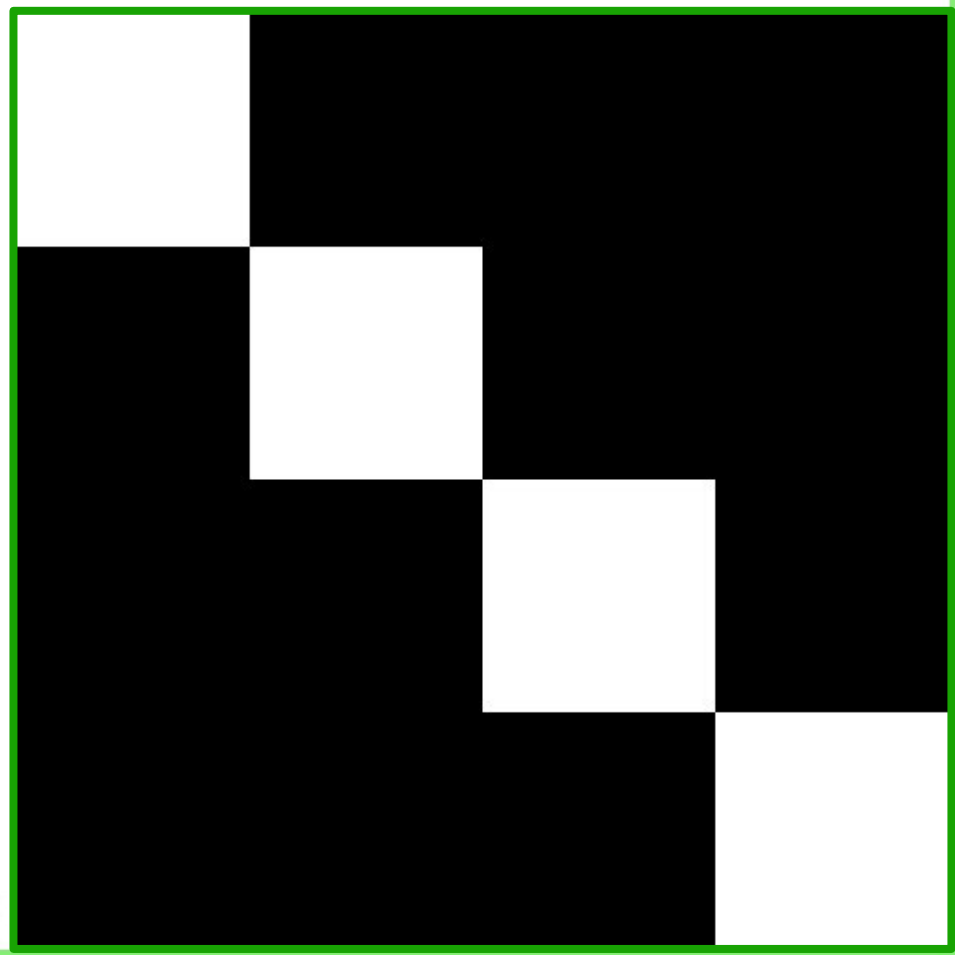


- výstupní znaky (**aG9q**) se přenáší jako normální byty (**8** bitů), tzn. z **8x3** bitů se stane **8x4** bitů

Michal Pecalka

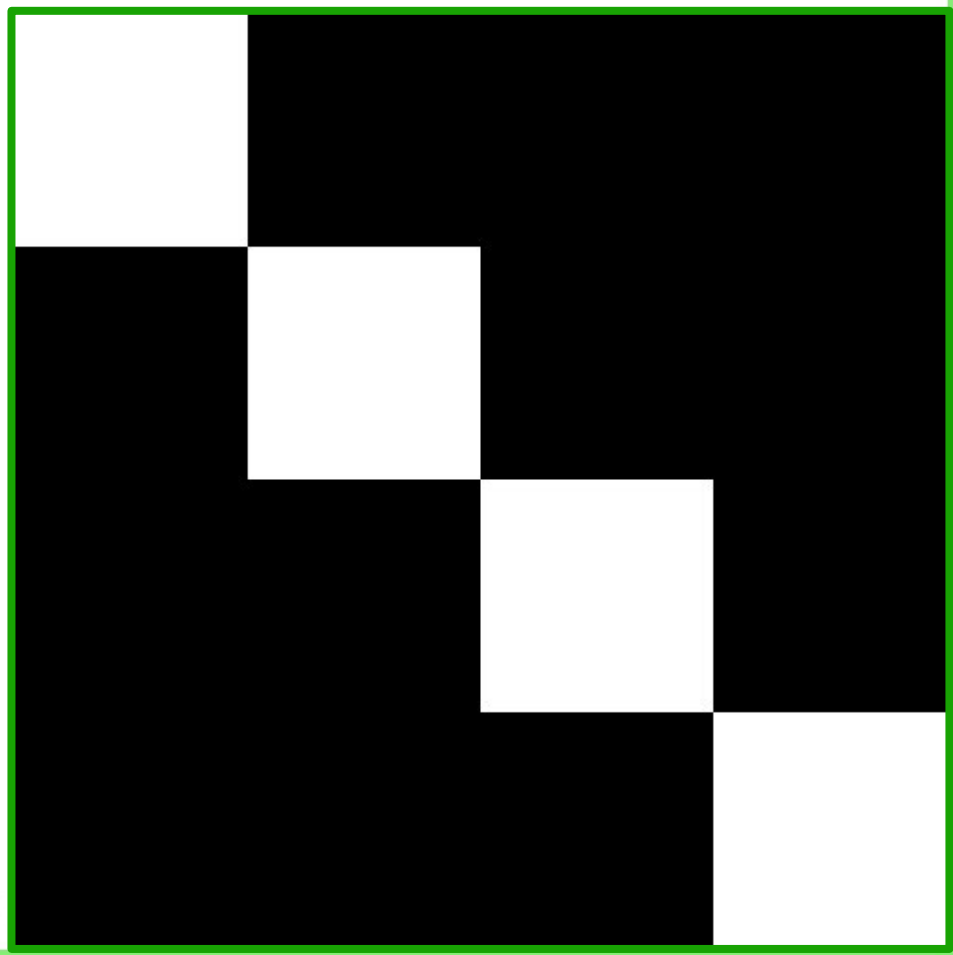
Binární data (8bitů na bajt), obrázek

```
01000111 01001001 01000110 00111000 00110111 01100001 00000100 00000000
00000100 00000000 10000000 00000001 00000000 00000000 00000000 00000000
11111111 11111111 11111111 00101100 00000000 00000000 00000000 00000000
00000100 00000000 00000100 00000000 00000000 00000010 00000101 00001100
00001110 10000110 01111010 01010001 00000000 00111011
```



Binární data (8bitů na bajt), obrázek

```
01000111 01001001 01000110 00111000 00110111 01100001 00000100 00000000
00000100 00000000 10000000 00000001 00000000 00000000 00000000 00000000
11111111 11111111 11111111 00101100 00000000 00000000 00000000 00000000
00000100 00000000 00000100 00000000 00000000 00000010 00000101 00001100
00001110 10000110 01111010 01010001 00000000 00111011
```

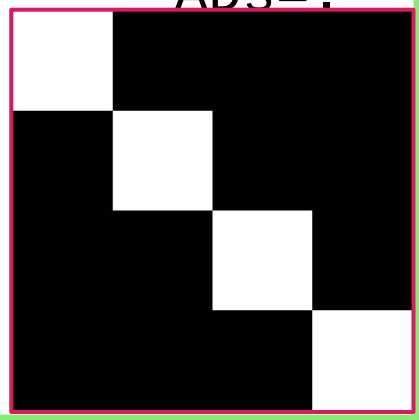


Binární data (8bitů na bajt), obrázek

Zakódováno do base64:

R0lGODdhBAAEAIABAAAAAP//ywAAAAABAAEAAACBQwOhnpRADs=

```
01010010 00110000 01101100 01000111 01001111 01000100  R0lGOD
01100100 01101000 01000010 01000001 01000001 01000101  dhBAAE
01000001 01001001 01000001 01000010 01000001 01000001  AIABAA
01000001 01000001 01000001 01010000 00101111 00101111  AAAP//
00101111 01111001 01110111 01000001 01000001 01000001  /ywAAA
01000001 01000001 01000010 01000001 01000001 01000101  AABAAE
01000001 01000001 01000001 01000011 01000010 01010001  AAACBQ
01110111 01001111 01101000 01101110 01110000 01010010  wOhnpR
01000001 01000100 01110011 00111101 00001010           ADs=.
```



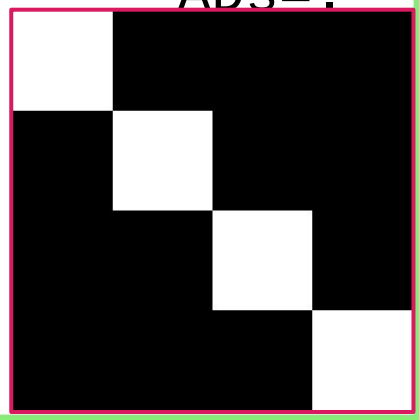
Michal Bořátal

Binární data (8bitů na bajt), obrázek

Zakódováno do base64:

R0lGODdhBAAEAIABAAAAAP//ywAAAAABAAEAAACBQwOhnpRADs=

01010010	00110000	01101100	01000111	01001111	01000100	R0lGOD
01100100	01101000	01000010	01000001	01000001	01000101	dhBAAE
01000001	01001001	01000001	01000010	01000001	01000001	AIABAA
01000001	01000001	01000001	01010000	00101111	00101111	AAAP//
00101111	01111001	01110111	01000001	01000001	01000001	/ywAAA
01000001	01000001	01000010	01000001	01000001	01000101	AABAAE
01000001	01000001	01000001	01000011	01000010	01010001	AAACBQ
01110111	01001111	01101000	01101110	01110000	01010010	wOhnpR
01000001	01000100	01110011	00111101	00001010		ADs=.



Kódování metodou base64

- věta „TCP je stavovy protokol.“ má právě 24 znaků
- Kolik bude mít po zakódování do base64?

Michal Řeřicha

Kódování metodou base64

- věta „TCP je stavovy protokol.“ má právě 24 znaků
- Kolik bude mít po zakódování do base64?

```
TCP je stavovy protokol.
VENQIGplIHN0YXZvdnkgcHJvdG9rb2wu
```

- každé 3 byty na vstupu se zakódují jako 4 byty na výstupu:

  24 : 3 = **8**

  24 + **8** = 32

Michal Bořecký

Quoted printable vs base64

- věta „TCP je stavový protokol.“ má **24** znaků
  - při použití kódování **ISO-8859-2** zůstane **24** znaků
  - při použití kódování **UTF-8** bude mít **25** znaků
- použijeme kódování **ISO-8859-2**
- kolik bytů bude věta zabírat, pokud použijeme:
  - kódování base64?
  - kódování quoted printable?

Quoted printable vs base64

- věta „`TCP je stavový protokol.`“ má **24** znaků
  - při použití kódování **`ISO-8859-2`** zůstane **24** znaků
  - při použití kódování **`UTF-8`** bude mít **25** znaků
- použijeme kódování **`ISO-8859-2`**
- kolik bytů bude věta zabírat, pokud použijeme:
  - kódování `base64`? **32** znaků, viz dříve
  - kódování `quoted printable`? 24 + 2 = **26**
    - každý znak nepatřící do ASCII (zde „**ý**“) je nahrazen rovnítkem a hexadecimální hodnotou (3 znaky), tj. za každý takový znak připočtu 2 extra byty navíc

Dekódování base64

- v příkazové řádce `4iz110.vse.cz` je k dispozici kodér/dekodér base64 nazvaný: `base64`
- zakódování souboru do base64:

```
base64 soubor.txt
```

- dekódování souboru z base64:

```
base64 -d soubor.txt
```

- dekódování bez ukládání do souboru:

```
echo "text k dekódování" | base64 -d
```

- dekódovat lze i pomocí online dekodérů

Michal Račák

Kódování textu

Michal Bližňák

Kódování textu

- všechno v počítači je číslo, tedy i znaky jsou reprezentovány čísly
- jednobytová kódovací schémata
  - **`ASCII`**
  - **`ISO 8859`** – série 15 znakových sad (kromě opuštěné 12ky)
  - **`ISO 8859-2`**: čeština a další vybrané středo/východoevropské jazyky
  - Proprietární kódy
- vícebytová kódovací schémata
  - **Unicode**
    - **`UTF-8`**

