

Úvod do počítačových sítí (NSWI141)

Libor Forst, SISAL MFF UK

- Základní pojmy z oblasti komunikací
- Vrstevnatý model sítě (OSI vs. TCP/IP, adresace, multiplexing, ...)
- Aplikační vrstva (DNS, FTP, email, web, VoIP, ...)
- Transportní vrstva
- Síťová vrstva (IPv4, IPv6, směrování, firewallly, ...)
- Linková a fyzická vrstva (switch vs. repeater, Ethernet, Wi-Fi, kabeláž, ...)

Literatura

- D. E. Comer, D. L. Stevens: Internetworking With TCP/IP; Prentice Hall 1991
- A. S. Tanenbaum: Computer Networks; Prentice Hall 2003
- C. Hunt: TCP/IP Network Administration; O'Reilly & Associates 1992
- P. Satrapa, J. A. Randus: LINUX - Internet server; Neokortex 1996; ISBN 80-902230-0-1
- L. Dostálek, A. Kabelová: Velký průvodce protokoly TCP/IP a systémem DNS; Computer Press 2002
- zdroje na internetu
- Request For Comment (RFC)
- <http://www.warriorsofthe.net>

Obecné atributy komunikace

- Identifikace
 - komunikující strany se musí „najít“ (telefonní čísla), představit
- Metoda
 - př.: hluchoněmý u přepážky, zkusí znakovou řeč, recepční napíše na papír, že nerozumí a navrhne psanou formu komunikace
- Jazyk
 - obě strany se musí dohodnout na jazyku, který použijí
- Rychlost
 - obě strany se musí dohodnout na rychlosti komunikace
- Proces
 - požadavky, odpovědi, potvrzení

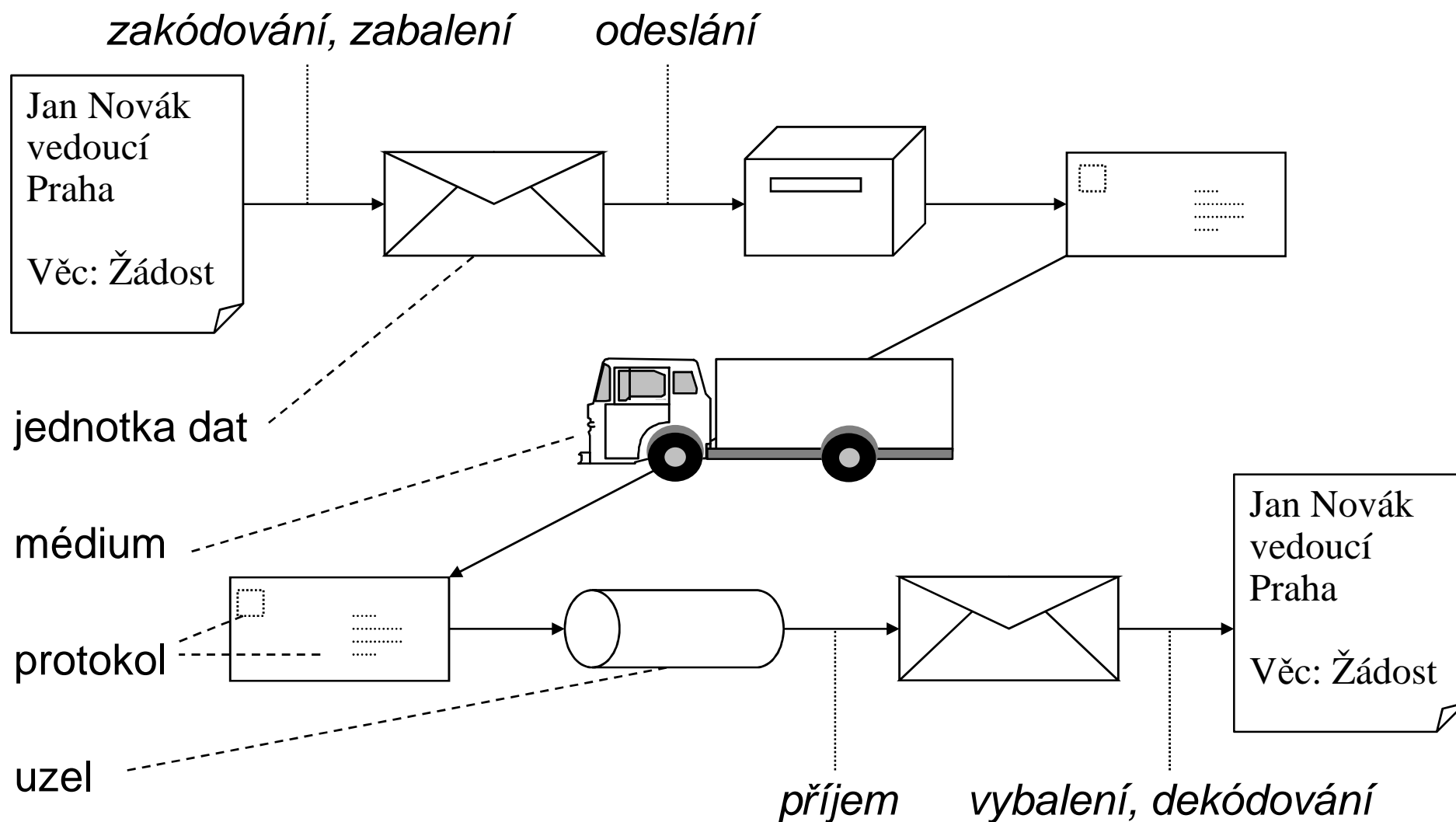
Porovnání komunikací

- Běžná komunikace
 - hlas, signály, písmo
 - volná intuitivní pravidla
- Telekomunikace
 - složitá technologie se zabudovanými pravidly
 - řízení má na starosti síť, řídí i koncová zařízení
- Počítačová síť
 - pravidla jsou volně dostupná
 - značná část logiky je v koncových zařízeních
 - síť se stará jen o přenos
- Konvergovaná síť
 - spojuje svět spojů a počítačů (cena, efektivita...)
 - úspěšnější je konvergence na bázi počítačové sítě

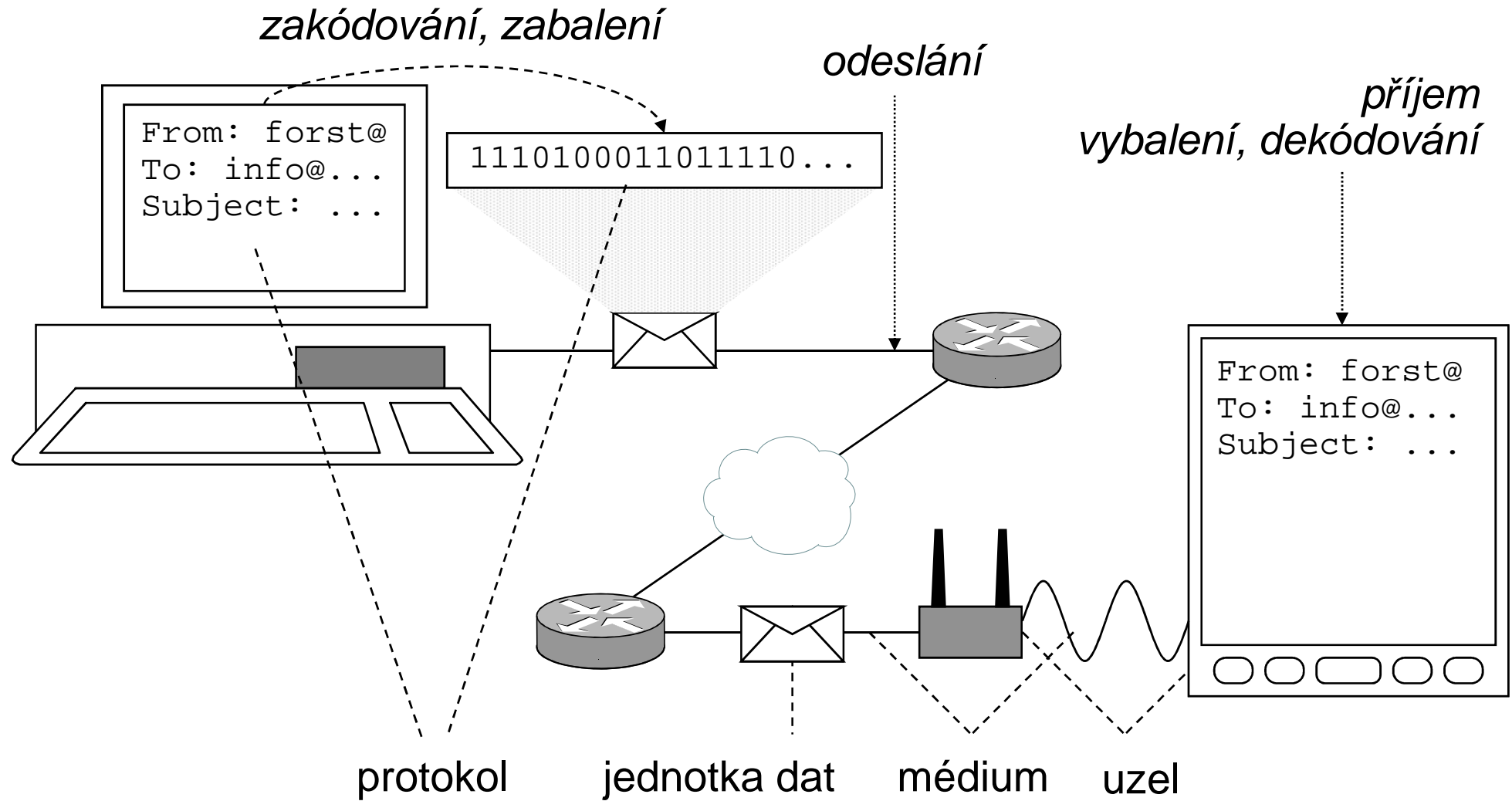
Prvky síťové komunikace

- Protokoly (pravidla)
 - normy
 - standardy
 - doporučení
- Média
 - drát
 - optika
 - „vzduch“
- Jednotky dat
 - zpráva
 - paket
 - bit
- Uzly
 - koncová zařízení
 - síťová zařízení

Přenos zprávy (pošta)

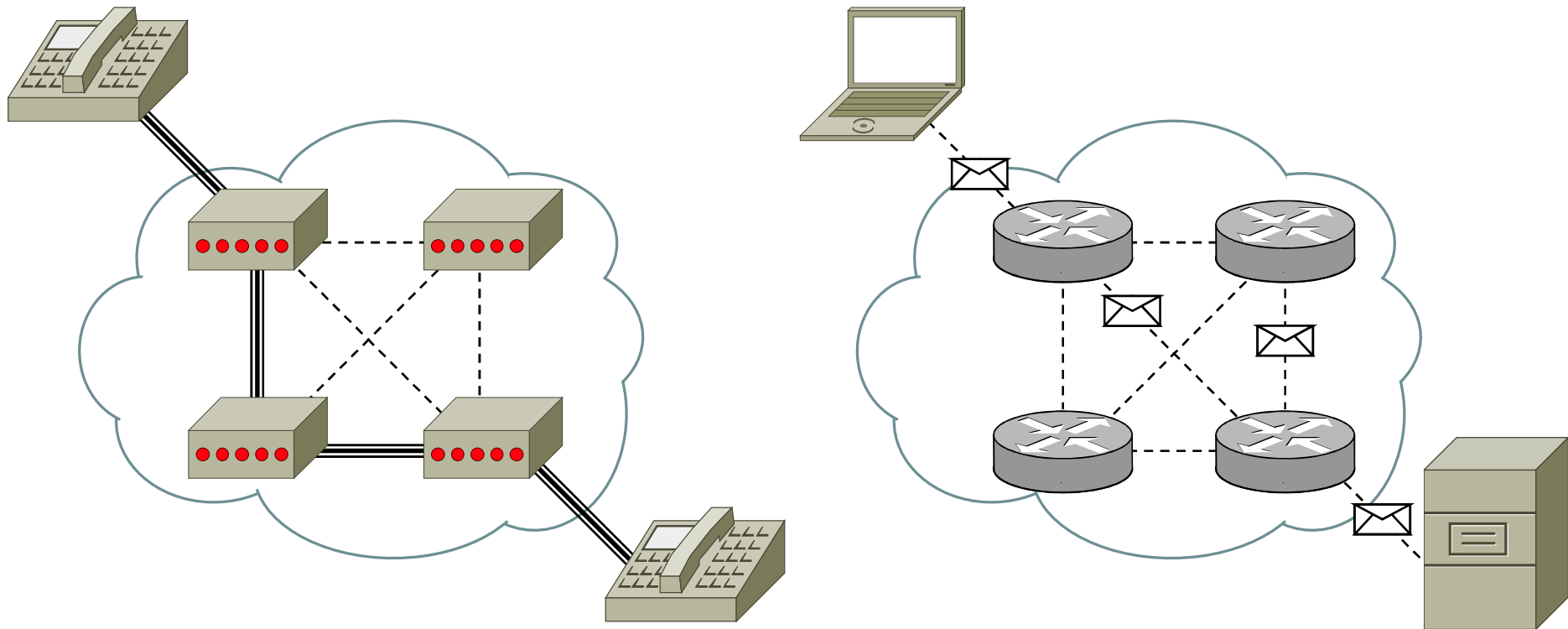


Přenos zprávy (e-mail)



Požadavky - odolnost

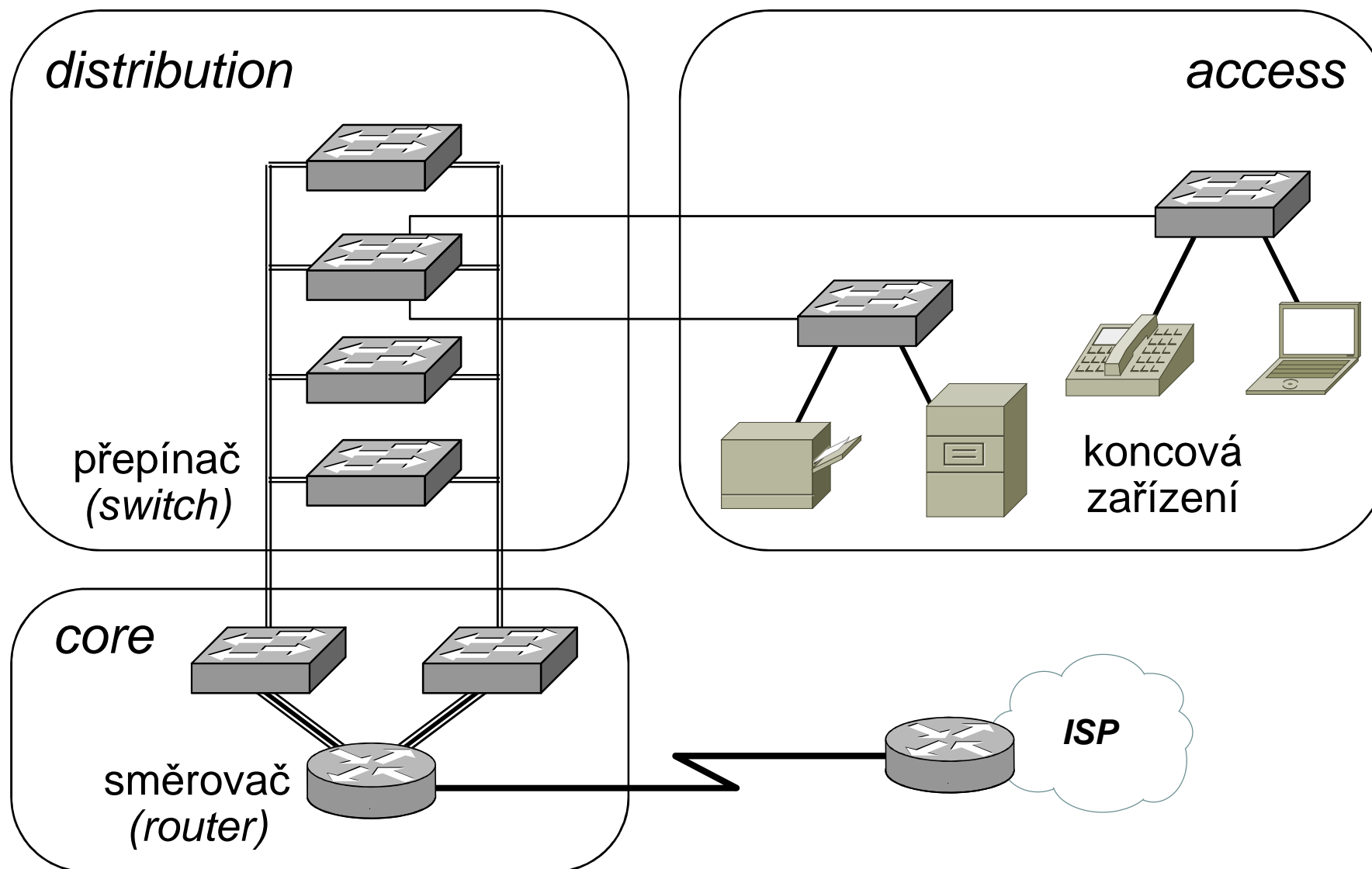
- přepojování okruhů: rychlejší, plynulejší, ale při výpadku uzlu se spojení rozbíjí
- přepojování paketů: každý může jít jinou cestou, liší se doba přenosu, ale výpadek uzlu není fatální



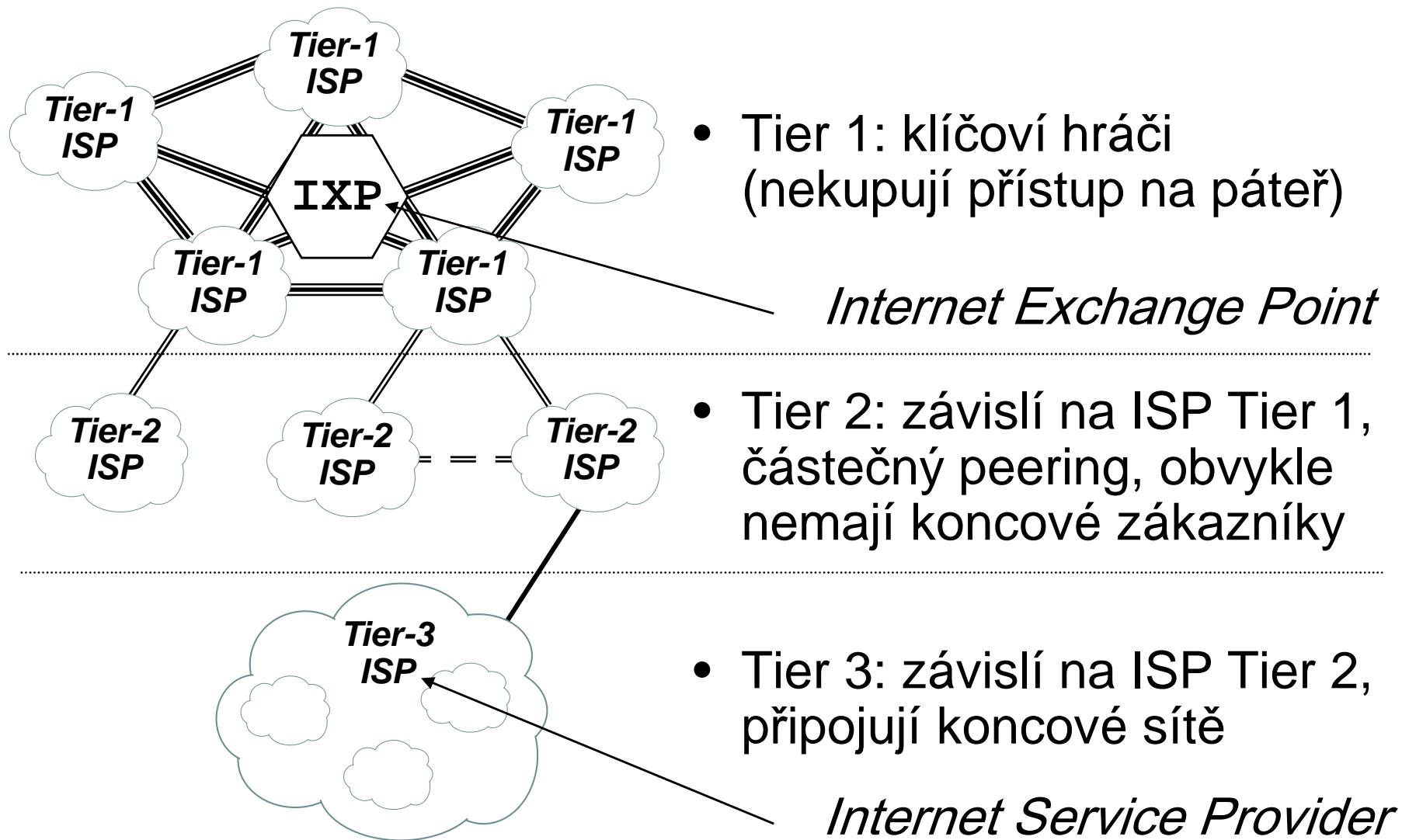
Požadavky - bezpečnost

- Relativně nové kritérium, staré technologie byly naivní:
 - otevřená komunikace (odposlech)
 - důvěra v identitu protistrany
 - důvěra ve správnost obsahu
- Základní dělení:
 - bezpečnost infrastruktury
 - bezpečnost dat
- Současné metody:
 - ověřování uživatelů a kontrola přístupových práv
 - ověřování počítačů (serverů, příp. i klientů)
 - inspekce dat (aplikační proxy, antiviry, antispamy, ...)
 - kryptografie (šifrování a podpisy)

Požadavky - rozšiřitelnost (LAN)



Požadavky - rozšiřitelnost (WAN)



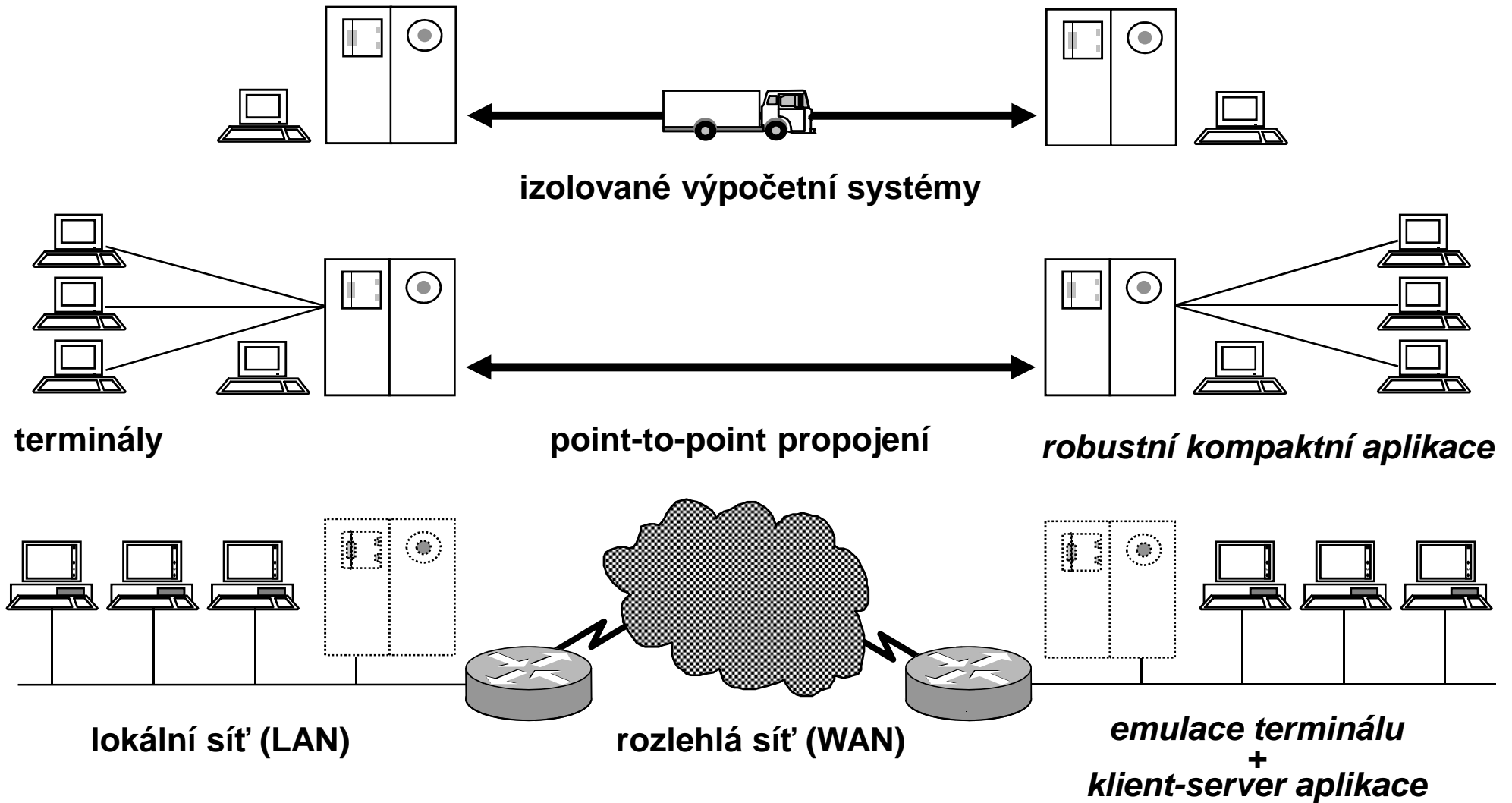
Požadavky - kvalita služeb

- Různé aplikace mají různé požadavky
 - zpoždění (*latence, delay*)
 - pravidelnost doručování (*jitter*, rozptyl zpoždění)
 - oba parametry kritické pro multimediální aplikace
 - ztrátovost dat
 - kritická pro přenos dat (WWW, pošta)
 - šířka pásma (*bandwidth*, „rychlost“)
- Cíl:
 - garance vymezeného toku pro konkrétní typ provozu
 - garance rychlejšího doručení prioritních zpráv

Quality of Service

- Externí faktory:
 - kvalita a zaplnění komunikačního kanálu
 - změny formy (hlas \Rightarrow text \Rightarrow obrázek)
 - přeposílání (změny adresy)
 - čas vymezený pro komunikaci
- Interní faktory:
 - velikost, složitost, důležitost zprávy
- Implementace:
 - data obsahují klasifikaci QoS
 - strategie *garance kvality*: vyhrazená šířka pásma
 - zaručená kvalita, plýtvání kapacitou
 - strategie *best effort*: prioritní fronty
 - efektivní využití média, není záruka kvality

Vznik počítačových sítí

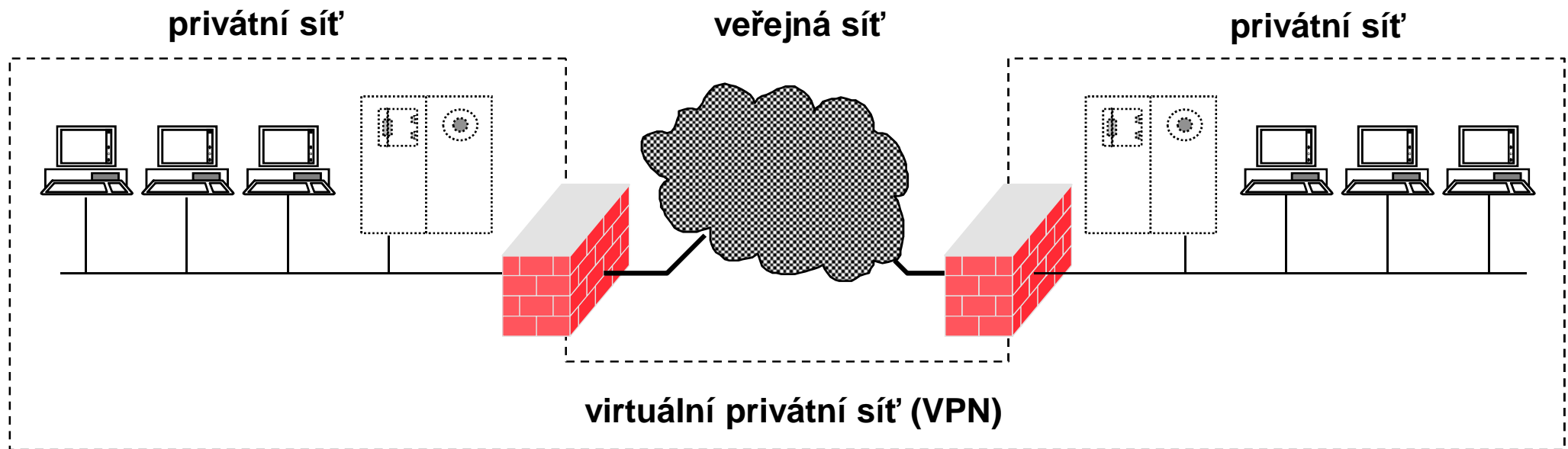


Základní dělení sítí

- Lokální síť (Local Area Network)
 - sdílení prostředků (souborové a databázové servery, tiskárny)
 - menší vzdálenosti (budova, kampus), malé zpoždění
 - jednotné vlastnictví a řízení
- Rozlehlé síť (Wide Area Network)
 - vzdálený přístup, komunikace
 - velké vzdálenosti, větší zpoždění
 - mnoho vlastníků, distribuované řízení
- Dnes:
 - rozdíly se stírají (nejmarkantnější jsou ve vlastnictví)
 - vznikají mezistupně (MAN)
- Není to dělení technické (neexistuje definice), ale logické

Veřejné a privátní sítě

- Většina LAN je privátních (uživatel je vlastníkem)
- Většina non-LAN je veřejných (uživatel není vlastníkem)

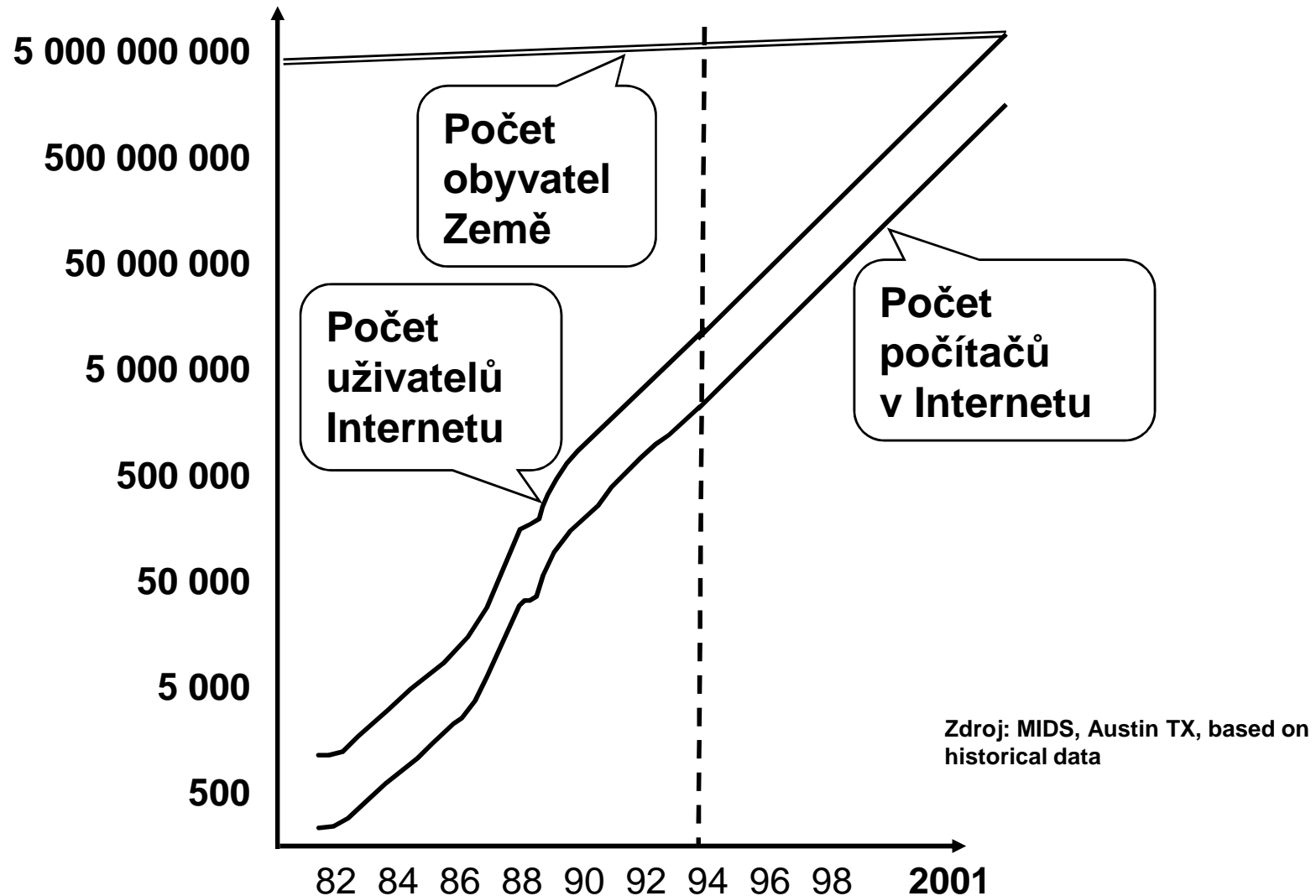


- Motivace VPN: bezpečnost, cena
- Typické použití VPN: propojení poboček, připojení (mobilních) koncových uživatelů

Historie Internetu

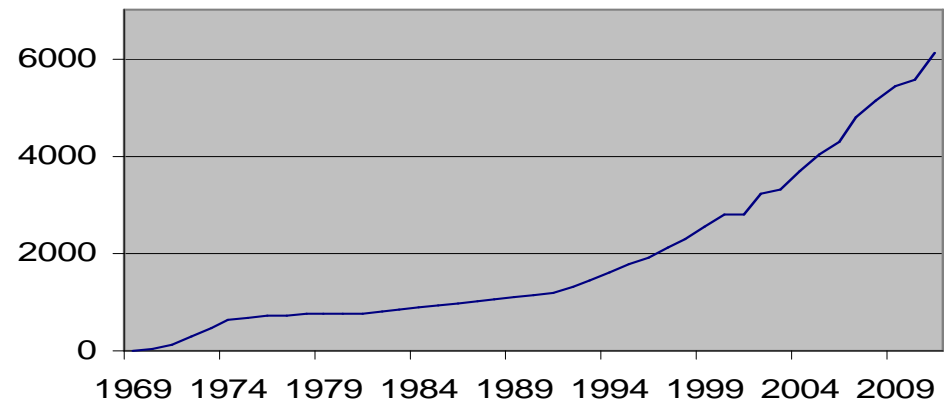
- zač. 60. let - koncepce „packet switching“
- 60.léta - US DoD podporuje koncept „packet switching“ pro odolnost proti fyzickému útoku
- 1969 - ARPANET - financuje Defense Advanced Research Project Agency, provozují akademická pracoviště, point-to-point, pevné linky
- 1974 - termín „Internet“ (zkratka „internetworking“) použit v RFC 675 definujícím TCP
- 1977 - na ARPANET páteř se připojuje první síť
- 1983 - TCP/IP nahrazuje NCP v ARPANETu
- pol. 80. - TCP/IP součástí BSD UNIXu

Vývoj Internetu v číslech



Request for Comments (RFC)

- Prostředek „standardizace” Internetu
- RFC 1 zveřejněno 7.4.1969



- Jsou volně šiřitelné (<http://www.ietf.org/rfc.html>)
- Různý charakter: standardy, informace, návody
- Návrh textu se předkládá IAB \Rightarrow IETF, IRTF \Rightarrow WG
- Dokumenty se nemění, aktualizace mají nové číslo (SMTP: 772, 780, 788, 821, 2821, 5321)
- Aktuální stav lze najít v indexovém souboru
- Zdaleka ne všichni RFC dodržují

Vrstevnatá filozofie

- Př.: rozeslání zápisu z obchodní porady
 - vrstva Zapisovatel
 - vytvoří zápis z porady
 - pravidla: formát zápisu
 - požadavek na Sekretářku: poslat dopis [zápis;osoba]
 - vrstva Sekretářka
 - vyhledá adresu, doplní záhlaví, podpis ... vloží do obálky
 - pravidla: formát obchodního dopisu
 - požadavek na Podatelnu: odeslat poštou [dopis;adresa]
 - vrstva Podatelna
 - dopis ofrankuje a zařadí do balíku pro transport na poštu
 - pravidla: odesílání pošty
- Výhody:
 - snazší dekompozice a popis
 - snadná změna technologie (pošta/email, pošta/kurýr)

Síťový model, síťová architektura

- Síťový (referenční) model:
 - počet a struktura vrstev
 - rozdělení práce mezi vrstvy
 - př.: ISO/OSI
- Síťová architektura (protocol suite):
 - síťový model
 - komunikační technologie
 - služby a protokoly
 - př.: TCP/IP

OSI model

- Budovaný shora, megalomanský, nepraktický
- Vhodný jako teoretický model

Pořadí	Vrstva	Úkol
7	aplikační	komunikace mezi programy
6	prezentační	datové konverze pro aplikace
5	relační	řízení dialogu mezi koncovými uzly
4	transportní	end-to-end přenos datových celků
3	síťová	dosažení cílového počítače
2	linková	přenos dat mezi sousedy
1	fyzická	fyzický přenos (bitů) mezi uzly

X.400, X.500

- Implementace služeb na základě OSI se opírala o řadu podobně (shora) navržených standardů
 - X.400: Message Handling System (pošta), nějakou dobu byl základem Microsoft Exchange Serveru, př. adresy:
G=Libor; S=Forst;
O=Charles University;
OU=Faculty of Mathematics and Physics;
OU=SISAL;
C=cz
 - X.500: Directory Access Protocol (adresářové služby, telefonní seznam), *perlička: implicitní položkou osoby je oblíbený nápoj*
- Následovníci:
 - LDAP (Lightweight DAP), např. MS Active Directory
 - X.509 Public Key Infrastructure (identifikace vlastníků klíčů)

Rodina protokolů TCP/IP

- Vyrostly z potřeb praxe, od jednoduchých ke složitějším

OSI	Vrstva	Příklady protokolů			
7	aplikační	FTP, HTTP, SMTP	DNS, SIP	NFS	
6				XDR	
5				RPC	
4	transportní	TCP		UDP	
3	síťová	IP			ICMP
2	síťové rozhraní	Ethernet, FDDI, ATM, WiFi, SLIP, PPP, ...			ARP
1					

Spojované/nespojované služby

- Spojované (connection-oriented) služby
 - obdoba telefonního spojení
 - zaručeno spolehlivé (*reliable*) doručení dat
 - aplikace je jednodušší, ale nemůže řídit komunikaci
 - v TCP/IP se používá TCP
- Nespojované (connectionless) služby
 - obdoba poštovního spojení
 - není zaručeno pořadí ani doručení paketů, služba se označuje jako „nespolehlivá“ (*unreliable*)
 - kontrolu musí provádět aplikace, zato může řídit komunikaci
 - v TCP/IP se používá UDP (IP samo je také nespolehlivé)

Aplikační modely

- Model klient-server
 - klient zná pevnou adresu serveru
 - klient navazuje komunikaci, zadává požadavky
 - server obvykle obsluhuje více klientů
 - tok dat server \Rightarrow klient: download
 - tok dat klient \Rightarrow server: upload
 - př. DNS, WWW, SMTP
- Model peer-to-peer (P2P)
 - partneři neznají pevné adresy „zdroje dat“
 - nejsou vyhraněné role
 - každý je zároveň klientem i serverem (=šíří data!)
 - Napster, Gnutella, BitTorrent

Adresování počítačů

- **HW**
(linková vrstva)
- **fyzická, MAC adresa**
(např. ethernetová: 8 : 0 : 20 : ae : 6 : 1f)
 - dána výrobcem (dříve), nastavitelná (dnes)
 - nerespektuje topologii
- **SW**
(síťová vrstva)
- **IP adresa**
(např.: 194 . 50 . 16 . 71, ::1)
 - přidělována podle topologie sítě
 - určuje jednoznačně síť a v jejím rámci počítač
- **Lidé**
(aplikační vrstva)
- **doménová adresa**
(např.: whois.cuni.cz)
 - přidělována podle organizační struktury
 - snazší zapamatování

Adresování služeb

- Uniform Resource Identifier (URI, RFC 3986)
 - jednotný systém odkazů
 - jeden klient pro více služeb (FTP ve WWW)
 - historické členění: URL (umístění), URN (název)

URI = schéma: [/ /] autorita [cesta] [? dotaz] [# fragment]

autorita = [jméno [: heslo] @] adresa [: port]

př.: ***ftp: // sunsite.mff.cuni.cz / Net / RFC***
http: // 1.2.3.4:8080 / q?ID=123#Local
mailto:forst@cuni.cz
sip:221911111@voip.cz

Port, socket

- **Port**

... 16bitové číslo identifikující jeden konec spojení - aplikaci, proces, který má zpracovávat příchozí pakety

- destination-port musí klient znát, typicky je to některý z tzv. *well-known services*
- source-port navazovatele spojení přiděluje lokální systém (původně ≥ 1024)

- **Socket**

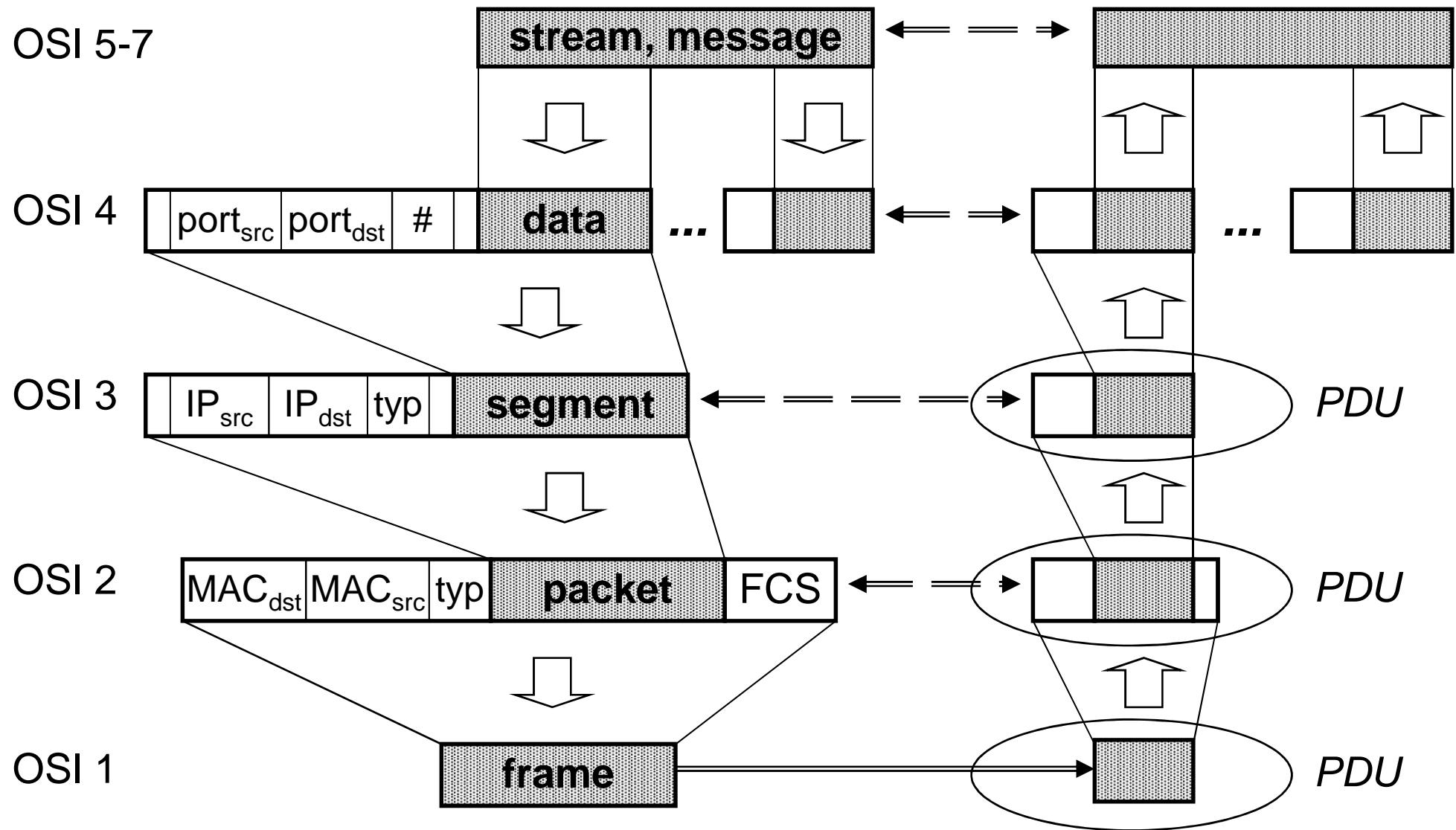
... jeden konec komunikačního kanálu mezi klientem a serverem

... označení (adresa) jednoho konce kanálu
<IPadresa, port>

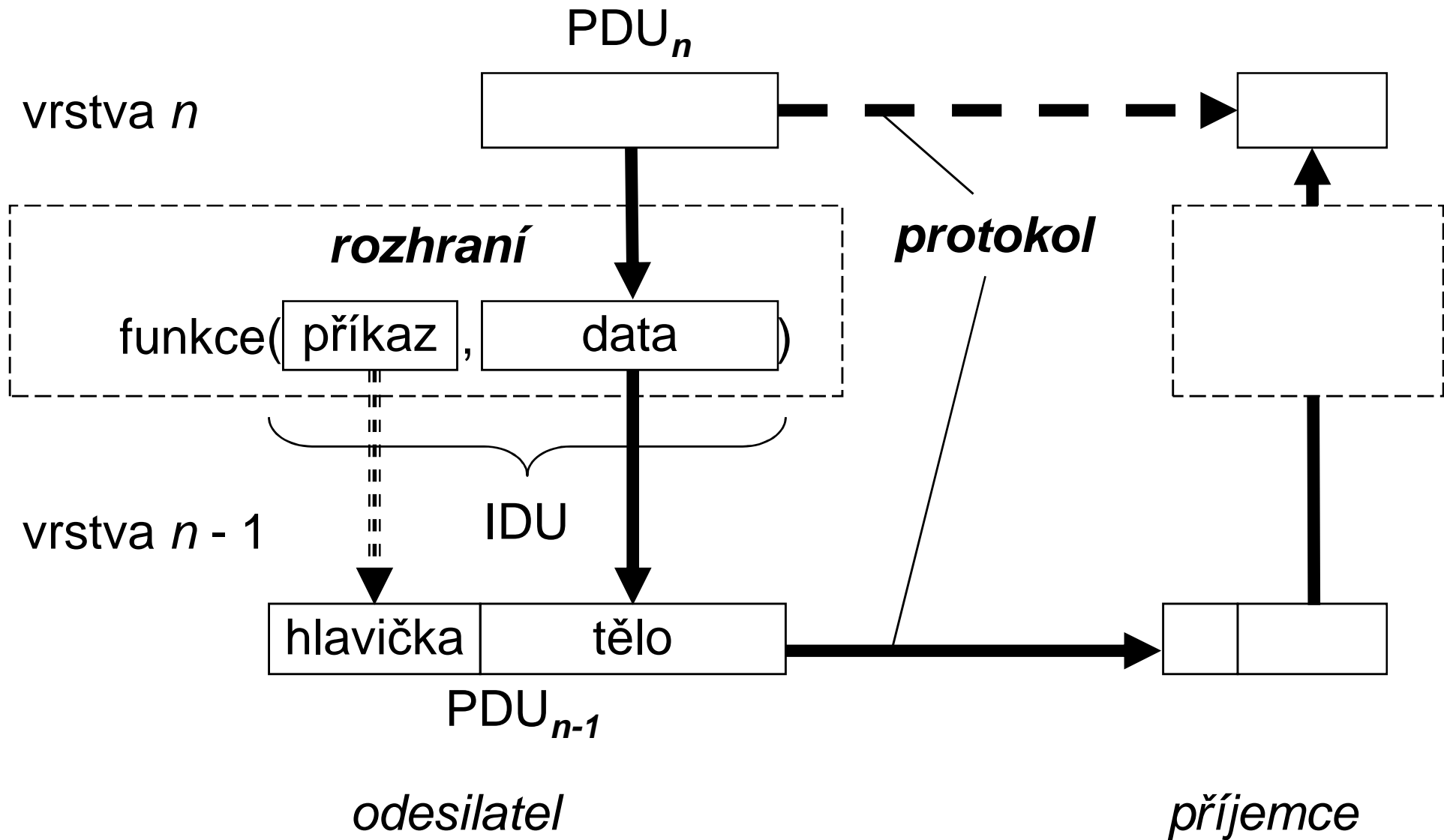
Příklady well-known services

- **21/TCP: FTP - File Transfer Protocol**
(přenos souborů)
- **22/TCP: SSH - Secure Shell**
(vzdálené přihlášení a přenos souborů)
- **23/TCP: telnet - Telecommunication network**
(vzdálené přihlášení)
- **25/TCP: SMTP - Simple Mail Transfer Protocol**
(přenos elektronické pošty)
- **53/*: DNS – Domain Name System**
(překlad jmen na IP adresy a naopak)
- **67,68/UDP: DHCP - Dynamic Host Configuration Protocol**
(vzdálená konfigurace)
- **80,443/TCP: HTTP - HyperText Transfer Protocol**
(přenos stránek informačního systému WWW)
- **5060,5061/*: SIP - Session Initiation Protocol**
(VoIP, IP telefonie)

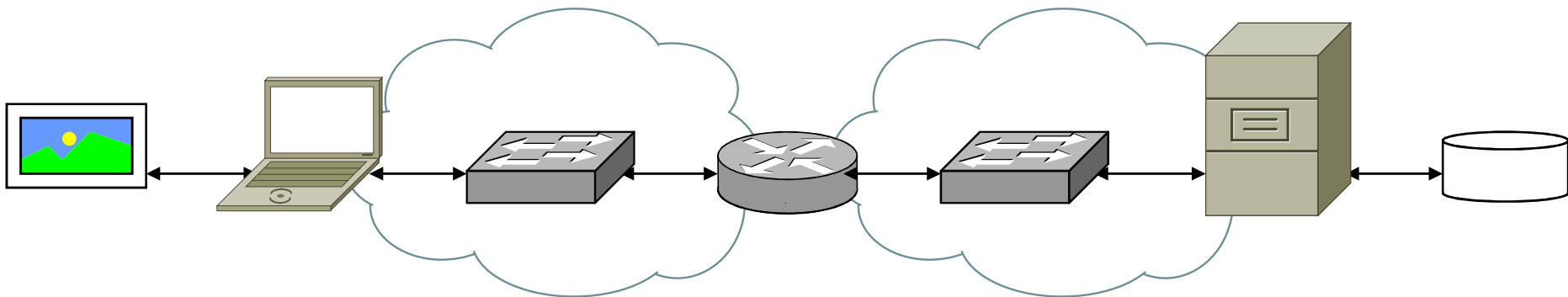
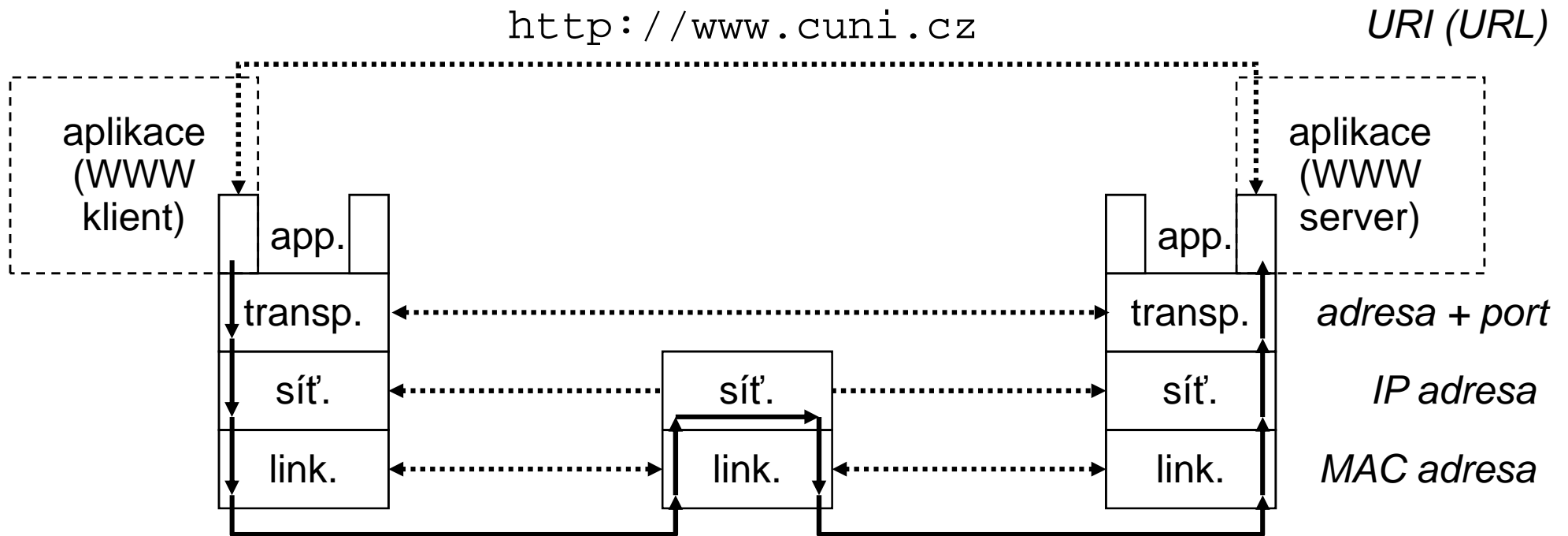
Multiplexing, zapouzdření



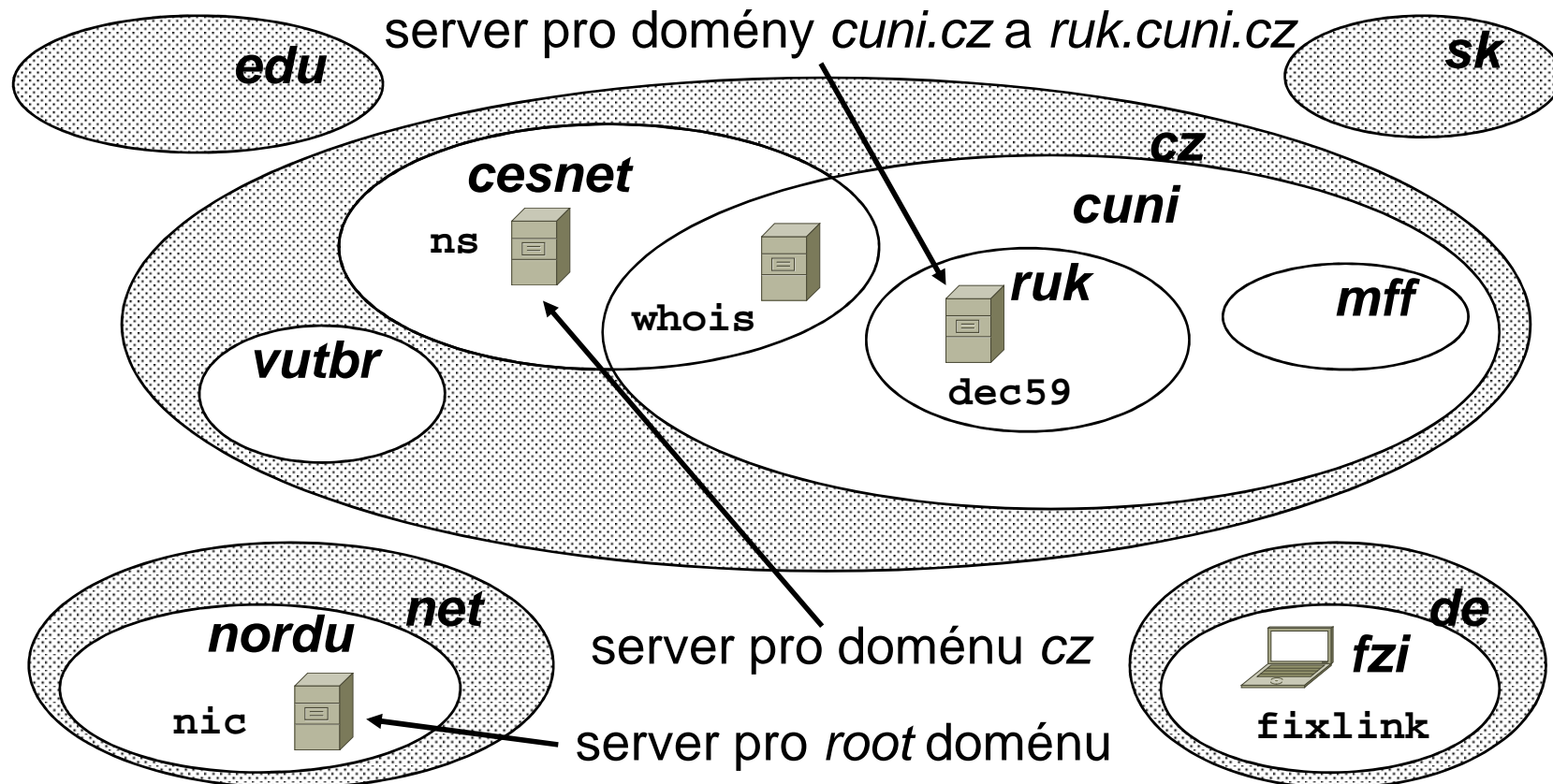
Komunikace a vrstvy



Adresace v TCP/IP



Doménový systém



Správa domén

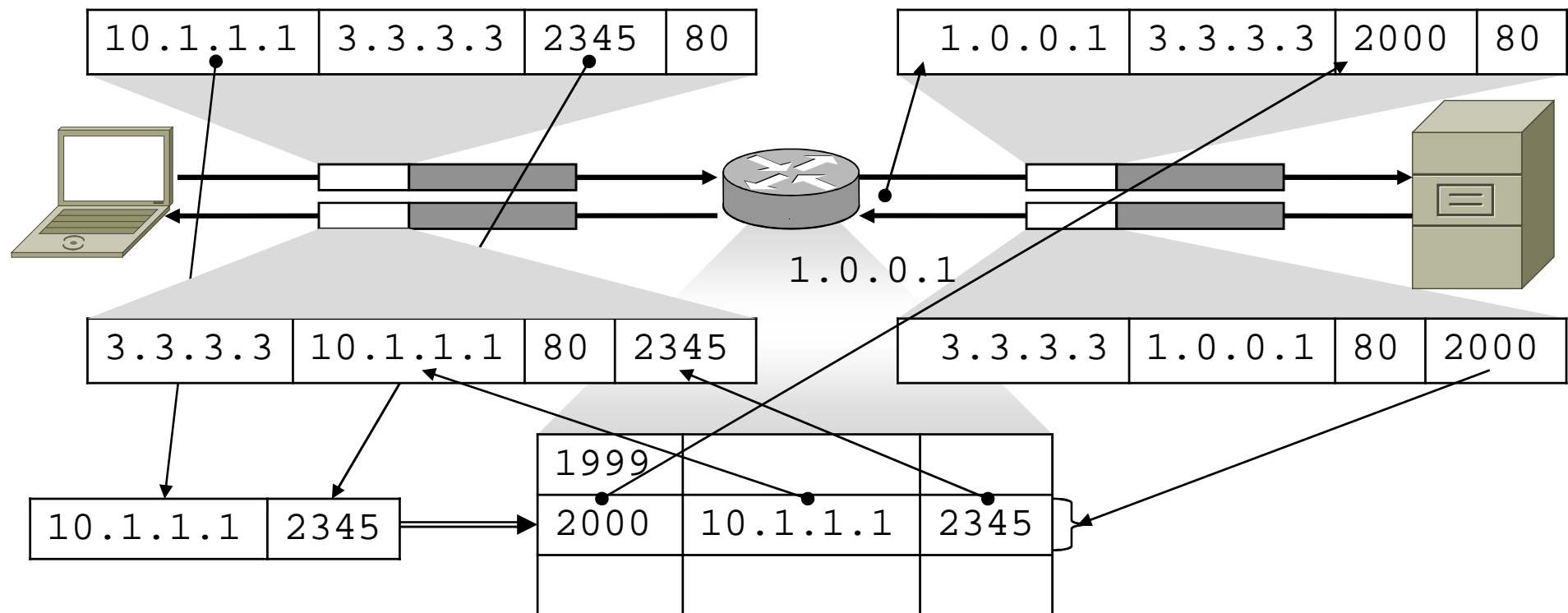
- Domény nejvyšší úrovně (spravuje ICANN, Internet Corp. for Assigned Names and Numbers):
 - původně čistě americké „rezortní“ (**com**, **net**, **org**, **edu**, **mil**, **gov**); postupně uvolněny a doplněny o další (**info**, **biz**, **aero**, ...); o další už mohou žádat privátní subjekty
 - ISO kódy zemí (**cz**, **sk**, ...) a několik výjimek (**uk**, **eu**); některé „zajímavé“ státečky jména prodávají (**nu**, **to**)
 - internacionalizované kódy (**.中国** = **.xn--fiqs8s**, **.pφ**)
- TLD **.cz**:
 - CZ.NIC (sdružení ISP), dohoda s vládou o správě
 - není zavedena struktura, cca 3/4 mil. jmen pod **.cz**
 - nejsou podporována lokalizovaná jména (IDN)
- Nižší domény:
 - spravuje sám vlastník (**ms** . [**mf** . [**cuni** . **cz**]])

IP adresy

- Každý koncový uzel v síti TCP/IP musí mít IP adresu
- V současnosti:
 - IP verze 4: 4 byty (např. 195.113.19.71)
 - IP verze 6: 16 bytů (např. 2001:718:1e03:a01::1)
- Přiřazení adresních bloků:
 - veřejné adresy přiděluje síti její ISP
 - uvnitř LAN lze používat privátní adresy nedostupné zvenku (bezpečnost vs. interoperabilita)
- Přiřazení adresy počítači:
 - o způsobu (pevné vs. dynamické, volné vs. omezené) rozhoduje správa LAN
 - platí i pro privátní, neplatí pouze pro *link-local* adresy

Překlad adres (NAT)

- Obecný princip, kdy lokální síť používá *privátní* adresy a ven se představuje *veřejnými* adresami (nebo jinými privátními)
- Jiný termín: *IP masquerading*
- Implementace i terminologie se v detailech liší



Tok dat v TCP/IP

- Multiplexing:
 - sdílení komunikačního kanálu více službami
- Demultiplexing:
 - přijímající strana musí data správně distribuovat podle řídicích informací uložených v PDU (protocol data unit)
- Zapouzdření (encapsulation):
 - do PDU jedné vrstvy se uloží data i řídicí informace jiné vrstvy (typicky $n+1 \Rightarrow n$, jsou možné i jiné kombinace)
- Segmentace:
 - rozdělení aplikačních dat na transportní vrstvě
- Fragmentace:
 - další dělení dat na síťové vrstvě díky malé MTU (maximum transmission unit) linkové vrstvy

Autentikace, autorizace

- Autentikace (autentizace, autentifikace) je proces ověření identity subjektu. Autorizace je vymezení rozsahu služeb pro identifikovaný subjekt.
- Lokálně lze autentikovat pomocí:
 - znalostí (heslo, PIN)
 - technických prostředků (klíč, HW token, ...)
 - biometrie (otisky prstů, ...)
- Vzdálená autentikace:
 - ochrana proti odposlechu: systém jednorázových hesel (OTP), kryptografie
 - přenos dat v protokolu: např. pomocí SASL (obecný model, zařazený do protokolů na základě *profilů*, např. v SMTP)
 - možnost použití autentikačního serveru a autentikačního protokolu (LDAP, RADIUS, NTLM, Kerberos)

One-Time Password (OTP)

- Obecné označení pro mechanismy umožňující nereplikovatelnou plain-textovou autentikaci uživatele
- Původní varianta:
Vytištěný seznam jednorázových hesel.
- Starší systém:
Server vyšle jedinečný náhodný kód, uživatel na klientovi z něj určeným způsobem vyrobí odpověď (např. pomocí speciální HW či SW kalkulačky, kam zadá přijatý kód a svoje heslo a dostane odpověď) a klient ji pošle servru.
- Modernější řešení:
Uživatel dostane speciální autentikační předmět (*token*), který na základě přesné časové synchronizace se servrem generuje jednorázový časově omezený kód.