Michal Bořek

Znakové sady

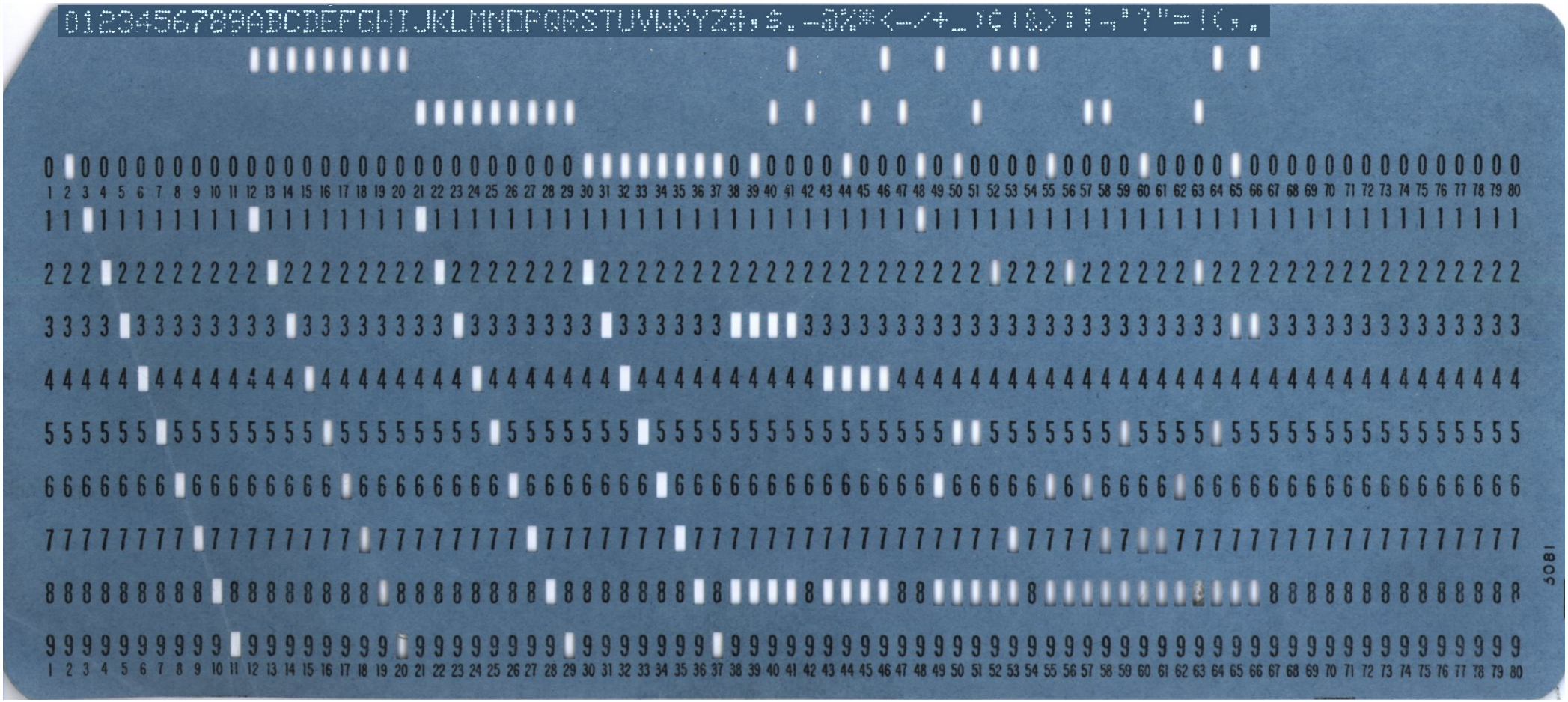
- Anglické kódování
  - **ASCII** kód (**A**merican **S**tandard **C**ode for **I**nformation **I**nterchange)
  - **EBCDIC** kód – mainframy
- České kódování
  - Kód Kamenických
  - Kód PC Latin – 2
  - Kód ISO 8859-2
  - Kód Win 1250

Michal Bořek

ASCII: 60. léta

$b_7 b_6 b_5$ →	000	001	010	011	100	101	110	111
Bits / b_4 b_3 b_2 b_1 / Row ↓ / Column →	0	1	2	3	4	5	6	7
0 0 0 0	0	NUL	DLE	SP	0	@	P	`	p
0 0 0 1	1	SOH	DC1	!	1	A	Q	a	q
0 0 1 0	2	STX	DC2	"	2	B	R	b	r
0 0 1 1	3	ETX	DC3	#	3	C	S	c	s
0 1 0 0	4	EOT	DC4	$	4	D	T	d	t
0 1 0 1	5	ENQ	NAK	%	5	E	U	e	u
0 1 1 0	6	ACK	SYN	&	6	F	V	f	v
0 1 1 1	7	BEL	ETB	'	7	G	W	g	w
1 0 0 0	8	BS	CAN	(	8	H	X	h	x
1 0 0 1	9	HT	EM	)	9	I	Y	i	y
1 0 1 0	10	LF	SUB	*	:	J	Z	j	z
1 0 1 1	11	VT	ESC	+	;	K	[	k	{
1 1 0 0	12	FF	FS	,	<	L	\	l	|
1 1 0 1	13	CR	GS	—	=	M	]	m	}
1 1 1 0	14	SO	RS	.	>	N	^	n	~
1 1 1 1	15	SI	US	/	?	O	_	o	DEL

Děrný štítek, EBCDIC



Čeština a její kódování

- 8-bitové – historicky i nehistoricky mnoho standardů:
  - **`KOI8-CS`** – kolem 1979
  - **`bratří Kamenických`** – kolem 1985
  - **`ISO-8859-2`** (ISO Latin 2) – od 1987
  - **`CP852`** (PC Latin 2) – asi 1990, 1991 (MS-DOS)
  - **`Windows 1250`** (CP1250) – asi 1992 (Windows 3.1)
- *"Embrace, extend, extinguish"*
  - strategie užívaná Microsoftem: adopce existujících standardů, jejich rozšíření o proprietární funkce a využití těchto rozdílů k silnému znevýhodnění konkurence

Michal Pecánek

Čeština a její kódování

- **Unicode** – jednotné kódování pro všechny znaky národních abeced
  - Unicode 1.0 již v roce **1991**, původně 16-bitový
  - Unicode 2.0 (rok 1996), již 32-bitový a od této verze funguje zpětná kompatibilita (znaky se jen přidávají)
  - **UTF-8** – variabilní délka kódovaných znaků, jeden znak se kóduje do jednoho až čtyřech znaků (dle pozice znaku v kódovací tabulce), zpětně kompatibilní s ASCII (nejpoužívanější)
  - *UTF-16* – kódování 1 znaku do jedné až dvou 16-bitových jednotek
  - *UTF-32* – kódování 1 znaku do jednoho 32-bitového znaku

Michal Perešíni

FTP

File
Transfer
Protocol

Michal Dobeš

FTP

- klasický protokol RFC 959 (1985)
- **klient-server** protokol
- spojované spojení **nutnost loginu**
  - přihlášení je textové, údaje se nešifrují
    - až později rozšířeno o možnost šifrování (FTPS)
  - anonymní režim: také vyžaduje přihlášení, ale bez hesla (nebo s e-mailem jako heslo)
    - server musí podporovat
    - pro veřejné FTP servery

FTP

- Cíle protokolu:
  - Sdílení souborů se světem
  - Podpora vzdáleného užití počítačů
  - Přenos souborů mezi vzdáleným a lokálním počítačem
- V dnešní době již **citelný úpadek**
- Podpora FTP ze strany prohlížečů u Firefoxu a Chromu skončila v roce 2021
- dnes nahrazeno protokoly **SFTP** či **HTTP**

Michal Bořek

FTP – způsob komunikace

- Používá dva **`TCP`** porty: **`20`** a **`21`**
- **Příkazový** / **řídící kanál** pracuje na portu **`21`**
  - přenáší **příkazy** z klienta na server a **hlášky** serveru na klienta
- Vytváří **datové spojení** na portu **`20`**
  - přenáší data přenášených souborů i **výpisy příkazů** (např. příkaz `LIST` k vylistování obsahu adresáře)
  - dva režimy vytváření datového spojení
    - **Aktivní režim**: klient vytváří řídící spojení na server, server vytváří datové spojení na klienta (nefunguje když je klient za NATem!)
    - **Pasivní režim**: klient vytváří obě spojení

FTP příkazy

- přenášejí se přes řídící spojení (port **21**)
- **Příkazy FTP klienta** (výběr)
  - `cd`, `lcd`, **`dir`**, `ls`, `mkdir`, `rmdir`
  - **`get`**, **`put`**, `mget`, `mput`
  - `ascii`, `binary`, `prompt`, `hash`
- **Příkazy FTP protokolu** (výběr)
  - **`USER`**, **`PASS`**, `TYPE`, **`STOR`**, `SIZE`, **`RETR`**, `REST`, `QUIT`, `PWD`, `PORT`, `PASV`, `NOOP`, **`LIST`**, `MKD`, `HELP`, `CWD`, `ABOR`

Michal Rešetka

FTP – aktivní a pasivní režim

Aktivní režim

- klient se připojuje z neprivilegovaného portu **`N>1023`** na řídící port serveru (**`21`**)
- klient začíná naslouchat na portu **`N+1`** a posílá FTP příkaz **`PORT N+1`** na server
  - **`PORT h1,h2,h3,h4,p1,p2`** `h1..h4`: IP adresa, `p1,p2`: port
  - číslo portu = `p1*256+p2`
- server se připojí zpět na datový port klienta z datového portu serveru (**`20`**)
- problém s firewally a NATem

Michal Bečák

Ukázka aktivního režimu

```
telnet 4iz110.vse.cz 21
USER anonymous
PASS void                          netcat -l -p 5000
PORT 127,0,0,1,19,136
LIST                               netcat -l -p 5001
PORT 127,0,0,1,19,137
RETR textovy_soubor
```

Vzoreček pro kalkulaci čísla portu

číslo portu = p1*256+p2

FTP – aktivní a pasivní režim

Pasivní režim

- **klient** otevírá **obě spojení** (**příkazové** i **datové**)
- klient otevře **dva** náhodné neprivilegované porty **`N>1023`** a **`N+1`**
- první port kontaktuje server na portu **`21`**, ale místo příkazu PORT použije příkaz **`PASV`**
- server následně otevře náhodný neprivilegovaný port **`P>1023`** a pošle **`P`** zpět klientovi jako odpověď na příkaz **`PASV`**
- klient otevře spojení ze svého **`N+1`** portu na port **`P`** serveru pro přenos dat

Ukázka pasivního režimu

```
telnet 4iz110.vse.cz 21
USER anonymous
PASS void
PASV                              telnet localhost port
LIST
PASV                              telnet localhost port
RETR textovy_soubor
```

Vzoreček pro kalkulaci čísla portu

číslo portu = p1*256+p2

FTP

- **Nutnost přihlášení** pro přístup k datům
- **Jméno** a **heslo** se posílá **nešifrovaně**
- **Aktivní režim** přenosu interferuje s **NAT**em u domácích ISP
- **Bezpečnostní zranitelnosti protokolu** (viz RFC 2577)
  - **FTP bounce attack** - zneužití FTP serveru jako proxy pro skenování portů oběti či síťové útoky
    - příkaz PORT akceptuje jako parametr **IP adresu!** a číslo portu
  - **Port stealing** - uhodnutí portu útočníkem a získání kontroly nad legitimním spojením
  - a další!

Michal Rešetka

FTP – současnost

- V současné době FTP upadá, často se používal pro:
  - veřejný repozitář souborů pro stahování
  - přístup k souborům webhostingu
  - může být stále ještě používán v rámci IoT nebo embedded zařízení
- FTP je postupně nahrazován protokoly
  - **HTTP** – při přístupu k veřejným datům
  - **SFTP** – při přístupu k soukromým datům

Michal Račák

Telnet

Telnet

Michal Dočekal

Telnet protokol

- Klasický protokol (rok 1983)
- Definován jako STD8 a v RFC854 a 855
- Cíl telnetu je nabídnout dvousměrnou dosti obecnou komunikační službu založenou na 8-bitových bytech
- Emulace terminálu tj. standardní přístup k počítači v 70. letech

Michal Bečák

Telnet protokol

- Připojí TCP kanál na klávesnici a obrazovku
- Různé typy terminálů
  - IBM svět - 3270
  - DEC svět - VT 100, VT 220, VT 320, …
  - Unix vznikl na DEC HW a proto VT častější.

Michal Bečák

Telnet – bezpečnost

- Telnet má závažné bezpečnostní nedostatky.
- Celý protokol komunikuje nešifrovaně
- Nepřítomnost zabezpečovacích mechanismů umožňuje zachycovat a měnit komunikaci

Michal Bečák

Posuďte sami tento zachycený úrývek komunikace

```
rosnatka login: karelkarel
.
Password: neuhadnutelneheslo
.
Linux rosnatka 3.2.0-4-amd64 #1 SMP Debian 3.2.68-1+deb7u3 x86_64
.]0;karel@rosnatka: ~.karel@rosnatka:~$ llss  --ll.[D.aal.
.
celkem 20
drwxr-xr-x  2 karel karel 4096 lis 11 12:44 .[0m.[01;34m..[0m
drwxr-xr-x 57 root       root       4096 lis 11 12:44 .[01;34m...[0m
-rw-r--r--  1 karel karel  220 lis 11 12:44 .bash_logout
-rw-r--r--  1 karel karel 3392 lis 11 12:44 .bashrc
-rw-r--r--  1 karel karel  675 lis 11 12:44 .profile
.]0;karel@rosnatka: ~.karel@rosnatka:~$
```

Posuďte sami tento zachycený úrývek komunikace

```
rosnatka login: karelkarel
.
Password: neuhadnutelneheslo
.
Linux rosnatka 3.2.0-4-amd64 #1 SMP Debian 3.2.68-1+deb7u3 x86_64
.]0;karel@rosnatka: ~.karel@rosnatka:~$ llss  --ll.[D.aal.
.
celkem 20
drwxr-xr-x  2 karel karel 4096 lis 11 12:44 .[0m.[01;34m..[0m
drwxr-xr-x 57 root       root       4096 lis 11 12:44 .[01;34m...[0m
-rw-r--r--  1 karel karel  220 lis 11 12:44 .bash_logout
-rw-r--r--  1 karel karel 3392 lis 11 12:44 .bashrc
-rw-r--r--  1 karel karel  675 lis 11 12:44 .profile
.]0;karel@rosnatka: ~.karel@rosnatka:~$
```

Telnet – současnost

- Používání se omezuje pouze na situace, kdy je kryptografická bezpečnost nahrazena fyzickou
  - komunikace uvnitř servrovny, v domácí síti, apod.
- Nahrazen protokolem SSH na vzdálený přistup přes nezabezpečenou linku.