Kryptografie – symetrické šifrování

- Historie: aditivní, transpoziční, substituční šifry, šifrovací mřížky a tabulky atd.
- Současnost: analogické principy převedené do digitální podoby a podložené matematickou teorií
- Podstata: pro šifrování a dešifrování se používá stejný klíč
- Příklady: DES, Blowfish, AES, RC4
- Výhoda: rychlé, vhodné na velká data
- Nevýhoda: partneři si musí klíč předat bezpečnou cestou

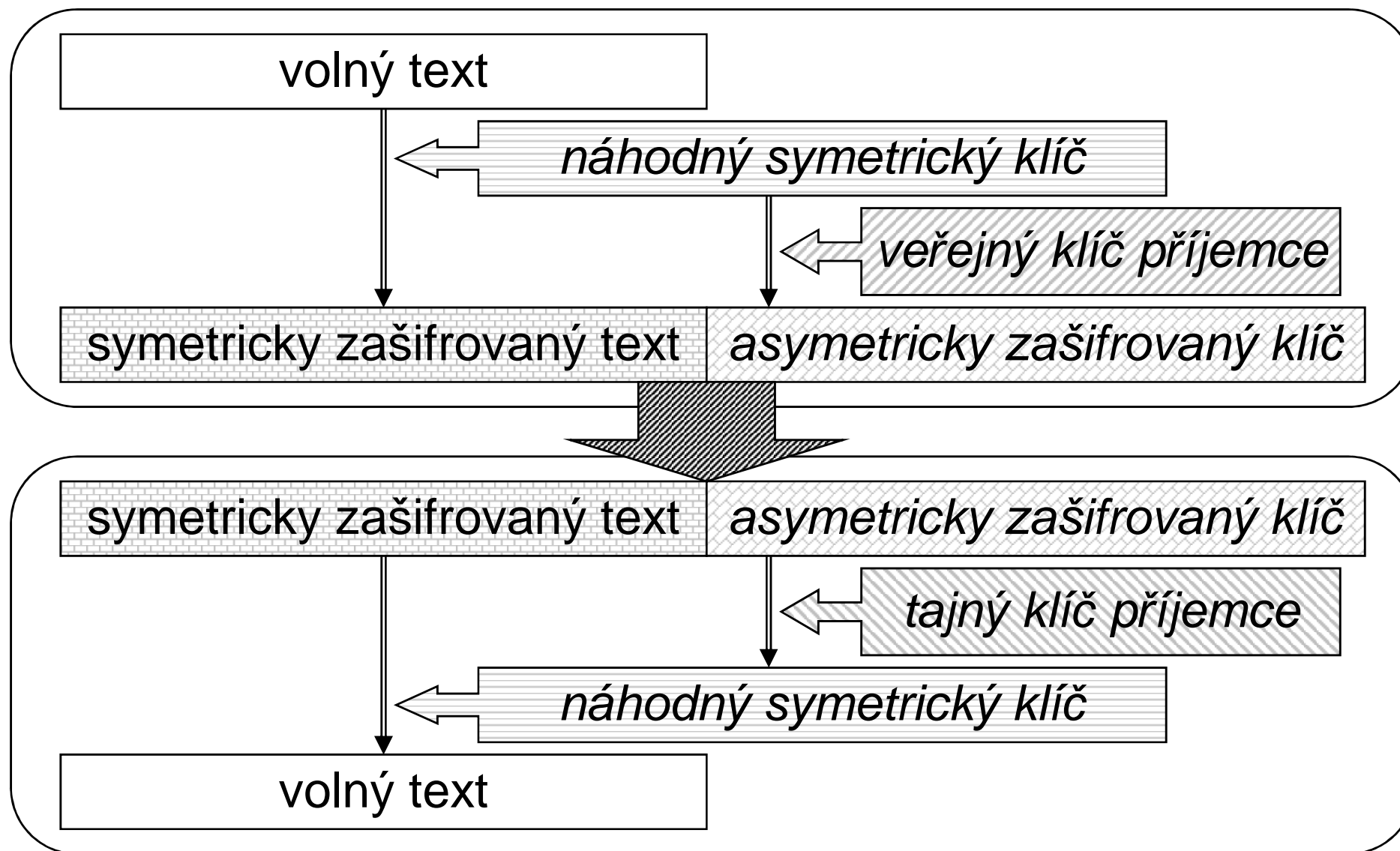
Kryptografie – asymetrické šifrování

- Podstata: pro šifrování a dešifrování se používá pár navzájem neodvoditelných klíčů
- Matematický základ: jednocestné funkce
 - násobení vs. rozklad na prvočinitele
 - diskrétní logaritmus $m = p^k \bmod q$
- Příklady: RSA, DSA
- Výhoda: odpadá problém sdíleného tajemství - jeden klíč (veřejný) lze šířit, druhý (tajný) pečlivě uschovat
- Nevýhoda: pomalé algoritmy, lze šifrovat jen malá data
- Zásadní problém: veřejný klíč je třeba **pečlivě ověřovat**

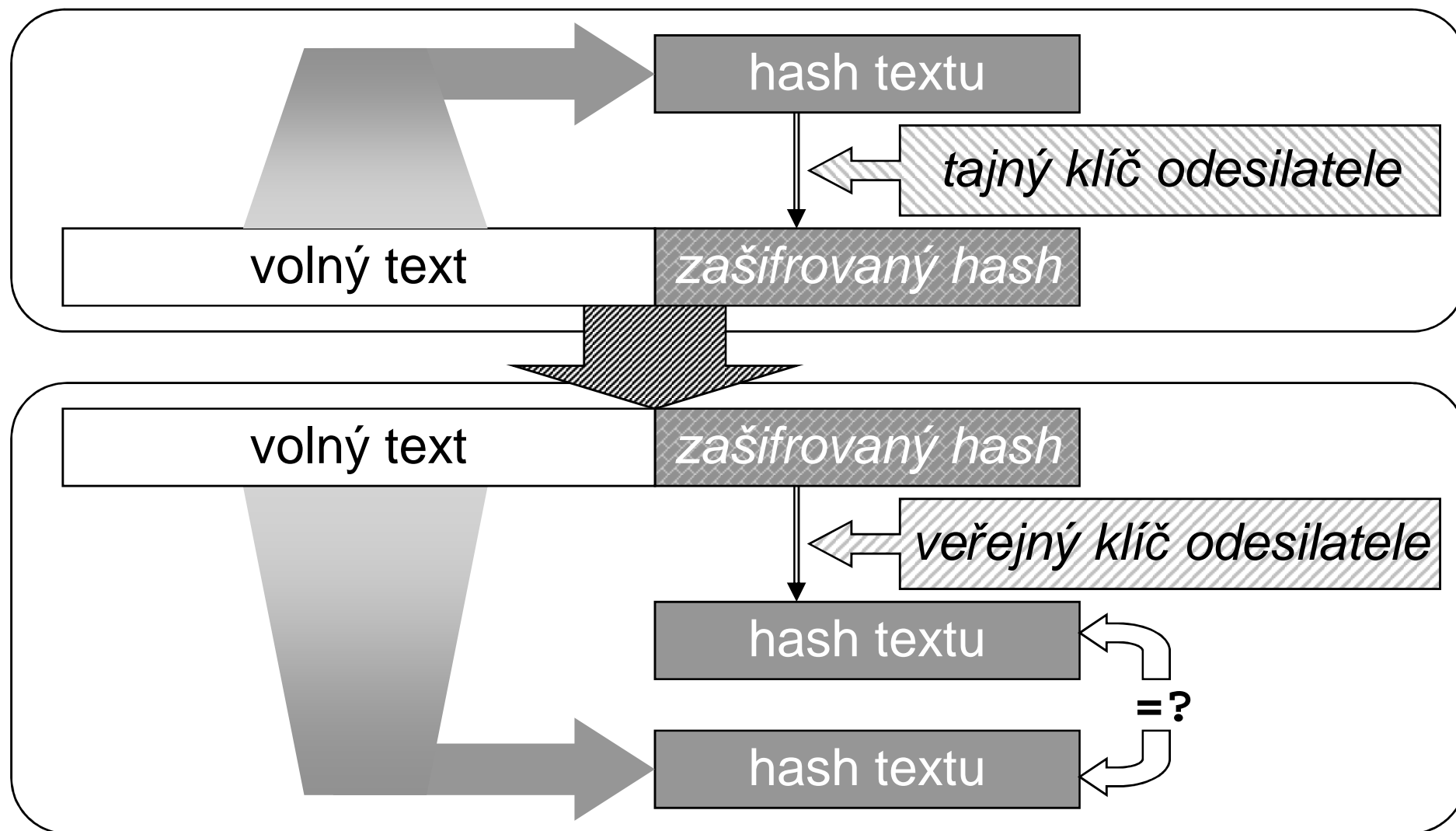
Kryptografie - hashovací funkce

- Hashovací funkce
 - vytvoření pevného „kódu“ z daného textu
 - široké uplatnění (kontroly shody, výběr z tabulky,...)
 - příklady: CRC, MD5
- Použití v kryptografii
 - doplnění požadavků na algoritmus o bezpečnostní prvky
 - malá změna textu = velká změna hashe, „skoro jednoznačný“
 - jednosměrnost, text je z hashe „neodvoditelný“
 - nalezení textu se shodným hashem je obtížné
 - příklad: SHA

Šifrování dat



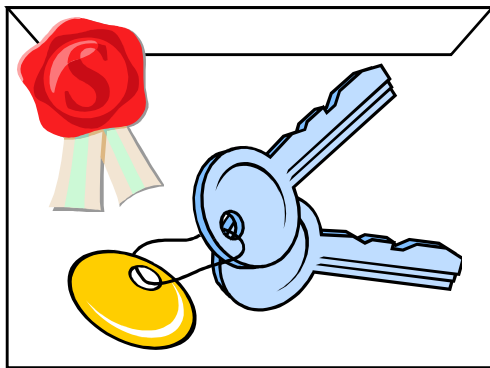
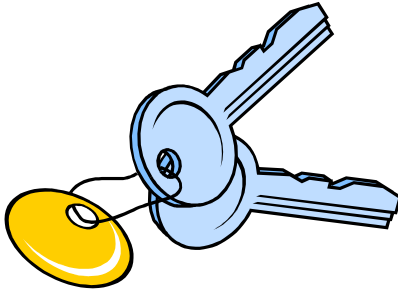
Elektronický podpis



Diffie-Hellman algoritmus

- Způsob výměny informací mezi dvěma partnery posílanými nezabezpečeným kanálem tak, aby oba získali sdílenou tajnou informaci (např. symetrický šifrovací klíč)
- Základ řady protokolů založených na symetrické kryptografii
- Postup:
 1. Alice vygeneruje tajné číslo a a veřejná (prvo)čísla p a q .
 2. Spočítá číslo $A = p^a \bmod q$ a pošle p , q , a A Bobovi.
 3. Bob zvolí tajné číslo b , spočte $B = p^b \bmod q$ a pošle B Alici.
 4. Alice spočítá $s = B^a \bmod q$ a Bob totéž $s = A^b \bmod q$.
- Princip:
 - $A^b = (p^a)^b = p^{ab} = p^{ba} = (p^b)^a = B^a$
 - Bez znalosti tajných čísel a a b a při volbě dostatečně velkých prvočísel p a q je i při odchycení čísel A a B spočítání čísla s považováno za neřešitelnou úlohu.

Autenticita veřejných klíčů



- Je třeba ověřit, že jmenovka patří ke klíči
 - Mezi lidmi lze obvykle snadno ověřit, že komunikuji se správným partnerem dřív, než sdělím tajné informace
 - Klíč lze ověřit z více nezávislých zdrojů
 - Mezi komponentami SW je nutno nějak automatizovat
- Autenticitu ověří třetí strana a připojí svůj podpis; je to buďto
 - někdo, koho já osobně znám a mám jeho resp. její klíč ověřený („pavučina důvěry“)
 - veřejně uznávaná certifikační autorita; jejich seznam je např. v prohlížečích, ale věrohodnost takového seznamu není zcela stoprocentní

Certifikát

- Certifikát je klíč doplněný o identifikaci vlastníka a podepsaný vydavatelem, např. certifikační autoritou (CA)
- Pokud důvěřujeme vydavateli, můžeme věřit klíči vlastníka (**ověřovat věrohodnost CA!**)
- Struktura certifikátu podle X.509 (RFC 3280, SSL, ne SSH):
 - certifikát
 - verze certifikátu
 - sériové číslo certifikátu
 - vydavatel
 - doba platnosti
 - vlastník veřejného klíče
 - informace o veřejném klíči vlastníka (algoritmus a klíč)
 - algoritmus pro elektronický podpis
 - elektronický podpis

SSL, TLS

- Secure Socket Layer 3.0 ~ Transport Layer Security 1.0, *dnes již nedoporučovaná*, novější verze 1.1 a 1.2
- Mezivrstva mezi transportní a aplikační vrstvou umožňující autentikaci a šifrování
- Využívá řada protokolů (např. HTTPS na portu 443)
- Princip:
 1. Klient pošle požadavek na SSL spojení + parametry.
 2. Server pošle odpověď + parametry + certifikát serveru.
 3. Klient ověří server a vygeneruje základ šifrovacího klíče, zašifruje ho veřejným klíčem serveru a pošle mu ho.
 4. Server rozšifruje základ šifrovacího klíče. Z tohoto základu vygenerují jak server, tak klient hlavní šifrovací klíč.
 5. Klient a server si navzájem potvrdí, že od teď bude jejich komunikace šifrovaná tímto klíčem.

Aplikační vrstva TCP/IP

- Spojuje funkce OSI 5, 6 a 7
 - pravidla komunikace mezi klientem a serverem
 - stav dialogu
 - interpretaci dat
- Protokol na aplikační vrstvě definuje
 - průběh zpracování na obou stranách
 - formát zpráv (textový/binární, struktura,...)
 - typy zpráv (požadavků a odpovědí)
 - sémantiku zpráv, sémantiku informačních polí
 - varianty průběhu dialogu
 - interakci s transportní vrstvou

Domain Name System

- Klient-sever aplikace pro překlad jmen na adresy a naopak
- Binární protokol nad UDP i TCP, port 53, RFC 1034, 1035
 - Běžné dotazy (do 512B v non EDNS) se vyřizují pomocí UDP
 - Větší datové výměny probíhají v TCP
- Klient se obrací na servery definované v konfiguraci, postupně získává informace o dalších, dokud neví odpověď
- Jednotkou dat je „záznam“ (resource record - RR), př.:
`ns.cuni.cz. 3600 IN A 195.113.19.78`
 - jméno záznamu
 - doba platnosti (TTL)
 - typ a data

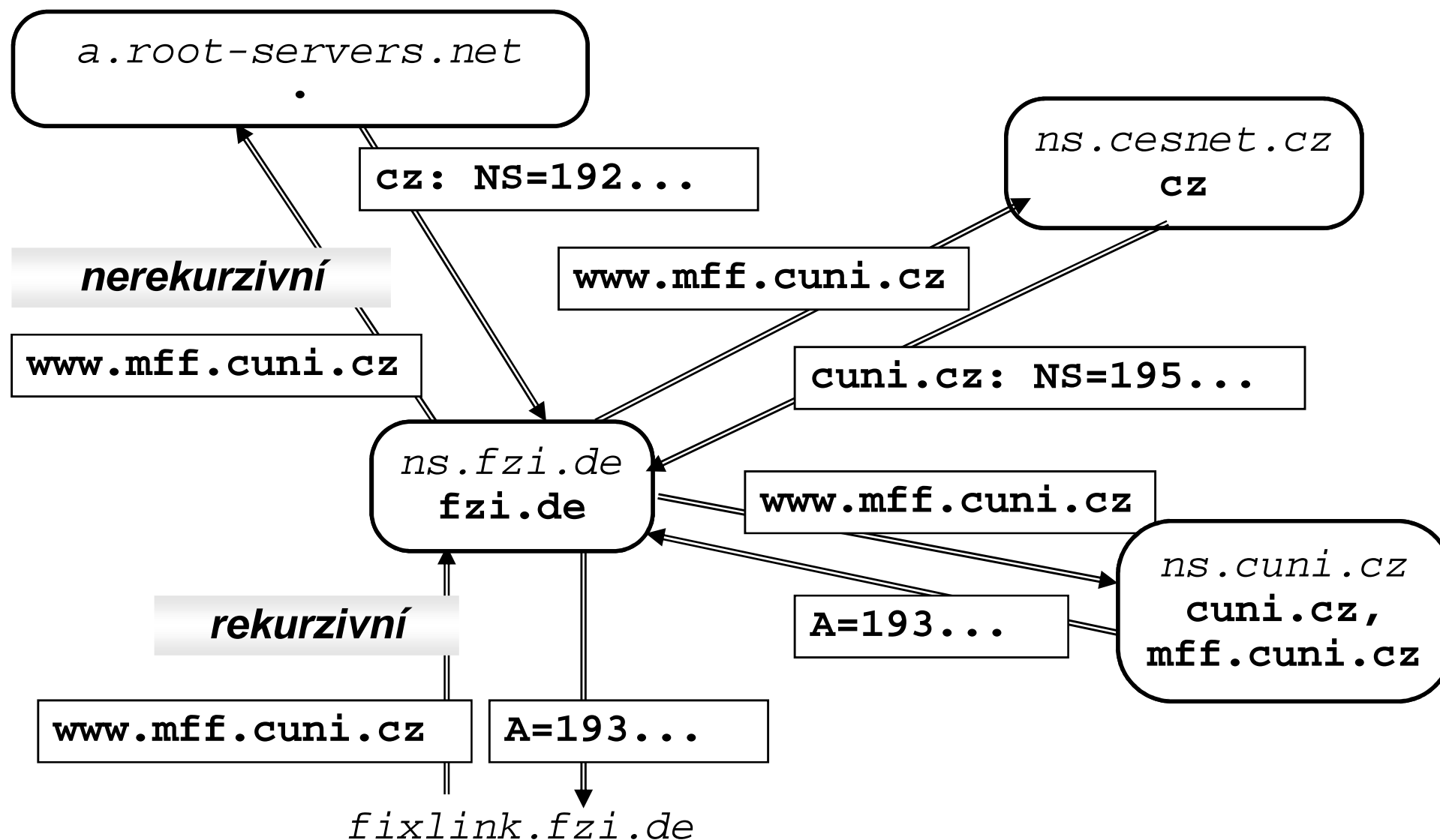
DNS záznamy

Typ	Jméno	Data
SOA	jméno domény	obecné informace o doméně
NS	jméno domény	jméno nameserveru domény
A	jméno počítače	IPv4 adresa počítače
AAAA	jméno počítače	IPv6 adresa počítače
PTR	reverzní jméno (např. pro IP adresu 1.2.3.4 je to 4.3.2.1.in-addr.arpa, pro ::1 je to 1.0...0.ip6.arpa)	doménové jméno počítače
CNAME	jméno aliasu	kanonické jméno počítače
MX	jméno domény/počítače	jméno poštovního serveru a jeho priorita

Servery DNS

- Typy serverů:
 - primární: spravuje záznamy o doméně
 - sekundární: stahuje a uchovává kopii dat o doméně
 - caching-only: udržuje jen (ne)vyřešené dotazy po dobu platnosti
- Každá doména (zóna) musí mít alespoň jeden, ale raději více *autoritativních* (primárních nebo sekundárních) nameserverů
- Pro výměnu dat se používá TCP, ale normální formát dotazu a odpovědi (data se posílají jako DNS RR)
- Aktualizaci zónové databáze vyvolává sekundární server, je ale možné z primárního serveru signalizovat její potřebu

Vyřizování DNS dotazu



DNS dotaz a odpověď

- **Dotaz:**

ID: *n*
FLAGS: Recursion Desired
QUERY: www.cuni.cz. IN A

- **Odpověď:**

ID: *n*
FLAGS: Authoritative Answer
QUERY: www.cuni.cz. IN A
ANSWER: www.cuni.cz. IN CNAME tarantula
tarantula IN A 195.113.89.35
AUTHORITY: cuni.cz. IN NS golias
ADDITIONAL: golias IN A 195.113.0.2

Bezpečnost DNS

- Problém útočníka: jak se dostat ke znění dotazu?
 - volba náhodného zdrojového portu
 - volba náhodného ID
- Příklad útoku („cache poisoning“): Do korektní odpovědi může server do sekce `AUTHORITY` a `ADDITIONAL` přidat falešné údaje o jiné doméně.
- Možné řešení: postupovat od root serverů a ptát se pouze autoritativních serverů.
- Komplexní řešení:
 - podpisy zabezpečené DNS (DNSSEC) - delegující doména obsahuje hash podepisovacího klíče, který je uložen na autoritativním serveru domény
 - je ale komplikované a rozšiřuje se pomalu.

Konfigurace DNS

UNIX

lokální doména a nameserver: `/etc/resolv.conf`

```
domain jméno_domény  
nameserver IP_adresa_nameserveru  
nameserver IP_adresa_nameserveru
```

Windows

Control Panel ⇒ Network and Internet
⇒ Network Connections
⇒ Local Area Connection ⇒ Properties
⇒ TCP/IPv4 ⇒ Properties
⇒ General ⇒ Advanced ⇒ DNS

Diagnostika DNS

- Program **nslookup**

- podpříkazy: **set type, server, name, IPadr, ls, exit**

```
> set type=ns
```

```
> cuni.cz
```

```
Server:          195.113.19.71
```

```
Address:         195.113.19.71#53
```

```
Non-authoritative answer:
```

```
cuni.cz nameserver = golias.ruk.cuni.cz.
```

```
cuni.cz nameserver = ns.ces.net.
```

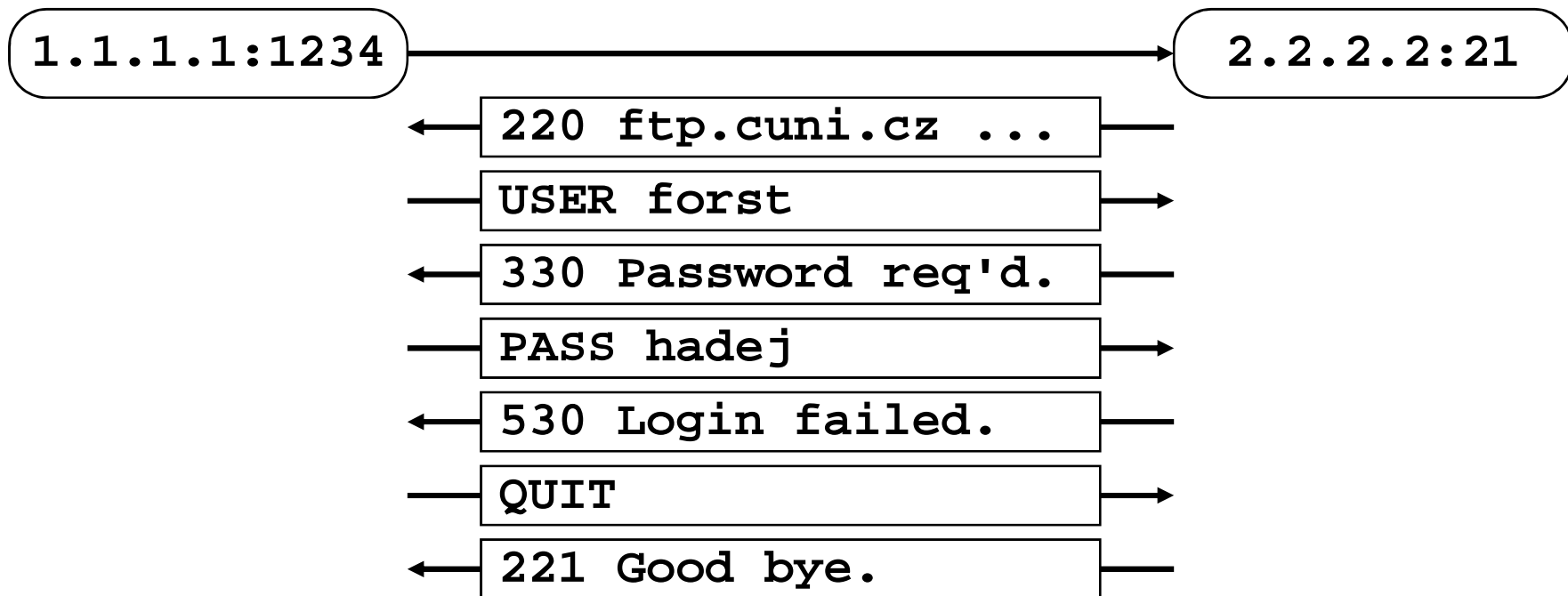
```
Authoritative answers can be found from:
```

- Program **dig**

- **dig** [*@server*] *jméno* [*typ_RR*] [*options*]

File Transfer Protocol

- Jeden z nejstarších protokolů (RFC 959, dodnes platí!)
- Původně přístup k vlastnímu účtu, **otevřený přenos hesla!**
- Dnes hlavně anonymní přístup (uživatel **anonymous** nebo **ftp**, heslo je email) k volně šiřitelným datům
- Ukázka řídicího spojení (příkazy a odpovědi):

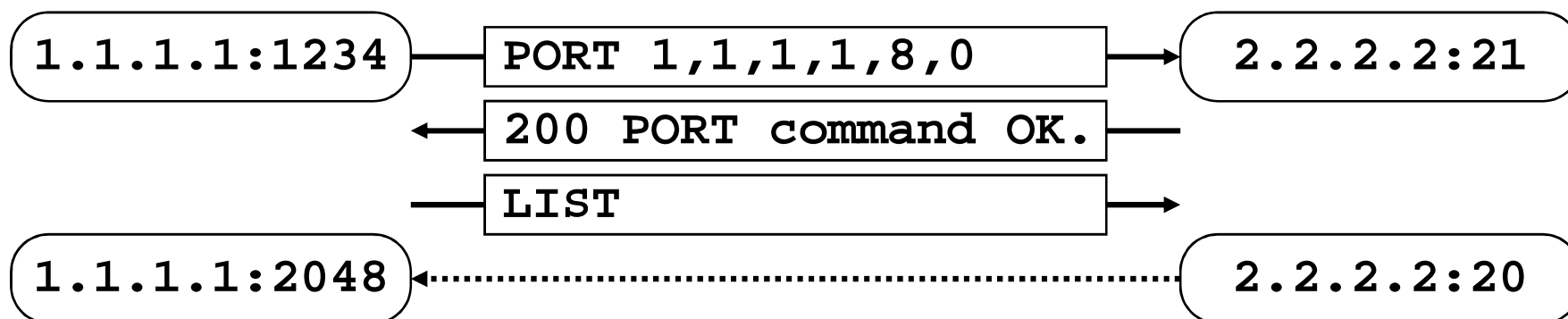


Kódy odpovědí

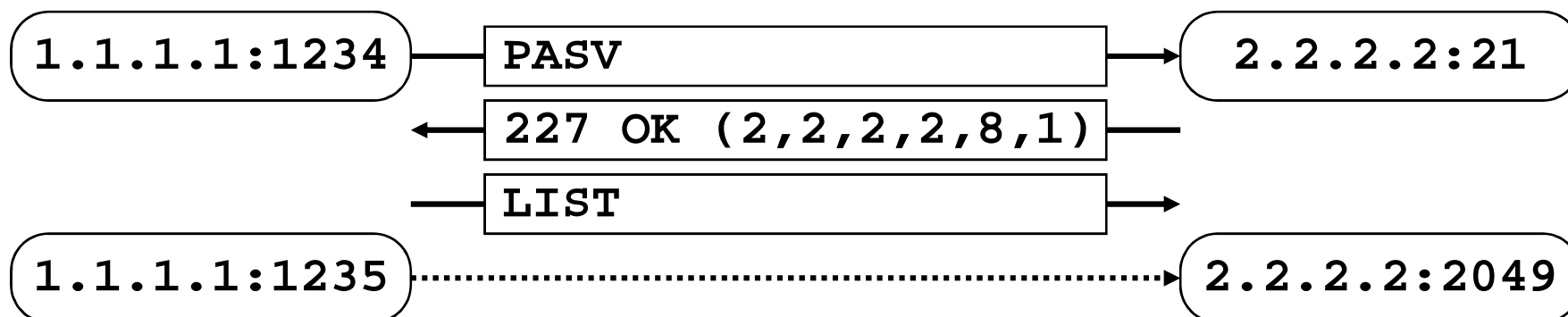
- Pro usnadnění automatického zpracování začíná každá odpověď trojmístným číslem
- První číslice vyjadřuje závažnost odpovědi:
 - 1xx **předběžná kladná odpověď** (akce byla zahájena, budou ještě další odpovědi)
 - 2xx **kladná odpověď** (definitivní)
 - 3xx **neúplná kladná odpověď** (jsou nutné další příkazy)
 - 4xx **dočasná záporná odpověď** (nepodařilo se, ale je možné příkaz opakovat)
 - 5xx **trvalá záporná odpověď** (nepodařilo se a nemá smysl příkaz opakovat)
- Podobné schéma převzala řada následníků

Aktivní/pasivní datové spojení

- Přenos dat probíhá po jiném (datovém) spojení
- Aktivní navázání datového spojení:



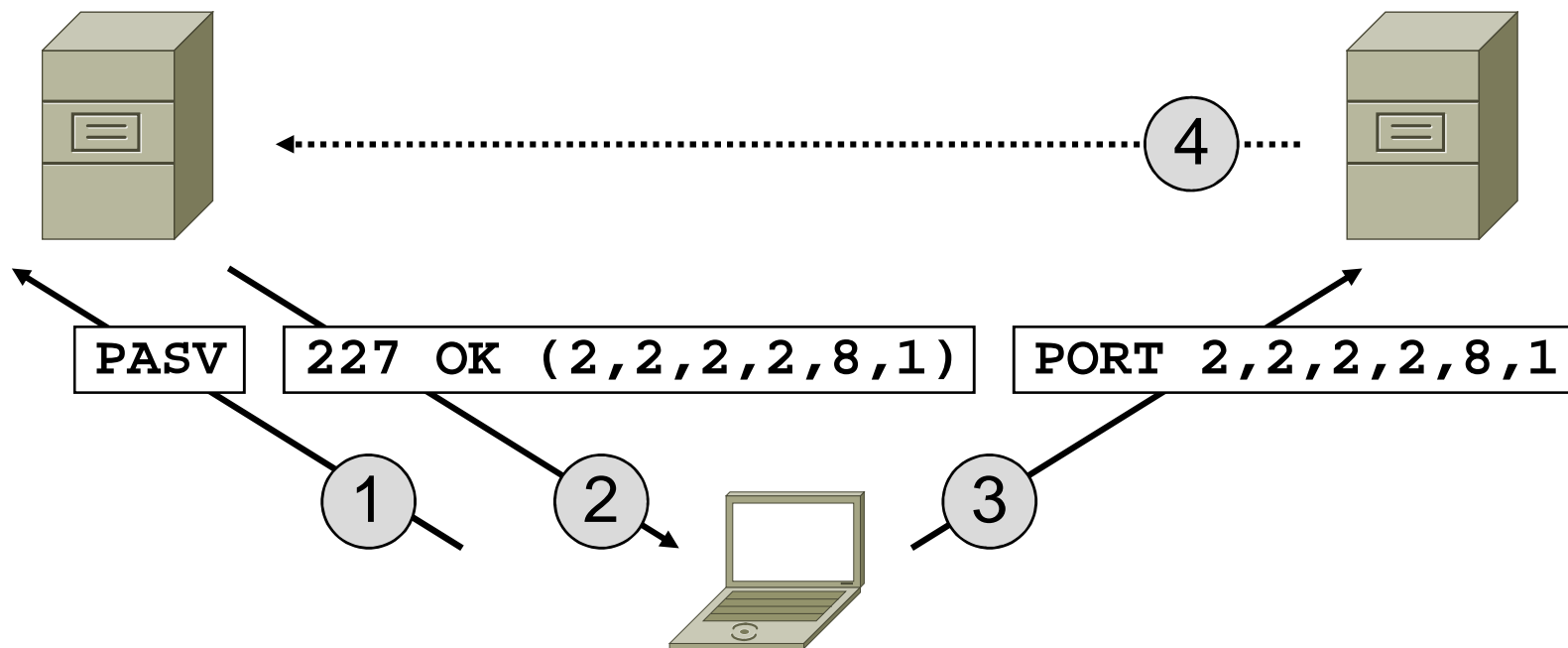
- Pasivní navázání datového spojení:



- Po skončení přenosu se datové spojení uzavře

Third Party Transfer

- Přímý přenos dat mezi servery (z výkonových, kapacitních nebo bezpečnostních důvodů)



- Bezpečnostní riziko: útočník může podvrhnout adresu a port

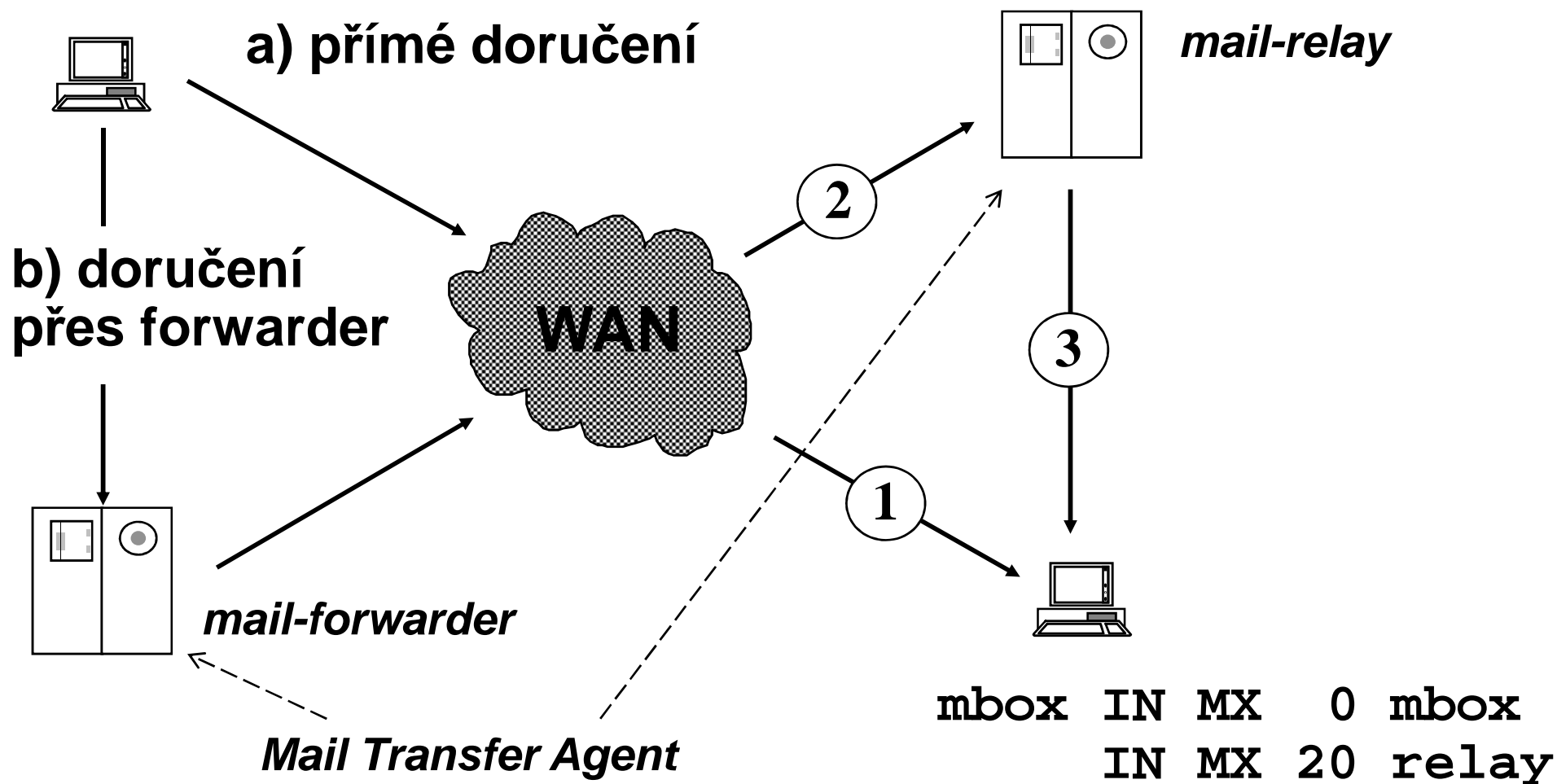
Aplikace pro FTP

- WWW prohlížeče
- správci souborů (Total Commander)
- řádkový interaktivní příkaz **ftp**
 - navazování relace: **open, user**
 - ukončování relace: **close, quit, bye**
 - vzdálené příkazy: **cd, pwd, ls, dir**
 - práce se soubory: **delete, rename, mkdir, rmdir**
 - lokální příkazy: **lcd, !command**
(!cd obecně nefunguje!)
 - přenos souborů: **get, put, mget, mput**
 - typ přenosu souborů: **ascii, binary**
(pozor na textové/binární soubory mezi různými OS!)
 - pomocné příkazy: **prompt, hash, status, help,...**

Elektronická pošta

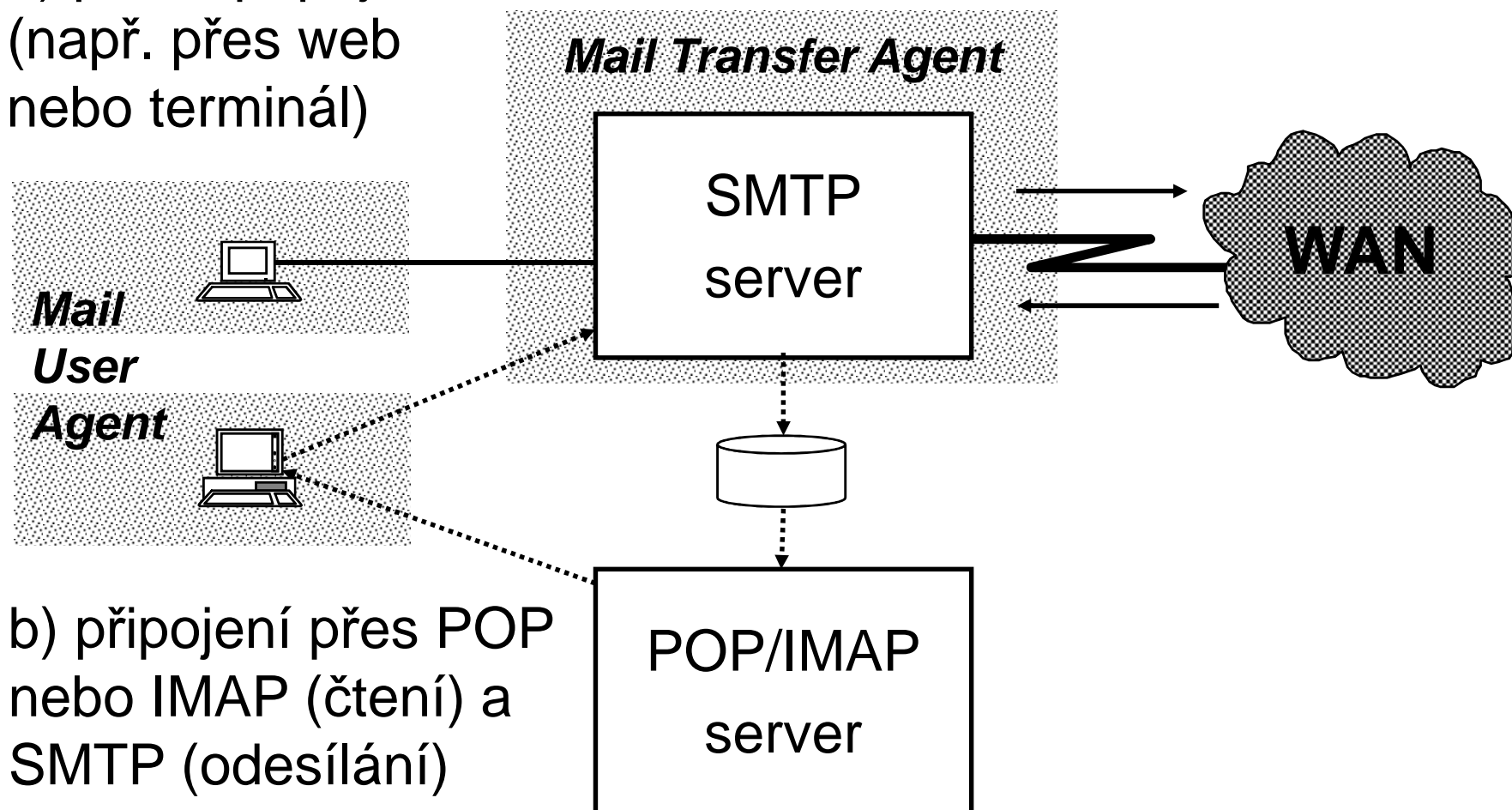
- Obecná služba, existuje i mimo Internet
 - off-line předávání zpráv příp. souborů
 - off-line použití informačních služeb
 - diskusní kluby (mailing-listy, konference)
 - komunikace mimo Internet
- Na Internetu funguje na základě RFC 821, 2821 a 5321 (protokol SMTP resp. ESMTP) a RFC 822, 2822 a 5322 (formát zpráv) na portu 25
- E-mailová adresa v Internetu (typicky):
login@počítač nebo *alias@doména*
např.:
forst@ms.ms.mff.cuni.cz, Libor.Forst@cuni.cz

Příjem a odeslání pošty v SMTP



Přístup k poště z pohledu uživatele

a) přímé připojení
(např. přes web
nebo terminál)



b) připojení přes POP
nebo IMAP (čtení) a
SMTP (odesílání)

Ukázka SMTP protokolu

⇐ 220 alfik.ms.mff.cuni.cz ESMTP Sendmail ...
⇒ HELO betynka
⇐ 250 alfik Hello betynka, pleased to meet you
⇒ MAIL FROM: <forst@cuni.cz>
⇐ 250 2.1.0 <forst@cuni.cz>... Sender ok
⇒ RCPT TO: <libor@forst.cz>
⇐ 250 2.1.5 <libor@forst.cz>... Recipient ok
⇒ DATA
⇐ 354 Enter mail, end with "." on a line by itself
⇒ From: <forst@cuni.cz>
⇒ To: <libor@forst.cz>
⇒ ...
⇒ .
⇐ 250 2.0.0 h98G9Fxt Message accepted for delivery
⇒ QUIT
⇐ 221 2.0.0 alfik closing connection

obálka

dopis

Elektronický dopis

Received: from alfik.ms.mff.cuni.cz
by betynka.ms.mff.cuni.cz...
Date: Thu, 16 Nov 1995 00:54:31 +0100
To: student1@ms.mff.cuni.cz
From: Libor Forst <forst@cuni.cz>
Subject: Test posty
Cc: student2@ms.mff.cuni.cz
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary=="_XXX_="

--=_XXX_="

Content-Type: text/plain; charset=Windows-1250
Content-Transfer-Encoding: 8bit

Čau Petře!

...

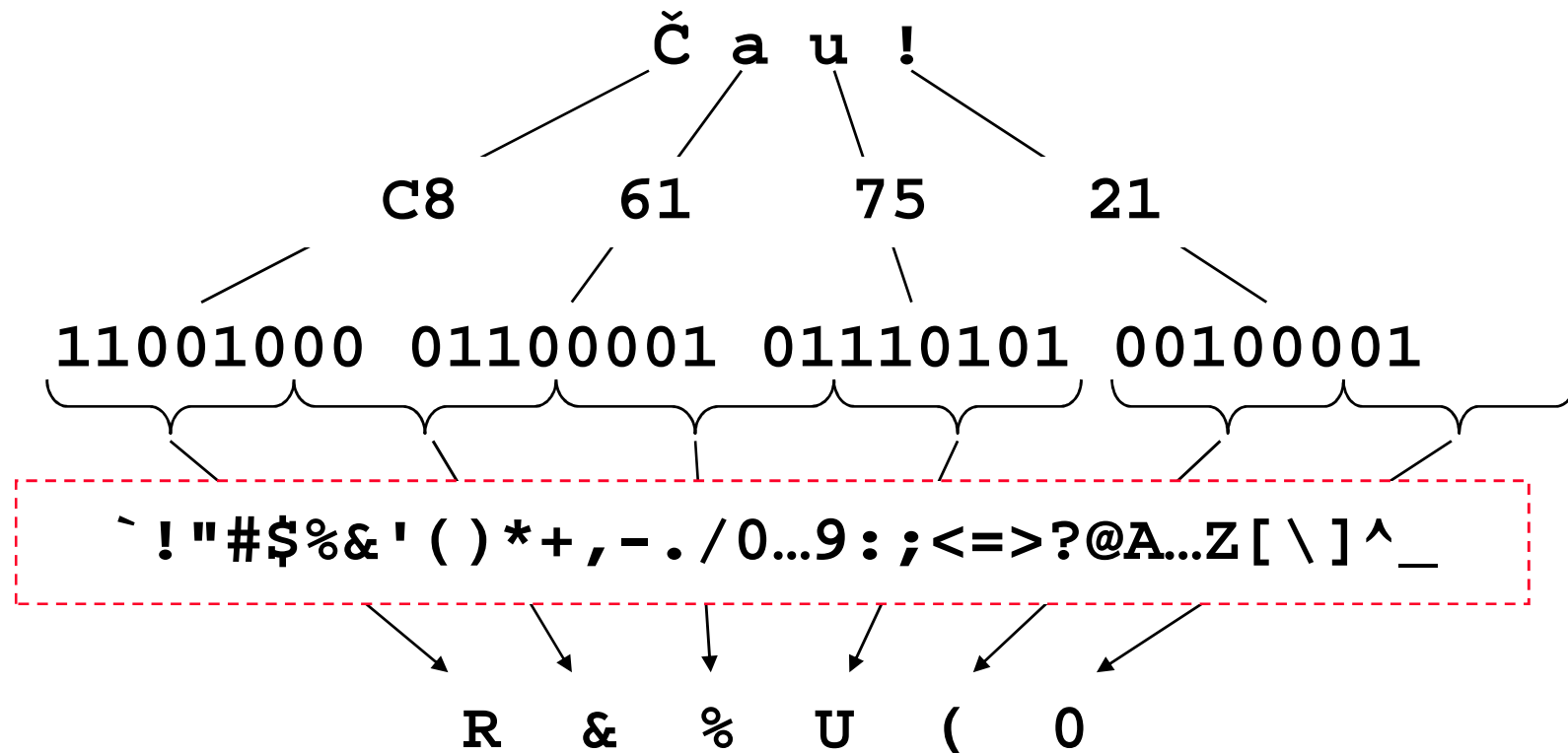
--=_XXX_="

Hlavičky dopisu

Date :	datum pořízení dopisu
From :	autor (autoři) dopisu
Sender :	odesílatel dopisu
Reply-To :	adresa pro odpověď
To :	adresát(i) dopisu
Cc :	(carbon copy) adresát(i) kopie („na vědomí:“)
Bcc :	(blind cc) tajní adresáti kopie
Message-ID :	identifikace dopisu
Subject :	předmět dopisu
Received :	záznam o přenosu dopisu

Soubory a diakritika v poště

- Původně pouze 7-bit ASCII, kódování souborů pomocí UUENCODE (pochází z UUCP, unix-to-unix-copy)



- Kódování OK, ale chybí systematické začlenění do dopisu

Multipurpose Internet Mail Extension

- RFC 2045-2049, umožňuje:
 - Strukturovat dokument
 - Pro každou část
 - Popsat typ a formát obsahu (př. `text/html`)
 - Zadat znakovou sadu a kódování dokumentu
 - Doplnit další informace ke zpracování
 - Používat diakritiku i v (některých) hlavičkách:
`Subject: =?utf-8?b?SVRBVCAyMDEyIC0gcG96?=`
- Kódování:
 - **Base64**: vychází z uuencode, jiná tabulka a formát řádek
 - **Quoted-Printable**: nonASCII znaky jsou uloženy jako řetězec „=HH“, kde HH je jejich hexadecimální hodnota
- Dnes široce používaný i mimo poštu

Etika poštovního styku

- RFC 1855 (Netiquette Guidelines)
 - přečíst všechny maily, než odpovíte
 - zvažovat zásah do konverzace, pokud jste jen Cc
 - nechat příjemci čas na odpověď (ale ověřit doručení lze)
 - odpovídat rychle, alespoň jako potvrzení
 - pečlivá volba Subjectu, kontrola adresátů
 - volba jazyka, výrazových prostředků, emocí
 - míra zachování původního textu v odpovědi
 - respektování ©, souhlas autora při přeposílání
 - účelné a ověřené posílání souborů, češtiny
 - kontrola toho, co mailer posílá (ne HTML!)
 - přetěžování uživatelů a sítě, řetězové dopisy
 - podpis

Bezpečnost pošty (uživatel)

- Dopis je vždy **otevřená listovní zásilka**
(z různých příčin se může dostat do ruky mnoha lidem)

Řešení: šifrovat obsah dopisu (např. PGP - Pretty Good Privacy)

- Nikdy není jistý **odesílatel**, ani shoda údajů v obálce a textu

Částečná řešení: Sender Policy Framework, pokus o zpětné doručení

Řešení: systém výzva/odpověď, elektronický podpis

- Neotevírat soubory neznámého původu!

Bezpečnost pošty (klient, server)

- Běžný server by měl posílat maily lokálních klientů/uživatelů komukoliv, ostatní maily pouze lokálním uživatelům; jinak se jedná o tzv. *open-relay* a hrozí riziko zneužití pro rozesílání hromadných mailů a díky tomu zablokování komunikace od jiných serverů.
- Ze stejného důvodu může při prvotním vložení mailu do systému (*mail submission*) server (někdy označovaný jako MSA) požadovat, aby se klient autentikoval pomocí ESMTP příkazu AUTH (je to součást SASL profilu pro SMTP).
- Klient může pomocí ESMTP příkazu STARTTLS požádat o zahájení SSL/TLS spojení (např. mezi pobočkami firmy, jinak je šifrování spíše problém uživatele).

Ochrana proti spamu

- Spam („kořeněná šunka“) je nevyžádaná pošta, jejímž smyslem je buď inzerce nebo prostě jen obtěžování lidí
 - Grey-listing: spam-engine obvykle neopakuje pokus o doručení, takže server udržuje databázi tripletů <klient, sender, recipient> a napoprvé mail odmítne odpovědí 450, opakované doručení již akceptuje.
 - Sender Policy Framework: doména publikuje (pomocí SPF příp. TXT DNS RR) algoritmus, jak ověřit, že stroj, který odesílá dopis z dané domény, má na to právo; problém při přeposílání dopisů.
 - DomainKeys Identified Mail (DKIM): server domény podepisuje text a některé hlavičky dopisu
 - Antispam: server na základě nastavitelné heuristiky odhaduje pravděpodobnost, že mail je spam; diskutabilní účinnost a riziko *false positive*

Post Office Protocol

- Protokol pro přístup uživatelů k poštovní schránce
- Aktuální je verze 3, RFC 1939, port 110
- Hlavní nevýhody:
 - Otevřené posílání hesla; existuje rozšiřující příkaz pro posílání šifrovaného hesla (APOP), ale řada klientů ho nemá implementovaný
 - Dopisy je nutno stahovat ze serveru celé; existuje příkaz TOP pro stažení začátku dopisu, ale opět je jen řídce implementovaný
 - Není možné pracovat se strukturou dokumentů
- Dnes podporován spíše kvůli zpětné kompatibilitě a nahrazován protokolem IMAP

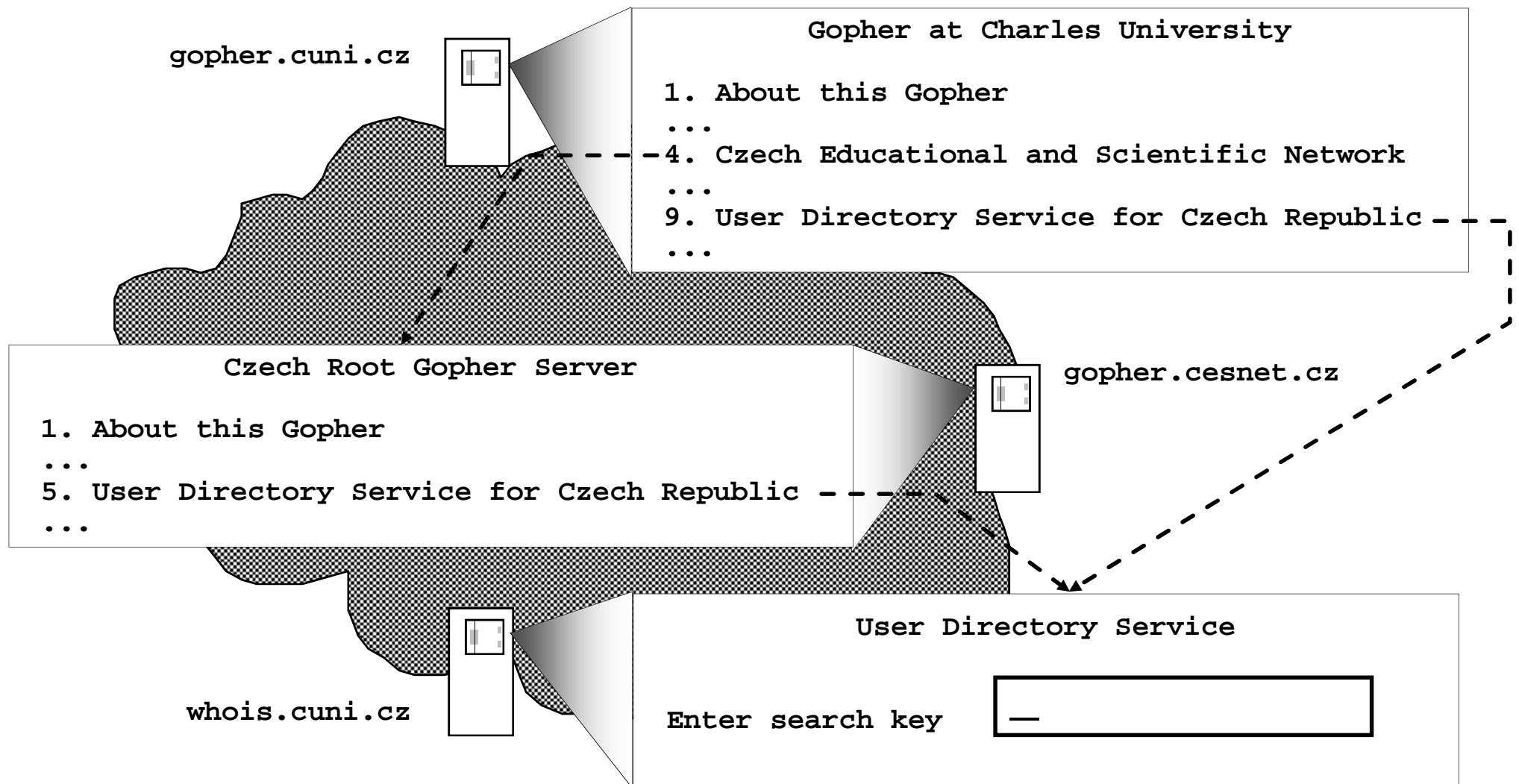
Ukázka POP3 protokolu

```
⇐ +OK POP3 server ready ...
⇒ USER forst
⇐ +OK User accepted
⇒ PASS heslo
⇐ +OK Pass accepted
⇒ LIST
⇐ +OK 2 messages (1234 octets)
⇐ 1 1111
⇐ 2 123
⇐ .
⇒ RETR 1
⇐ +OK 1111 octets
⇐ From: ...
⇐ .
⇒ DELE 1
⇐ +OK message 1 deleted
```

Internet Message Access Protocol

- Modernější, ale složitější nástupce POP
- Aktuální verze 4rev1, RFC 3501, port 143
- Hlavní výhody:
 - Zabudována možnost používat šifrované spojení
 - Server uchovává informace o dopisech (stav)
 - Podpora více schránek (složek)
 - Protokol umožňuje vyžádat pouze část dopisu
 - Je možné nechat na serveru v dopisech vyhledávat
 - Možnost zadat paralelní příkazy
- Šifrování:
 - a) navázání spojení na port 993
 - b) vyvoláno příkazem STARTTLS
- IMAP má implementována většina stávajících MUA