Michal Bečák

Telnet na cvičeních 4IZ110

- telnet je jednoduchý textový **`TCP klient`**
- lze jej využít k popovídání si se servery, které poskytují služby prostřednictvím textových protokolů (SMTP, HTTP, atd.)
- na školních Windows není
- ale na serveru **`4iz110.vse.cz`** ano

Michal Bližňák

SSH

Secure
SHell

Michal Růžek

```
SSH-2.0-OpenSSH_7.4p1 Debian-10+deb9u7
SSH-2.0-OpenSSH_7.9p1 Debian-10+deb10u3
.........E......CEy.3;...0curve25519-sha256,curve25519-sha256@libssh.org,ecdh-sha2-nistp256,ecdh-sha2-
nistp384,ecdh-sha2-nistp521,diffie-hellman-group-exchange-sha256,diffie-hellman-group16-sha512,diffie-
hellman-group18-sha512,diffie-hellman-group-exchange-sha1,diffie-hellman-group14-sha256,diffie-hellman-
group14-sha1,ext-info-c..."ecdsa-sha2-nistp256-cert-v01@openssh.com,ecdsa-sha2-nistp384-cert-
v01@openssh.com,ecdsa-sha2-nistp521-cert-v01@openssh.com,ssh-ed25519-cert-v01@openssh.com,ssh-rsa-cert-
v01@openssh.com,ecdsa-sha2-nistp256,ecdsa-sha2-nistp384,ecdsa-sha2-nistp521,ssh-ed25519,rsa-sha2-
512,rsa-sha2-256,ssh-rsa....chacha20-poly1305@openssh.com,aes128-ctr,aes192-ctr,aes256-ctr,aes128-
gcm@openssh.com,aes256-gcm@openssh.com,aes128-cbc,aes192-cbc,aes256-cbc....chacha20-
poly1305@openssh.com,aes128-ctr,aes192-ctr,aes256-ctr,aes128-gcm@openssh.com,aes256-
gcm@openssh.com,aes128-cbc,aes192-cbc,aes256-cbc....umac-64-etm@openssh.com,umac-128-
etm@openssh.com,hmac-sha2-256-etm@openssh.com,hmac-sha2-512-etm@openssh.com,hmac-sha1-
etm@openssh.com,umac-64@openssh.com,umac-128@openssh.com,hmac-sha2-256,hmac-sha2-512,hmac-sha1....umac-
64-etm@openssh.com,umac-128-etm@openssh.com,hmac-sha2-256-etm@openssh.com,hmac-sha2-512-
etm@openssh.com,hmac-sha1-etm@openssh.com,umac-64@openssh.com,umac-128@openssh.com,hmac-sha2-256,hmac-
sha2-512,hmac-
sha1....none,zlib@openssh.com,zlib....none,zlib@openssh.com,zlib..............................  .5,.E^
.#........curve25519-sha256,curve25519-sha256@libssh.org,ecdh-sha2-nistp256,ecdh-sha2-nistp384,ecdh-
sha2-nistp521,diffie-hellman-group-exchange-sha256,diffie-hellman-group16-sha512,diffie-hellman-group18-
sha512,diffie-hellman-group14-sha256,diffie-hellman-group14-sha1...!rsa-sha2-512,rsa-sha2-256,ssh-
rsa...lchacha20-poly1305@openssh.com,aes128-ctr,aes192-ctr,aes256-ctr,aes128-gcm@openssh.com,aes256-
gcm@openssh.com...lchacha20-poly1305@openssh.com,aes128-ctr,aes192-ctr,aes256-ctr,aes128-
gcm@openssh.com,aes256-gcm@openssh.com....umac-64-etm@openssh.com,umac-128-etm@openssh.com,hmac-sha2-
256-etm@openssh.com,hmac-sha2-512-etm@openssh.com,hmac-sha1-etm@openssh.com,umac-64@openssh.com,umac-
128@openssh.com,hmac-sha2-256,hmac-sha2-512,hmac-sha1....umac-64-etm@openssh.com,umac-128-
etm@openssh.com,hmac-sha2-256-etm@openssh.com,hmac-sha2-512-etm@openssh.com,hmac-sha1-
etm@openssh.com,umac-64@openssh.com,umac-128@openssh.com,hmac-sha2-256,hmac-sha2-512,hmac-
sha1....none,zlib@openssh.com....none,zlib@openssh.com.....................,.....  .s..F.T;..#.....\...F
...'.Y7...C.........d..........ssh-rsa.............
.k:X.....N...2{=....._..a....X..5ZJ..]...|..N.h.....qYuR..F.i0.b.?
s.o[O.1..B...  ..L......'.<.a.....>_D..p..R...#.u...*.m9.n.I.....OX...-..B...
[3..5..e..!...i....T8.....E.E...2...5P....h'(O.......H.l7._ko.D-R.......q...~..4
.c....e.........B.QY3..:.4.......  ..M".HN...j.t.yf...x..Z:..&!J>o:........rsa-sha2-
512....k.\..).t..o....;.b7...i..W...........H\...I.[0Q+B..!..l.+  ....Qu...^.8.)rK.`.e..9.....)b.
+0....`:.|.m...l..S..F......H.....C...Y...F@.peM.V3=.q._.PMJ.r_.s..
. .W.= }]dY.%..Mn......
+........_P@..]x!.........e.d..\...__.2.>..~...Q.#u.uc....*.K...c7oFB.pA*.Wh.V..............
............$...**.e.(h./.....yt.W...&{..
```

```
>Xh.C.
...
.:2.e*.5...2L.47.........).,.......(o".........s.H.8[Z..XY.[.x....dd&..<..Q.2...l.m...  N;&4C[B.........T.D......_.._.-Mfm.........
..........r.X.x....<.`=9..2..&....N.dic[>...J.h6I?..._v..=.M{.....7j9.8
.....Q.......1...V|..........B*....E.g..?Di
......0q..i|7..z....
.#...c.."}.....D.=M.....$.kw....]J....l;h..S;.......v...-l`...m..O....B.o..$..y..,%\.#G..P..M.b.Zz'.&...(U...|./"d...p...*.'...L
.XsIyH..J.....^.}Ex........#..........h.A.9...`..Lo..v...Ev.N.qm..nmP.../.V'....Z....S..N.c...\........\\3X.^]
+K#...*4.2$:...M.....n'ZP5..$..Q.....u.@RM......j_.!..b.0..S#.D...d.m.}.U..J. ..u...j
...V....qf.~....D}z.UUPu.....0'......  %..#..Y...wQA.......#/3....i...p.pq..  P......t...LG).....-.5x..2.6[..j....M...
[Zu........D.T...m.....i...MP...
|.6y.....U..[..1v......r.~..M......Y.....J..].^...0..A.u...=..DY.......h./w._..\....t1.g..i....4..@..[.ko....4...M.I...
%..0~.'.9z.6......:.+F.....@E..V...05&.......P.?.....W.A.....Z@.......xh^....~..01f9b...4(.Z.u....(..j.l...Y...a..\....-H....k......F$.\
U.w....b4..G./.... .o'.oD...X.L.7:.....>.N..\..19......}.Ctv..........Mo=>...V.|.k........,Z...A..vG..;.......F...... ..I.v.`...).n.`.O>...-...
i....%8..s.r.g..=H.t.h...|RVY..f.7H.QC....:.~..k.]h./.b...>j.j.%.......5.i.M...o....,~..e........7..-@y.v.....i.Oh........m.;..9.a..9.....`..Z.
{wk
Q!.[....#...M......=. ....l.N ..a..#...ftI..sM.Y...2.>.UAT.M..)m.p.?..GU.^u.....iN@......zF~...!..::@.1...f.....N..!{..ydY
.ww.;...%....r.....?mw..%..K....s.i.>..=f9.....].n..l.|.,...%y.
{.Jw....].......DO.Ut......-.E.s.4..+....p....&'....QT..0FgT.1.....U.........Fz.....%.\.r.t.v.BU.bH.z..g~.}!.........Hj.  ..=c..OU....#..
n@.(...g".o.
..Df~!.%..<.r.up.|7.}....e.G(K.%...X..
..,.3.......[.4.;.kb..0R.R.......6MkX.mB.
]..m,;.....xG.-.n..g_=..G3..[A.z...j]N.*.~..Q..4. V..;..|c.......-Mxw.....e.{.P..4.....?.A......_;....!zB.....G...WA..~@.IP.
$90.x.......j.Jy....f.t.[........;|..@.._.3.k...x ....8...B"B.?.\W..U..6...%p....z.....M.az2....Y.ot..p.Y..7.2.5......N0).c..=..=;.}.}v
...
.p"h...?.q..6.\4......m.d..H//%\R.......M.EP1?....z..nL...U.......,..m(...i.9sE..5.k!.94.?...NF..}......&.o.f.....$....OS.. .k]P.!..2,^.N
~....N.g~.U.....t!.9.H.).........HL.O|.nS...?M....1%2^NO...I.%.....4.....oXT./ql.'.....+Ke...........a.G_8.....zgaxU+..X.)..ON.G.
&.S.uQ&m............U.t,z..Q.H......l7..zz}..A.H....m%XK.$..eV...{...3...A:.....5.:s.S..W.f...#...z.....U.Tv..........*.Z-.#
......w.+...\.-.....C.$...
.o..9......kC..7.^1.b.kL.....sVN........2.H...*..b3..y@.......?^i.N..{......k......c........y.U..e..o.......<..(.....+....r:........b.Gd...
+..@.
f*.h$Mgu.4...y1...(..f.o..yT.Y.e.\..*H?g9.I4.."!.....d..x$.z..Q....y^.....K.(.4?..Ue^...$...v... ..x.o.ckQ../t.\..ETRf...W|t.sE+.D..
{..#.....2.'..~..{#U.8...4...4.W./'.IE.I=..t.......-..b.I..dw^B;..;..xU<}.._
..Vz:uC.5    P...H$f...W...I4:g^kA}..X......):c.......m..th...8..=...r....o<....e.wZ".!...\......|..x>..U..X&Dj#w.=.....R........
{.?..j....q.C./..;......E$u.Q........
.&t.]..U.T..-*..F[...t.=q.v..p....]M...b.9JHo>).....B0'.V..k^.q...)..0........Q:^.1(r:$..v.MT.C[..'...|R.~
...f..d.....tGz8..W.+Ix.h3....f9.....Aj6.d..~..".{.p.@O..L%..
.l..R..$........OE0....ET.K.i.Nh....S.F....2.....<}.-.zx.q_..E.Y.|...P....~>..q.\._?...o...)...>.Pe...............?$
%.m..........t...|.vB..uC5..I..........F.D... ....
.H#..D^X...5{..L.W.....x....C3....K..O.;[..|... .6..;.o..:......u....+....0j..T..>... ...{O1v'.?
m.....H..WG...~i.E5... .q...EIi7.......:."..fC.i..!..a.....Yl..I...X...k..n....G../...-.V.r0...t.[wC.s.'...........l...>  =.
$..6........rB....s.z;.j%..r...H....>..........s.....2$........I..(......n...j@...MeM..7..%t\.u%g.%.j....`0L...g...&.H|.QO.% ....$........
$m..cU......C.\8....W...sFT.....................~*q......pn[.fJ..._..~aU.
.......Q..].]............[...r. 2HF.j!7w....Z......0..N..e+.pY"J5H.....bw.?vA.._.!.P3..O...:.V.-..F...eH........7H.(......
(\....wk.....^G.i.T.>....Q..p.Bi*p&..x......XD......8.....g8
M.:.xZ.G...?.QQ.m`.bm.......&`.a.....:=2=..\F.q..u.=..z.
...5".......
[..G..$.L......+....&....P....y.....&./..YT..9%.....      .bpb.V.E...P.@6
{.
```

SSH protokol

- **Secure SHell**: bezpečný komunikační protokol, náhrada protokolů `telnet`, `rsh`, `rlogin`
- „Švýcarský nůž"
  - Umožňuje přístup k příkazové řádce (shellu)
  - Umožnuje bezpečný přenos souborů SFTP
  - Umožnuje tunelování portů a X windows
  - Umožňuje vytvořit SOCKS proxy i VPN
- Bezpečná je pouze **verze 2** protokolu.
- Série RFC dokumentů RFC **4251-4256**

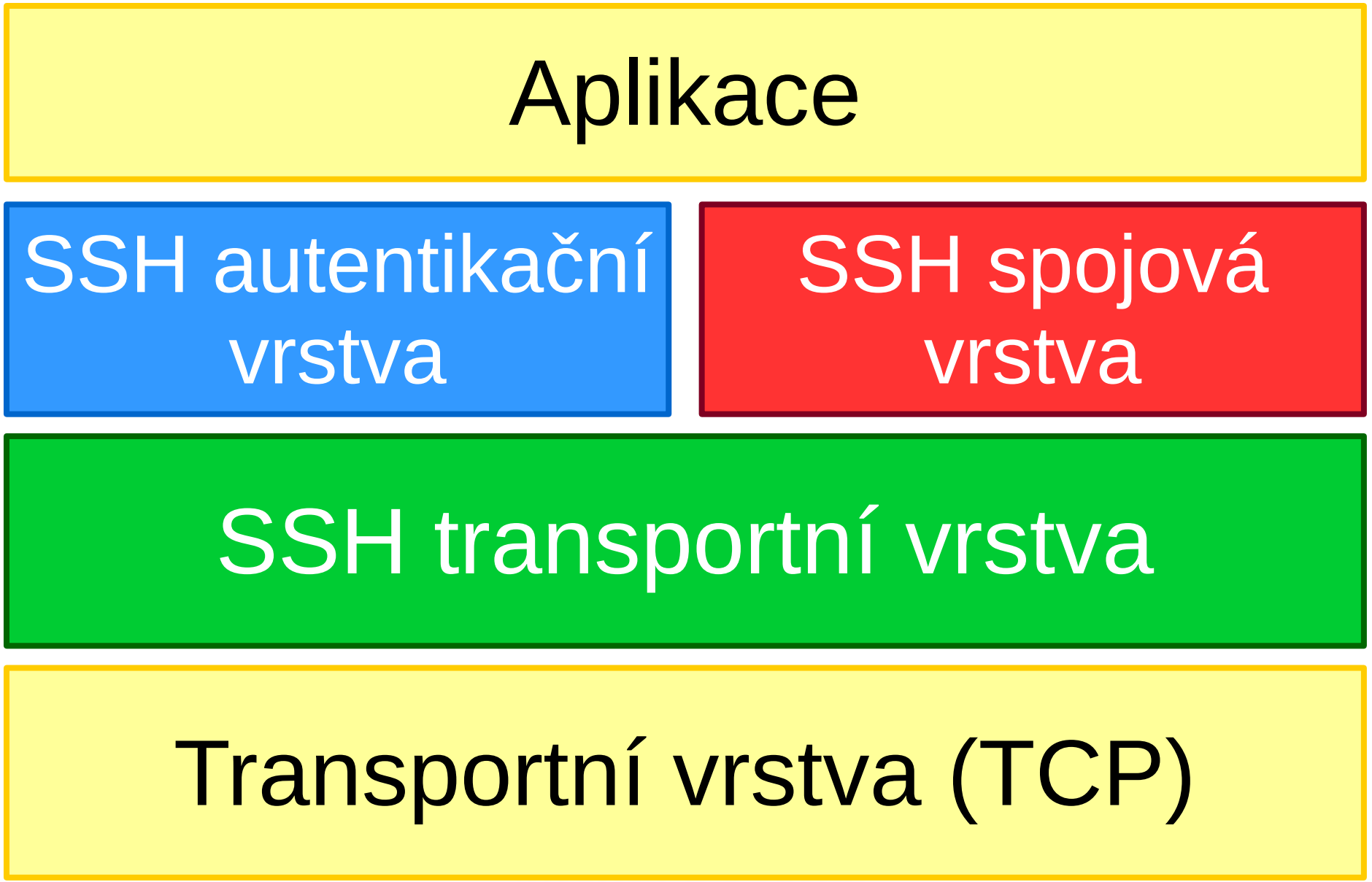
Michal Račák

SSH protokol

- vzniklo více implementací, komerční i FOSS
- nejrozšířenější: **OpenSSH** (z OpenBSD)
- protokol vyrobí **šifrovaný autentikovaný tunel**, který je možný využít mnoha způsoby od přístupu k příkazové řádce, jiným službám až po kompletní VPN postavenou nad SSH
- SSH protokol používá **TCP** port **22**
- protokol má vrstvenou architekturu

Michal Rešetka

Vrstvy SSH protokolu



Vrstvy SSH protokolu

- **transportní vrstva** (transport layer, RFC 4253)
  - úvodní **výměna klíčů**, autentikace **serveru**
  - nastavuje a zahajuje **šifrování**, **kompresi** a ověření **integrity** přenášených dat (že nebyla cestou pozměněna – klíčové pro bezpečnost šifrování!)
- **autentikační vrstva** (user authentication layer, RFC 4252)
  - autentikaci řídí klient, může se ověřit libovolnou metodou podporovanou serverem
- **spojová vrstva** (connection layer, RFC 4254)
  - SSH spojení může být multiplexováno do více logických **kanálů**, kde každý přenáší data současně a obousměrně

Michal Poceta

Vrstvy SSH protokolu

- možnosti autentikační vrstvy:
  - autentizace **heslem** (bývá vypnuta)
  - autentizace **veřejným klíčem**
    - veřejný klíč je na serveru v `$HOME/.ssh/authorized_keys`
    - soukromý klíč má klient, chráněný heslem, např. v: `$HOME/.ssh/id_ecdsa`
  - **interaktivní klávesnicová** autentizace (RFC 4256)
    - např. jednorázové kódy
  - externí autentizace přes **GSSAPI**
    - Generic Security Service Application Program Interface

Michal Račák

Vrstvy SSH protokolu

- **spojová vrstva**
  - implementuje **kanál** (channel), **žádost kanálu** (channel request) a **globální žádosti** (global requests)
  - kanály operují **současně** a komunikace je **obousměrná**, každý kanál představuje **jednu** službu
  - **každý** kanál řídí svůj vlastní **datový tok** pomocí velikosti okna příjemce (receive window size)
  - typy kanálů: **shell** (pro příkazovou řádku, SFTP, atd.), **direct-tcpip** (tunely client → server, `-L`) a **forwarded-tcpip** (tunely server → client, `-R`)

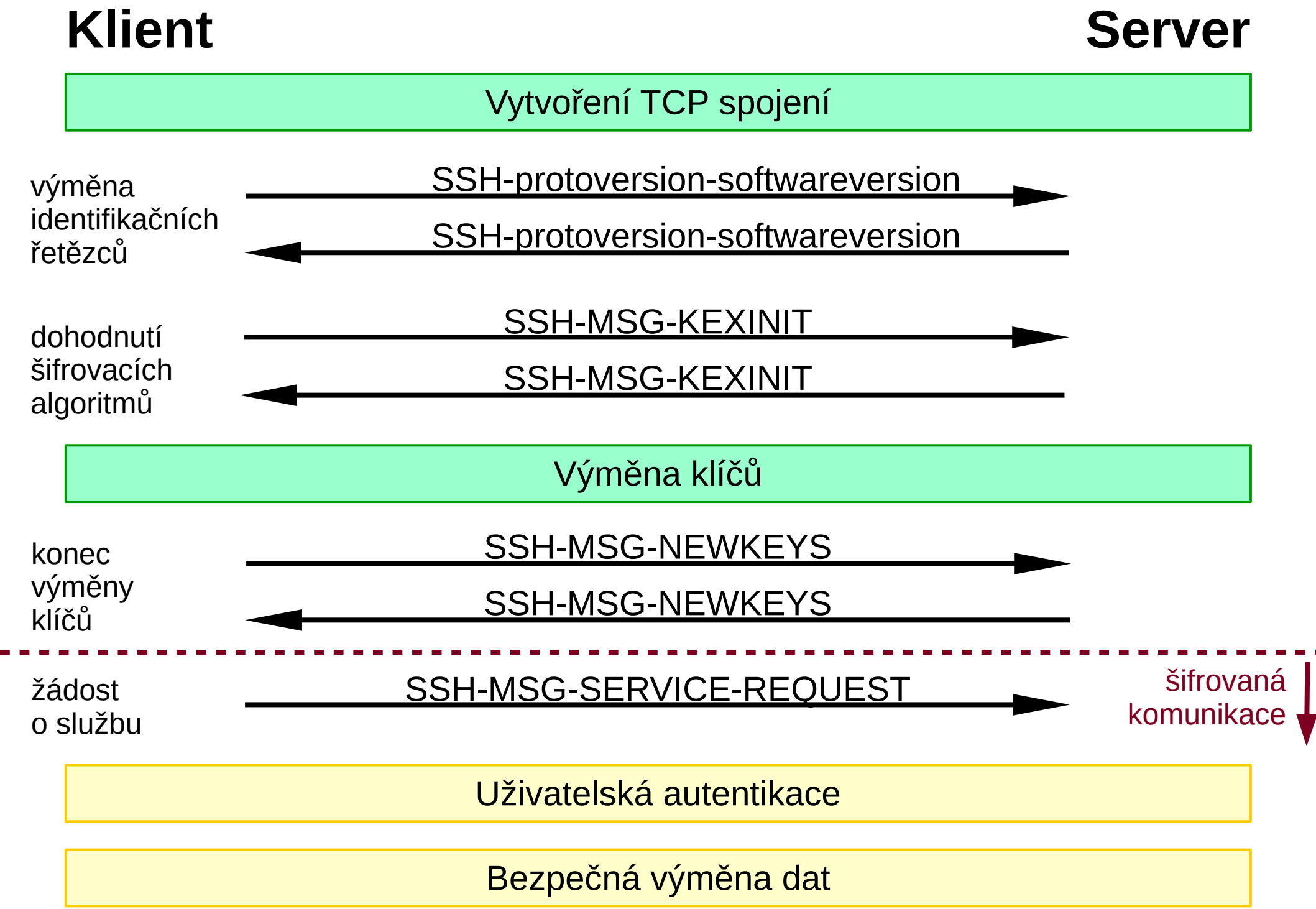
Michal Ďačeľ

SSH ověření identity

- ověření identity je **obousměrné**, nejprve se autentikuje **server klientovi** (v rámci fáze výměny klíčů, klíče jsou podepsané serverovými **host keys**):

```
The authenticity of host '4iz110.vse.cz
(2001:718:1e02:18::43)' can't be established.

RSA key fingerprint is
SHA256:sv4okaXPnhxiCZl3kls4nQhHve1jsK8ViOT7vwDckZM.
```

- pozor, na ověření serveru stojí bezpečnost celého SSH spojení – nepodceňovat!
- poté se klient už v rámci vytvořeného SSH spojení nějakým způsobem autentikuje serveru


Klient
Server
Vytvoření TCP spojení
výměna identifikačních řetězců
SSH-protoversion-softwareversion
SSH-protoversion-softwareversion
dohodnutí šifrovacích algoritmů
SSH-MSG-KEXINIT
SSH-MSG-KEXINIT
Výměna klíčů
konec výměny klíčů
SSH-MSG-NEWKEYS
SSH-MSG-NEWKEYS
žádost o službu
SSH-MSG-SERVICE-REQUEST
šifrovaná komunikace
Uživatelská autentikace
Bezpečná výměna dat

Přes SSH je možné provozovat ledacos

- příkazová řádka (shell)
- připojení k X11 (X11 forwarding) a možnost lokálně pouštět vzdálené **GUI** aplikace
- klient-server forwarding
- server-klient forwarding
- dynamický forwarding přes SOCKS proxy
- SFTP
- VPN
- atd.

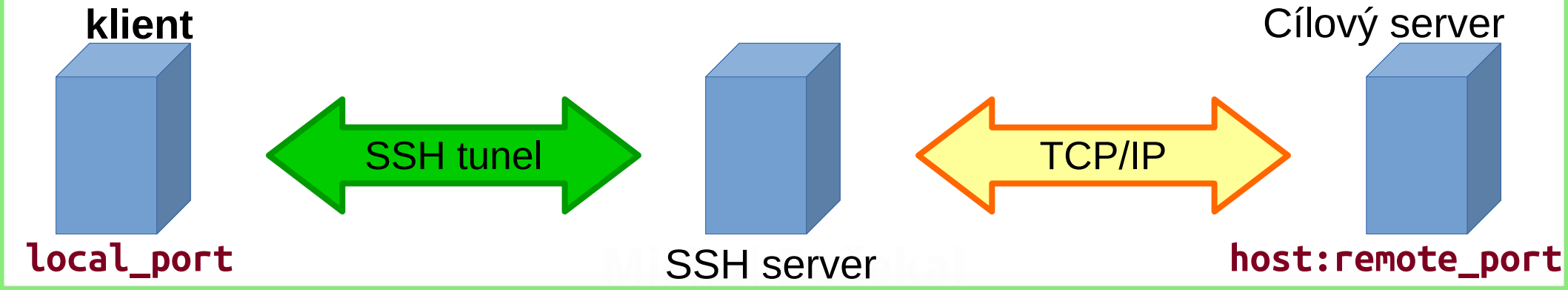
Michal Ručaj

Klient – server tunelování (lokální port forwarding)

možnost zpřístupnění vzdálené služby jako služby místní, tzv. „lokální port forwarding“:

```
ssh -L local_port:host:remote_port usr@srv
```

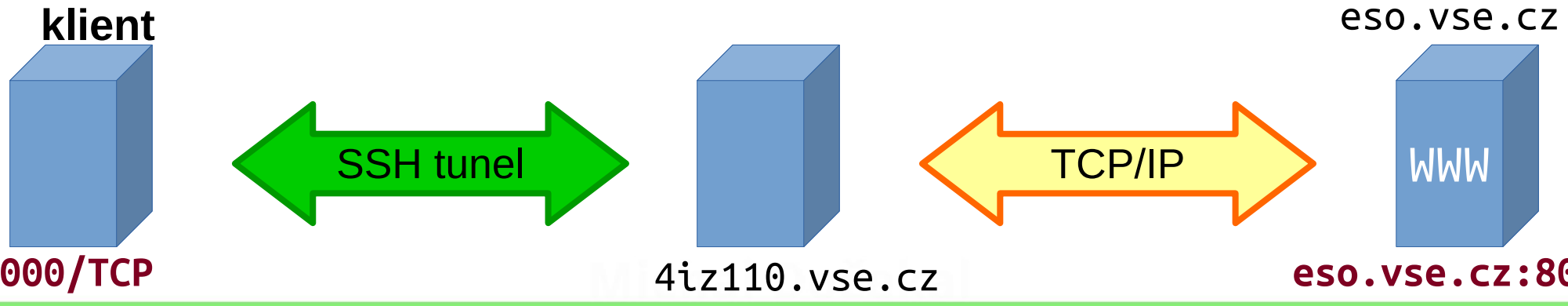
- **`local_port`**: místní port, na kterém bude zpřístupněna vzdálená služba
- **`host`**: hostitel vzdálené služby
- **`remote_port`**: port vzdálené služby