Princip distribuované databáze



Hypertext

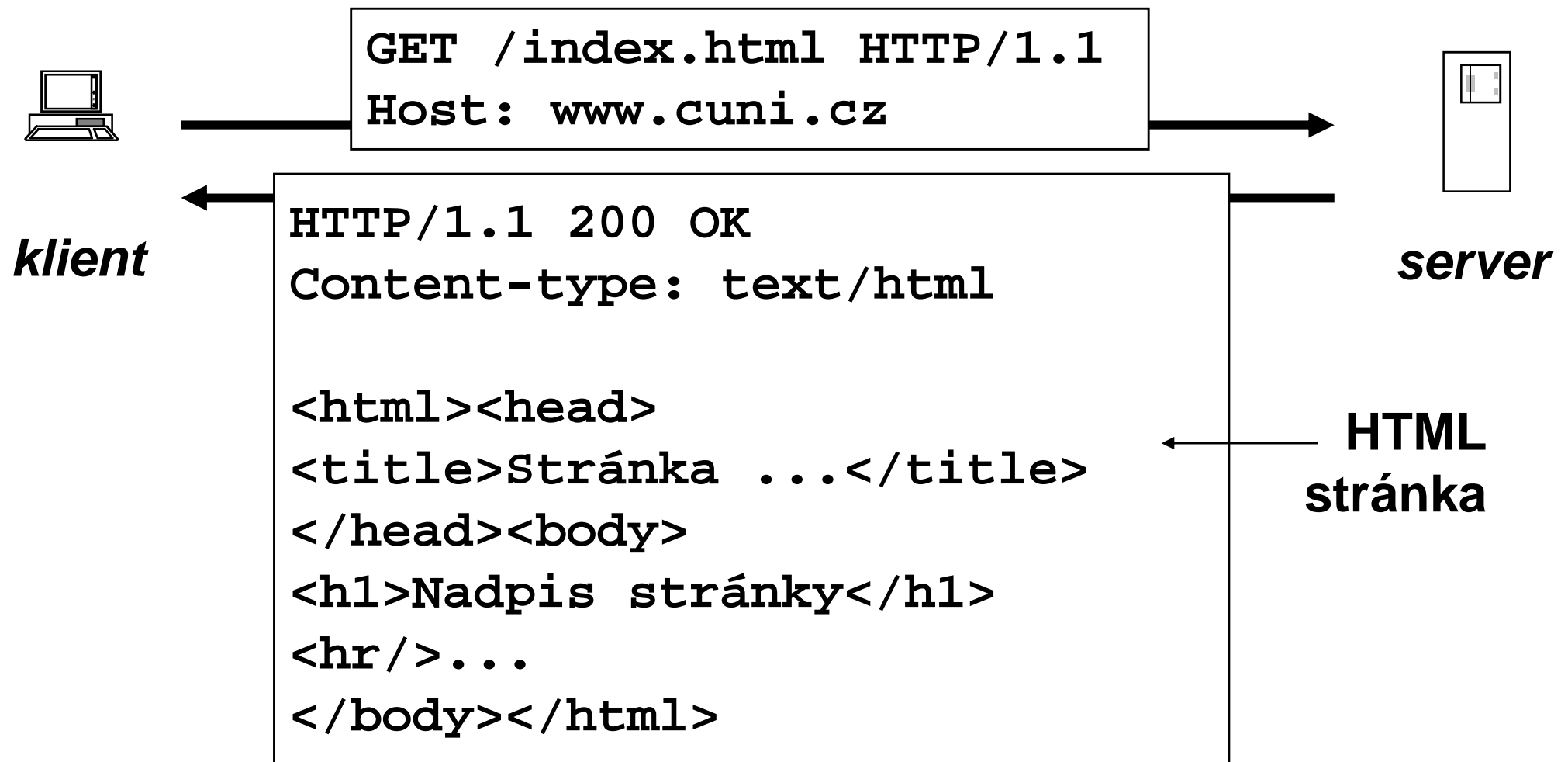
- Základní myšlenka (1945):
nelineární hierarchický text obsahující vazby, které umožňují pokračovat čtením podrobnější informace nebo příbuzného tématu
- Pozdější rozšíření (1965):
doplnění samotného textu o netextové informace (obrázky, zvuk, video...), někdy se používá pojem *hypermediální text*
- Praktická implementace (1989):
systém World Wide Web vyvinutý v CERNu

World Wide Web

- WWW je distribuovaná hypertextová databáze
- Základní jednotkou je hypertextová *stránka* (dokument), kterou server posílá na žádost klientům
- Dokumenty jsou psány v textovém jazyce HTML (Hypertext Markup Language)
 - popisuje obsah i formu
 - konkrétní zobrazení je v režii klienta resp. uživatele
- Dokumenty existují staticky (cesta v URL pak obvykle odpovídá skutečné relativní cestě na disku serveru) nebo se vytvářejí dynamicky dle požadavků klienta
- Přenos stránek se odehrává pomocí protokolu HTTP (Hypertext Transfer Protocol)

Ukázka protokolu HTTP

URL: `http://www.cuni.cz/index.html`



Hypertext Transfer Protocol v.1

- V současnosti převažuje verze 1.1, RFC 7230, port 80
- Obecný formát zpráv:
 - úvodní řádka (požadavek/odpověď)
 - doplňující hlavičky
 - požadavek: jazyk, kódování, stáří stránky, autentikace,...
 - odpověď: typ dokumentu, kódování, expirace,...
 - (volitelné) tělo dokumentu
- Kódy odpovědí:
 - 1xx **informativní odpověď** (požadavek přijat, zpracovává se)
 - 2xx **kladná odpověď** (definitivní)
 - 3xx **přesměrování** (očekává se další požadavek od klienta)
 - 4xx **chyba na straně klienta** (nesprávný požadavek)
 - 5xx **chyba na straně serveru** (nepodařilo se vyhovět požadavku)

Metody HTTP

Metoda	Tělo požadavku	Tělo odpovědi
GET	---	požadovaná stránka
HEAD	---	---
POST	parametry stránky	požadovaná stránka
PUT	uploadovaný soubor	---
CONNECT	← <i>tunel</i> →	

Vlastnosti HTTP v.1

- Odpovědí na jeden požadavek je obvykle jeden dokument (stránka, obrázek,...)
- Po jednom (perzistentním) spojení může jít postupně více požadavků, klienti si obvykle otevírají současně několik spojení
- Požadavky jsou nezávislé, komunikace je bezstavová; stav je nutno přenášet jako dodatečná data, tzv. *cookies*:
 - server vygeneruje cookies s identifikací spojení a pošle je v hlavičkách klientovi
 - klient při dalších požadavcích na stejný server tato data přidává do hlaviček požadavku

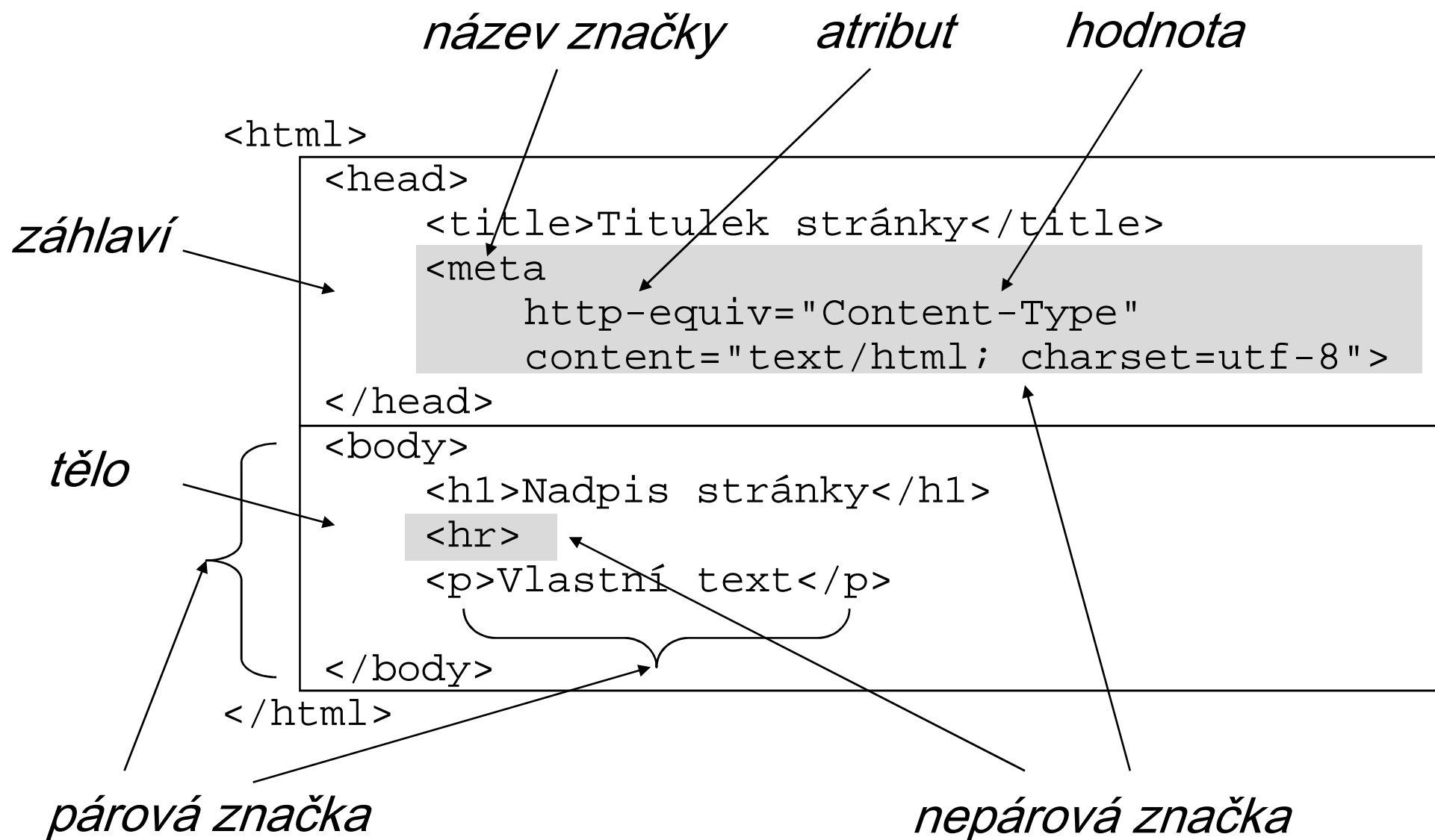
Hypertext Transfer Protocol v.2

- Web je dnes úzce svázán s komercí, takže kolem vývoje inovace HTTP bylo trochu rušno
- Momentálně se prosazují implementace dle RFC 7540
 - binární protokol, lze na něj přejít v rámci HTTP/1 spojení
- Hlavní motivace: větší propustnost
- Metody:
 - vlastní multiplexing více *streamů* v rámci jednoho TCP spojení (streamy se neblokují, dají se prioritizovat)
 - server může poslat (*push*) více dat, než požadoval klient, pokud usoudí, že je klient bude potřebovat
 - v současné době narůstá rozsah hlaviček, navíc často mají podobný obsah - lze je efektivně komprimovat
- Neprošlo: povinné šifrování

Jazyk HTML

- Hypertext Markup Language, vývoj v posledních letech poněkud dramatický, 2014 vyšla kompromisní verze 5
- Vlastní textový obsah stránky je doplněn doplňujícími informacemi, značkami: strukturálními (např. odstavec), sémantickými (např. adresa), formátovacími (např. tučně)
- Je aplikací staršího SGML (Standard Generalized ML) a předchůdcem XML (Extensible ML)
- Formát značky: `<znacka [atributy]>`
- Volný formát řádek (bílé znaky nevýznamné)
- Speciální znaky - entity (`<`, `>`, `&`, ` `;...)
- Komentáře (`<!-- ... -->`)

HTML - struktura dokumentu



HTML - hypertext

- Odkazy - značka *anchor*.
 - odkaz na jinou stránku: `...`
 - označení místa v dokumentu: ``
 - odkaz na část dokumentu: `...`
- Obrázky - značka *image* (`img`), atributy:
 - `src` URI obrázku
 - `alt` alternativní text pro textové klienty
 - `width, height`
 cílové rozměry obrázku
 - `border` okraje obrázku

HTML - formátování

- Základní formátování:
 - odstavec (`<p>...</p>`)
 - nadpis (`<h1>` až `<h6>`)
 - pevné odřádkování (`
`)
 - vodorovná čára (`<hr>`)
 - vycentrování (`<center>`)
- Písmo:
 - určení fontu: `...`
 - fyzický formát: tučné (``), kurzíva (`<i>`), podtržení (`<u>`), pevná šířka (`<tt>`), index (`<sub>`)...
 - logický formát: zvýraznit (``, ``), ukázka kódu (`<code>`)...

HTML - seznamy

```
<ul>
<li>položka A</li>
<li>položka B</li>
</ul>
```

```
<ol>
<li>položka A</li>
<li>položka B</li>
</ol>
```

```
<dl>
<dt>termín A</dt>
<dd>vysvětlení</dd>
<dt>termín B</dt>
<dd>vysvětlení</dd>
</dl>
```

- položka A
- položka B

1. položka A
2. položka B

termín A
vysvětlení
termín B
vysvětlení

HTML - tabulky

```
<table border="1">
  <tr>
    <td colspan="2">Období</td>
    <td>Zisk</td>
  </tr><tr>
    <td rowspan="2">2012</td>
    <td>I - III</td>
    <td align="right">10</td>
  </tr><tr>
    <td>IV - VI</td>
    <td>2000</td>
  </tr>
</table>
```

Období		Zisk
2012	I - III	10
	IV - VI	2000

HTML - formuláře

```
<form action="mailto.cgi" method="post">  
Jméno: <input name="jmeno">  
Zpráva: <textarea name="zprava"  
        rows="3" cols="40"></textarea>  
Poslat  
<input type="radio"  
      name="kdy" value="hned">  
hned  
<input type="radio"  
      name="kdy" value="zitra">  
zítra  
<input type="submit"  
      value="Odeslat">  
</form>
```

Jméno:

Zpráva:

Poslat ☐ hned ☒ zítra

Kaskádové styly

- Složitější formátování přímo v HTML je komplikované
- Kaskádové styly (CSS) je prostředek, jak
 - definovat vlastnosti pro celé oblasti stránky
 - vytvářet vlastní styly
 - dědit a upravovat vlastnosti jiných stylů
- Umožňují snazší údržbu rozsáhlých souborů stránek dodržujících zadané formátovací konvence
- Př.:

```
<style type="text/css">
h2 {color: blue; font-style: italic;}
</style>
```

Zodpovědnost za vzhled stránky

1. Autor stránky

- vkládá do stránky svou ideu
- hloubka detailu záleží na něm

2. Typ a verze prohlížeče

- různé (verze) prohlížeče mohou interpretovat stejný kód mírně odlišným způsobem
- je žádoucí ověřit vzhled na různých prohlížečích

3. Nastavení klienta

- uživatel obvykle má možnost nastavením ovlivnit některé atributy vzhledu (např. zvolit strategii používání fontů, barev)

Dynamické stránky (server)

Dynamika řízena na serveru, na klientovi neběží žádný kód.

- V HTML lze vytvořit formulář, jeho odesláním se na serveru spouští tzv. *cgi-skript*, který za pomoci dat od uživatele (přenášejí se v URI nebo v těle požadavku) vygeneruje text dynamické stránky
- Autor stránky může nechat SW na serveru vložit do textu stránky určité části (tzv. *server-side include*)
- Do textu stránky je možné vložit kód, který zpracuje *HTML preprocesor* (PHP), klient už vidí jen výsledek (datum a čas)

```
<?php  
    echo date( DATE_RFC822 ) ;  
?>
```

PHP obsahuje širokou podporu funkcí, např. pro zacházení s databázemi

Dynamické stránky (klient)

Přenesení dynamiky (spuštění kódu) na klienta.

- Java - jazyk myšlenkově vycházející z C++, s vyššími nároky na bezpečnost, s knihovnamí pro jednoduchou tvorbu uživatelského rozhraní

Java programy (*applety*) se na klienta přenášejí ve formě **mezikódu** nezávislého na platformě, ten klient interpretuje a vykonává za pomoci lokálních knihoven

- Javascript - analogický princip, na klienta se ale přenáší **zdrojový kód** a on ho interpretuje přímo, př.:

```
<script>
    document.write( "<b>POZOR</b>" );
</script>
```

Dnes umí i komunikovat se serverem.

Bezpečnost WWW

- Bezpečnost uživatele
 - komunikace probíhá **otevřeně**, přenos citlivých informací (hesla, údaje ve formulářích) představuje riziko
 - obsah stránky může být podvržen
 - spouštění nebezpečného Java(script) kódu
 - autentikace a šifrování (HTTPS: HTTP+SSL)
 - cookies se ukládají na klientovi, jsou čitelné a mohou být poslány jinému serveru
- Bezpečnost serveru
 - přes WWW server vede řada útoků
 - pečlivě udržovaný systém, minimální práva
- Bezpečnost sítě
 - pokud se klient a server domluví, lze do HTTP zabalit libovolný provoz

Telnet

- Protokol pro přihlašování na vzdálené stroje, port 23
- Zkratka z *Telecommunication Network*
- Jeden z nejstarších protokolů, poprvé v RFC 97 (1971!)
- Uživatel má k dispozici síťový virtuální terminál (NVT), protokol přenáší oběma směry znaky a příkazy pro řízení NVT (slabiny: např. nerozlišuje příkaz a odpověď)
- Hlavní nevýhoda: otevřený přenos dat (řeší až rozšíření podle RFC 2946, které ale přichází pozdě)
- Dnes:
 - přístup na síťová zařízení v rámci odděleného segmentu LAN
 - ladění jiných protokolů:

```
> telnet alfik 25
220 alfik.ms.mff.cuni.cz ESMTP Sendmail ...
HELO betynka
250 alfik Hello betynka, pleased to meet you
```

Secure Shell (SSH)

- Bezpečná náhrada starších protokolů pro vzdálené přihlašování resp. přenos souborů
 - klient ověřuje server
 - komunikace je šifrovaná
- Aktuální verze 2, RFC 4250-4254, port 22
- SSHv2 kromě základní funkce umožňuje:
 - otevírat paralelně více zabezpečených kanálů
 - tunelovat zabezpečeným kanálem jiný provoz
 - zpřístupnit souborový systém (SSHFS)
- Klienti (windows): putty, winscp
- Příkazy (unix):

```
ssh [user@]host [command]
scp [-pr] [user@[host:]]file1 [user@[host:]]file2
```

Bezpečnost SSH

- Klient ověřuje server
 - na základě klíče (potvrzuje uživatel)
 - certifikátu (ověřen autoritou)
- Server ověřuje uživatele
 - pomocí hesla
 - pomocí výzev a odpovědí (OTP)
 - pomocí veřejného klíče (server posílá výzvu zašifrovanou klíčem uživatele, klient odpovídá plain textem)
- Strategie používání klíčů
 - důkladně ověřovat klíč serveru, pozor zvl. při změně (nebezpečí útoku „*man-in-the-middle*“)
 - přihlášení bez hesla vázat na privátní klíč s heslem
 - na méně důležité cíle je možné i bez hesla, ale rozhodně nikoliv recipročně ($A \rightarrow B$ i $B \rightarrow A$) - ochrana proti *červům*

Voice over IP

- Obecné označení technologií pro přenos hlasu po IP
- Lze realizovat různými navzájem nekompatibilními způsoby:
 - standard H.323
 - standard SIP
 - proprietárně (Skype)
- Celá řada problémů:
 - digitalizace hlasu
 - dohadování vlastností zařízení
 - nalezení partnera
 - propojení s běžnou telefonní sítí

H.323

- Komplexní řešení multimediální komunikace od ITU
- Založeno na ASN.1 (binární, bitové protokoly)
- Zahrnuje celou řadu dílčích protokolů, mj.:
 - H.225/RAS (Registration/Admission/Status) pro vyhledávání partnera pomocí tzv. *gatekeeper* uzlů
 - Q.931 (síťová vrstva ISDN) řeší navazování spojení
 - H.245 řeší řízení hovoru (dohodu o používaných vlastnostech zařízení)
 - RTP kanály (Realtime Transport Protocol, RFC 3550) se používají pro vlastní přenos multimediálních dat
 - RTCP (RTP Control Protocol) zabezpečuje jejich řízení
- Dnes postupně nahrazováno SIP

Abstract Syntax Notation 1

- Formální definice datové struktury, př.:

```
Answer ::= CHOICE {  
    word PrintableString,  
    flag BOOLEAN }
```

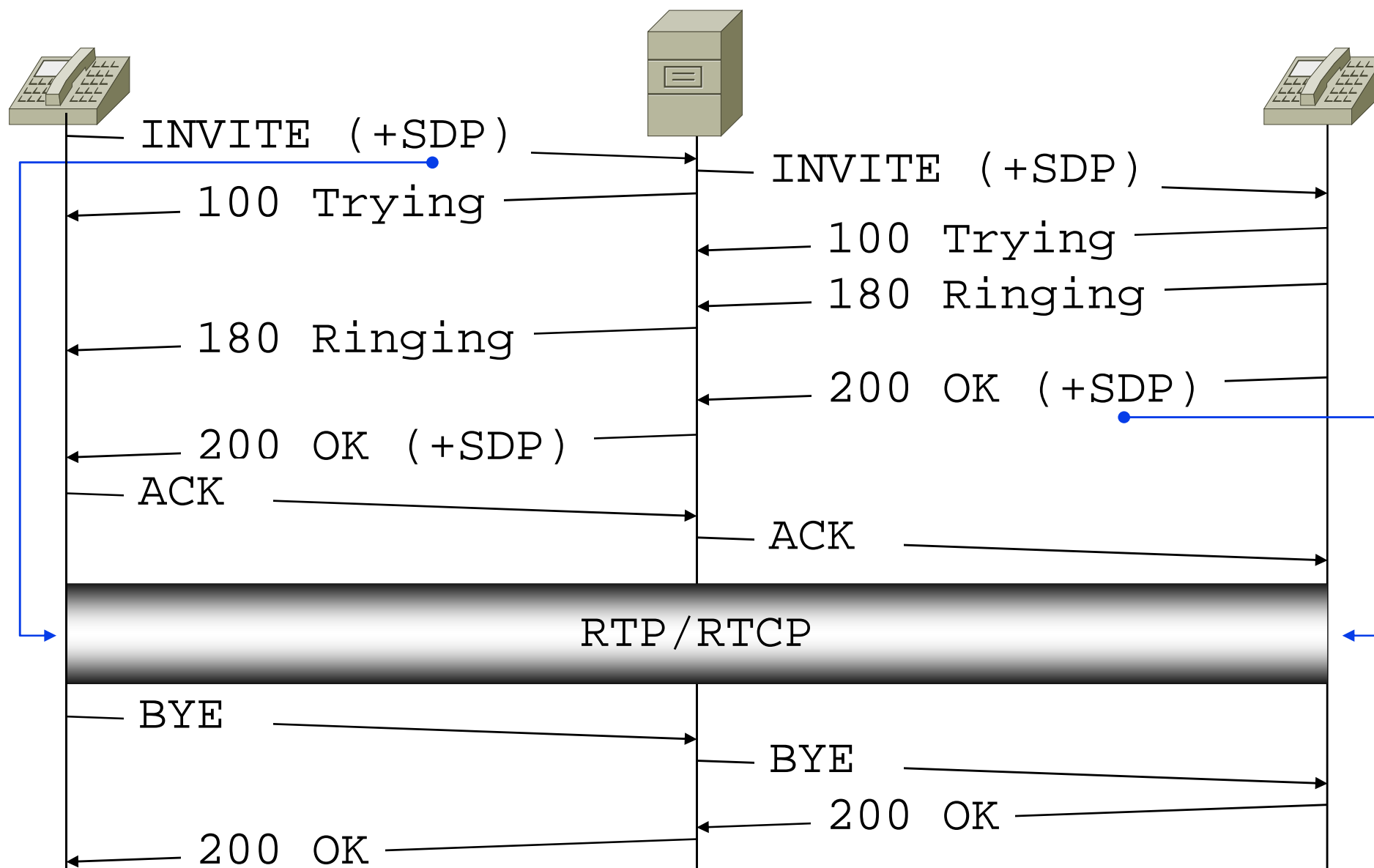
```
SignedData ::= SEQUENCE {  
    version Version,  
    digestAlgorithms DigestAlgorithmIdentifiers,
```

- Pochází z 80. let (a je to na ní znát)
 - př.: výčtový typ (enumerace) se zapíše do tolika **bitů**, kolik je třeba, dopředu se přidá bit s hodnotou 0, ale pokud bude mít hodnotu 1, je typ rozšířen a zapsán **jiným** počtem bitů
- Je možné automaticky generovat parser
- Umožňuje přenášet menší objemy dat, ale neprůhledně
- Příklady použití: H.323, X.509

Session Initiation Protocol

- Náhrada složitého H.323 jednodušším protokolem
- RFC 3261, port TCP i UDP 5060
- Architektura protokolu se podobá HTTP, informace se přenášejí ve formě hlaviček
- Neřeší vlastní přenos dat (obvykle používá RTP/RTCP)
- Řeší jen signalizaci (vyhledání partnera a navázání spojení)
- Dohodu o parametrech datových kanálů obvykle řeší SDP (Session Description Protocol, RFC 4566), jeho data se přenášejí zabalená do těla SIP zpráv
- Koncový uzel se může registrovat u registrátora, tím lze uskutečnit propojení na běžnou telefonní síť

Příklad SIP session



Sdílení systému souborů

- Připojení cizího filesystemu transparentně do lokálního
- Network File System (NFS)
 - původně vyvinut v Sun Microsystems, dnes IETF
 - poslední verze 4.1, RFC 5661, port 2049 (UDP i TCP)
 - identifikace zdroje: server:cesta
 - autentikace: Kerberos
 - zajímavost: relační (RPC) a prezentační (XDR) vrstva
- Server Message Block (SMB)
 - původně vyvinut v IBM, posléze přejal Microsoft
 - open implementace Samba (UNIX)
 - identifikace zdroje: UNC (\\jméno_serveru\jméno_zdroje)
 - autentikace: obvykle uživatelské jméno a heslo

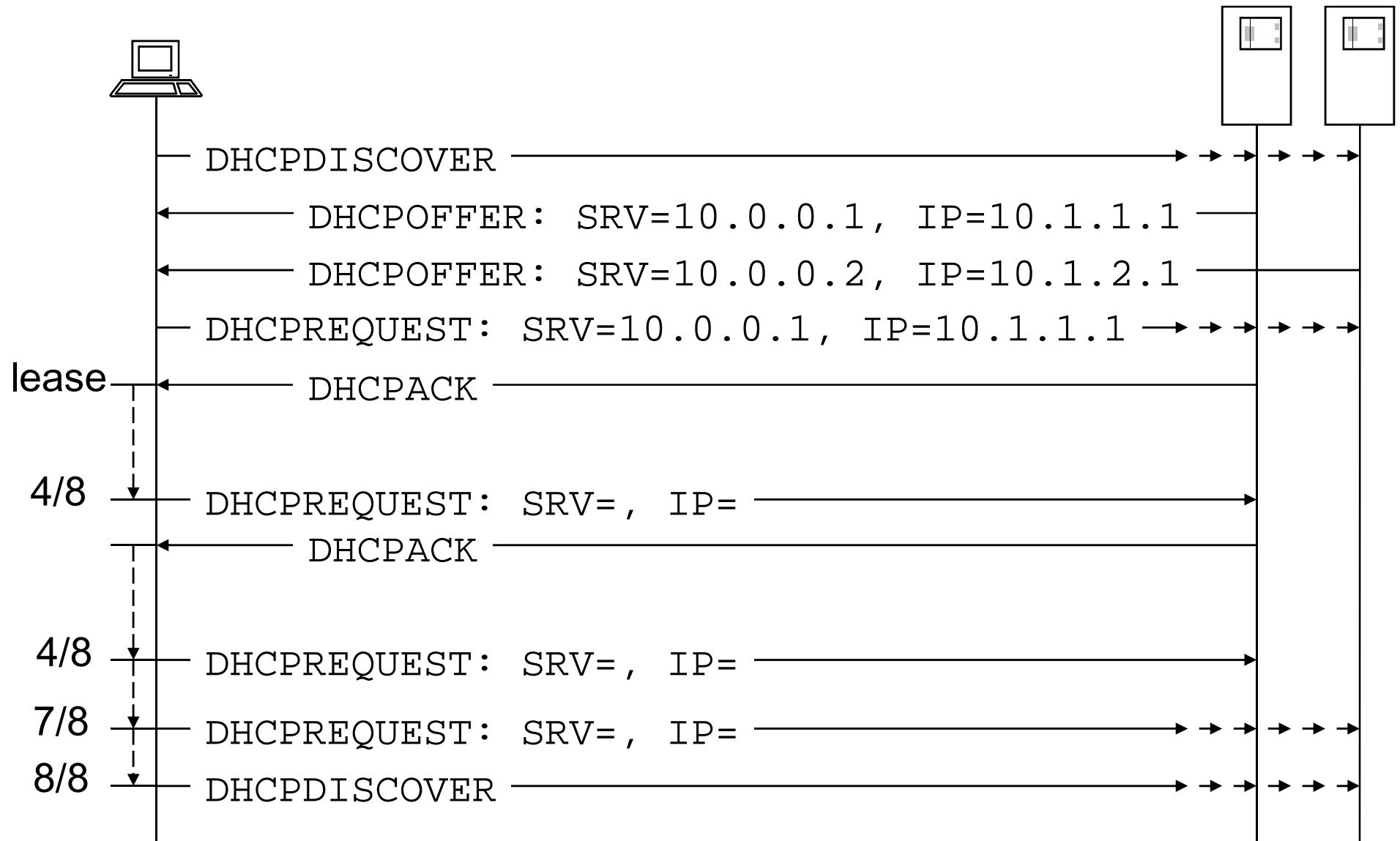
Network Time Protocol

- Synchronizace času mezi uzly sítě
 - stejné timestamps souborů
 - porovnávání času událostí na různých počítačích
- Aktuální verze 4, RFC 5905, port 123 (UDP)
- Klient kontaktuje servery uvedené v konfiguraci
- Zdroje mají kvůli přesnosti a prevenci cyklů klasifikaci:
 - přesné zařízení, stratum 0: např. atomové hodiny
 - server stratum N : řízený podle zdroje stratum $N-1$
- Problém: odpovědi od serverů mají (různé) zpoždění
 - podle časových známek se pro každý spočítá interval, do něhož pravděpodobně spadá jím udaný čas
 - pomocí Marzullova algoritmu se najde nejlepší průnik intervalů

BOOTP a DHCP

- Bootstrap Protocol, RFC 951, byl vyvinut pro automatickou konfiguraci bezdiskových stanic
 - stanice pošle (všem) fyzickou adresu síťové karty
 - server najde klienta v seznamu a pošle IP adresu, jméno...
 - pokud je odděluje router, musí umět BOOTP forwarding
- Nahrazen DHCP (Dynamic Host Configuration Protocol)
 - stejný formát zpráv
 - kromě statické alokace adres i dynamická
 - časově omezený pronájem
 - možnost zapojení více serverů
- IPv4: RFC 2131, UDP porty 67 (server) a 68 (klient)
- IPv6: RFC 3315, UDP porty 546 (server) a 547 (klient)
- Klient si vybírá nabídku (podle adresy, délky pronájmu...)