Klient – server tunelování (lokální port forwarding)

Příklad č. 1:

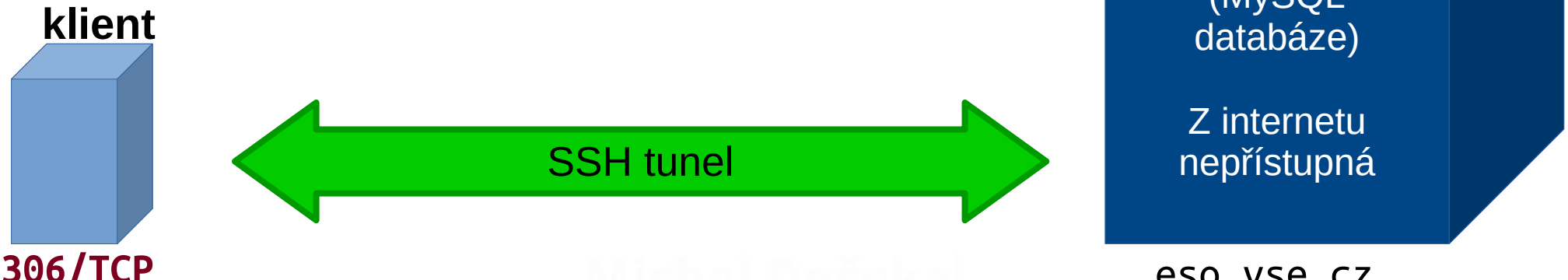
```
$ ssh -N -L 8000:eso.vse.cz:80 usr@4iz110.vse.cz
$ telnet localhost 8000
Connected to localhost.
Escape character is '^]'.
GET / HTTP/1.1
Host: eso.vse.cz

HTTP/1.1 302 Found
Date: Sat, 28 Oct 2023 13:07:42 GMT
Server: Apache/2.4.56 (Debian)
Location: https://eso.vse.cz/
```



Klient – server tunelování (lokální port forwarding)

Příklad č. 2:

```
$ ssh -N -L 3306:localhost:3306 usr@eso.vse.cz
$ mysql -u mysql_usr -p -h 127.0.0.1
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 404198
Server version: 10.5.19-MariaDB-0+deb11u2 Debian 11

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help.

MariaDB [(none)]>
```

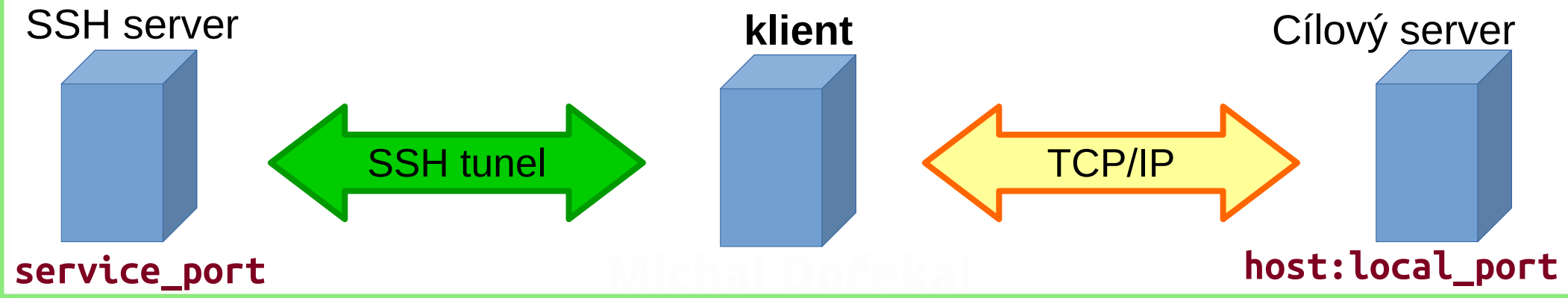


Server – klient tunelování (vzdálený port forwarding)

možnost zpřístupnění místní služby na vzdáleném počítači, tzv. „vzdálený (remote) port forwarding":

```
ssh -R service_port:host:local_port usr@srv
```

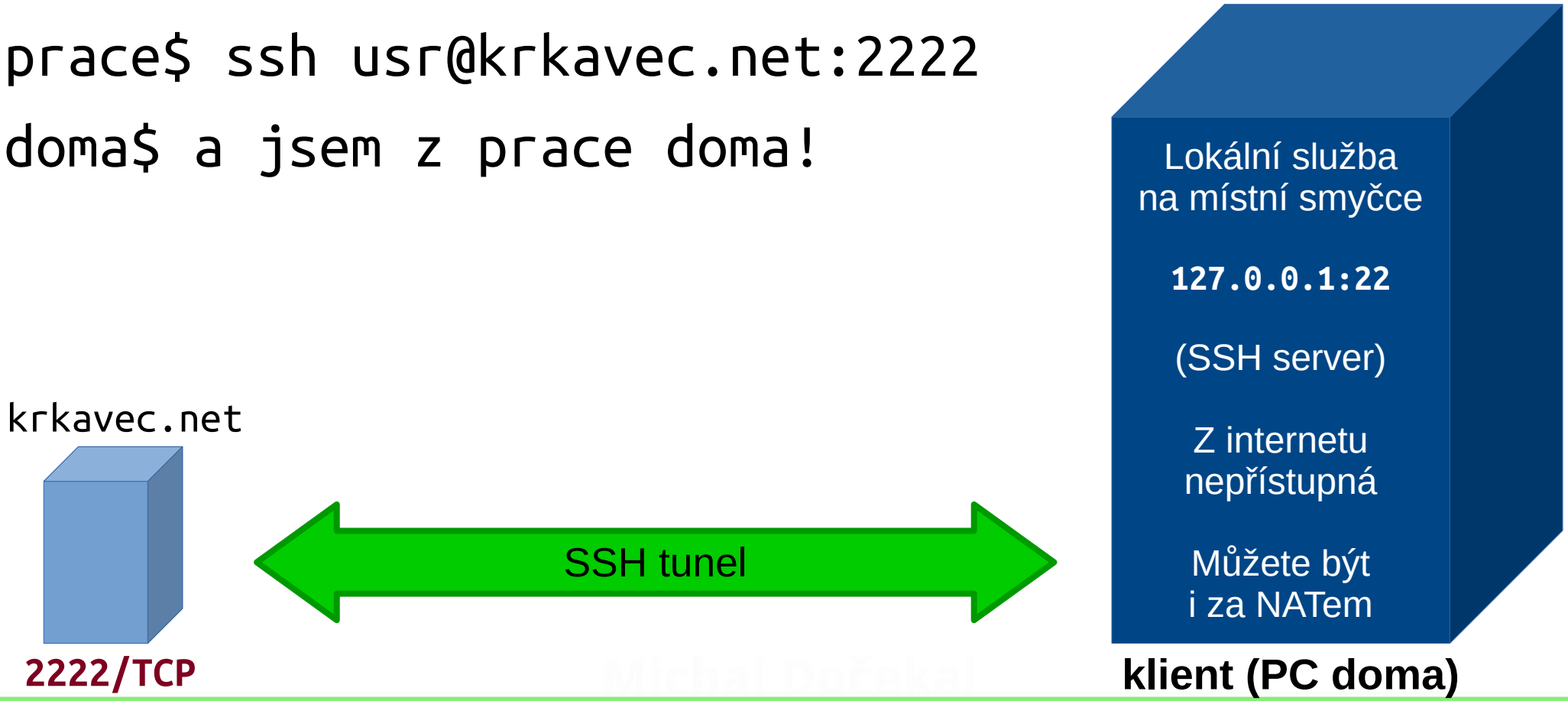
- **`service_port`**: port na SSH serveru, na kterém bude zpřístupněna lokální služba
- **`host`**: hostitel lokální služby
- **`local_port`**: port lokální služby



Server – klient tunelování (vzdálený port forwarding)

Příklad:

```
doma$ ssh -N -R 2222:localhost:22 usr@krkavec.net
prace$ ssh usr@krkavec.net:2222
doma$ a jsem z prace doma!
```



SSH SOCKS proxy

- **SOCKS** je protokol pro předávání paketů mezi klientem a serverem přes proxy server (RFC 1928)
  - `IP datagram → SOCKS proxy → cílový server`
  - `cílový server → SOCKS proxy → IP datagram`
- SSH umí tzv .dynamický forwarding paketů právě pomocí vlastní implementace SOCKS proxy:

  ```
  ssh -D cislo_portu
  ```

- přes dané **číslo portu** na localhostu je pak možné SOCKS protokolem předávat pakety a SSH server pak pracuje jako proxy server

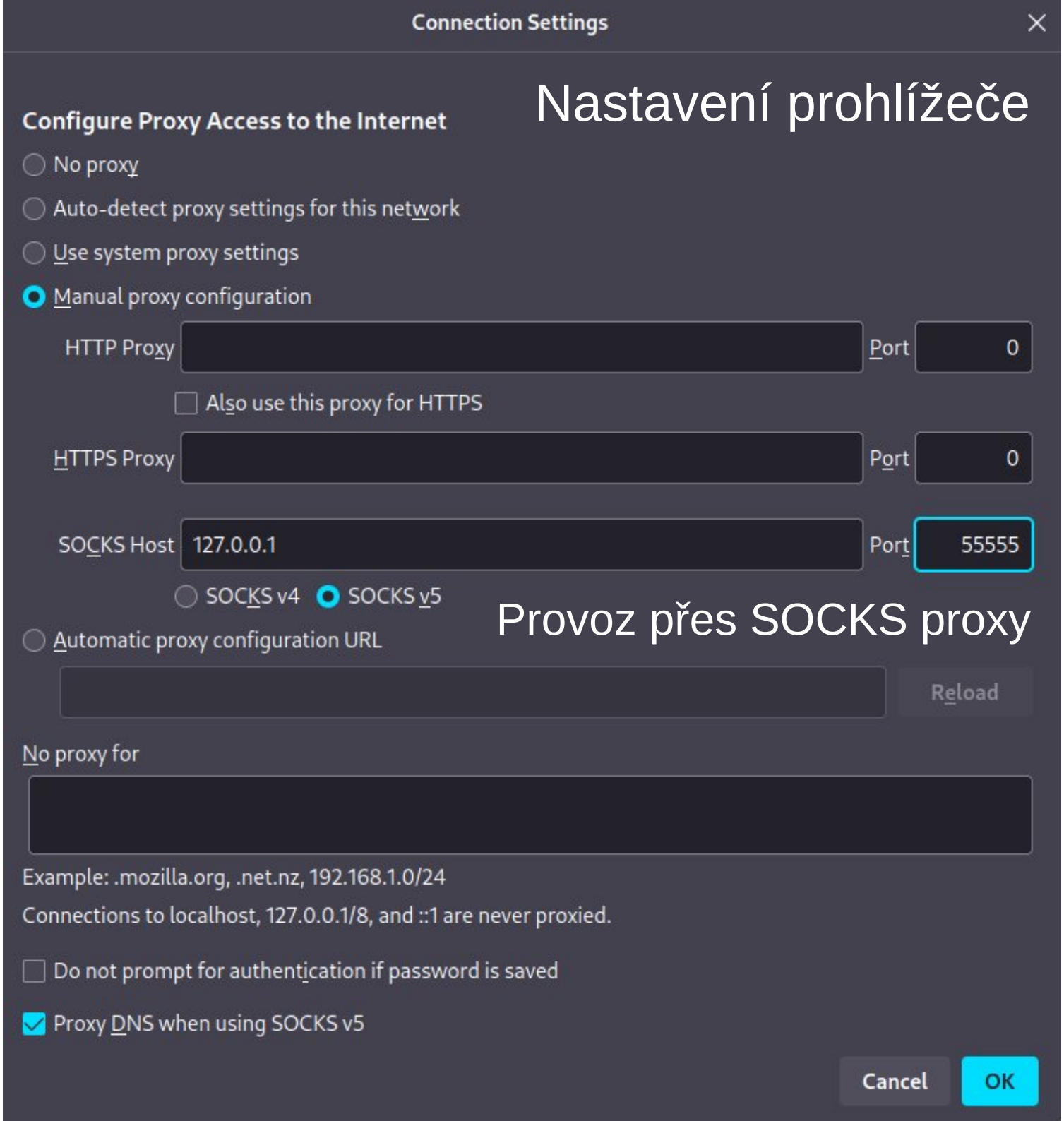
Michal Rešetka

SSH SOCKS proxy

- SOCKS protokol verze 5 umí navíc přenášet i UDP pakety a podporuje autentizaci (zde nepoužitá)
- předávat pakety na privilegované porty (číslo portu `< 1024`) je možné pouze s právy roota
- je možné specifikovat i místní IP adresu a teoreticky poskytovat SOCKS proxy pro další počítače:

```
ssh -D lokální_ip_adresa:port
```

Michal Račák


Connection Settings
Nastavení prohlížeče
Configure Proxy Access to the Internet
No proxy
Auto-detect proxy settings for this network
Use system proxy settings
Manual proxy configuration
HTTP Proxy
Port
0
Also use this proxy for HTTPS
HTTPS Proxy
Port
0
SOCKS Host
127.0.0.1
Port
55555
SOCKS v4
SOCKS v5
Provoz přes SOCKS proxy
Automatic proxy configuration URL
Reload
No proxy for
Example: .mozilla.org, .net.nz, 192.168.1.0/24
Connections to localhost, 127.0.0.1/8, and ::1 are never proxied.
Do not prompt for authentication if password is saved
Proxy DNS when using SOCKS v5
Cancel
OK

sshfs

- `sshfs` poskytuje možnost připojit vzdálený souborový systém jako `FUSE` (Filesystems in Userspace) – Linux/Unix

  ```
  sshfs uživatel@server:/adresář/k/připojení kam
  ```