Průběh DHCP



Prezentační vrstva (OSI 6)

- Představa o všeobecném modelu popisujícím kódování
 - datových typů: celých čísel, řetězců,...
 - datových struktur: polí, záznamů, pointerů,...
- Obecně velmi složité: kdo a kdy (de)kóduje
- Pokus o realizaci: ASN.1
- TCP/IP obecnou potřebu potlačilo, začlenilo definici výměnného formátu přímo do aplikačních protokolů, konverzi musí provádět aplikace
- Praktické problémy:
 - konce řádek: CRLF (0x0D, 0x0A)
 - pořadí bytů: *big endian* (1 = 0x00, 0x00, 0x00, 0x01), např. Intel má *little endian* (1 = 0x01, 0x00, 0x00, 0x00)

Relační vrstva (OSI 5)

- Představa o obecném modelu dialogu
 - jeden dialog může obsahovat více spojení
 - po jednom spojení může probíhat více dialogů
- TCP/IP obecnou potřebu potlačilo, začlenilo princip dialogu přímo do aplikačních protokolů, př.:
 - v rámci jednoho SMTP spojení mezi klientem a serverem může být vyřízeno několik mailů
 - SIP inicializuje dialog za pomoci více parciálních spojení pro přenos audio či video dat

Transportní vrstva (OSI 4)

- Funkce OSI 4:
 - zodpovídá za end-to-end přenos dat
 - zprostředkovává služby sítě aplikačním protokolům, které mají rozdílné požadavky na přenos
 - umožňuje provozování více aplikací (klientů a serverů) na stejném uzlu sítě
 - (volitelně) zabezpečuje spolehlivost přenosu dat
 - (volitelně) segmentuje data pro snazší přenos a opětovně je skládá ve správném pořadí
 - (volitelně) řídí tok dat (*flow control*, „rychlost vysílání“)

Transportní vrstva v TCP/IP

- TCP (Transmission Control Protocol):
 - používá se pro spojované služby
 - klient naváže *spojení*, data tečou ve formě *proudu (streamu)*
 - *spojení* (relaci) řídí a zabezpečuje TCP, nikoliv aplikace
 - TCP je komplikované, má velkou režii
 - příp. méně pravidelné, ale bezeztrátové doručování
- UDP (User Datagram Protocol):
 - používá se pro nespojované služby
 - neexistuje *spojení*, data se posílají jako nezávislé *zprávy*
 - UDP je jednoduché, relaci musí řídit aplikace
 - pravidelný tok, za cenu vyšší ztrátovosti
- Další modifikace či kombinace: SCTP, DCCP, MPTCP

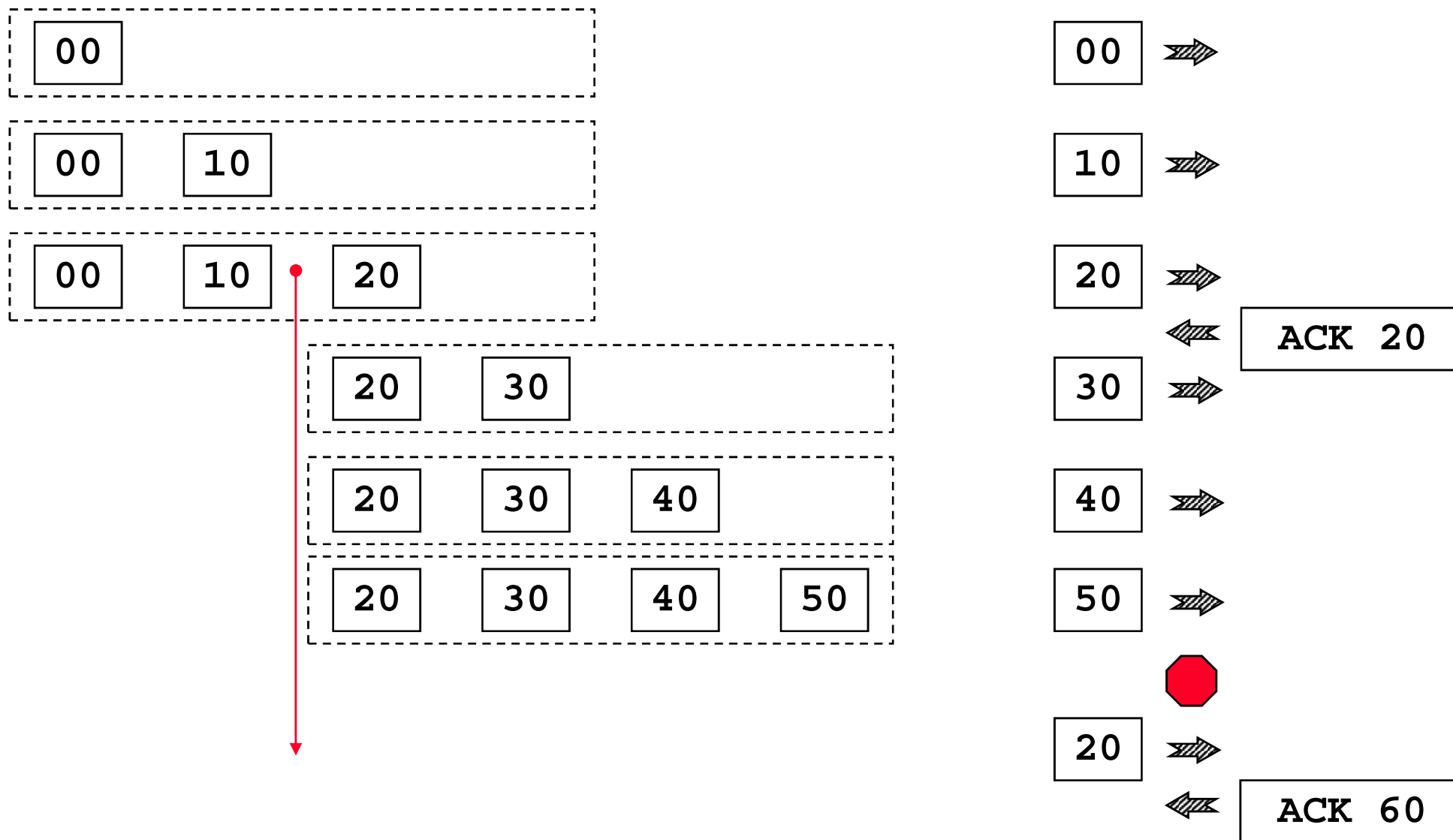
Struktura UDP datagramu

Source Port	Destination Port
Length	Checksum
Data	

Struktura TCP paketu

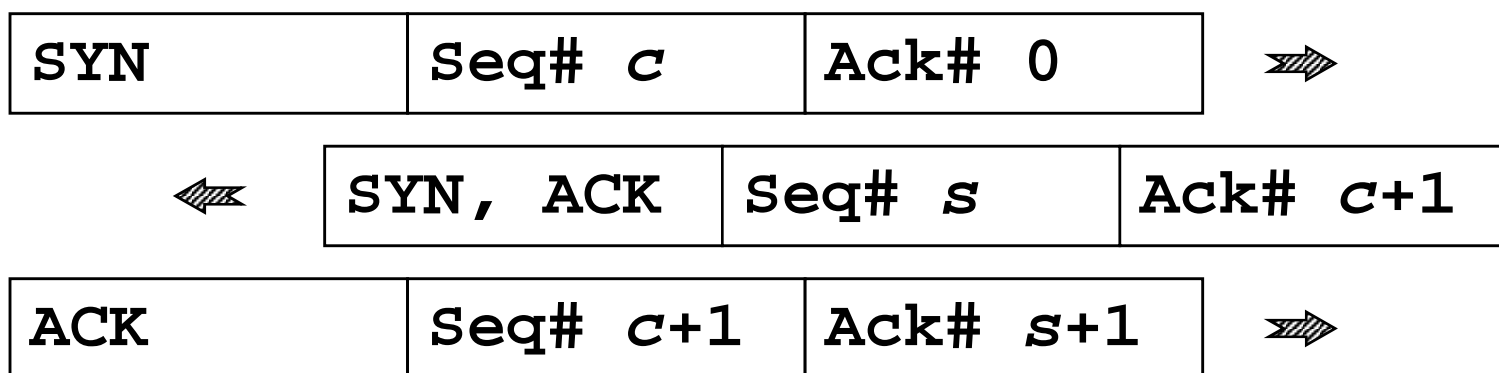
Source Port		Destination Port	
Sequence Number			
Acknowledgement Number			
Data Offset	(rsvd)	Flags	Window
Checksum		Urgent Pointer	
Options			
Data			

TCP okna

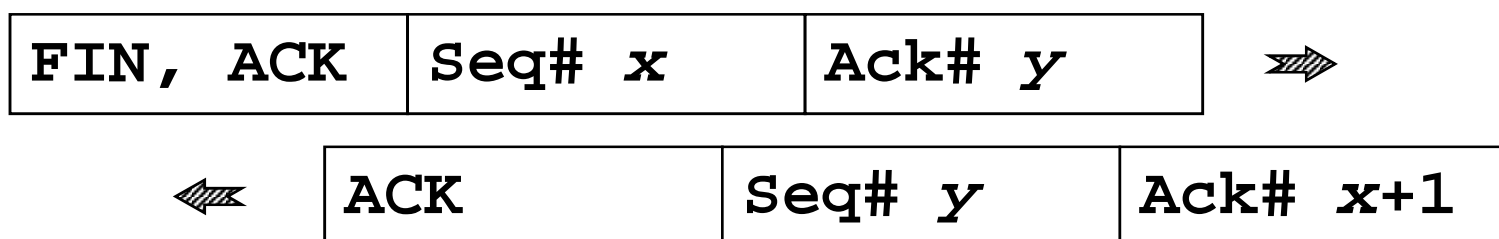


Zahájení a ukončení spojení

- Navázání TCP spojení (three-way handshake):



- Uzavření spojení (jednostranné):



Protistrana (hned nebo později) provede totéž.

TCP příznaky

- **SYN** - paket slouží k synchronizaci čísel segmentů (inicializace „Sequence number“)
- **ACK** - paket potvrzuje doručení všech paketů až po „Acknowledgement number“ (nevčetně); paket může ale nemusí obsahovat i data
- **PSH** - informuje příjemce, že obdržel kompletní blok a má ho předat aplikaci („push“)
- **FIN** - odesílatel uzavírá svoji stranu spojení, nehodlá už posílat žádná data
- **RST** - odesílatel odmítá přijmout spojení resp. oznamuje okamžité přerušení spojení („reset“)
- **URG** - paket obsahuje urgentní (*out-of-band*) data, jejich adresa je v „Urgent pointer“

Výpis programu tcpdump

10.1.1.1.5471 > 1.2.3.4.25: Flags [SYN],
seq 1620916916, win 8192 ←

skutečné SEQ 1.2.3.4.25 > 10.1.1.1.5471: Flags [SYN,ACK],
seq 2525839733, ack 1620916917, win 65535

relativní SEQ 10.1.1.1.5471 > 1.2.3.4.25: Flags [ACK],
ack 1, win 64240 ←

změna window

← 220 alfik.ms.mff.cuni.cz ESMTP Sendmail 8.15.2

1.2.3.4.25 > 10.1.1.1.5471: Flags [PSH,ACK],
seq 1, ack 1, win 65535, length 48

10.1.1.1.5471 > 1.2.3.4.25: Flags [ACK],
ack 49, win 64192, length 0

⇒ HELO betynka

10.1.1.1.5471 > 1.2.3.4.25: Flags [PSH,ACK],
seq 1, ack 49, win 64192, length 14

← 250 alfik Hello betynka, pleased to meet you

1.2.3.4.25 > 10.1.1.1.5471: Flags [PSH,ACK],
seq 49, ack 15, win 65535, length 46

Výpis existujících socketů

```
C:\Users\forst> netstat -an
```

Active Connections

Proto	Local Address	Foreign Address	State
TCP	0.0.0.0:135	0.0.0.0:0	LISTENING
TCP	0.0.0.0:623	0.0.0.0:0	LISTENING
TCP	127.0.0.1:49209	127.0.0.1:49210	ESTABLISHED
TCP	127.0.0.1:49210	127.0.0.1:49209	ESTABLISHED
TCP	192.168.28.73:139	0.0.0.0:0	LISTENING
TCP	192.168.28.73:49167	195.113.19.78:22	ESTABLISHED
TCP	192.168.28.73:49183	195.113.19.78:80	ESTABLISHED
UDP	0.0.0.0:3702	*:*	
UDP	127.0.0.1:1900	*:*	
UDP	192.168.28.73:1900	*:*	

TCP spojení: místní adresa / port vzdálená adresa / port
poslouchající server

Síťová vrstva (OSI 3)

- Hlavní funkce OSI 3: přenos dat předaných transportní vrstvou od zdroje k cíli
- Základem této činnosti jsou
 - adresace* - protokol síťové vrstvy definuje tvar a strukturu adres komunikujících partnerů
 - encapsulation (zapouzdření)* - řídicí data potřebná pro přenos (zjm. adresy) se musí vložit do PDU
 - routing (směrování)* - vyhledání nejvhodnější cesty k cíli přes mezilehlé sítě
 - forwarding (přeposílání)* - předání dat ze vstupního síťového rozhraní na výstupní
 - decapsulation* - vybalení dat a předání transportní vrstvě
- Příklady protokolů: **IPv4**, **IPv6**, IPX, AppleTalk

Internet protokol (IP)

- Vlastnosti:
 - nespojovaná služba (každý datagram běží svou cestou)
 - best effort (nespolehlivá, spolehlivost řeší vyšší vrstvy)
 - nezávislá na médiu (vyšší vrstvy neřeší typ média)
- Adresy:
 - obsahují část s adresou sítě a část s adresou uzlu
 - IPv4: 4 byty, IPv6: 16 bytů
- Přidělování:
 - centrální: IANA (Internet Assigned Numbers Authority), oddělení ICANN
 - regiony: RIR (5x, náš: RIPE NCC)
 - dále: ISP různých úrovní
 - v lokální síti: lokální správa sítě (ručně nebo automaticky)

Struktura IPv4 datagramu

Vers.	Header Length	Service Type (priorita, QoS)	Packet Length	
Fragment Identification			Flags	Fragment Offset
Time-to-live		Protocol	Header Checksum	
Source IP Address				
Destination IP Address				
Options				Padding
Data				

IPv4 adresy

- Původně: jeden byte
- 1975 (RFC 687): tři byty („*This expansion is adequate for any foreseeable ARPA Network growth.*“)
- 1976 (RFC 717): jeden byte (sít') + tři byty (počítač)
- 1981 (RFC 791): třídy A, B a C

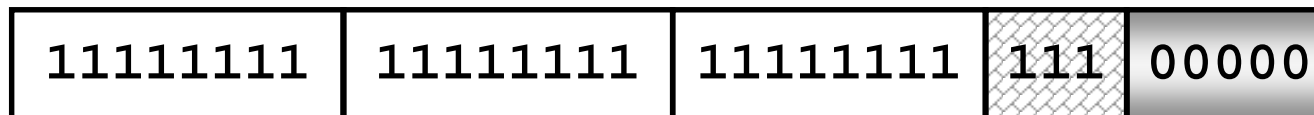
Třída	1. byte	2. byte	3. byte	4. byte	1. byte	Sítí	Adres
A	0	net	host		1-126	126	~16 M
B	10	net		host	128-191	~16 k	~64 k
C	110	net		host	192-223	~2 M	254
D	1110	net			224-239	multicast	
E	1111				240-255	experimental	

Subnetting

- Rozdělení sítě na podsítě rozšířením síťové části adresy:



pomocí specifikace tzv. síťové masky (*netmask*),
v tomto případě 255.255.255.224:



- Nedoporučuje se používat subnet "all-zeros" a "all-ones", takže zde máme jen 6 x 30 adres (70%)
- Je přípustná nespojitá maska, obvykle se nepoužívá
- V současnosti se často ignorují třídy (*classless* mód) a uvádí jen počet bitů prefixu (např. 193.84.56.71/27)
- Pokud se v síti používají různé masky, hovoříme o síti s *variable length subnet mask* (VLSM)
- Posun hranice sítě opačným směrem: *supernetting*

Speciální IPv4 adresy (RFC 5735)

- Speciální adresy „by design“
 - **this host** (smí být použita pouze jako zdrojová): $0.0.0.0/8$
 - adresa rozhraní s dosud nepřřazenou adresou
 - **loopback** (RFC 1122): $127.0.0.1/8$
 - adresa lokálního počítače, umožňuje vytvoření smyčky
 - **adresa sítě**: $\langle \text{adresa sítě} \rangle . \langle \text{samé nuly} \rangle$
 - **network broadcast** (RFC 919): $\langle \text{adresa sítě} \rangle . \langle \text{samé jedničky} \rangle$
 - „všem v dané síti“, normálně se doručí do cílové sítě
 - **limited broadcast** (RFC 919): $255.255.255.255$
 - „všem v této síti“, nesmí opustit síť
- Speciální adresy „by definition“
 - **privátní adresy** (RFC 1918):
 $10.0.0.0/8$, $172.16-31.0.0/16$, $192.168.*.0/24$
 - pro provoz v lokální síti, přiděluje správce, nesmí opustit síť
 - **link-local adresy** (RFC 3927): $169.254.1-254.0/16$
 - pouze pro spojení v rámci segmentu sítě, uzel si ji sám volí

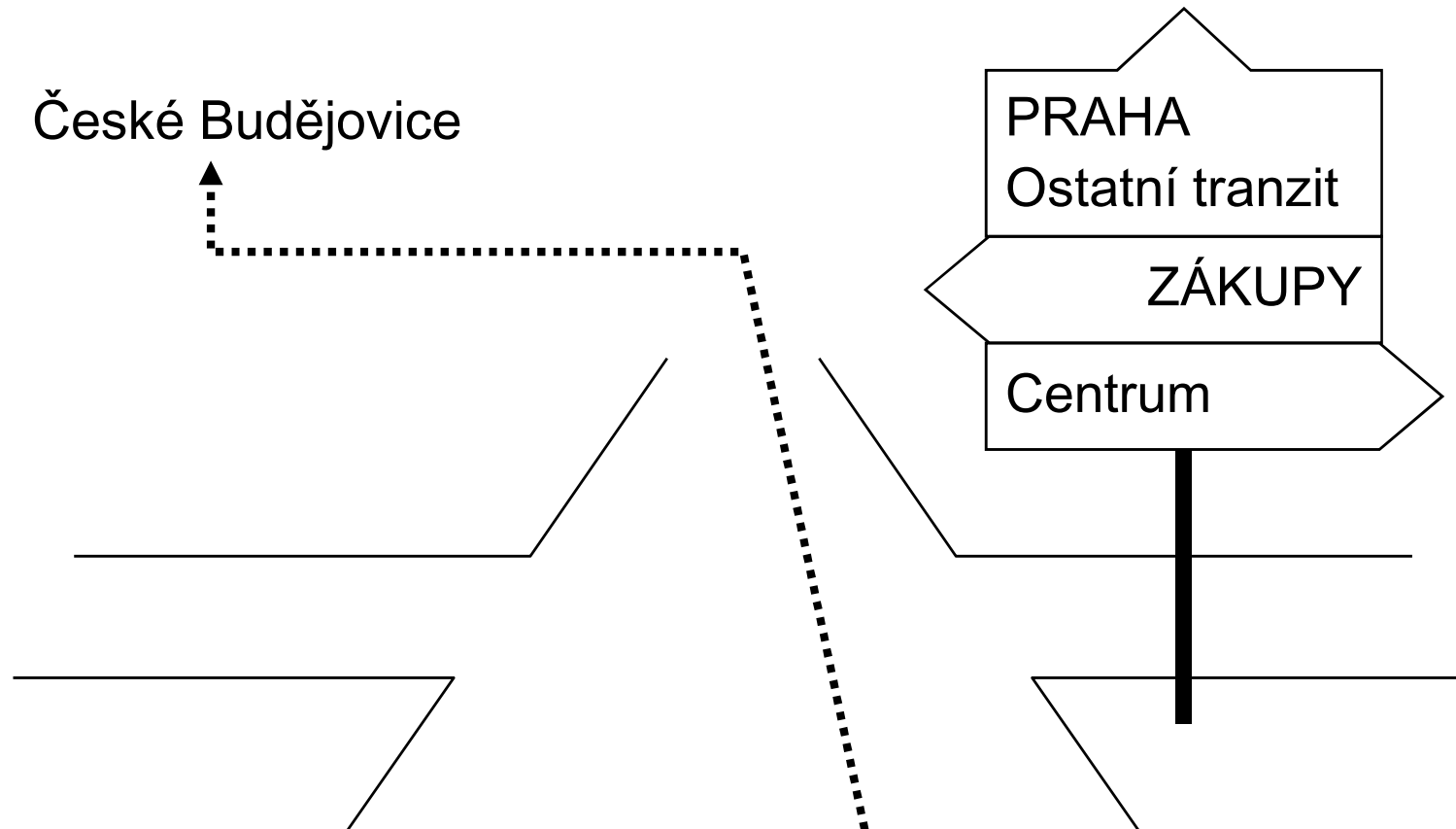
Krize Internetu

- Vyčerpávání adresního prostoru
 - Podstata problému: díky hrubému členění dochází k „plýtvání“
 - Částečné řešení: přidělování bloků adres bez ohledu na třídy (tzv. *classless*), vracení nepoužívaných bloků, privátní adresy + NAT
 - IANA už prostor vyčerpala, APNIC 2011/04, RIPE NCC 2012/09, LACNIC 2014/04, ARIN 2015/09, AFRINIC ?
- Přepřehování směrovacích tabulek
 - Podstata problému: velký počet nesouvisle přidělených bloků rychle plní směrovací tabulky
 - Částečné řešení: realokace adres, CIDR (Classless InterDomain Routing) agregace

IPv6 adresy

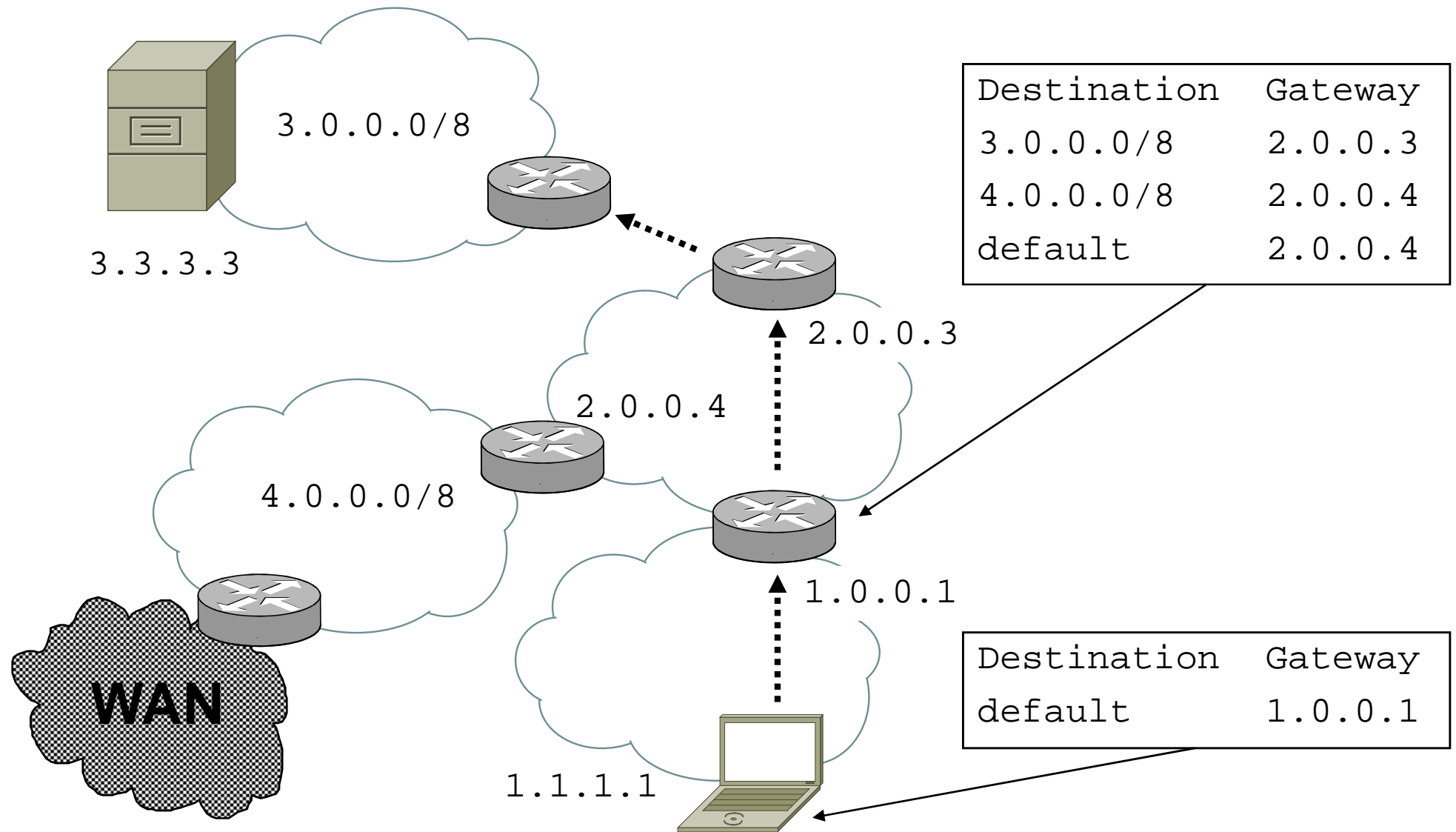
- Dlouhý vývoj, konečná podoba: 128 bitů (16 bytů)
- Zápis: `fec0::1:800:5a12:3456`
- Druhy adres:
 - unicastová - adresa jednoho uzlu; zvláštní adresy (RFC 5156):
 - *Loopback* (`::1/128`)
 - *Link-Scope* (`fe80::/10`), dříve *link-local*
 - *Unique-Local* (`fc00::/7`), dříve *site-local*, obdoba privátních adres v IPv4
 - multicastová (`ff00::/8`) - adresa skupiny uzlů (rozhraní)
 - anycastová - de facto unicastová adresa, přidělená více uzlům; doručení řeší směrování; účel: distribuce serverů po světě
 - chybějí broadcastové
- Přechod z IPv4 usnadňují různé varianty tunelování IPv4 a IPv6

Směrování (silnice)

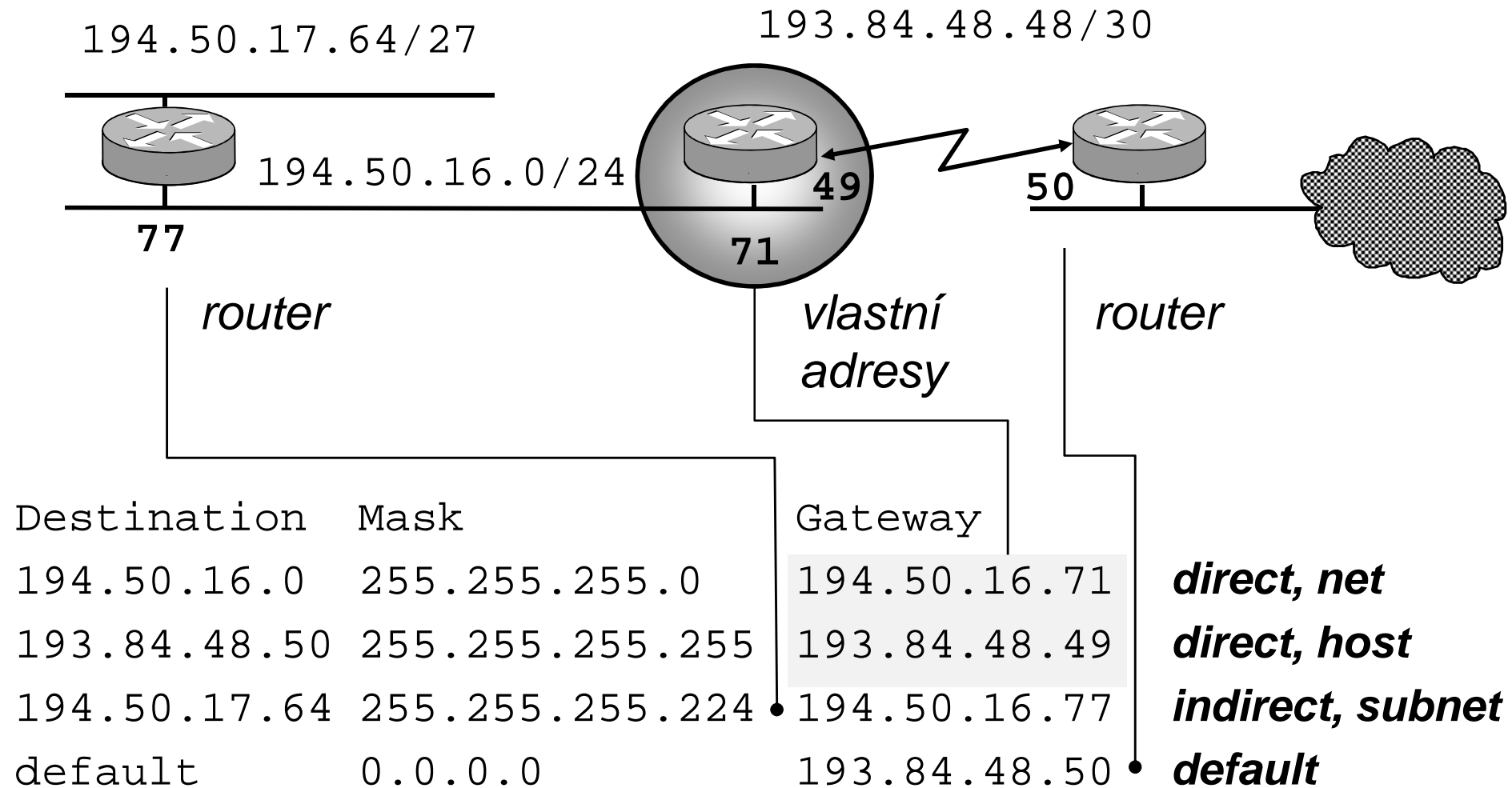


- Na každé křižovatce se rozhodujeme podle směrovek
- Ke správné interpretaci potřebujeme lingvistickou a geografickou znalost

Směrování (sít')