- implementace využívá protokol SFTP

Michal Bořek

SFTP

Secure
File
Transfer
Protocol

Michał Bořátka

SSH FTP protokol (SFTP)

- rozšíření protkolu SSH verze 2
- nástupce **S**ecure **C**opy **P**rotokolu (SCP)
- pracuje nad bezpečnou transportní vrstvou (SSH)
  - a v rámci ní pak dochází k přesunu dat nebo jiným operacím
- oproti SCP, který podporuje jen přenosy souborů, SFTP podporuje další operace nad vzdálenými soubory (navázání po přerušení přenosu souboru, zjištění obsahu adresáře, mazání souborů, atd.)
  - podporuje všechny Unixové typy souborů včetně symlinků
- **TCP** port **22** (jako SSH)

Michal Račák

Wi-Fi

Bezdrátové sítě Wi-Fi

Michal Pořízek

Wi-Fi: Bezdrátové sítě

- **`WLAN`**: **W**ireless **LAN**, založeno na `IEEE 802.11`
- rádiové vlny
  - ultrakrátké vlny UKV (0,3 – 3 GHz)
  - superkrátké vlny SKV (3 – 30 GHz)
  - extrémně krátké vlny EKV (30 – 300 GHz)
- `Wi-Fi` je ochranná známka Wi-Fi Alliance
- `Wi-Fi Certified` lze použít na produkty, které projdou testováním a získají certifikaci

Michal Račák

Wi-Fi pásma

- WiFi využívá tato pásma:
  - 2,4 GHz
  - 5 GHz
  - 6 GHz
  - 60 Ghz
- bezlicenční pásma: homologované zařízení lze používat bez poplatků
  - všeobecné oprávnění **VO-R/12**
- https://spektrum.ctu.cz/
- https://rlan.ctu.cz/cs/

Michal Bečák

Wi-Fi

standardy Wi-Fi
Wi-Fi pásma
sdílené médium

Michal Rešetka

Wi-Fi standardy

- technologie Wi-Fi se vyvíjí v jednotlivých iteracích, které kulminují vydáním nového standardu
- zařízení Wi-Fi pak podporují zpravidla více standardů, a tak jsou schopny se domluvit i se staršími zařízeními
- starší standardy nemusí být v moderních sítích podporovány (Eduroam)

Michal Rešetka

Standard	Označení	Rok vydání	Pásmo [GHz]	Max. rychlost [Mbit/s]
IEEE 802.11	Wi-Fi 0	1997	2,4	2
IEEE 802.11b	Wi-Fi 1	1999	2,4	11
IEEE 802.11a	Wi-Fi 2	1999	5	54
IEEE 802.11g	Wi-Fi 3	2003	2,4	54
IEEE 802.11n	Wi-Fi 4	2009	2,4/5	600
IEEE 802.11y	-	2008	3,7	54
IEEE 802.11ac	Wi-Fi 5	2013	5	6928
IEEE 802.11ad	-	2012	60	6757
IEEE 802.11ax	Wi-Fi 6	2019	2,4/5	600–9608
	Wi-Fi 6E	2020	2,4/5/6	
IEEE 802.11be	Wi-Fi 7	N/A		40000

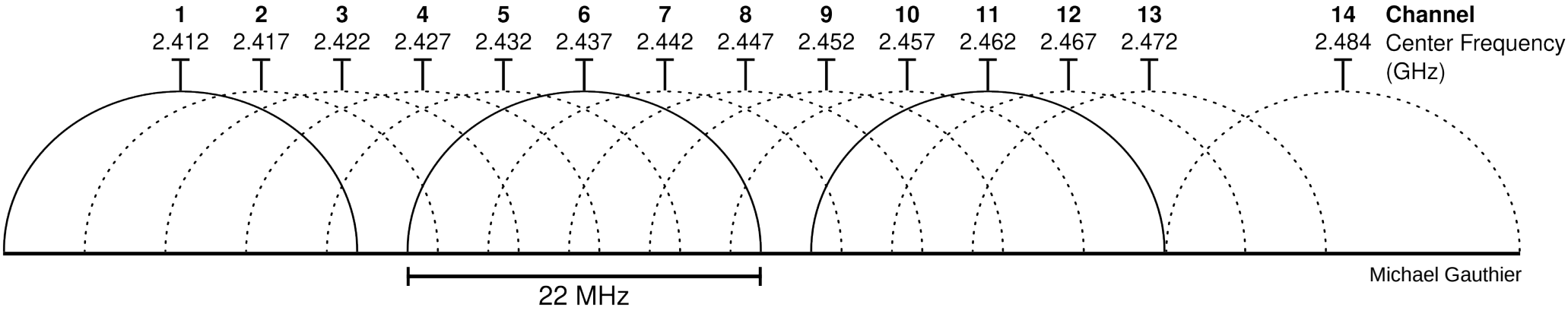
Rádiové vlny

- ve spektru elektromagnetického záření
- čím je kratší vlnová délka, tím více je omezen dosah, a tím více signálu pohltí překážky
- Wi-Fi používá krátkou vlnovou délku, přičemž 5Ghz Wi-Fi má samozřejmě kratší vlnovou délku než 2,4GHz Wi-Fi
- dosah Wi-Fi je tedy až 20 m v obytných prostorech se stěnami (překážkami) nebo až 150 metrů v případě přímé viditelnosti mezi AP a klientem

Michal Pačes

2,4Ghz Wi-Fi pásmo

- komunikační **pásmo** je rozdělené do *kanálů*:



- kanály se částečně překrývají
- na jednom kanálu může vysílat jen jeden vysílač
- šířka pásma – vyšší šířka → větší průtok
  - pro šířku 20 Mhz lze použít kanály 1,6,11 (72Mb/s)
  - pro šířku 40 Mhz lze použít kanály 1 a 9 (150Mb/s)

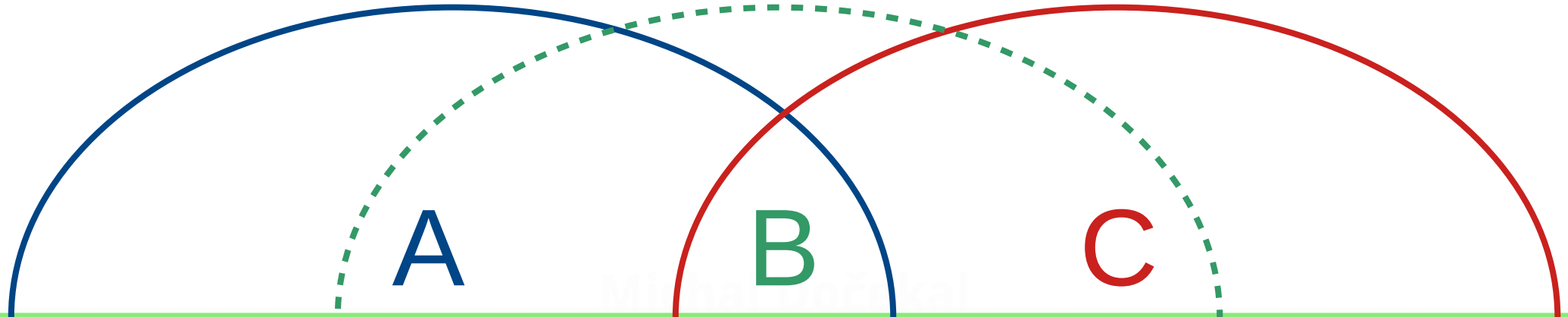
5Ghz Wi-Fi pásmo

- v EU rozsah 5Ghz Wi-Fi: 5180 – 5700 MHz
- 19 kanálů o šířce pásma 20MHz
  - kanály se nepřekrývají
- 9 kanálů o šířce pásma 40 MHz (802.11n)
- 2 kanály s rychlostí 1300 Mb/s (802.11ac)
  - v pásmu mohou být pouze dva routery vedle sebe

Michal Rešetka

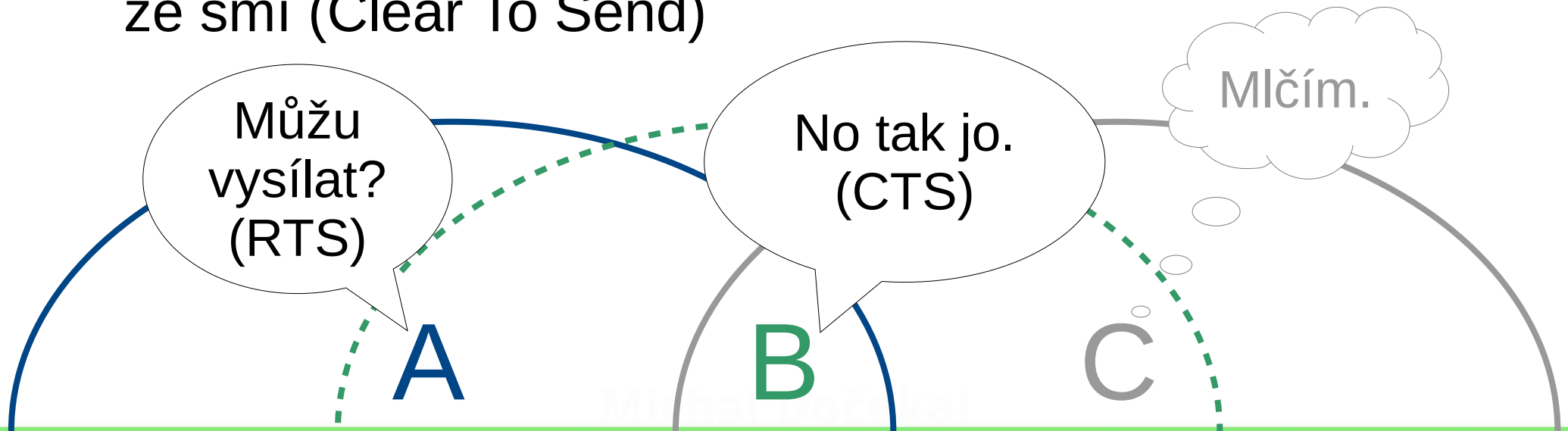
Řešení kolizí sdíleného média (opakování pro připomenutí)

- Wi-Fi využívá pro přenos dat **sdílené médium**
- CSMA/CA: Carrier Sense Multiple Acces with Collision Avoidance (opakování)
  - **Carrier sense**: před vysíláním posloucháme, jestli už někdo jiný nevysílá
    - pokud ne, nemusí to ovšem znamenat, že někdo jiný opravdu nevysílá (viz ***problém skrytého uzlu:***)



Řešení kolizí sdíleného média (opakování pro připomenutí)

- **Collision avoidance**: když někdo vysílal, čekáme náhodně zvolenou dobu až dotyčný vysílat přestane, a teprve pak znovu posloucháme, jestli někdo nevysílá
- **IEEE 802.11 RTS/CTS** – částečné řešení problému skrytého uzlu – odesílatel se nejdřív ptá, jestli může vysílat (Request To Send), a vysílá, až dostane signál, že smí (Clear To Send)



Wi-Fi

Principy fungování

Michal Dočekal

Dva základní režimy Wi-Fi

- WLAN může pracovat ve dvou režimech:
  - **infrastrukturní**: mobilní zařízení komunikují prostřednictvím přístupového bodu Wi-Fi (Access Point), který pak může fungovat jako brána do LAN a internetu
    - nejčastější režim funkce WLAN
  - **ad-hoc**: mobilní zařízení fungují v režimu peer-to-peer
    - každý uzel se účastní směrování přeposíláním dat pro ostatní uzly

Michal Račák

Klient a přístupový bod

- infrastrukturní režim používá typy zařízení:
  - Wi-Fi **A**ccess **P**oint (**AP**) – přístupový bod Wi-Fi
  - Wi-Fi **klient**, který se připojuje k **AP**
- AP i klient mají své Wi-Fi NIC, které mají MAC adresu
- AP má definováno svoje **SSID**
  - **S**ervice **S**et **Id**entifier
  - identifikátor sítě Wi-Fi
  - až 32 ASCII znaků
  - pravidelně vysílán jako broadcast
  - skryté sítě SSID nevysílají

Wireless LAN

- WLAN implementuje linkovou vrstvu a přenáší rámce
  - různé typy řídících (control) rámců (ACK, RTS, CTS, atd.) a správních (management) rámců (Authentication, Association, Beacon, Probe, atd.)
- klient se asociuje s AP, a teprve po úspěšné asociaci může klient s AP komunikovat (posílat data)
- roaming: jedna síť může používat mnoho AP, při pohybu klienta dojde k přepojení na jiné AP
- k propojení bezdrátové LAN s drátovou LAN se používá síťový most (bridge)

Michal Račátal

Soukromí a bezpečnost

- kdokoliv, kdo je v dosahu vaší Wi-Fi sítě
  - může odposlouchávat vaši Wi-Fi komunikaci
  - může prolamovat bezpečnost vaší Wi-Fi sítě a eventuelně proniknout dovnitř
- pokud útočník prolomí zabezpečení, může páchat odposlech i jiné MITM útoky
- Wi-Fi sítě je tedy třeba adekvátně zabezpečit!
- i při komunikaci se zabezpečenou sítí může docházet k vyzrazení určitých informací (vaše MAC adresa)

Michal Rešetka

Bezpečnostní mechanismy Wi-Fi

- bez zabezpečení (otevřená síť)
- MAC filtr
- WEP
- WPA 1
- WPA 2
- WPA 3

Michal Rešetka

Bezpečnostní mechanismy Wi-Fi

- × bez zabezpečení (otevřená síť) – nepoužívat!!!
- × MAC filtr - MAC lze odposlechnout a změnit
- × WEP - prolomení v reálném čase
- × WPA 1 - mnoho zranitelností
- • WPA 2 - zranitelnosti, jen s velmi silným heslem
- ✓ WPA 3 - zatím OK, stejně raději silné heslo