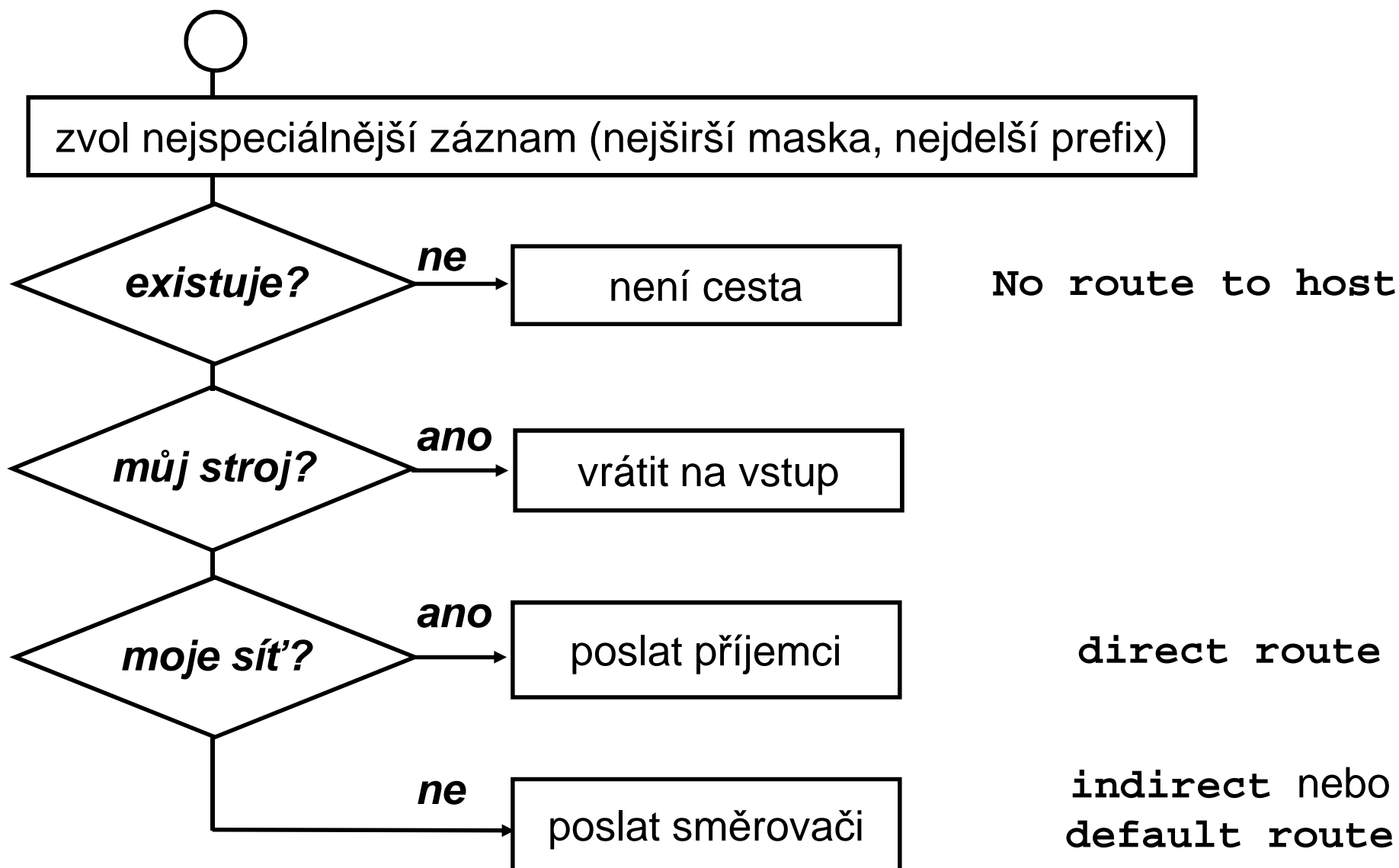
Příklad směrovací tabulky



Principy směrování

- Směrování by měla umět každá stanice v TCP/IP síti
- Záznam ve směrovací tabulce obsahuje „sloupce“:
cíl, maska, gateway
- Maska vyjadřuje „uvažovanou část“ adresy cíle
- Dřívější členění cílů: host (/32), net, default (/0)
- Typy záznamů:
 - *direct* (přímo připojená síť, “gateway” je vlastní adresa)
 - *indirect, default*
- Vznik záznamu:
 - *implicitní* (automaticky po přiřazení adresy rozhraní)
 - *explicitní* („ručně“ zadán příkazem)
 - *dynamický* (v průběhu práce od partnerů v síti)

Směrovací algoritmus



Konfigurace sítě

UNIX

- IP adresa: `ifconfig interface IP_adr [netmask maska]`
- defaultní router: `route add default router`
- DHCP: `dhclient interface`
- často uložené v konfiguračním souboru, liší se podle typu OS

Windows

Control Panel ⇒ Network and Internet
⇒ Network Connections
⇒ Local Area Connection ⇒ Properties
⇒ TCP/IPv4 ⇒ Properties
⇒ General

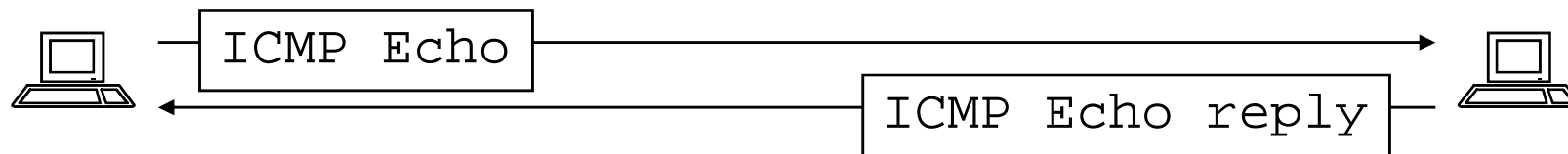
Internet Control Message Protocol

- ICMP slouží pro posílání řídicích informací pro IP:
 - Echo, Echo Reply** ... testování dosažitelnosti počítače (**ping**)
 - Destination Unreachable** ... nedostupný stroj, služba, síť, zakázaná fragmentace
 - Time Exceeded** ... vypršel Time-to-live (chyba v routování)
 - Source Quench** ... žádost o snížení rychlosti toku datagramů
 - Router Solicitation, Router Advertisement** ... vyhledávání routerů
 - Redirect** ... výzva ke změně záznamu v routovací tabulce
 - Parameter Problem** ... chyba v záhlaví datagramu
- Používá IP datagramy, ale není to transportní protokol
- ICMPv6 podstatně doplněn a rozšířen (např. o zprávy pseudoprotokolu Neighbor Discovery Protocol)

Ping

- Základní prostředek pro diagnostiku sítě

```
betynka:~> ping alfik
```

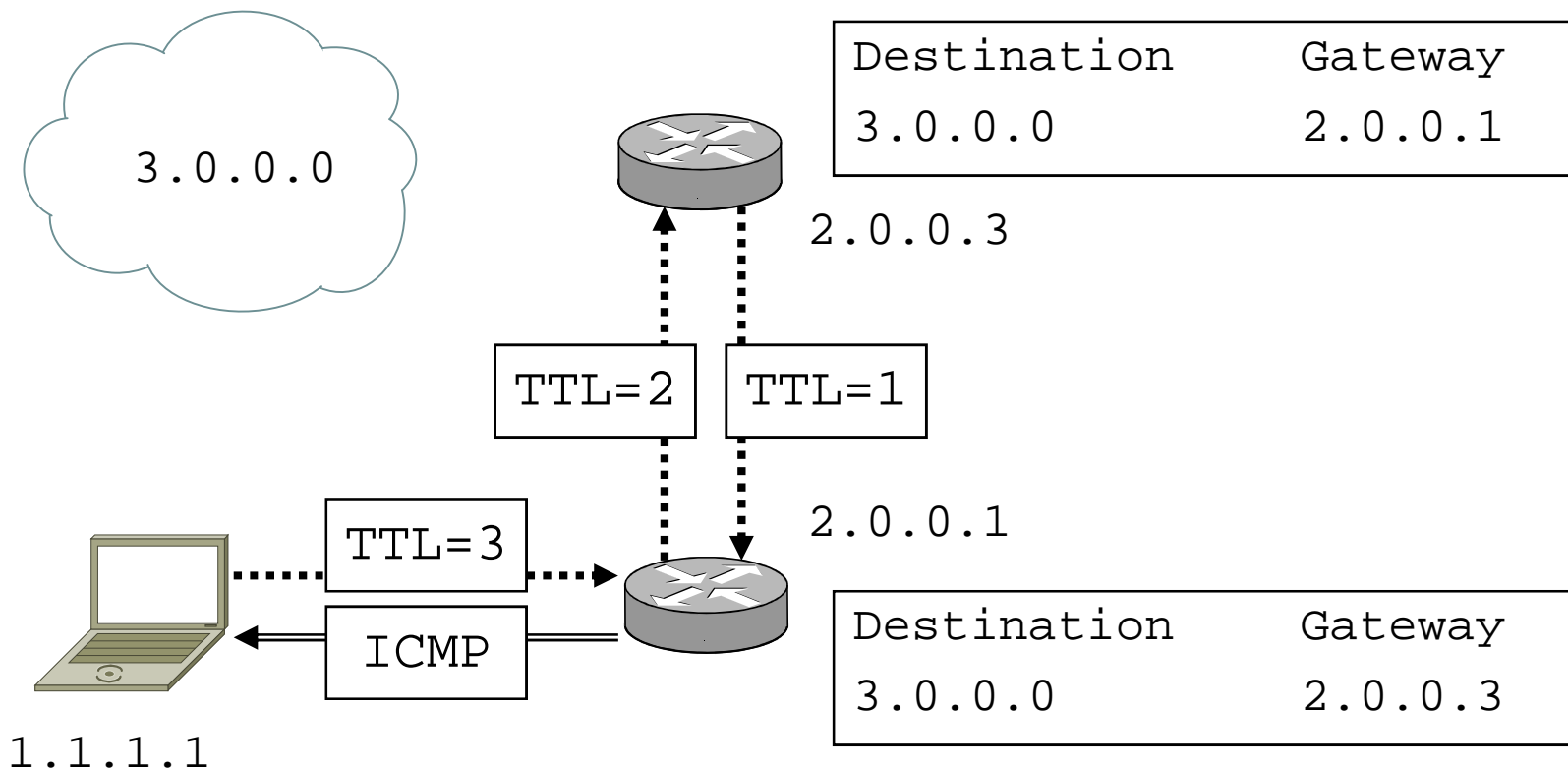


```
PING alfik.ms.mff.cuni.cz (195.113.19.71): 56 data bytes
64 bytes from 195.113.19.71: icmp_seq=0 ttl=64 time=0.214 ms
64 bytes from 195.113.19.71: icmp_seq=1 ttl=64 time=0.323 ms
64 bytes from 195.113.19.71: icmp_seq=2 ttl=64 time=0.334 ms
^C
--- alfik.ms.mff.cuni.cz ping statistics ---
3 packets transmitted, 3 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 0.214/0.290/0.334/0.054 ms
```

- na cílovém uzlu nemusí běžet žádný speciální program
- nezaručuje dostupnost služeb (pouze síťové vrstvy)

Time To Live (IP)

- Prostředek pro ochranu před zacyklením v případě routovací smyčky (chybné konfigurace routerů)
- Udává počet hopů, které smí paket ještě přeskočit
- Při dosažení 0 se posílá ICMP Time Exceeded



Diagnostika směrování

- Výpis směrovací tabulky: `netstat -r[n]`
příp.: `route print`

Destination	Gateway	Flags	Ipkts	...	Colls	Interface
194.50.16.0	this	U	15943	...	0	tu0
127.0.0.1	loopback	UH		...		lo0
default	gw	UG		...		tu0
193.84.57.0	gate	UGD		...		tu0

- Kontrola cesty: `tracert`, `tracert`

```
1 gw.thisdomain (194.50.16.222)  2 ms  1 ms  1 ms
2 gw.otherdomain (193.84.48.49) 12 ms 15 ms 15 ms
3 * * *
```

Statické řízení směrovacích tabulek

Cesty se nastavují při startu podle konfigurace

- nepružné při změnách
 - problémy se subnettingem
 - nesnadné zálohování spojení
 - + méně citlivé na problémy v síti
 - + dostupné i ve zcela heterogenním prostředí
- ⇒ vhodné pro jednodušší, stabilní sítě

```
route {  $\left\{ \begin{array}{l} \text{add} \\ \text{delete} \\ \text{flush} \end{array} \right\} \left| \begin{array}{l} -f \end{array} \right. \left\{ \begin{array}{l} [[-]\text{host}] \text{ host} \\ [[-]\text{net}] \text{ net } [[-]\text{netmask}] \text{ mask} \\ \text{default} \mid 0 \end{array} \right\}$ 
```

$\left. \begin{array}{l} \text{[gw]} \left\{ \begin{array}{l} \text{router} \\ \text{interface } [-\text{interface}] \end{array} \right\} \text{ [metric]} \end{array} \right\}$

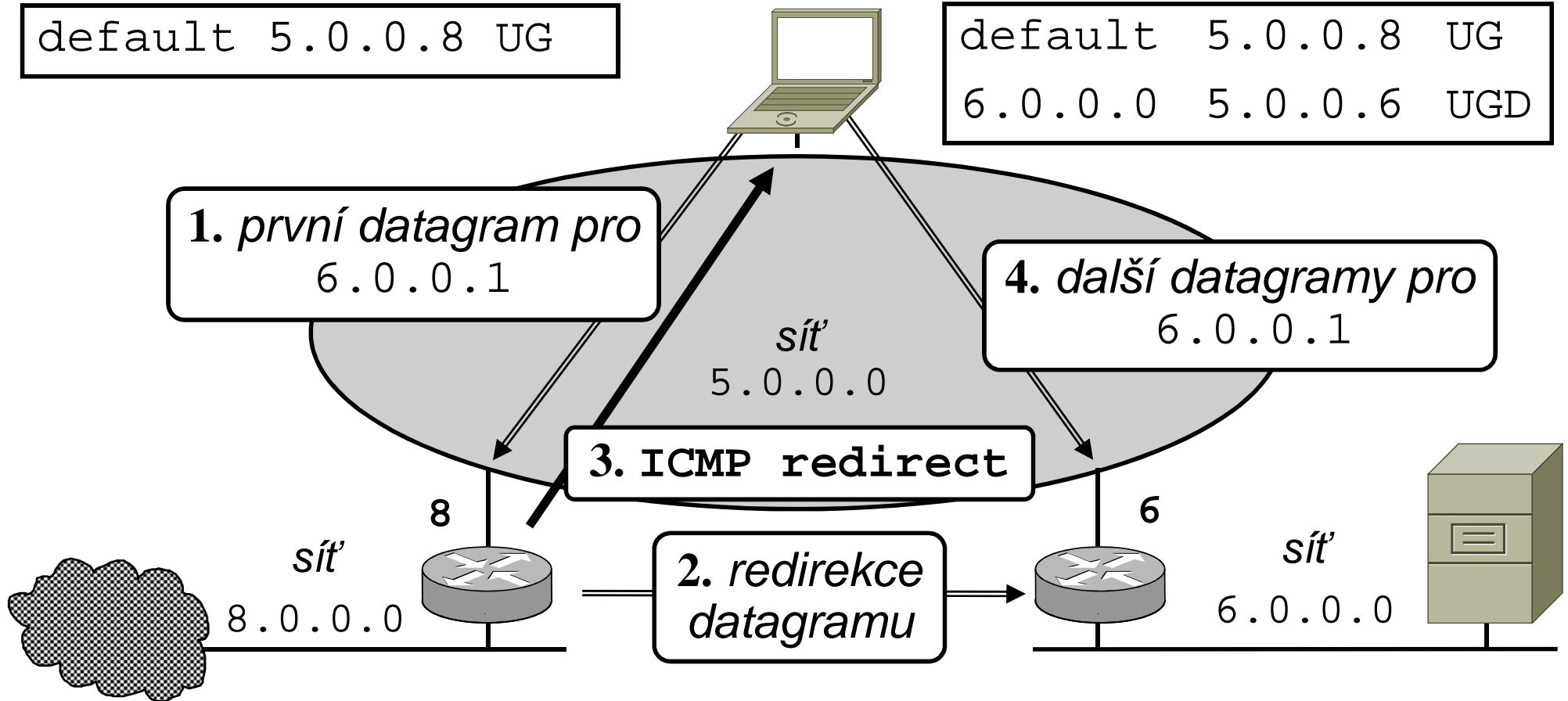
Redirekce

původní obsah tabulky:

```
default 5.0.0.8 UG
```

nový obsah tabulky:

```
default 5.0.0.8 UG  
6.0.0.0 5.0.0.6 UGD
```



Dynamické řízení směrovacích tabulek

Routery si navzájem vyměňují informace o síti pomocí *routovacího protokolu*, stanice se jím mohou řídit také, ale v režimu read-only

- + jednoduché změny konfigurace
 - + síť se dokáže sama „opravovat“
 - + směrovací tabulky se udržují automaticky
 - citlivější na problémy příp. útoky
-
- na počítači musí běžet program obsluhující protokol
 - př. routed, gated, BIRD (vyvinutý na MFF),...
 - pro lokální sítě (*interní routery*) se používají nejčastěji protokoly RIP a OSPF

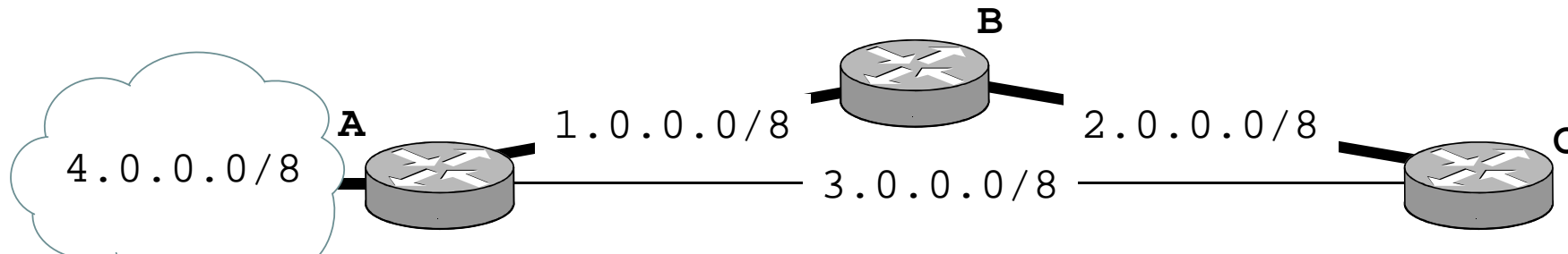
Distance vector protokoly

- Základní myšlenka:
 - uzel má u záznamů ve směrovací tabulce i „vzdálenosti“
 - svou tabulku periodicky posílá sousedům, ti si upraví svoji tabulku a v dalším taktu ji posílají dál
- Výhody:
 - jednoduché, snadno implementovatelné
- Nevýhody:
 - pomalá reakce na chyby
 - metrika špatně zohledňuje vlastnosti linek (rychlost, spolehlivost, cenu...)
 - omezený rozsah sítě
 - chyba ve výpočtu jednoho routeru ovlivňuje celou síť (možnost vzniku routovacích smyček)

Routing Information Protocol

- Nejstarší směrovací protokol, RFC 1058
- Vlastnosti:
 - metrikou je počet routerů v cestě (*hop count*)
 - rozsah sítě je omezen na 15 hopů, 16 je „nekonečno“
 - pro výpočet nejkratších cest používá Bellman-Fordův algoritmus
- Aktuálně verze 2, RFC 2453
 - používá UDP port 520, multicast adresu 224.0.0.9
 - umí subnetting vč. VLSM
 - obsahuje mechanismy na urychlení detekce chyb (triggered updates, split horizon, poison reverse)
- Dostupný na nejrůznějších systémech
- Nepoužitelný pro velké, složité nebo dynamické sítě

Metrika a kvalita linek



1../8	-	1
3../8	-	3
4../8	-	1

A rozesílá update:

1../8	-	1
2../8	-	1

2../8	-	1
3../8	-	3

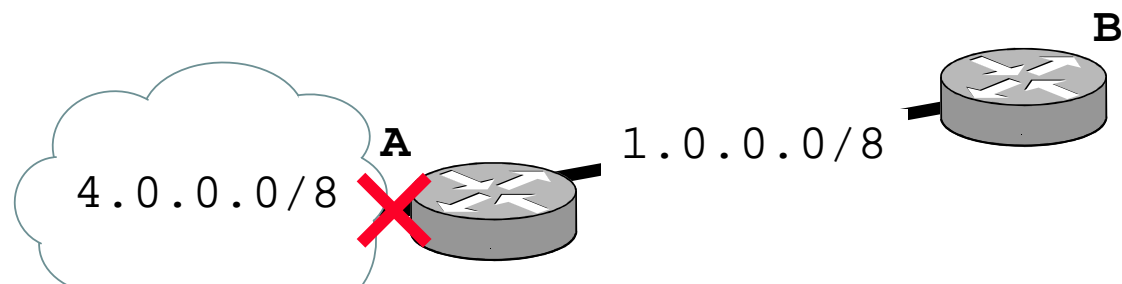
1../8	-	1
2../8	-	1
3../8	A	3+1
4../8	A	1+1

1../8	A	1+3
2../8	-	1
3../8	-	3
4../8	A	1+3

B rozesílá update:

1../8	B	1+1
2../8	-	1
3../8	-	3
4../8	B	2+1

Counting to infinity



1.. /8	-	1
2.. /8	B	2
3.. /8	-	3
4.. /8	-	1

1.. /8	-	1
2.. /8	-	1
3.. /8	A	4
4.. /8	A	2

Výpadek linky A/4:

4.. /8	-	16
--------	---	----

B rozesílá update:

4.. /8	B	2+1
--------	---	-----

A rozesílá update:

4.. /8	A	3+1
--------	---	-----

...

Stav po 14x30sec:

4.. /8	-	16
--------	---	----

4.. /8	-	16
--------	---	----

Link state protokoly

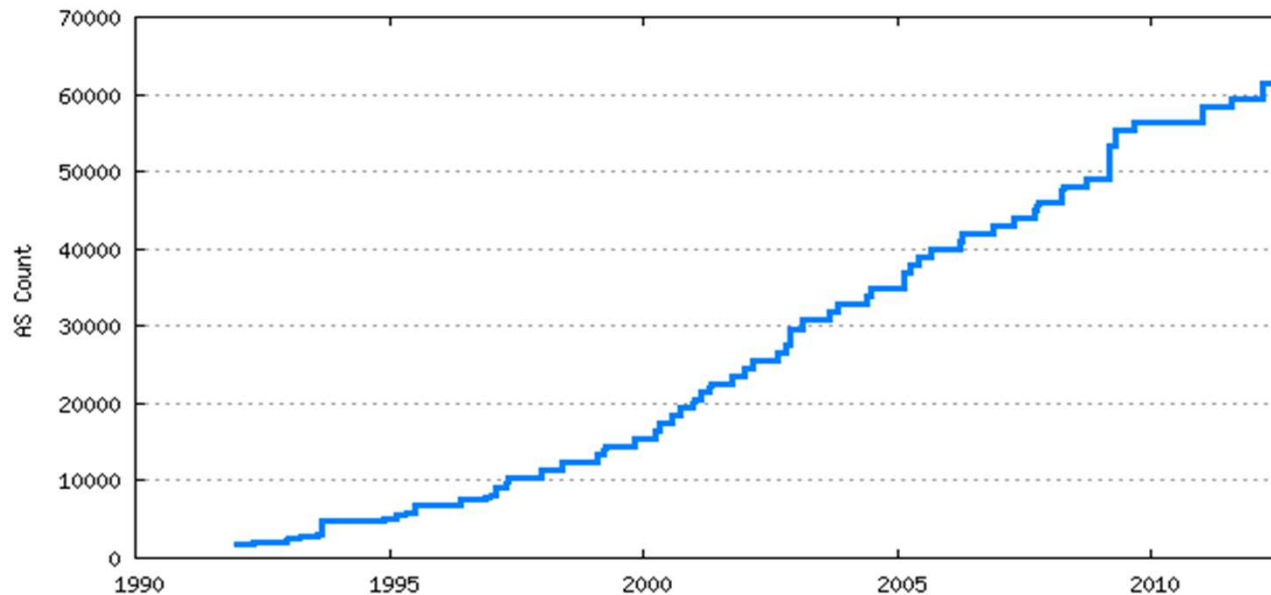
- Základní myšlenka:
 - každý router zná „mapu“ celé sítě
 - routery si navzájem sdělují stav svých linek a podle toho si každý modifikuje svoji mapu sítě
- Nevýhody:
 - výpočet mapy je náročnější na výkon CPU i na paměť
 - při startu a na nestabilních sítích může výměna dat znamenat významnou zátěž sítě
- Výhody:
 - pružná reakce na změny topologie
 - každý si počítá sám za sebe, chyba neovlivní ostatní
 - síť je možné rozdělit na menší podsítě (rychlost výpočtu!)
 - výměna dat probíhá pouze při změnách

Open Shortest Path First

- Nejrozšířenější link-state interní routovací protokol
- Vlastnosti:
 - používá Dijkstrův algoritmus nalezení nejkratší cesty
 - používá hierarchický model sítě:
 - oblast (area) 0 tvoří páteř
 - ostatní oblasti se připojují pouze na páteř
 - každý router zná mapu své oblasti a cestu k páteři
 - metriku je možné konfigurovat, implicitně je to *path cost*, součet „cen“ na cestě, kde cena je dána šířkou pásma
- Používá samostatný protokol transportní vrstvy 89 a multicast adresy 224.0.0.5 a 224.0.0.6
- Aktuální je verze 2 pro IPv4 (RFC 2328) a revize pro IPv6 označovaná jako verze 3 (RFC 5340)

Autonomní systémy

- Definice: blok sítí se společnou routovací politikou
- Zavedeny v r. 1982: snazší routování na globální úrovni, nasazení *externích routovacích protokolů* (EGP)
- Jako EGP se dnes používá Border Gateway Protocol (BGP)
- Identifikátor: 16bitové číslo, dnes přechod na 32bitová
- V ČR: na počátku 2, nyní stovky



IP filtrování

- Router na perimetru (intranet/internet) má v konfiguraci uvedeno, jaký provoz je povolen a za jakých podmínek
- Přísná konfigurace: ven vybrané, dovnitř nic
 - dobré pro protokoly s jedním kanálem (HTTP, SMTP)
 - problém u protokolů s více kanály (FTP, SIP)
- Obvyklá konfigurace: ven cokoliv, dovnitř nic
 - naráží např. u FTP s aktivním přenosem
 - nepoužitelné u protokolů s mnoha kanály (SIP)
- Lépe se dá řešit nastavením aplikací a SW na routeru, který musí částečně rozumět aplikační vrstvě
- Problém se službami „uvnitř“ (např. www server, pošta)
 - povolení výjimek je riskantní
 - lepší je oddělený segment, DMZ, demilitarizovaná zóna

Proxy server

- Transparentní varianta:
 - SW na **routeru** zachytí spojení, uloží požadavek, naváže „svým jménem“ spojení na server a požadavek odešle.
 - Odpověď přijde zpět na router, ten ji uloží (pro další klienty) a zároveň odešle původnímu žadateli.
 - Není třeba konfigurovat na klientovi.
- Netransparentní varianta:
 - Klienty je třeba **nakonfigurovat**, aby se požadavky neposílaly přímo, ale proxy-serveru v lokální síti (lze i automaticky po síti).
 - Proxy server nemusí být nutně router.
 - Je nutná podpora v protokolu.
- Významný bezpečnostní a výkonnostní prvek:
 - umožňuje správě sítě efektivně kontrolovat činnost klientů
 - umožňuje omezit objem provozu na přípojně lince

Address Resolution Protocol

- Konverze MAC (např. Ethernet) a síťových (např. IP) adres
- Neznámé adresy se zjišťují broadcastovou výzvou:

Ethernet=1	IP=0x0800		ARPreq=1
Sender MAC			Sender IP
FF:FF:FF:FF:FF:FF			Target IP

- Výsledky se ukládají na stanici do *ARP cache*
- Unicastová odpověď (odpovídající si nejprve musí přidat informace o tazateli do svojí ARP tabulky)
- Neexistuje metoda, jak ověřit správnost odpovědi
- Gratuitous ARP: nevyžádané ARP (rychlejší změny, riziko)
- Výpis ARP tabulky: **arp -a**
- Omezení na linkový segment, mezi sítěmi je v činnosti OSI 3

Proxy ARP

1. **host A** posílá broadcastem **ARP request** s IP adresou **B**

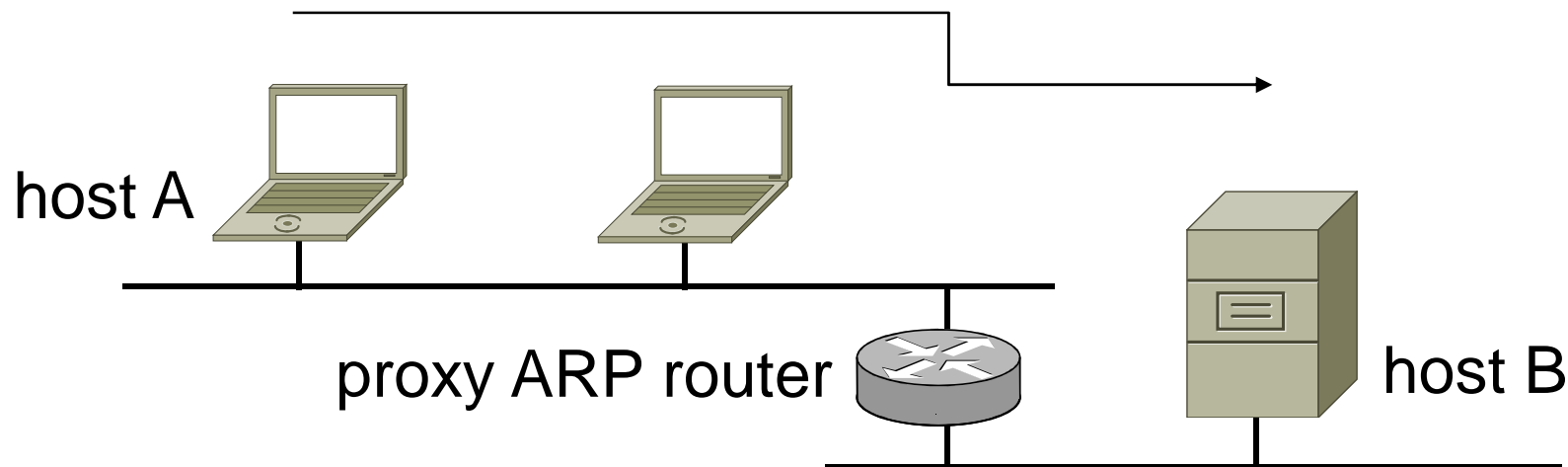


2. **router** pozná, že dotaz nebude zodpovězen, proto sám posílá **ARP reply** s MAC adresou **routeru**



3. MAC **routeru** přiřazena k IP adrese **B** v ARP cache na **A**

4. **host A** posílá data pro **B** s MAC adresou **routeru**



Linková vrstva (OSI 2)

- Dělí se na dvě podvrstvy:
 - Logical Link Control (LLC) umožňuje různým protokolům síťové vrstvy přístup ke stejnému médiu (multiplexing)
 - Media Access Control (MAC) řídí adresaci uzlů a přístup k médiu: kdo, kdy a jak může data odesílat a jak je přijímat
- TCP/IP už se touto vrstvou („síťového rozhraní“) nezabývá
- Síťový segment (fyzická síť):
 - množina uzlů sdílející stejné médium
- PDU na linkové vrstvě: rámec (frame)
 - liší se podle použitého média
 - obecně obsahuje: synchronizační pole, hlavičku (adresy, typ, příp. řídicí data), datové pole a patičku (Frame Check Sequence - detekce chyb)

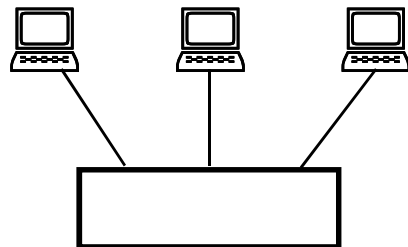
Typy fyzických topologií

Multipoint

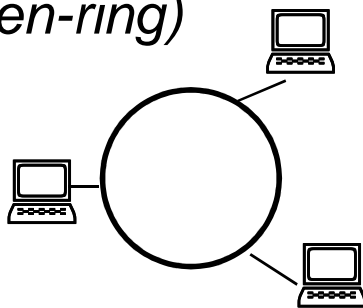
Sběrnice (např. Ethernet)



Hvězda (např. ATM)

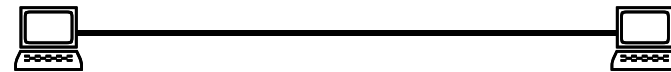


Kruh (např. FDDI, Token-ring)

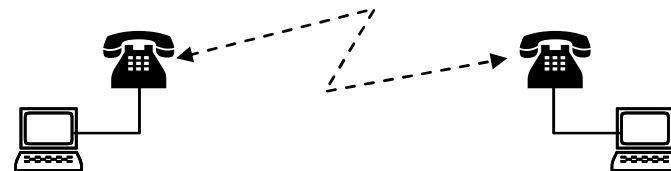


Point-to-point

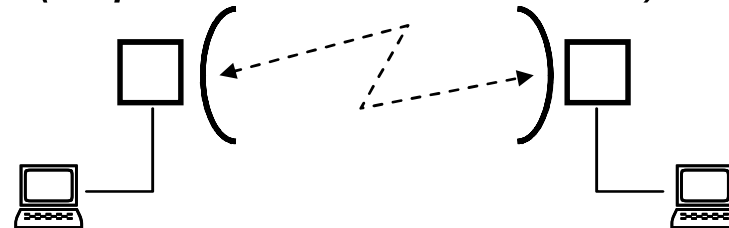
Přímé propojení kabelem (např. sériový, koaxiální, UTP, optický)



Propojení přes modemy



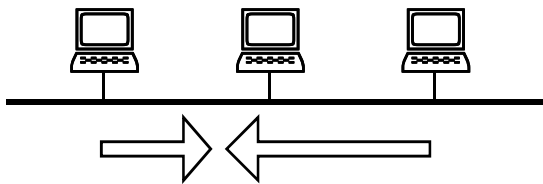
Bezdrátové propojení (např. laser, radioreléové)



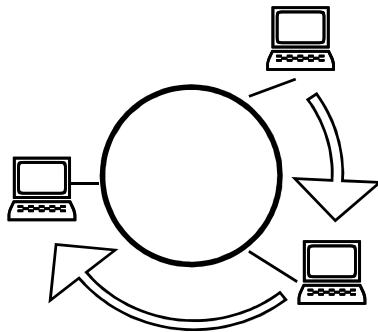
Typy přístupu k médiu

Multipoint

Nedeterministický - kolize

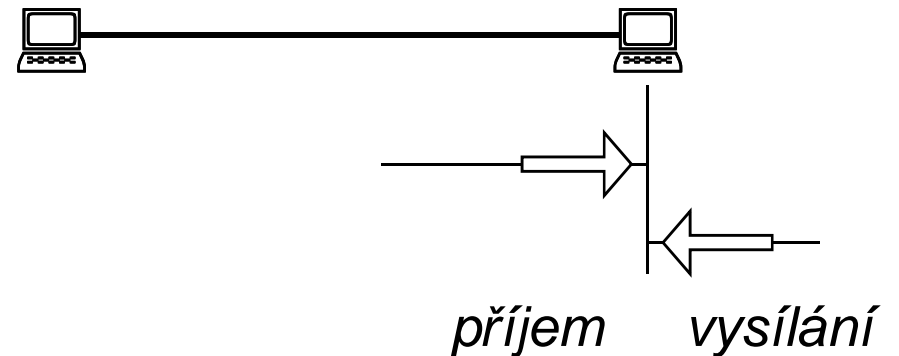


Deterministický - režie

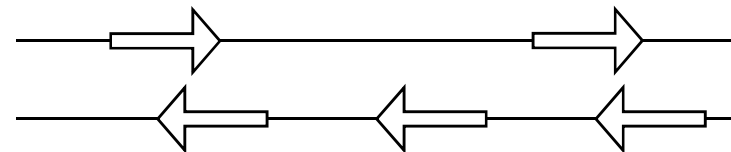


Point-to-point

Half duplex



Full duplex (nejsou kolize)



Řešení kolizí

- CSMA (Carrier Sense with Multiple Access)
 - uzel poslouchá „nosnou“, a pokud není volno, čeká
- CSMA/CD (Collision Detection), např. Ethernet
 - během vysílání uzel současně detekuje případnou kolizi
 - při kolizi stanice zastaví vysílání, upozorní ostatní, počká určitou (náhodnou!) dobu a pokus opakuje, obvykle se postupně prodlužuje interval čekání (*exponenciální čekání*)
 - podmínka: doba vysílání rámce > doba šíření po segmentu (*kolizní okénko*); limituje max. délku segmentu a min. velikost rámce
- CSMA/CA (Collision Avoidance), např. WiFi
 - když je volná nosná, vysílá se celý rámec a čeká se na ACK
 - pokud není volná nosná nebo nedorazí ACK, zahájí se exponenciální čekání

Ethernet

- Historie:
 - první pokusy o realizaci LAN ve firmě Xerox
 - standardizaci převzalo IEEE (únor 1980 → IEEE 802)
 - dva nejběžnější formáty Ethernet II, IEEE 802.3
- Momentálně vůdčí technologie pro lokální sítě
 - dokáže pružně reagovat na progresivní vývoj HW
 - přizpůsobí se širokému spektru přenosových médií
- Na multipoint spojích řízení přístupu metodou CSMA/CD
 - při detekci kolize uzel vysílá „jam signal“
 - exponenciální čekání končí po 16 pokusech chybou
- Adresy:
 - 3 byty prefix (výrobce, multicast...), 3 byty adresa
 - dříve „vypálená“ v kartě, dnes nastavitelná

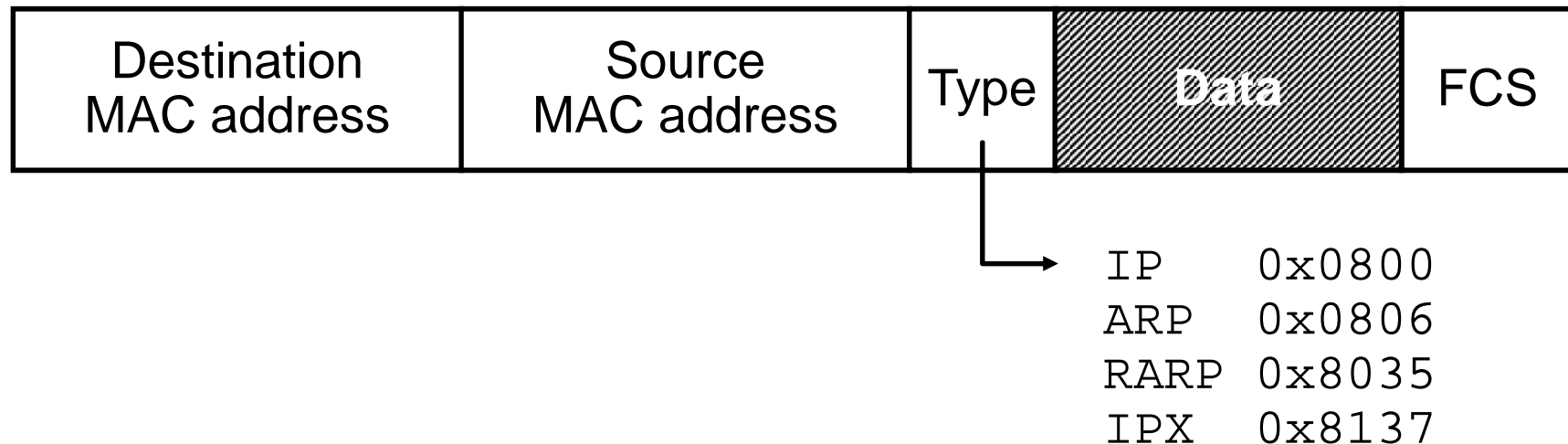
Standardy IEEE 802.3

Standard	Rok	Označení	Rychlost	Médium
802.3	1983	10BASE5	10 Mbit/s	tlustý koaxiální kabel
802.3a	1985	10BASE2	10 Mbit/s	tenký koaxiální kabel
802.3i	1990	10BASE-T	10 Mbit/s	kroucená dvoulinka (UTP)
802.3j	1993	10BASE-F	10 Mbit/s	optický kabel
802.3u	1995	100BASE-TX,FX	100 Mbit/s	UTP nebo optický kabel
802.3z	1998	1000BASE-X	1 Gbit/s	optický kabel
802.3ab	1999	1000BASE-T	1 Gbit/s	kroucená dvoulinka
802.3ae	2003	10GBASE-SR,...	10 Gbit/s	optický kabel
802.3an	2006	10GBASE-T	10 Gbit/s	kroucená dvoulinka
802.3ba	2010	100GBASE-SR	100 Gbit/s	optický kabel

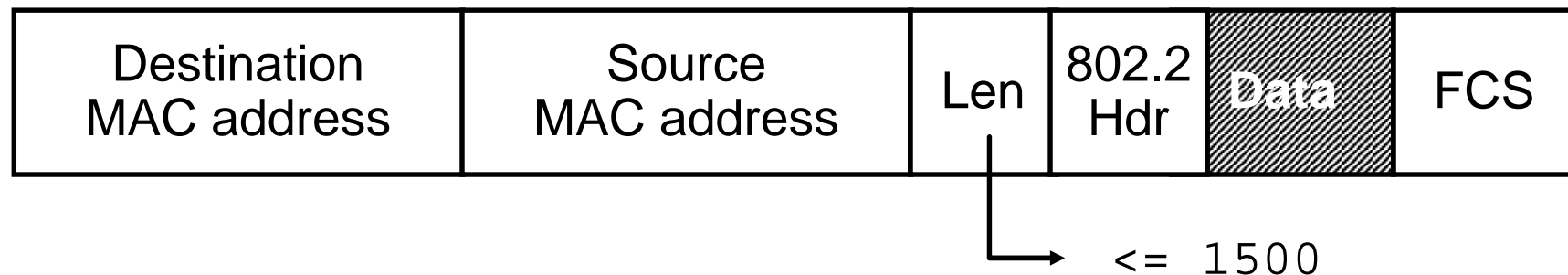
Na rozdíl od RFC jsou normy IEEE vázány licenci.

Struktura ethernetového rámce

Ethernet v2:

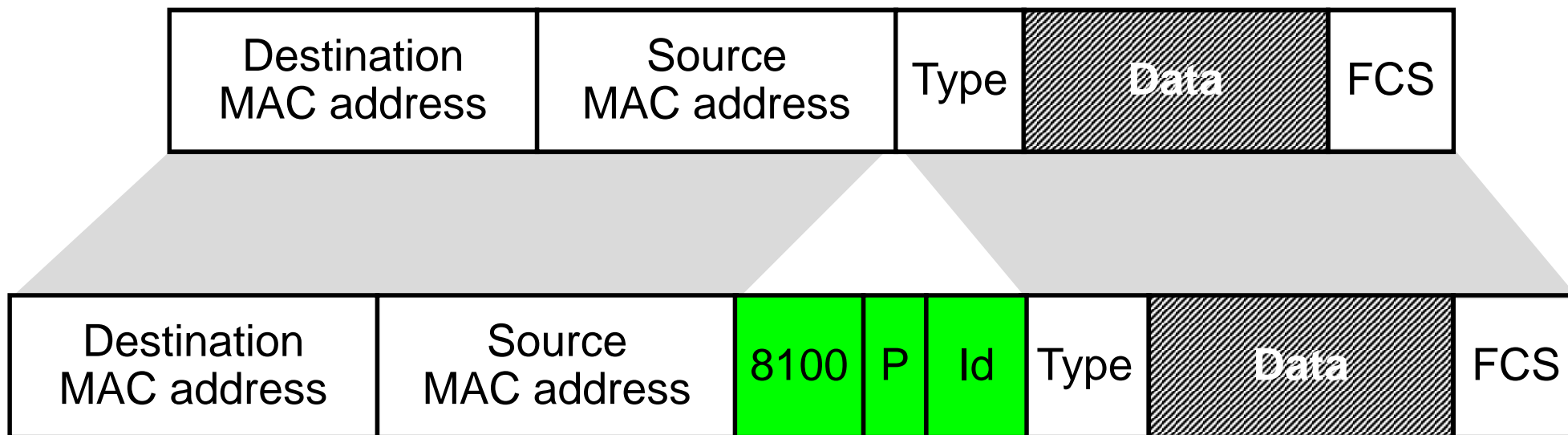


IEEE 802.3



Virtuální sítě (VLAN)

- Prostředek, jak po jedné fyzické síti provozovat více nezávislých lokálních sítí
- Sítě jsou označeny 12bitovým identifikátorem (VLANID)
- Ethernetový rámec se prodlouží o 32 bitů dlouhý tag (tag protocol identifier 0x8100, QoS prioritu a VLANID)
- Tagovat může koncová stanice nebo switch (pro koncovou stanici transparentně)



Cyklický kontrolní součet (CRC)

- CRC (Cyclic Redundancy Check) je hashovací funkce široce používaná pro kontrolu konzistence dat (např. FCS)
- Posloupnost bitů je považována za koeficienty polynomu (ve dvojkové soustavě)

$$\begin{array}{|c|c|c|c|c|} \hline \dots & 1 & 1 & 0 & \dots \\ \hline \end{array} \quad \Rightarrow \quad \dots + 1 \cdot x^{28} + 1 \cdot x^{27} + 0 \cdot x^{26} + \dots$$

- Ten se vydělí tzv. *charakteristickým polynomem* (např. pro CRC-16 je to $x^{16} + x^{15} + x^2 + 1$)
- Zbytek po dělení se převede zpět na bity a použije jako hash
- Jednoduchá implementace (i pomocí HW)
- Velká síla, n -bitový CRC detekuje:
 - na 100% chyby s lichým počtem bitů, chyby kratší než n bitů
 - s vysokou pravděpodobností i delší chyby

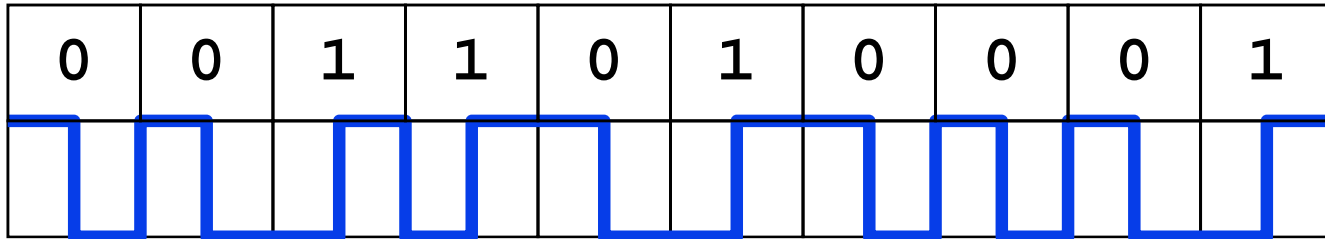
Wi-Fi

- Bezdrátová síť, jiný název: WLAN (wireless LAN)
- Mnoho různých variant pod souhrnným označením IEEE 802.11 (802.11a, b, g, n, y,...):
 - různá pásma (2,4 až 5 GHz)
 - různé rychlosti (2 až 600 Mbps)
- WiFi zařízení dnes prakticky v čemkoliv
- Struktura sítě:
 - ad-hoc peer-to-peer síť
 - infrastruktura přístupových bodů (access pointů)
- SSID (Service Set ID): řetězec (až 32 znaků) pro rozlišení sítí
- Problém: **zabezpečení!**

Fyzická vrstva (OSI 1)

- Funkce vrstvy:
 - přenos dat po konkrétním fyzickém médiu
 - převod digitální informace na analogovou a obráceně
- Různé typy médií
 - metalické: elektrické pulzy
 - optické: světelné pulzy
 - bezdrátové: modulace vln

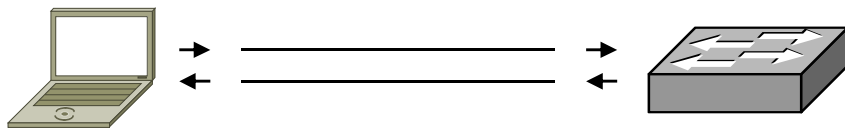
Druhy přenosu dat

- Analogový vs. digitální
 - ve skutečnosti je vše analogové (přenáší se např. proud)
 - digitální: rozhoduje, zda hodnota signálu spadá do nějakého intervalu (menší vliv zkreslení)
 - převody: D→A a zpět *modem* (modulator/demodulator), A→D *codec* (coder/decoder)
- Baseband vs. broadband
 - baseband přenáší přímo signál a kóduje ho, Ethernet používá tzv. Manchester:


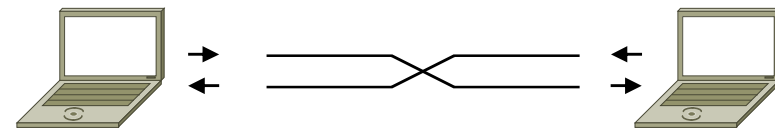
0	0	1	1	0	1	0	0	0	1
---	---	---	---	---	---	---	---	---	---
 - broadband přenáší základní signál a moduluje ho (fázi, amplitudu, frekvenci)

Nestíněná kroucená dvoulinka (UTP)

- Dnes standardní prostředek strukturované kabeláže
- 4 páry Cu vodičů navzájem pravidelně zakroucené
 - zakroucení snižuje vyzařování i příjem elektromagnetického záření (nižší rušení)
- 100Mb Ethernet používá jen dva páry (je možno rozdělit)
- Konektory: RJ 45
- Při propojení je třeba zohlednit povahu zařízení
 - dnes obvykle už autodetekce MDI/MDIX



přímý kabel

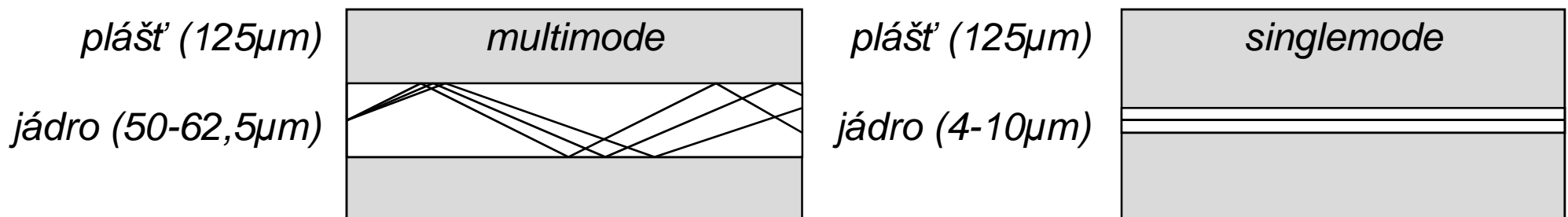


křížený kabel (crossover)

- Alternativa: kabel s kovovým stíněním (STP)

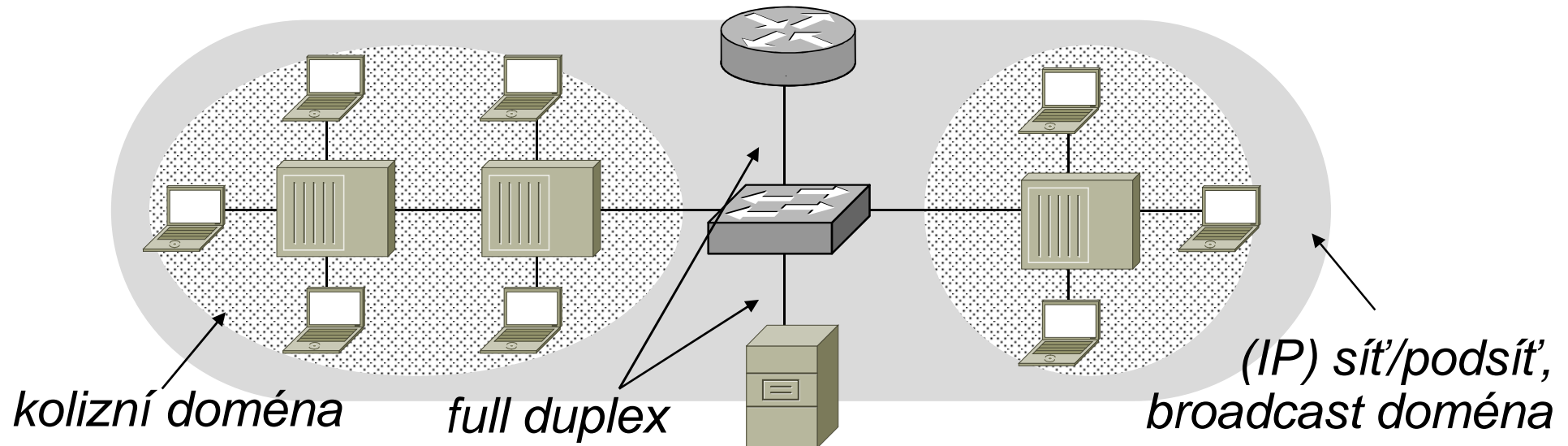
Optická vlákna

- Signál se šíří jako viditelné světlo vláknem z SiO_2
 - vysoké frekvence, velká šířka přenosového pásma
 - nízký útlum, žádné rušení
- Nevýhody:
 - vyšší cena, náročnější manipulace, **nekoukat do kabelu**
- Druhy vláken:
 - jednovidová (singlemode): svítí se laserem => jeden paprsek, větší dosah, šířka pásma („rychlost“, ne rychlost), cena
 - mnohovidová (multimode), svítí se LED

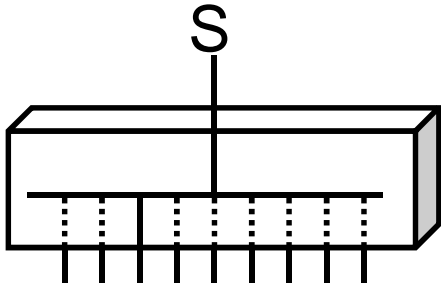


Segmentace sítě

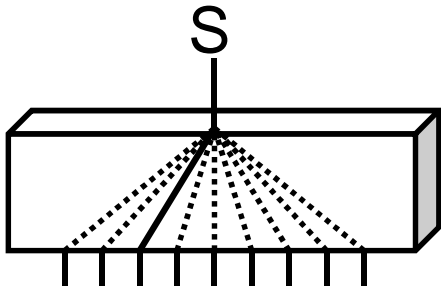
- Repeater (opakovač) spojuje segmenty na fyzické vrstvě
 - řeší: větší dosah (překonává útlum kabelu)
 - neřeší: propustnost (problém kolizí naopak zhoršuje)
 - ve strukturované kabeláži se nazývá *hub*, *rozbočovač*
- Bridge (most) spojuje segmenty na linkové vrstvě
 - řeší: větší propustnost (rozděluje kolizní doménu)
 - ve strukturované kabeláži se nazývá *switch*, *přepínač*



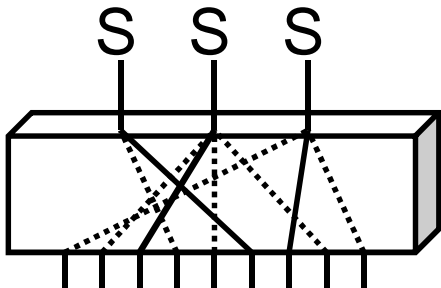
Porovnání hub vs. switch



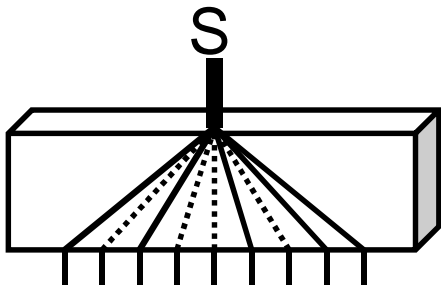
- HUB
 Σ 10 Mbit/s



- Switch
 Σ 10 Mbit/s

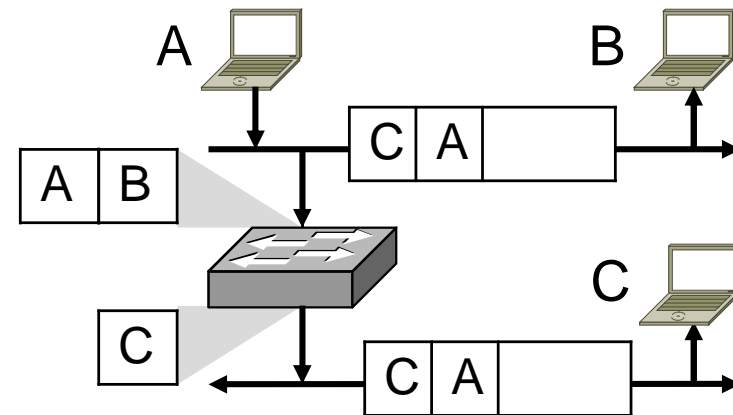
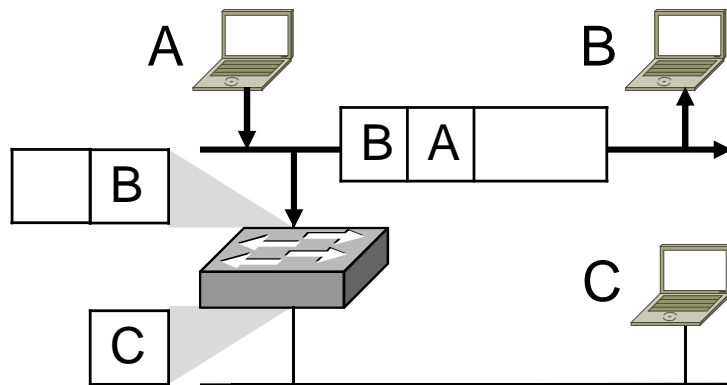
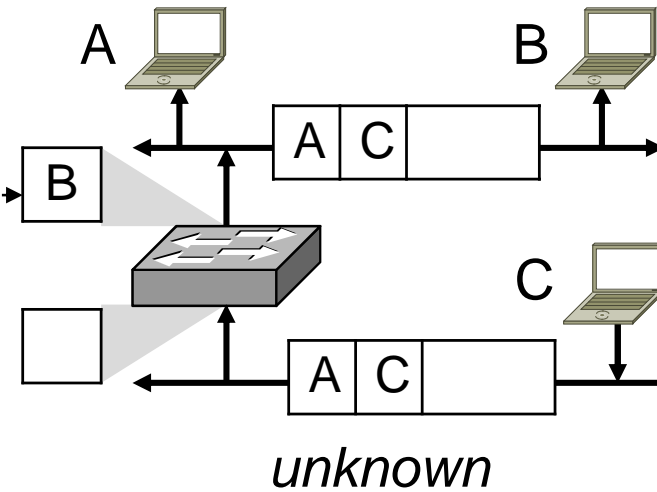
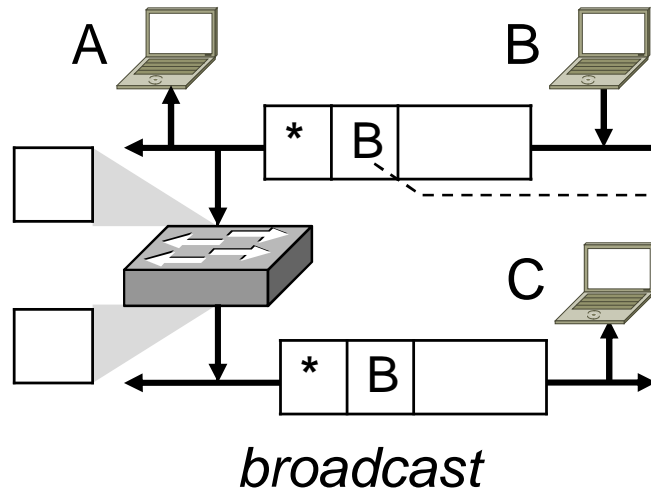


- Switch, více serverů
 $\Sigma > 10$ Mbit/s



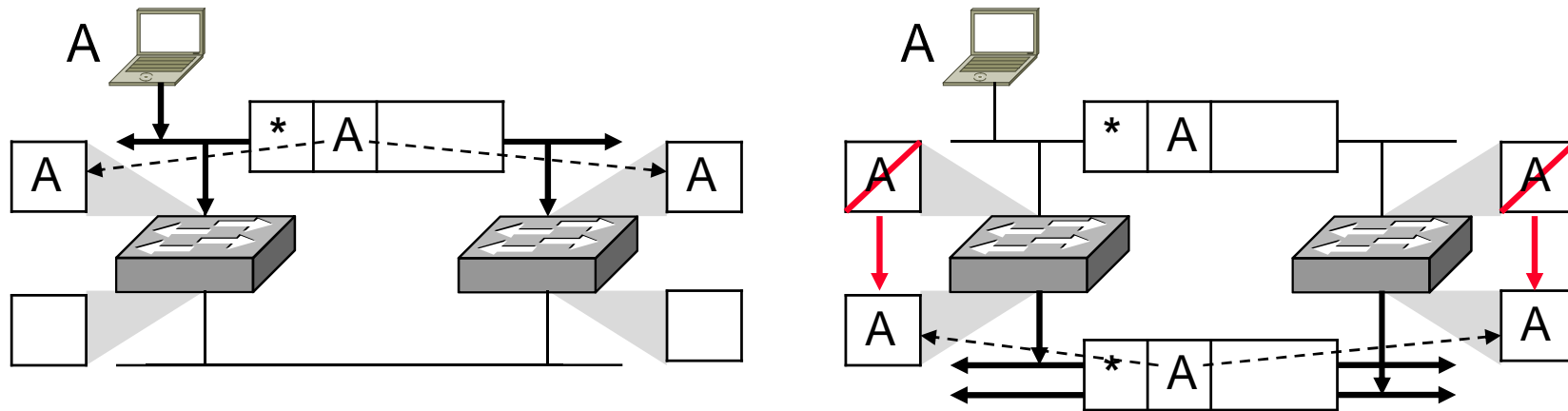
- Switch s uplinkem
 Σ up to 100 Mbit/s

Learning bridge



Spanning Tree Algorithmus

- Motivace: pokud je v síti záložní switch, learning nefunguje a síť se zahltí přeposíláním rámců



- Důvod: graf je cyklický
- Řešení: najít acyklickou podmnožinu, kostru (spanning tree)
- Switche se musejí dohodnout, který z nich bude mít potlačeno forwardování a bude pouze monitorovat provoz
- Protokol (STP) má nezbytné timeouty, start portů je pomalý
 - obvykle lze STA na portu potlačit („faststart“), nutno zvážit

The End