Michal Rešetka

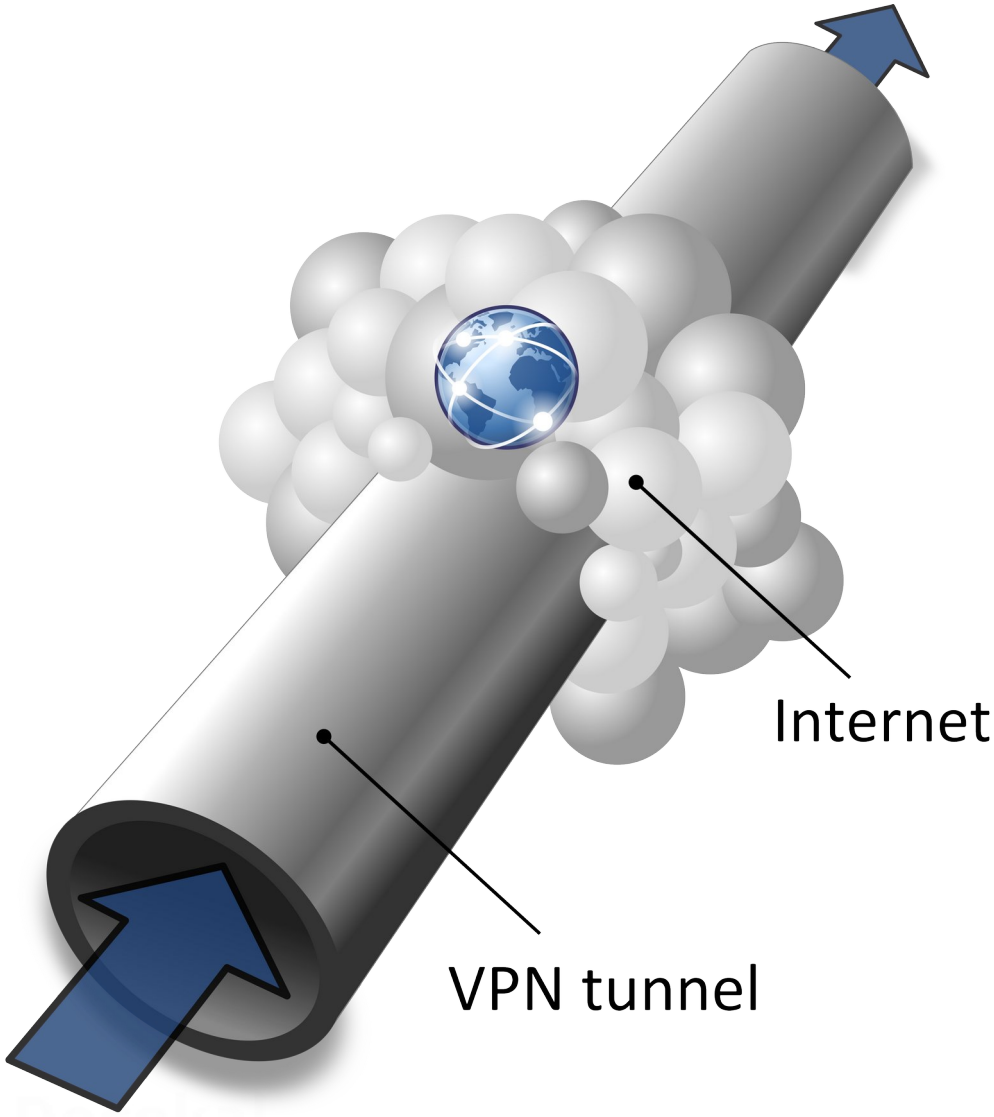
Bezpečnost Wi-Fi

- vlastní Wi-Fi síť důkladně zabezpečte
  - WPA3 + silné heslo
  - přinejhorším WPA2 + supersilné heslo
- nepřipojujte se k sítím, u kterých nevíte, kdo je provozuje, nebo provozovateli nedůvěřujete
- nepřipojujte se k otevřeným (nezabezpečeným) sítím, ani když provozovatele znáte
- používejte pouze šifrované a zároveň autentikované služby (např. HTTPS)

Michal Poček

VPN

Virtual
Private
Network



Tunelování

- tunelování a tunelovací protokoly
  - jedno/více spojení se přenese přes jiné spojení
  - principem je zapouzdření jednoho protokolu do jiného (např. ethernetový rámec vložený do datové části UDP paketu)
  - technika rozbíjí OSI model
  - široké využití (legitimní i šedá zóna), např. IPv4 přes IPv6, ale i IP přes HTTP, DNS, ICMP, atd.
  - v případě VPN se tuneluje přes šifrované spojení

Virtual Private Network

- do *většinou* šifrovaného spojení (typicky UDP) se balí rámce (L2 VPN) nebo IP datagramy (L3 VPN)
- nad *tunelem* je postavená celá nová (virtuální) síť
- široké užití
  - bezpečné připojení zařízení do firemní sítě z domova (i VŠE má svou VPN) a získání přístupu k systémům, které nejsou z internetu dostupné
  - tunelování celého síťového provozu stanice (nebo celé sítě) přes VPN server (který vystupuje jako proxy)
  - obcházení cenzury, regionálních omezení, místní data retention, jiných omezení (např. filtrovací proxy servery firemních sítí, apod.)

Michal Rešetka

Virtual Private Network

- mnoho implementací: IPsec, OpenVPN, WireGuard, SSH, atd.
- **OpenVPN**
  - FOSS s proprietární firemní verzí
  - `UDP` port `1194`, lze použít i `TCP`
  - L3 i L2 tunel
  - šifrování a autentizace pomocí TLS včetně klientských i serverových certifikátů
  - Solaris, Linux, OpenBSD, FreeBSD, NetBSD, QNX, macOS, Windows, Android, iOS
  - integrováno do firmwaru mnoha routerů (vč. např. OpenWrt)

Michal Pořízek

NAT

Network
Address
Translation

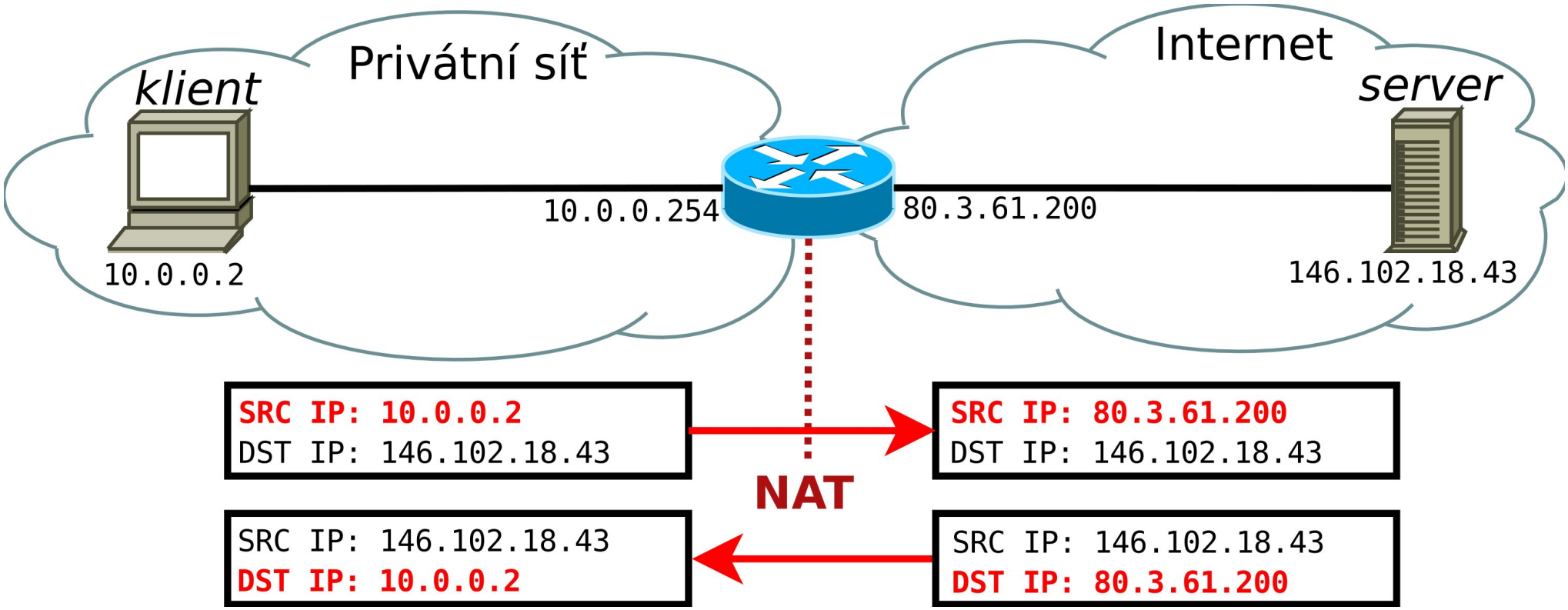
Michal Rešetka

Network Address Translation

- řešení problému s neroutovatelností privátních rozsahů
- **překlad adres**: NATovací router mění zdrojové či cílové IP adresy v hlavičkách IP datagramů, které routerem prochází
  - součástí může být i změna zdrojových či cílových portů v TCP hlavičkách
- existují různé varianty NATu pro různá použití

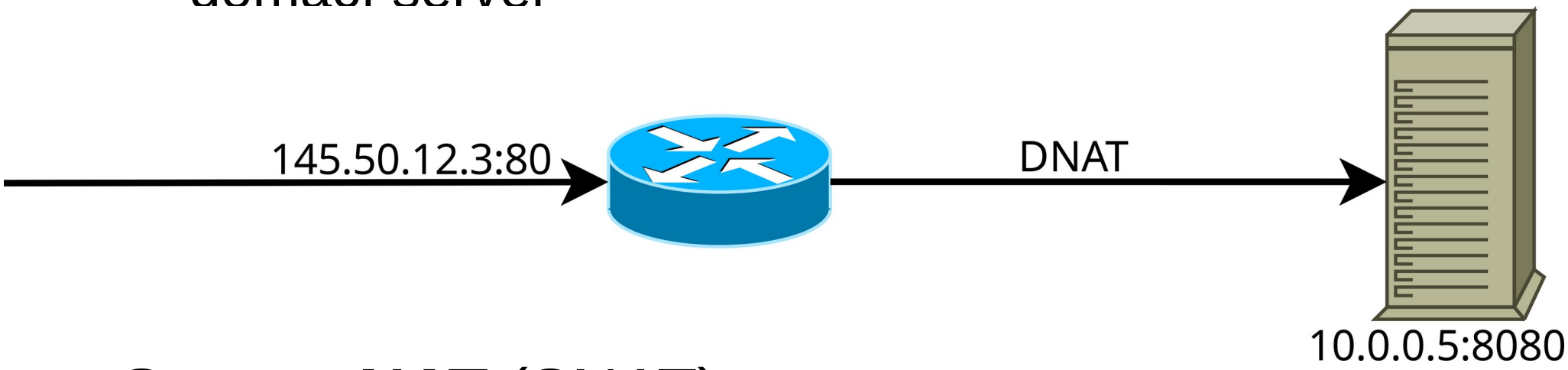
Michal Řeřábek

Princip funkce NATu



Různé typy NATů

- **Destination NAT** (DNAT)
  - změna pouze **cílové IP adresy** (popř. i portu)
  - využití: port forwarding z domácího routeru na domácí server

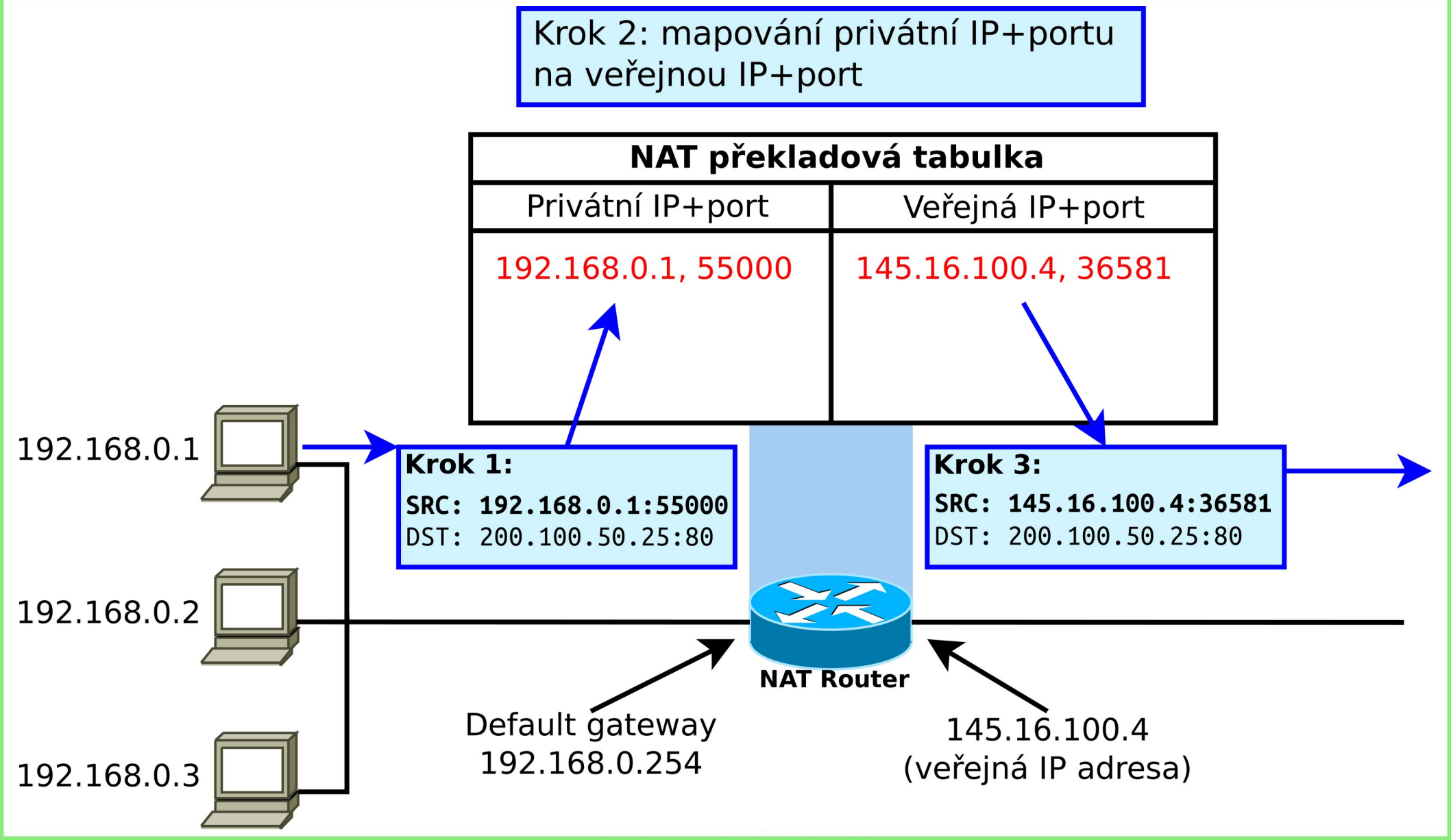


- **Source NAT** (SNAT)
  - analogie k DNAT, změna pouze **zdrojové IP adresy**

Různé typy NATů

- **Statický NAT (NAT 1:1)**
  - jedna privátní IP adresa je mapována na jednu veřejnou
  - poskytování veřejných IP adres když ISP používá CGNAT
- **Dynamický NAT**
  - pro mapování adres se používá rozsah veřejných adres, které se přidělují dynamicky klientům s privátními IP adresami
  - mapování se vytváří, až když klient navazuje spojení s vnějším uzlem
- **Network Address and Port Translation** (NAPT, PAT)
  - *IP maškaráda*, použití **pro domácí sítě** – podpora více zařízení za NATem (NAT 1:n)
  - někdy bývá označována jako *NAT overload*
- **Carrier Grade NAT** (CGNAT)
  - velkorozsahové řešení NATu pro ISP

Network and Port Translation



NAT: shrnutí

- do přechodu na IPv6 nás bude pronásledovat
- narušuje end-to-end princip
- komunikace zevnitř NATu ven *většinou* funguje
- komunikace zvenčí do NATu nikoliv
- odstavuje některé protokoly (SIP – VoIP)
- pro komunikaci zvenčí do NATu je potřeba komunikačních mezičlánků → hrozba MITM nebo narušení soukromí

Michal Rešetka

NAT: ~~bezpečnost~~

- NAT poskytuje mechanismus, kterým je počítač jakoby „skryt" před internetem (maškaráda)
- NAT však **není** zabezpečovací mechanismus!
  - NAT se dá obejít: NAT traversal, hole punching, prolomení zabezpečení ISP, prolomení zabezpečení domácí sítě (kompromitace routeru či jiného zařízení, malware, atd.), sociální inženýrství
- síťová bezpečnost využívá jiné mechanismy: **firewally**, IDS/IPS, DMZ, autentizace, šifrování, atd.

Michal Račák

Bezpečnost

Security by obscurity

Secure by desgin

Michal Ďuračka

Security by obscurity

- bezpečnost **skrze neznalost** – strategie, která považuje systém za bezpečný, pokud útočník neví, jak funguje, nebo jak je sestaven
  - **utajení** protokolů, postupů, algoritmů, nastavení, atd.
- mnohdy však stačí jeden zkoumavý útočník
  - SSH na nestandardním portu → portscan → útok
  - substituční šifry → frekvenční analýza → prolomení
- jedná se o špatný přístup k bezpečnosti
- opačný a lepší přístup: secure by design

Michal Račák

Secure by design

- systém je s ohledem na bezpečnost navržen
  - tj. je navržen tak, aby znalost principů, postupů, algoritmů či nastavení **neohrozila** bezpečnost
  - např. šifrovací algoritmy jsou **veřejně známé**, utajený je jen šifrovací klíč
    - co byste si vybrali? utajený šifrovací algoritmus, který vyvinul a zná 1 člověk, versus veřejný šifrovací algoritmus, který neúspěšně zkusilo prolomit nespočet expertů
  - FOSS: více očí více vidí, ale některé oči jsou zlé
- princip nejnižších privilegií, sandboxing, ošetření uživatelských vstupů, silná hesla, penetrační testování, atd.

Michal Račinský

Bezpečnost

Nástroje síťové bezpečnosti *(některé)*

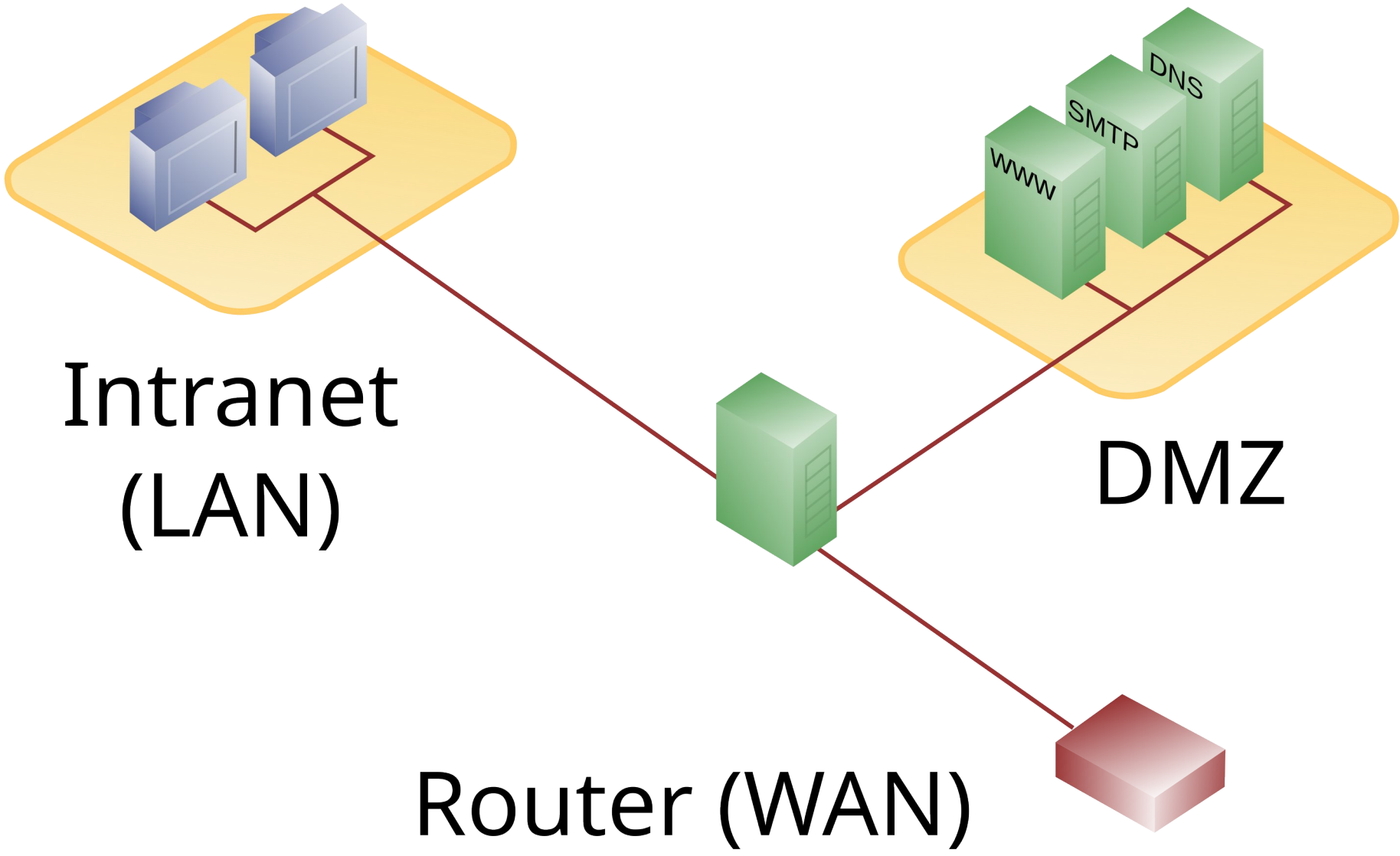
Michal Rešetka

Nástroje síťové bezpečnosti

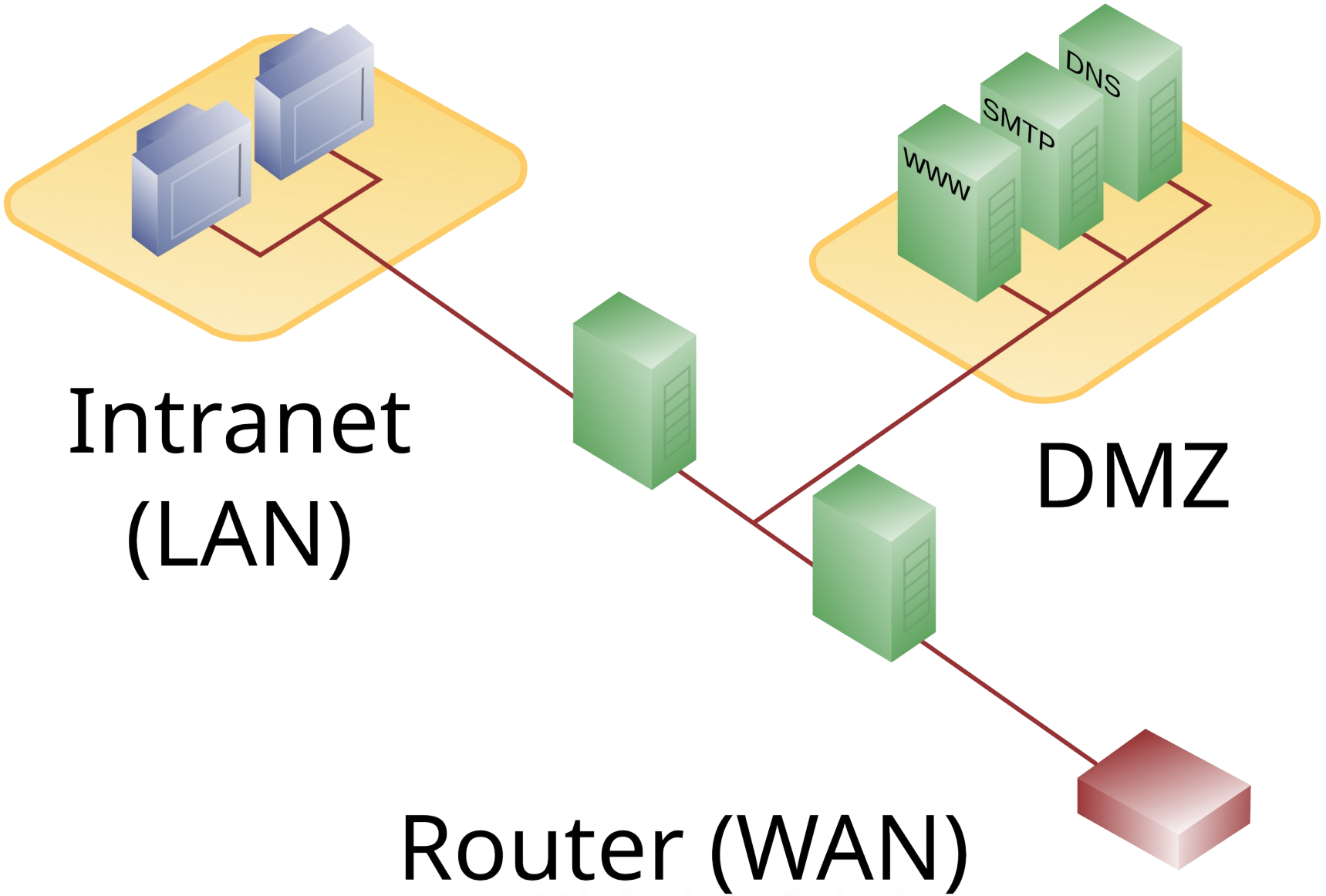
- **firewall** – filtruje pakety dle definovaných pravidel
- **IDS/IPS** – intrusion detection system / intrusion prevention system – nástroj monitorující síťový provoz a detekující podezřelou a škodlivou aktivitu s možností škodlivou aktivitu blokovat a jinak potírat
- **DMZ** – demilitarizovaná zóna – oddělení sítě obsahující veřejně dostupné služby od LAN

Michal Račinský

DMZ s jedním firewallem



DMZ se dvěma firewally



Nástroje síťové bezpečnosti

- **autentizace** – ověřování identity protistrany
  - elektronický podpis, certifikáty, certifikační autority, PKI: infrastruktura veřejných klíčů
- **šifrování** – zabezpečení dat jejich převodem z čitelné podoby do podoby, která je čitelná pouze se speciální znalostí (v podobě dešifrovacího klíče)
- atd.

Michal Ročkai

<EOF>